

Student Work

5-1-2005

Web Auction System

Praveen K. Narra
University of Nebraska at Omaha

Follow this and additional works at: <https://digitalcommons.unomaha.edu/studentwork>
Please take our feedback survey at: https://unomaha.az1.qualtrics.com/jfe/form/SV_8cchtFmpDyGfBLE

Recommended Citation

Narra, Praveen K., "Web Auction System" (2005). *Student Work*. 1405.
<https://digitalcommons.unomaha.edu/studentwork/1405>

This Thesis is brought to you for free and open access by DigitalCommons@UNO. It has been accepted for inclusion in Student Work by an authorized administrator of DigitalCommons@UNO. For more information, please contact unodigitalcommons@unomaha.edu.

Web Auction System

A Thesis-Equivalent Project

Presented to the

Department of Computer Science

and the

Faculty of the Graduate College

University of Nebraska

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

University of Nebraska at Omaha

by

Praveen K Narra

May, 2005

UMI Number: EP73445

All rights reserved

INFORMATION TO ALL USERS

The quality of this reproduction is dependent upon the quality of the copy submitted.

In the unlikely event that the author did not send a complete manuscript and there are missing pages, these will be noted. Also, if material had to be removed, a note will indicate the deletion.



UMI EP73445

Published by ProQuest LLC (2015). Copyright in the Dissertation held by the Author.

Microform Edition © ProQuest LLC.

All rights reserved. This work is protected against unauthorized copying under Title 17, United States Code



ProQuest LLC.
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106 - 1346

THESIS-EQUIVALENT PROJECT ACCEPTANCE

Acceptance for the faculty of the Graduate College,
University of Nebraska, in partial fulfillment of the
requirements for the degree Master of Science in Computer Science,
University of Nebraska at Omaha.

Committee

Deborah W. H. 4/19/05
L. L. D. 4/19/05
Stulenmorf. 4/19/2005

Chairperson

V2 Dick

Date

4/17/05

WEB AUCTION SYSTEM

Praveen K Narra, MS

University of Nebraska, 2005

Advisor: Dr. Ken Dick

Auctions are perhaps the oldest market places in history. For centuries, people were allowed to place a price on an item for sale. The person who pledged the highest amount won. Although there were no legal contracts and laws in place, the underlying concept remained the same; both buyer and seller gained something from the transaction. This Computer Science Masters project is aimed at designing an online auction system that is scalable, robust, and flexible to meet the increasing demands of secure and efficient transactions.

This project proposes the use of a three-tier architecture system to conduct online auctions in order to ensure reliability, flexibility and scalability. The system uses HTML and JSP for the front end, WebLogic server as the middle tier, and Oracle as the backend database. This online auction system is built to ensure smooth and efficient transactions between buyers and sellers.

This report describes the basic concepts underlying auctions, specifically online auctions. The report goes on to explain in detail the design and logic of the system by incorporating figures and screen shots for illustration.

Acknowledgments

I am very grateful to my advisor, Dr. Ken Dick for his guidance, encouragement and support during the course of my thesis research.

My sincere thanks to the other members of my committee- Dr. Deepak Khazanchi and Prof. Stanley Wileman for all their helpful comments and suggestions.

I would like to express my humble gratitude to the faculty and staff at the Peter Kiewit Institute for all their support throughout my Master's education.

I would also like to thank my family and friends for their care, advice, and support throughout, without which this research work would not have been possible.

Table of Contents

CHAPTER 1	1
1. INTRODUCTION AND PROBLEM DEFINITION	1
1.1 INTRODUCTION	1
1.2 BACKGROUND	3
1.2.1 Straight Auctions	3
1.2.2 Dutch Auctions	3
1.3 PURPOSE OF THE PROJECT	4
1.4 SCOPE OF THE PROJECT.....	5
1.5 RELATED WORK.....	6
1.6 STRUCTURE OF THE PROJECT REPORT	6
CHAPTER 2	7
OVERVIEW OF AUCTION SYSTEM	7
2.1 DEFINITIONS AND TERMINOLOGY.....	7
2.2 INTERNET FRAUD	8
2.3 SECURING WEB SERVER.....	10
CHAPTER 3	13
AUCTION SYSTEM FRAMEWORK.....	13
3.1 INTRODUCTION	13
3.2 BEFORE AN AUCTION.....	13
3.3 AFTER AN AUCTION.....	16
3.4 TRANSACTION.....	16
3.5 OVERVIEW OF THE SYSTEM DESIGN.....	18
3.6 FEASIBILITY OF AUCTIONS	21
CHAPTER 4	22
AUCTION SYSTEM DESIGN	22
4.1 DATA FLOW DIAGRAMS	22
4.2 FLOWCHART FOR AUCTION SYSTEM.....	25
4.3 USE CASE DIAGRAM	26
4.4 DATABASE DESIGN	27
4.5 JAVA ENVIRONMENT	30
4.5.1 Java Database Connectivity (JDBC).....	30
4.5.2 Java Servlets	31
4.5.3 Client Interaction.....	33
4.5.4 Servlets in Auction system	33
4.5.5 Initializing a Servlet.....	35
4.5.6 The init Method.....	35
4.6 SERVLET AND ENVIRONMENT STATE	36
4.6.1 Threading Issues.....	38
4.6.2 Session Tracking.....	39
4.7 ENTERPRISE JAVA BEANS (EJBs)	42
4.8 ENTERPRISE APPLICATION MODELS	43
4.9 WEBLOGIC SERVER	44
4.9.1 Security.....	44
4.9.2 Authentication	45
4.9.3 Authorization	46
4.9.4 Auditing.....	47
4.9.5 Clustering with WebLogic Server.....	48
4.10 ADDITIONAL ENHANCEMENTS	50

4.10.1 Concurrency types	51
4.10.2 Creating a result set	51
4.10.3 Updates to result sets	54
4.10.4 Cursor movement	55
4.10.5 Batch Updates	56
4.10.6 Statements	56
4.10.7 Execution of batch updates	57
CHAPTER 5	59
IMPLEMENTATION OF WEB AUCTION SYSTEM.....	59
5.1 WEB AUCTION SYSTEM ARCHITECTURE	59
5.1.1 Client component.....	59
5.1.2 Middle Tier component.....	60
5.1.3 Backend tier component	60
5.2 USER INTERFACE OVERVIEW.....	61
CHAPTER 6	71
CONCLUSION.....	71
CHAPTER 7	73
REFERENCES	73

Table of Figures

Figure 1 System overview	20
Figure 2 Context Diagram	23
Figure 3 Level 1 DFD	23
Figure 4 Level 2 DFD	24
Figure 5 Level 2 DFD	24
Figure 6 Level 2 DFD	25
Figure 7 Level 2 DFD	25
Figure 8 Flowchart.....	26
Figure 9 Use Case diagram.....	27
Figure 10 Java environment.....	31
Figure 11 Clients interaction with Java Servlet	32
Figure 12 Servlets in three-tier architecture.....	38
Figure 13 Session Swapping and Persistence	40
Figure 14 Web auction system architecture.....	59
Figure 15 Registration page.....	62
Figure 16 Authentication	63
Figure 17 Account summary.....	64
Figure 18 Items for sale	65
Figure 19 Items that have been bid	66
Figure 20 Item details	67
Figure 21 Post an item for auction.....	68
Figure 22 Closing items (today/tomorrow).....	69
Figure 23 List of items.....	70

Chapter 1

1. Introduction and Problem Definition

1.1 Introduction

Auctions 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 project, we explore the efficiency of common auctions when values are interdependent i.e. the value to a particular bidder may depend on information available only to others. In this setting, it is well known that sealed-bid auctions do not achieve efficient allocations in general since they do not allow the information held by different bidders to be shared.

Typically, in an auction, say of the kind used to sell art, the auctioneer sets a relatively low initial price. This price is then increased until only one bidder is willing to buy the object, and the exact manner in which 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 auction gathers tremendous response

everyday, 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 an auction, all from the comfort of home, is a completely different experience.

Society cannot seem to escape the criminal element in the physical world, and it is the same with online auctions. This project, NETAUCS, an online auction system, addresses just this issue. With online auctions becoming a new trend in the way that business is conducted between individuals, it is absolutely essential to ensure that online auctions systems are safe, secure, robust, scalable, etc.

This project proposes the use of a three-tier architecture system to conduct online auctions; this ensures reliability, flexibility, and scalability. The system uses HTML and JSP for the front end, Web Logic server as the middle tier, and Oracle as the backend database.

Oracle has been chosen as the back end database due to its ability to handle enormous loads should the volume of the system increase in the future. Web logic is used as it reduces the cost and the complexity of the application infrastructure, transforming the technological assets into a critical business advantage.

1.2 Background

There are three types of auctions: Straight auctions, Dutch auctions and Sealed auctions.

1.2.1 Straight Auctions

In Straight auctions, the price of the product on the auction increases with every incremental bid. The price increase of the product is based on the minimum bid increment set by the seller. The seller must sell the item to the highest bidder at the close of the auction. For example, if a seller has put up product 'A' for sale with an initial price of \$50 and five buyers bid for it with the highest bid coming in from buyer 'B' who bids \$62, the product is sold to 'B' for \$62.

1.2.2 Dutch Auctions

During Dutch auctions, the number of units of a particular product for sale is more than one. This implies that multiple products are put up for sales which are identical. In this kind of auction there will be more than one buyer. Buyers can also bid for more than one unit of that product. At the end of the auction all the bidders pay the same price - the lowest winning bid amount (subject to the reserve price being met) for the product.

This is a perfect auction type for those who wish to sell many units of the same product. The Seller starts by listing a minimum price or starting bid for a product and the quantity or number of units for sale. Bidders specify both a bid price and the quantity they wish to buy. All winning bidders pay the same price per item as

the lowest successful bid. If there are more buyers than items, the earliest successful bids get the goods. Higher bidders are more likely to get the quantity they asked for.

Bidders have the option to refuse partial quantity. For example, if you place a bid for 10 items and only 8 are available after the auction, you are not obliged or bound to buy any of them.

Example: A seller places 10 pens on auction at \$20 each. 10 people bid \$20 for one pen each. In this case, all 10 bidders will win a pen for \$20. However if, let's say, five people bid \$25 for a pen each and 10 others bid \$20. The minimum bid for the pen will be raised to \$25, because demand exceeds supply.

Because the \$25 bidders' bid is higher than \$20 bidders, they will be guaranteed a pen. The other 5 pens will go to the earliest \$20 bidders. The final price for each pen will be \$20 (even though some participants placed a higher bid of \$25). Hence all the winning bidders pay the same price which is the lowest successful bid. In this type of auction the time of allocation of the quantities the preference is given to the buyer who puts in the highest bid. This buyer will get the quantity he bid for and this flows progressively down the bidding value stream.

1.3 Purpose of the project

The objective of this project is to develop an auction system that is scalable, robust and secure using some of the key technologies available. This project includes designing, developing, and implementing an auction system. The report

will include a detailed discussion about the System design, the database design and the implementation. This system will use three-tier architecture. The following technologies have been used in the system: Java, BEA WebLogic 7.1, Oracle 10g, HTML and JSP.

The main objectives of the project can be summarized as follows:

- Designing an auction system that is robust, scalable and reliable avoiding major problems surrounding online auction systems.
- Developing the architecture of the system - three-tier software system architecture
- Designing the database and configuring the WebLogic server.
- Identifying and implementing the key issues in online auctions

1.4 Scope of the project

The main scope of the project is to develop an online auction system. The system will be designed and implemented using key technologies. HTML, JSP's are used for the front end application. Java Servlets and Enterprise Java Beans (EJB's) are used to handle the transactions. Oracle is used to create the database and store the data. ✓

As online auctions have gained much prominence in the recent times, the implementation of this project could be beneficial to anyone looking for robust, scalable and reliable auction systems.

1.5 Related work

E-commerce has gained much importance in recent years. Although online-auctions have been around for some time now, it wasn't until recently that people started using them. "ebay" [3] has played a key role in making online auctions successful. Several companies have tried to follow in the steps of ebay, but have proven to be unsuccessful. As a result, there are a very limited number of resources available in the context of successful auction system implementations.

1.6 Structure of the Project Report

The project report is structured in the following way. Chapter 2 provides an overview of the system, including terms and definitions used for the project. Chapter 3 includes a framework of the system. Chapter 4 includes the design of the system. Chapter 5 covers the aspects of the implementation of the system. The conclusion of this project is given in Chapter 6 and finally the References in Chapter 7.

Chapter 2

Overview of Auction system

2.1 Definitions and Terminology

Some of the terms that are used in the auction system are listed below with appropriate definitions.

Auction: An auction is a sale in which a seller presents his product(s) on a public platform/ forum. The selling price in an auction 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 the close of the auction wins the right to purchase the item at that price.

Bid: A bid is the amount of money offered for an item that has been put on sale in an auction. 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.

Win: You win an auction by placing the winning (highest) bid and obtaining the item on auction. The person with the highest bid is the winner of the auction.

Below are some common terminology used in the context of online business transactions:

Electronic Data Interchange (EDI): It allows for transmission of information between private networks.

Open Buying on the Internet (OBI): Formed by the Internet Purchasing Roundtable, OBI makes communication possible between all ecommerce systems [11].

Open Trading Protocol (OTP): It's the standardization of payment transactions and was created to compete against the OBI standard.

Open Profiling Standard (OPS): It holds information about user's likes and interests and distributes it to merchants.

Secure Electronic Transactions (SET): Created by Visa and MasterCard, it encrypts credit card numbers located on servers.

2.2 Internet Fraud

Online-auctions are subject to fraud due to the amount of money involved. This section discusses some of the different types of online fraud. Most of the fraud with online auctions occurs on the seller's side, although some instances may occur on the buyer's side too. Below are a few scenarios where fraud occurs:

- a) **Seller fails to send the merchandise to the buyer:** The highest bidder in an auction gets the item that is for sale. A sale constitutes payment of

money in either credit/debit card or by other means. In some cases even after the payment is received, merchandise is not shipped to buyer.

- b) **Seller sends something of lesser value than advertised during the auction:** Sometimes the seller has an item advertised with better specifications and finally sends the buyer a lesser value item.
- c) **Seller fails to deliver in a timely manner to buyer:** In this kind of situation, seller sends the merchandise much later than the agreed time.
- d) **Sellers do not disclose all relevant information about a product or terms of the sale:** Consider an example where cars are sold on the internet. Seller shows the pictures and specifications of the car online. The buyer with highest bid gets the car. But after the delivery he notices that there are some problems with the car that were not disclosed during the sale. So hiding the facts of the merchandise being sold also constitutes an online fraud.

We can also observe problems that buyers can face. It is important to note that these problems may not be originated by buyer. These problems are mentioned below.

- (a) **Bid siphoning:** In this kind of fraud, sellers lure bidders off legitimate auction sites by offering to sell the item that is featured at legitimate sites for a lower price. They trick consumers into sending money without showing the actual details of the item. But as this transaction has occurred

off the legitimate site, any protection that the original site may provide, such as insurance, feedback forms or guarantees is lost.

- (b) **Shill bidding:** In this kind of bidding, buyers who do not intend to buy the item bid for it to drive up the price. By doing this, the intended buyer has to pay a higher amount of money to obtain the bidding item
- (c) **Bid shielding:** In this kind of bidding, fraudulent buyers submit very high bids to discourage other bidders from competing for the same item and then retract those bids so that people they know can get the item at a lower price.
- (d) **Fraud with escrow service:** In this type of fraud, sellers or buyers pose as escrow services to improperly obtain money or goods. The seller puts goods for sale on an Internet auction and insists that prospective buyers use a particular escrow service. Once buyers provide the escrow service with their payment information, the escrow service doesn't hold the payment; the payment goes directly to the seller. Even if the buyer doesn't receive the merchandise, the seller gets the money.

2.3 Securing Web Server

One of the major concerns in E-commerce is security. Online auction systems involve credit card transfers. This results in a high risk of internet fraud. So, appropriate measures must be taken to avoid online frauds and breaching. The first step that must be taken to ensure security is to secure the web server from

intrusion. The following are a few steps that must be kept in mind when dealing with a web server:

- The web server must be accessible to only authorized personnel who have access to it.
- Keep the server in a secure location.
- Limit the number of users who can log on.
- Do not use the web server as mail server since a mail server can be the source of an unauthorized access.
- Limit the number of people who have access to the server by using two separate machines for the server related functions.

After securing the web server, the second step would be protecting information as it travels over the Internet. One way to secure data while transmitting it over the Internet is through the use of SSL (Secure Socket Layer). SSL encrypts credit card numbers, usernames, passwords, and other valuable information before it is sent over the wire and then it is decrypted when received.

Another method for securing information over the internet can be through the use of Digital Certificates. Digital Certificates work using a public key and a private key. As no two keys are ever identical, it allows for complete anonymity to its users. To keep data secure we can install a digital certificate onto our computer or server that will safeguard our information. After downloading the certificate, the server will contain the private key. Any given person who chooses to trade

secured data with the bidder can receive a public key. The two types of Digital Certificates available to web sites are server certificates and personal certificates.

Server Certificates are a great way to ensure confidentiality of the information exchanged between the vendor and the consumer. By using this kind of certificate, customers can feel secure that their personal information, such as credit card numbers, will be encrypted in the process of transmission to online auction sites. Also, the certificate will enable the customer to recognize the authenticity of the auction web site.

Personal Certificates can give businesses the advantage of allowing their customers, suppliers and partners controlled access to specified content on their web sites as well as disallowing access to specific areas. By implementing personal certificates we can offer different levels of service for premiere customers or request a fee for this service.

We also need to protect the network from outside intrusion. This can be accomplished easily by installing a firewall. Firewalls separate the company's network from the Internet. Implementing a firewall will help to ensure that unauthorized users cannot access information on the network.

Chapter 3

Auction system framework

3.1 Introduction

This section will outline in detail how a transaction will be carried out in a system. This section will explain how an item can be bid on, what happens after a bid has been placed, what happens after a bid has been accepted or rejected, and specifically how a transaction is carried out.

3.2 Before an Auction

The process of an online auction is much the same as a live auction. 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, that is, when you place a bid, when you've been outbid and when you've won an item.

To bid for an item, there is a bidding form through which you may place a bid 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.

Consider an example that outlines the above scenario, with a starting price of \$ 20, Bidding Increment of \$2, Current highest bid of \$22. Now, in order to bid, any bid anywhere above or equal to the current highest bid amount can be placed for this bid to be accepted. Hence a user can bid anywhere between \$22 and \$24 (or more) for the bid to be accepted. This means that he can bid at say \$24 and the next current minimum bid will get adjusted to \$24. If he wishes to place a higher bid, he may do so.

For example, he can place a bid of \$27. This would incorporate the current bid at \$24 and the minimum bid increment of \$2. The remaining \$1 is the additional amount he has bid. Therefore the next bidder should bid a minimum of \$29 as \$27 is now the minimum bid plus an additional \$2 for 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 has to be examined. A description of every item is put up on the item page. To know more about the item and its features, you can make use of the "Ask the seller or Read Q&A" facility provided on every 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.

Bidding can be carried out automatically by enabling the automatic bidding feature. The automatic Bid features bids on a particular item for you up to the maximum amount that you have specified. This amount is the maximum amount that you, the buyer, are willing to pay for an item. It is kept confidential. To activate the Automatic Bid, just enter your maximum bid amount in the "Enter Automatic Bid amount" box and click on the "Bid Now" button.

Automatic Bid is an optional facility. If you activate Automatic Bid, you need not be online to bid for and win an item you desire. The system will bid for you if necessary by the specified bid increment amount to the limit of your Automatic Bid. Please note that the Automatic Bid system will only bid for you when you have either been outbid or need to meet the reserve price at the end of the auction. You will receive an email notification every time the Automatic Bid bids for you.

To bid on multiple items is considered to be a special auction 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 auction. Whether the auction is a multiple auction or not can be found out by checking the quantity box of the item. If the number of quantities displayed is more than one, then we have a multiple quantity auction.

3.3 After an auction

At the end of every auction, the highest bidder receives an email with the item seller's contact details. The bidder should then contact seller to arrange for the delivery and complete the deal offline.

A bidder can know about the item he/she won in an auction once the auction of an item he/she has bid on has ended. The bidder will receive an auto-generated email if the 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 buyer's contact details. The buyer will need to contact the seller to inspect the goods and arrange delivery and payment.

Thus, once the auction 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.

3.4 Transaction

The transaction between the buyer and the seller can be carried out manually if the buyer and seller are located close to each other. In other cases, the transaction is carried out over the net via E-commerce Secured transaction systems.

On-line auctions could be a place of fraud as there is lot of money involved in the transactions. One of the concepts that the online auctions were started with was

the idea that people could police themselves, but those with the intent to take advantage of others have found 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 incorrectly trustworthy to do business with even when they are not.

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, one stands a better chance of resolving the problem satisfactorily for everyone.

The bidder should establish the top price that you are willing to give, keeping in mind how much more than the bid price he can give to cover shipping, insurance and tax. If possible, the identity of the seller should be established; Has the seller been operating with this auction service for an acceptable length of time? Does the auction service have any comments about them, either positive or negative? These things must be considered in an attempt to protect oneself from the risk of being scammed.

One of the better methods is using payment services such as PayPal or Billpoint. Many of these services offer some form of assurance and insurance against fraud. Also the bidder can consider an Escrow service to handle the transaction. Many of the auction sites have an escrow service to offer their clients, however there is usually an extra fee associated with these services as well.

Whatever the means one chooses to minimize the risk of loss due to fraud, in the end it still comes down to the individual consumer and the choices they make. Until the goods are received and all parties satisfied, one should keep record of any and all transactions, whether they are the buyer or the seller. It may help provide proof of what took place, and help get one's money or goods back, if one has the misfortune of being the victim of online auction fraud.

3.5 Overview of the system design

We can take an example of a typical auction scenario and design the system based on it. People tend to buy at seized property auctions 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.

The auction is attended by a limited audience who will not pay the full value of the items. 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 auctions 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, who then transfers the balance owed to the creditor, minus shipping costs and a percentage charged by the Collection Authority.

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

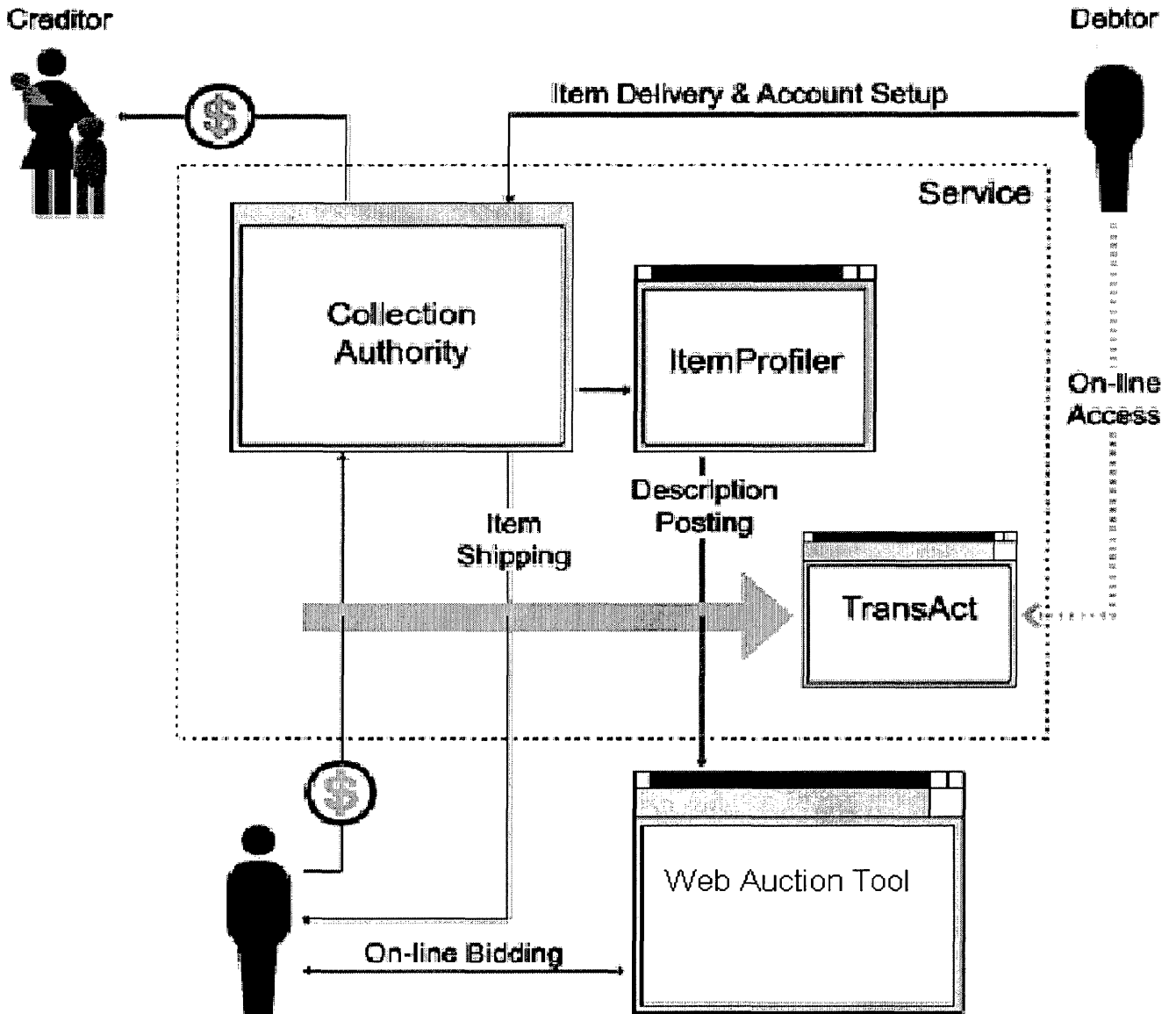


Figure 1 System overview

The item's description is logged under the appropriate category on the auction 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. Thus people can buy the items from the auction site.

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 auction system, can be designed and implemented in a very elegant way.

3.6 Feasibility of Auctions

There are very few situations in which auctions cannot be performed due to existing market conditions, like in 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 an auction would seem inappropriate. Also bidders may be under the impression that commitment to the outcome of an auction would reduce their flexibility.

Chapter 4

Auction system Design

4.1 Data Flow Diagrams

A data flow diagram is a structure analysis tool that is used for graphical representation of data processes through any organization. The data flow approach emphasizes on the logic underlying the system, by using a combination of only 4 symbols. It follows a top down approach. A full description of a system actually consists of set of DFD's, which comprises of various levels. Level two diagrams show additional features of the system. Further, each process can be broken down into one or more detailed data flow diagrams. This occurs repeatedly until sufficient details are described.

The data-flow diagrams for the Auction system are as shown in the figures below.

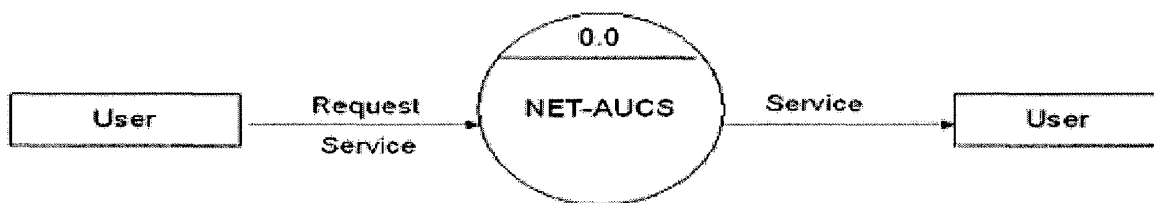
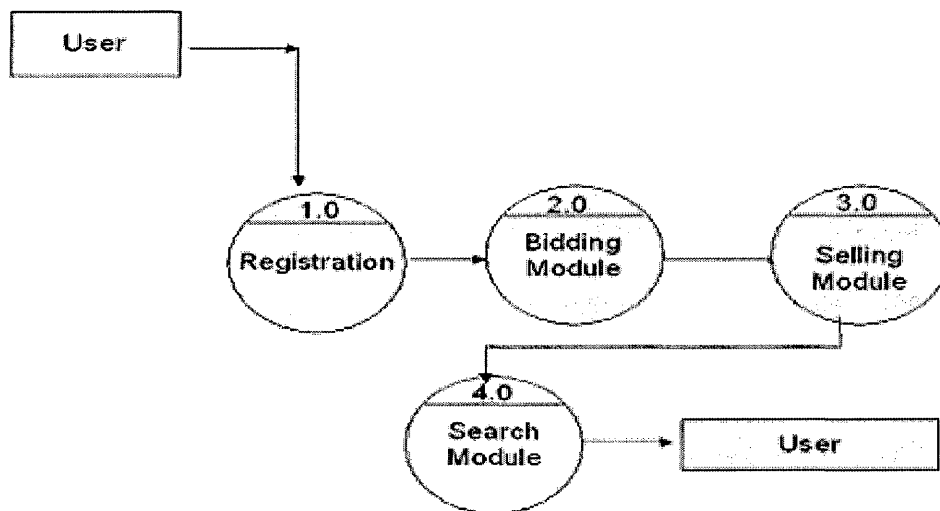


Figure 2 Context Diagram

The above figure shows a level zero diagram for the auction system. Any user can provide a request service to the auction system. After the request is processed by the auction system, the response is sent back to the user.

**Figure 3 Level 1 DFD**

The above figure shows the level 1 diagram for the auction system. Here the first step for the user on the auction system would be registration. After registration, the user can bid for any item. He can also sell any item that he wishes. There is a search feature available in the auction system that helps the user search for any specific items.



Figure 4 Level 2 DFD

The above figure shows a level 2 Data Flow Diagram. In this scenario, a user enters his information at the auction site. His information is then validated and then inserted into the database. If the user bids for an amount, then his information and session is validated and bid is accepted. After the bid is accepted it is updated in the database.



Figure 5 Level 2 DFD

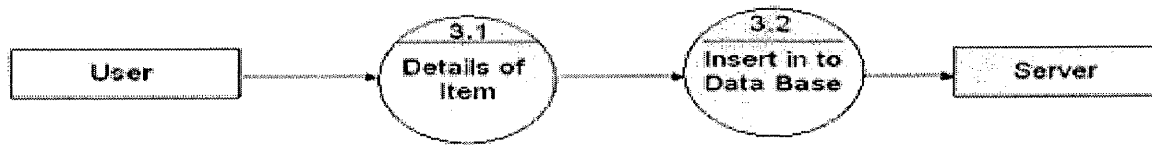


Figure 6 Level 2 DFD

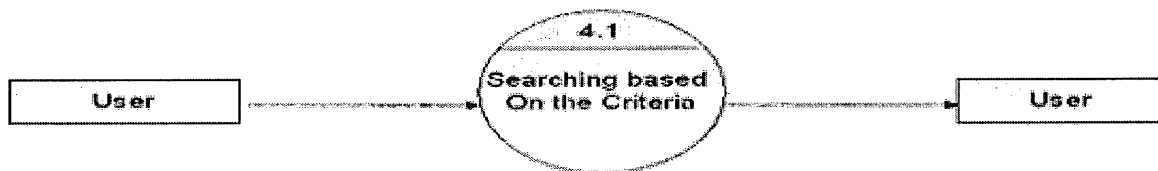


Figure 7 Level 2 DFD

In the above level 2 DFD diagrams, the user enters the detail of the items in the auction website. It is then validated and entered into the database. The user can also search for a particular item available for bidding. This operation can be performed by the 'search' feature and the results are made available to the user.

4.2 Flowchart for Auction system

The figure below shows the flowchart diagram for the auction system. It can be briefly described as follow. Any user can access the auction system through the website. The user cannot perform any actions such as bidding/ selling unless he registers. If he is not registered then he is taken to the user registration page

where user can register at the auction system. Once registered, any user can sell items, bid for items. He can also search for any items available for auction.

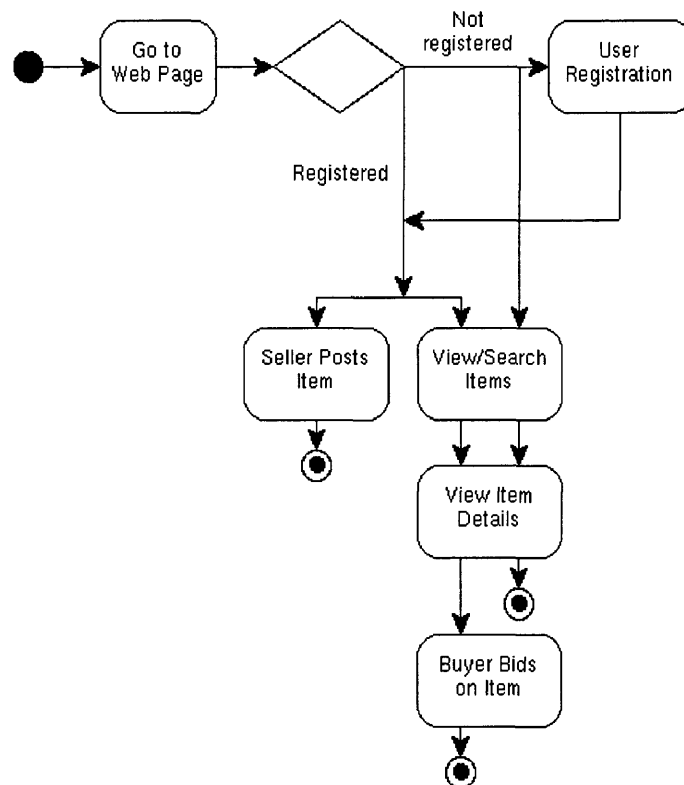


Figure 8 Flowchart

4.3 Use Case Diagram

The use case diagram for the online auction system is as shown in figure 9. The users of the system are classified based on how they use the auction system.

The users are seller, buyer and anyone. Sellers can post the items. Buyer can bid for items. Any person can search for items, view item lists and item details.

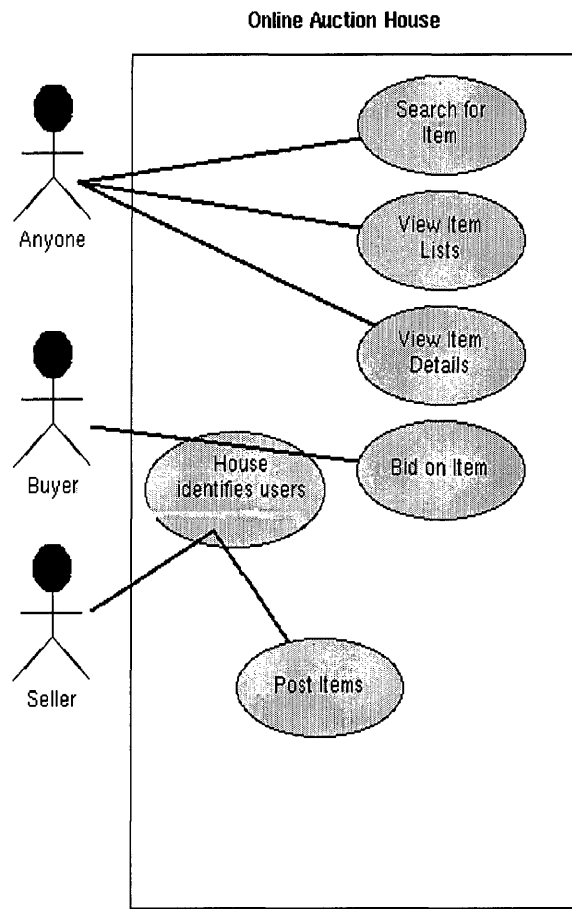


Figure 9 Use Case diagram

4.4 Database Design

A database must be designed for the auction system to hold the details of users, item information, bidding information and other pertinent information. The database is designed such that users do not have any contention and it has been

normalized. Four tables were necessary to hold all the details. Primary/ foreign key relations are then defined, such that tables get populated with valid transactions.

The structure of the tables is shown below. Uinfo_Master table stores the user information. Category_Master table holds the seed data for the different kinds of categories available. Item_Master table holds the items that are placed for bid. Bidding_Info table holds the bidding information of any particular bid.

Uinfo_Master Table (User information)

uname (username)	varchar2(20) primary key
pwd (password)	varchar(10) not null
fname (first name)	varchar2(10)
lname (last name)	varchar2(10)
Email (e-mail)	varchar2(25) not null
phno (phone number)	varchar2(15)
Address	varchar2(50)
City	varchar2(15)
State	varchar2(15)
Pin	varchar2(6)
Country	varchar2(20)
Ccardno (credit card number)	number(16)

Category_Master Table (Category details)

catid (category ID)	varchar2(10) primary key
cat_name (category name)	varchar2(20)

Item_Master Table (Item details)

Itemid	number(6) primary key
Catid (category ID)	varchar2(10) references category_master(catid)
Itemname	varchar2(20)
Description	varchar2(200)
Summary	varchar2(50)
Startprice	number(8,2)
incr_price	number(5)
Stdte	date
Enddate	date
Sellerid	varchar2(20) references uinfo_master(uname)
Bidcnt	number(4)

Bidding_Info Table (Bidding Information)

Bidderid	varchar2(20) references uinfo_master(uname)
Bamt	number(8,2)
Itemid	number(6) references item_master(itemid)
Biddate	date

4.5 JAVA environment

Various Java tools were used for the implementation of the project. These tools include Java Development Kit (JDK), Java servlets, Enterprise Java Beans (EJB's) etc. JDK usually includes hundreds of classes, methods, and interfaces grouped into packages to form part of Java standard library (JSL). Java architecture provides a portable, robust and high performing environment for development.

Java provides portability by compiling the bytecode which is interpreted on each platform by the runtime environment, Java Virtual Machine, JVM. Java also provides stringent compile and runtime checking and automatic memory management in order to ensure solid code.

When we compile the code, the Java compiler creates machine code (byte code) for a hypothetical machine called Java virtual machine (JVM). The JVM will execute the byte code and overcomes the issue of portability. The code is written and compiled for one machine and interpreted by all other machines.

4.5.1 Java Database Connectivity (JDBC)

The JDBC API used to provide result sets that have the ability to scroll in a forward direction only. But scrollable result sets allow for more flexibility in the processing of results by providing both forward and backward movement through their contents. In addition, scrollable result sets allow for relative and absolute positioning. For example, it's possible to move to the fourth row in a scrollable

result set directly, or to move directly to the third row following the current row provided the row exists. The JDBC API also allows the database result sets to be updated directly.

The batch update feature of JDBC allows an application to submit multiple data manipulation statements (insert/update/delete) in a single request to the database. This can provide a dramatic increase in performance when a large number of update statements need to be executed.

4.5.2 Java Servlets

Servlets provide a Java based solution used to address the problems currently associated with server-side programming. Servlets are objects that conform to a specific interface that can be plugged into a Java-based server. Servlets are to the server-side what applets are to the client-side. They differ from applets in that they are without graphics or a GUI component. Servlets extend request/response-oriented servers, such as Java-enabled web servers.

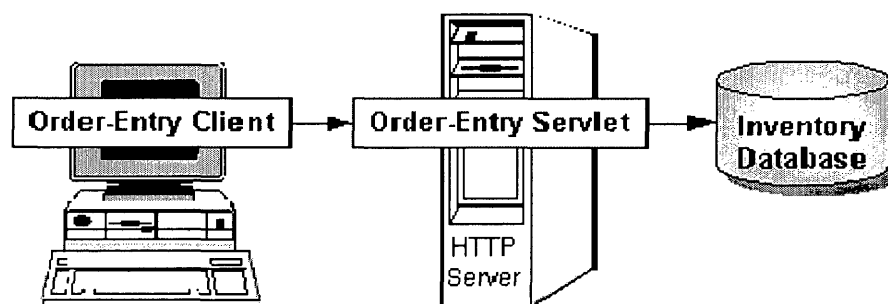


Figure 10 Java environment

Servlets have become most widely used within HTTP servers; many web servers support the Servlet API. For the auction system, servlets are used with WebLogic server.

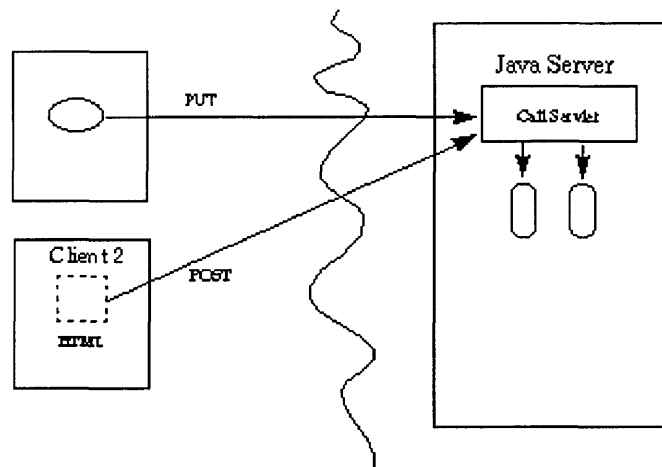


Figure 11 Clients interaction with Java Servlet

Servlets are most often provided by organizations which provide customized multi-user services to their customer bases. However, servlets are also flexible enough to support standardized services such as serving static web pages through the HTTP (or HTTPS) protocols, and proxying services. Since they are used for dynamic extensibility, they may be used in a plug-in style, supporting facilities such as search engines and semi-custom applications. Servlets also provide a strong platform for semi-custom applications, such as web-based order entry or inventory systems.

Since servlets handle multiple requests concurrently, the requests can be synchronized with each other to support collaborative applications such as on-line conferencing. One servlet could forward requests to other servers. This technique can balance the load among several servers which mirror the same content. Or, it could be used to partition a single logical service between several servers, routing requests according to task type or organizational boundaries.

4.5.3 Client Interaction

Client interaction with a servlet is important to complete any transaction. Usually when a servlet accepts a call from a client it receives two objects: `ServletRequest` and `ServletResponse`. A `ServletRequest` encapsulates the communication from the client to the server. It allows the servlet access to information such as the names of the parameters passed in by the client, the protocol being used by the client, and the names of the remote host that made the request and the server that received it. It also provides input stream, which is used to get data from clients that use application protocols such as the HTTP POST and PUSH method.

`ServletResponse` encapsulates the communication from servlet back to the client. It allows the servlet to set the content length and MIME type of the reply. It also provides an output stream, `ServletOutputStream`, and a `Writer` through which the servlet can send the reply data.

4.5.4 Servlets in Auction system

Various servlets are used in the Auction system to achieve the functionality. Servlets provide the base for addressing server-side programming. In the below example we show a sample servlet. In this servlet both the HTTP servlet request/response queues are used.

```
public class SimpleServlet extends HttpServlet
{
    /*Sample servlet to build a web page */
    public void doGet (HttpServletRequest request,
```

```

        HttpServletResponse response)
throws ServletException, IOException
{
    PrintWriter    out;
    String          title = "Simple Servlet Output";
    // Here the content type and other response header
    // fields are first
    response.setContentType("text/html");
    // here data is written to the response
    out = response.getWriter();

    out.println("<HTML><HEAD><TITLE>");
    out.println(title);
    out.println("</TITLE></HEAD><BODY>");
    out.println("<H1>" + title + "</H1>");
    out.println("<P>This shows a sample servlet.");
    out.println("</BODY></HTML>");
    out.close();
}
}

```

Sample servlet

We can observe in the above servlet that this servlet extends the `HttpServlet` class, which implements the servlet interface. Also this servlet overrides the `doGet` method in the `HttpServlet` class. The `doGet` method is called when a client makes a GET request. GET is default HTTP request method, and results in the simple HTML page being returned to the client. We can also observe in the

doGet method that the user's request is represented by an `HttpServletRequest` object.

The response to the user is represented by an `HttpServletResponse` object. Because text data is returned to the client, the reply is sent using the `Writer` object obtained from the `HttpServletResponse` object.

4.5.5 Initializing a Servlet

When a server loads a servlet, the server runs the servlet's `init` method. Initialization completes before client requests are handled and before the servlet is destroyed. Even though most servlets are run in multi-threaded servers, servlets have no concurrency issues during servlet initialization. The server calls the `init` method once, when the server loads the servlet, and will not call the `init` method again unless the server is reloading the servlet. The server can not reload a servlet until after the server has destroyed the servlet by calling the `destroy` method.

The `init` method provided by the `HttpServlet` class initializes the servlet and logs the initialization. A `HTTP Servlet` handles client requests through its `service` method. The `service` method supports standard HTTP client requests by dispatching each request to a method designed to handle that request.

4.5.6 The `init` Method

For initialization of the servlet the `init` method provided by the `HttpServlet` class can be used and it also logs the initialization. To do the initialization specific to

the servlet, we can override the `init ()` method by following some rules: If an initialization error occurs that renders the servlet incapable of handling client requests, throws a `UnavailableException`.

An example of this type of error is the inability to establish a required network connection. Also we do not need to call the `system.exit` method.

An example for the `init` method is as shown below.

```
public class BookDBServlet ... {  
    private BookstoreDB books;  
    public void init() throws ServletException {  
        // here we load the database to prepare for requests  
        books = new BookstoreDB();  
    }  
    ...  
}
```

4.6 Servlet and Environment State

Servlets have instance-specific data as they are Java objects. Servlets are independent applications running within servers. They run without needing the complexity of additional classes. Servlets have access to servlet-specific configuration data at initialization time. By having servlet specific configuration data, different instances of the same servlet class can be initialized with different data, and can be managed as differently named servlets. The data provided at

initialization time would include an area where each instance would keep its persistent instance-specific state. Servlets also have the ability to interact with their environment through a `ServletContext` object.

Servlets are always dynamically loaded, although servers will usually provide an administrative option to force loading and initializing particular servlets when the server starts up. They are loaded using normal Java class loading facilities and they can also be loaded from the local file system. This feature enables increased flexibility in system architecture and easier distribution of services in a network.

If we are using three-tier architecture as we used in the web auction system, then servlets follow a different approach. The first tier of the application can use any number of Java enabled browsers. Also we can have complex user interface tasks that would handle Java applets downloaded from the second tier servers. We can have simpler tasks handled using standard HTML forms.

The second tier of the three-tier system can consist of servlets which encapsulate the rules and logic of the application at hand. These rules could include arrangements to deliver a new shipment of parts. We can also define rules for secure applications at this level.

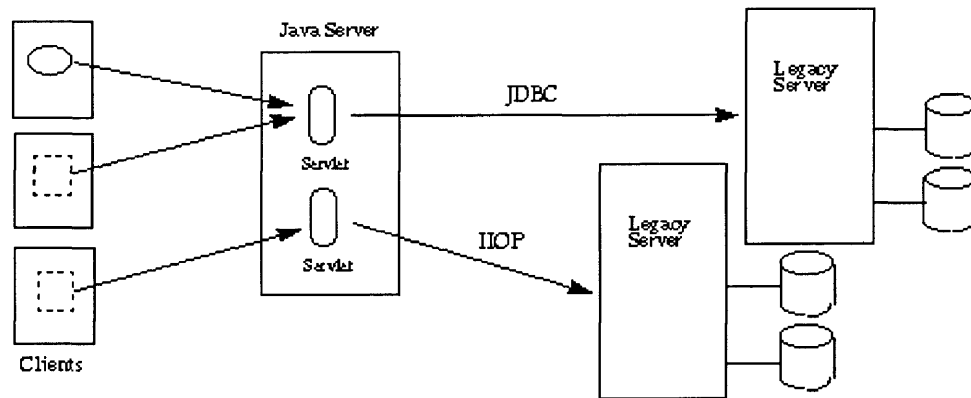


Figure 12 Servlets in three-tier architecture

Servlets can be used to connect the second tier of an application to the first tier. The third tier of such systems consists of databases. For the purpose of the project we have used an Oracle database for data storage, retrieval and other database related functions. This tier might be accessed using relational database interfaces such as JDBC.

4.6.1 Threading Issues

Multiple clients can be served concurrently by HTTP servlets. Servlets should have methods that can work for multiple clients using shared resource. If we want to implement concurrency by creating a servlet that handles only one client request at a time, then we have to implement a single thread model interface in addition to extending the `HttpServlet` class. Single thread model can be implemented by just declaring that the servlet implements the interface. The server makes sure that servlet runs only one service method at a time.

4.6.2 Session Tracking

Session tracking is a flexible, lightweight mechanism that enables stateful programming on the web. It helps maintain persistent user profiles and multi-user sessions. A session is a series of requests from the same user that occur during a time period. Session tracking gives servlets and other server-side applications the ability to keep state information about a user as the user moves through the site. Server-side applications can use this facility to create more stateful user experiences and to track the activity on the web-site.

Java Web Server maintains user state by creating a session object for each user on the site. These Session objects are stored and maintained on the server. When a user first makes a request to a site, the user is assigned a new session object and a unique session ID. The session ID matches the user with the session object in subsequent requests.

The Session object is then passed as part of the request to the servlets that handle the request. Servlets can add information to Session objects or read information from them. After the user has been idle for more than a certain period of time, the user's session can be made invalid, and the corresponding session object is destroyed.

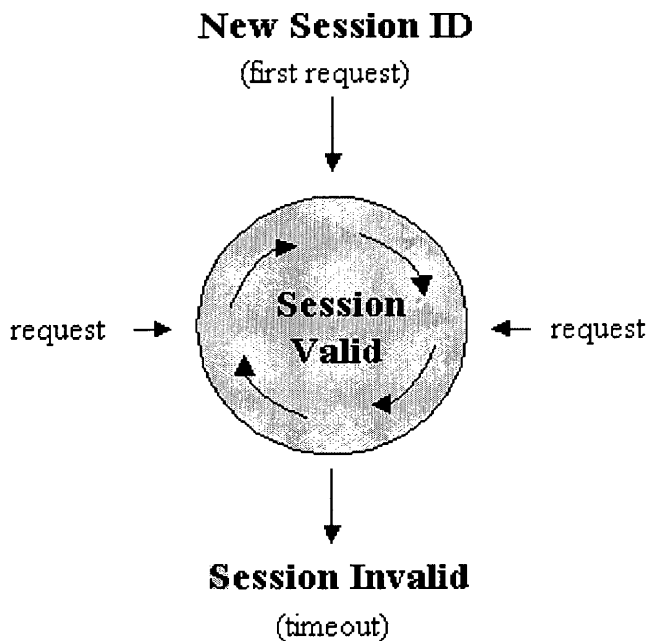


Figure 13 Session Swapping and Persistence

Any e-commerce site including the designed auction system must support many valid sessions. Any big commercial auction site can have hundreds, or even thousands, of simultaneously valid sessions. Because each session can contain arbitrary data objects placed there by the application servlets, the memory requirements for the entire system can grow prohibitively large.

The session tracking system can reduce this problem by placing a limit on the number of session objects that can exist in memory. This limit is usually set in the `session.maxresidents` property. When the number of simultaneous sessions exceeds this number, the Session Tracker swaps the least recently-used sessions out to files on disk. But these sessions will not be lost and will be

reloaded into memory if further requests come in for those sessions. Thus we can allow more sessions to be valid than allowed by memory.

This process of session swapping doesn't affect session invalidation. If a session goes unused for longer than the normal invalidation time, the session is invalidated, whether it is in memory or on disk. Session invalidation is set in the `session.invalidationinterval` property.

Sessions are written to and read from disk using Java serialization. So only serializable objects put into the session object will be written to disk. Any objects put into the session object that is not serializable will remain in memory, even if the rest of the session object has been written to disk. This doesn't affect session tracking but does reduce the memory savings that the session tracker gets from swapping a session to disk. So it would be desirable to put only serializable objects into the session objects.

The session-swapping mechanism is also used to implement session persistence. If the session persistence feature is enabled then when the server is shut down, sessions still in memory are written to the disk as specified in the `session.swapdirectory` property. When the server starts again, sessions that were written to disk will once again become valid. So by having session persistence any server can be restarted without losing existing sessions.

4.7 Enterprise Java Beans (EJBs)

EJB play an important role in the three-tier architecture used in web auction system. EJB architecture is a server-side component model for the Java platform. Its main purpose is to solve the problems of having all the server code at the same place, ability to work on different platform etc.

EJB help the developers to write business logic. The developer does not need to write code that handles transactional behavior, security, connection pooling, or threading, because the architecture assigns this task to the server vendor. The Enterprise JavaBeans (EJB) component architecture enables us to build scalable, secure, multiplatform, business-critical applications as reusable, server-side components.

There are some specific benefits of using EJB's in the system. These benefits include.

- a) **Productivity:** Development of code could be made faster by using this technology. Developers can focus on the writing the business logic as other aspects such as behavior, security, connection pooling, or threading can be assigned to server.
- b) **Industry Support:** Customers attempting to build EJB systems will have a range of solutions to choose from. Enterprise JavaBeans technology has been adopted, supported and put on the product roadmaps of over 25 companies.

- c) **Architectural Independence:** Enterprise JavaBeans technology insulates the developer from the underlying middleware. Developer can only be concerned about Java platform. We can also change the middleware layer. In addition to the cross-platform benefits described below, this allows the EJB server vendor the opportunity to improve and change the middleware layer without disturbing a user's EJB applications.
- d) **Server-Side Write Once, Run Anywhere:** By having the Java platform, EJB technology can be run on any platform. Considering that the server should support Enterprise JavaBeans APIs.

4.8 Enterprise Application Models

There are two fundamental models for building enterprise applications with EJB. In the first model, the client begins a session with an object that acts like an application, executing a unit of work on behalf of the client, possibly including multiple database transactions. In the second model, the client accesses an object that represents an entity in a database. EJB is designed to apply for both these kind of models.

- a) In the first model Session Beans is used. A session Bean is an object that represents a transient conversation with a client, and it executes database reads and writes for the client. These database accesses can be in the context of a transaction. A session Bean's fields contain the state of the conversation and are transient. But if either server or client is

crashed then session beans are gone. This model is typically used with database programming languages such as PL/SQL.

- b) In the second model Entity Beans are used. An entity Bean represents data in a database, along with the methods to act on that data. In a relational database context, for a table of employee information, there is one Bean for each row in the table. Entity Beans are transactional, and are long-lived. As long as the data remains in the database, the entity Bean exists. This model is mostly used in object-oriented databases.

4.9 WebLogic Server

It is very important to choose the right web server for the auction system as it plays an important role considering security, scalability and reliability. Various other factors have been considered to choose WebLogic server as the web server. The important features and the role it plays in the implementation of web auction system are discussed in detail. WebLogic Server provides a framework for all the applications, including J2EE objects, JSPs, Servlets, EJBs, JDBC connection pools, and JMS destinations. It also helps to differentiate security and business logic.

4.9.1 Security

It can be observed that WebLogic Server provides an integrated approach to solve the security problem for enterprise applications. Security architecture with WebLogic server can be achieved in two different ways. It can be secured either

through the security features included with the server out-of-the-box, or by extending the open Security Service Provider Interface to a custom security solution. We can also customize the security architecture by adding specialized security solutions from any other security vendor.

The two important goals that should be considered while designing the application security are to enforce business policies concerning which people should have access to which resources and information should be secured for unauthorized access. Security code should be combined with business logic to achieve both the goals. So even if developers make any mistakes with application code, the inbuilt security wouldn't allow intruders to get access to the system.

4.9.2 Authentication

The first step towards better security is having authentication. It is considered to be the first line of defense. By knowing the identity of requesters enables the application layer to decide whether to grant their requests or not. Authentication schemes establish the identity of the requestor and provide a means to verify that the request is credible. Authentication schemes depend on a number of factors, including the sensitivity of protected resources, expected modes of attack, and solution lifecycle cost. Even if the existing system has any other means of authentication, a middleware server should work with these existing security schemes.

One of the features that is currently available in many e-commerce web sites is Web single sign-on (SSO). Single sign-on is a mechanism whereby a single action of user authentication and authorization can permit a user to access all computers and systems where he has access permission, without the need to enter multiple passwords. SSO reduces system failures by avoiding human errors. So by having single sign-on, the number of times any user is authenticated is reduced dramatically and many problems of working with authentication can be avoided.

SSO can be established by sophisticated behind-the-scenes communication between the two servers involved in handing off a user session. So although SSO is highly desirable it is difficult to implement. SSO can be achieved in minimal effort by considering additional software's that implement it. It can be later added to the existing middleware server and customer can have the SSO feature.

4.9.3 Authorization

Authorization is next step after authentication. It happens after the application has established the requester's identity. Authorization should decide whether the set of existing security policies allow the user for access to the system. With middleware infrastructure such as J2EE a static role-based system can be used. By using this method, security administrators can assign roles to users and then update these assignments as conditions require. When additional components of

the application are deployed then security administrators can indicate the roles allowed to access the component.

During the runtime of application, if a request comes from a user with the necessary roles, the application grants the request. This static approach ignores the dynamic nature of many business policies. There are policies that are even more sophisticated where authorization depends on the combination of roles assigned to a user, as well as the content of the request. Middleware infrastructure should support these kinds of dynamic policies. This would greatly help in having better security for the system.

4.9.4 Auditing

Security auditing must be considered for the systems even if the existing security architecture provides efficient authentication and authorization to the users accessing the system. If an application can simultaneously use two different authorization services, then this would be a different scenario. Situations such as these should be studied to avoid any security loopholes.

For proper auditing, a security auditor must perform verification, detection, and investigation. Administrators should record of all security events in a single location. Keeping records of the important events in an efficient manner helps to search the records faster.

4.9.5 Clustering with WebLogic Server

The existing web auction system should be easily scalable and should have high availability. One of the better ways of achieving this would be by clustering the WebLogic server. By applying this clustering technique we can easily obtain application scalability. The server-side Java platform, encapsulated by J2EE, has emerged as one the better ways to deliver information services across the Web. High availability is desired as any failure can directly affect the business critical applications. Infrastructure is developed such that high availability can provide uninterrupted service to the users of the system.

A WebLogic Server cluster usually consists of a number of WebLogic Servers deployed on a network. These servers work together to provide a more powerful and reliable application platform when compared to a single server. But the server cluster is transparent to the end user. When a user tries to connect to the cluster, it would appear to him as if he is connecting to a single server. But there are a group of servers which act as a single super server.

The main functionality that can be achieved by considering a cluster would be

- a. **Scalability:** By having a cluster the capacity is not limited to a single server or a single machine. Whenever we observe additional load on the system new servers can be added to the cluster dynamically in order to increase capacity. If additional hardware such as CPUs and RAMs are

needed then a new server can be added to the cluster. Thus online systems can be made easily scalable.

- b. **High-Availability:** A cluster can be used to obtain the redundancy of multiple servers to avoid users having system failures. The same service can be provided on multiple servers in the cluster. If one server fails, another can take over. The ability to failover from a malfunctioning server to a functioning server ensures the seamless availability of the application to users. The failover is transparent and end users can have continuous access to the system.

This server clustering can be achieved in many ways. Some of the ways include Web clustering, Object clustering, JMS (Java Messaging Service) clustering, and clustering of JDBC connection pools (metapools) [13]. As WebLogic server cluster appears as a single server to its clients it can also be addressed using a simple URL. This URL maps into multiple servers in the cluster to ensure that connection requests can be load-balanced across the cluster and that failover can occur transparently.

One of the ways to create access to WebLogic server cluster is by using a single DNS (Domain Name System) name that maps to the IP addresses of all of the clustered servers. When a DNS name is mapped to multiple addresses, the DNS server will cycle through the list on each successive lookup of that name. Thus it provides a simple form of load balancing and failover. Each time a user resolves

the URL he will get the next address in the cycle. This ensures that end connections are evenly balanced across the cluster.

The second way that cluster access can be achieved is by setting up another web server that proxies back to the cluster. The proxy server can be the web server included in WebLogic Server. We can also build an Apache server that acts as a proxy within a WebLogic plug-in. The proxy server is set up to redirect requests based on the type of requests. If it gets a request for HTML pages then it can process the request else for other servlet or JSP requests it can redirect the request.

4.10 Additional Enhancements

This section presents the new functionality that can be used with result sets.

Two new enhancements can be added to the result sets. These include scrolling and updatability. Using the scrolling feature a result set created by executing a statement may support the ability to move backward through its contents, as well as forward. Result sets that support this capability are called scrollable result sets. Result sets that are scrollable also support relative and absolute positioning.

Absolute positioning is the ability to move directly to a row by specifying its absolute position in the result set, while relative positioning gives the ability to move to a row by specifying a position that is relative to the current row.

4.10.1 Concurrency types

An application may choose from two different concurrency types for a result set. They are read-only and updatable. A result set that uses read-only concurrency does not allow updates of its contents. This can increase the overall level of concurrency between transactions, since any number of read-only locks may be held on a data item simultaneously. A result set that is updatable allows updates and may use database write locks to mediate access to the same data item by different transactions. Since only a single write lock may be held at a time on a data item, this can reduce concurrency.

Alternatively, an optimistic concurrency control scheme may be used if it is thought that conflicting accesses to data will be rare. Optimistic concurrency control implementations typically compare rows either by value or by a version number to determine if an update conflict has occurred.

4.10.2 Creating a result set

Result sets are used in the auction system. So creation of result set is discussed here. This can be shown easily by showing an example that is forward-only and uses read-only concurrency. No performance hints are given by the example, so the driver is free to do whatever it thinks will result in the best performance. The transaction isolation level for the connection is not specified, so the default transaction isolation level of the underlying database is used for the result set that is created.

```
Connection con =  
DriverManager.getConnection("jdbc:my_subprotocol:my_subname");  
Statement stmt = con.createStatement();  
ResultSet rs = stmt.executeQuery("select itemid, itemname from Item_master");
```

In this example a scrollable result set that is updatable is created and is sensitive to updates. Rows of data are requested to be fetched fifty at-a-times from the database.

```
Connection con =  
DriverManager.getConnection("jdbc:my_subprotocol:my_subname");  
Statement stmt = con.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE,  
ResultSet.CONCUR_UPDATABLE);  
stmt.setFetchSize(25);  
ResultSet rs = stmt.executeQuery("select itemid, itemname from Item_master ");
```

This is an example of a scrollable result set is created that is updatable and sensitive to updates. Rows of data are requested to be fetched fifty at-a-times from the database.

```
Connection con =  
DriverManager.getConnection("jdbc:my_subprotocol:my_subname");  
Statement stmt = con.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE,  
ResultSet.CONCUR_UPDATABLE);
```

```
stmt.setFetchSize(50);  
ResultSet rs = stmt.executeQuery("SELECT emp_no,salary FROM employees");
```

The example below creates a result set with the same attributes as the previous example. However, a prepared statement is used to produce the result set.

```
PreparedStatement pstmt = con.prepareStatement("SELECT emp_no, salary  
FROM employees where emp_no = ?",  
ResultSet.TYPE_SCROLL_SENSITIVE,  
ResultSet.CONCUR_UPDATABLE);
```

```
pstmt.setFetchSize(50);  
pstmt.setString(1, "100010");  
ResultSet rs = pstmt.executeQuery();
```

There are various methods available to see which result sets set types are supported by JDBC driver.

The method `DatabaseMetaData.supportsResultSetType()` can be used for this purpose. However, an application may still ask a JDBC driver to create a `Statement`, `PreparedStatement`, or `CallableStatement` object using a result set type that the driver does not support. In this case, the driver should issue an SQL warning on the connection that produces the statement and choose an alternative value for the result set type of the statement according to the following rules:

1. If an application sends request for a scrollable result set type the driver should use a scrollable type that it supports, even if this differs from the exact type requested by the application.
 2. If the application requests for a scrollable result set type and the driver does not support scrolling, then the driver should use a forward-only result set type.
- Similarly, the method `DatabaseMetaData.supportsResultSetConcurrency()` can

be called to determine which concurrency types are supported by a driver. If an application asks a JDBC driver for a concurrency type that it does not support then the driver should issue an SQL warning on the connection that produces the statement and choose the alternative concurrency type. The choice of result set type should be made first if an application specifies both an unsupported result set type and an unsupported concurrency type.

4.10.3 Updates to result sets

A result set is updatable if its concurrency type is `CONCUR_UPDATABLE`. Rows in an updatable result set may be updated, inserted, and deleted. The example below updates the first row of a result set. The `ResultSet.update()` methods is used to modify the value of an individual column in the current row, but does not update the underlying database. When the `ResultSet.updateRow()` method is called the database is updated. Columns may be specified by name or number.

```
rs.first();  
rs.updateString(1, "100020");  
rs.updateFloat("salary", 10000.0f);  
rs.updateRow();
```

But the JDBC driver discards the updates if the application moves the cursor from the current row before calling `updateRow()` method. In addition, an application can call the `ResultSet.cancelRowUpdates()` method to explicitly cancel the updates that have been made to a row. The `cancelRowUpdates()`

method can be called after calling update() method and before calling updateRow(), otherwise it has no effect.

4.10.4 Cursor movement

A result set maintains an internal pointer called a cursor that indicates the row in the result set that is currently being accessed. It is analogous to the cursor on a computer screen which indicates the current screen position. The cursor maintained by a forward-only result set can only move forward through the contents of the result set. Thus, rows are accessed sequentially beginning with the first row.

Iterating forward through a result set is done by calling the ResultSet.next() method. In scrollable result sets where any result set whose type is not forward only implement the method, beforeFirst(), which may be called to position the cursor before the first row in the result set. The example below positions the cursor before the first row and then iterates forward through the contents of the result set. The get() methods are used to retrieve column values.

```
rs.beforeFirst();
while ( rs.next()) {
    System.out.println(rs.getString("emp_no")+ " " +rs.getFloat("salary"));
}
```

We can also iterate backward through a scrollable result set. It is shown below.

```
rs.afterLast();
while (rs.previous()) {
    System.out.println(rs.getString("emp_no") +
```



```
“ “ + rs.getFloat(“salary”));
}
```

4.10.5 Batch Updates

The batch update facility allows multiple update operations to be submitted to a data source for processing at once. By submitting multiple updates together, instead of individually, can greatly improve performance. Statement, PreparedStatement, and CallableStatement objects can be used to submit batch updates.

4.10.6 Statements

The batch update facility allows a statement object to submit a set of heterogeneous update commands together as a single unit, or batch, to the Database Management System (DBMS). If we consider an example where all update operations are required to insert a new employee details into a company's database. This update is done as a single batch.

```
// Here we turn the autocommit off
con.setAutoCommit(false);
Statement stmt = con.createStatement();
stmt.addBatch("INSERT INTO employees VALUES (1000, 'Bob')");
stmt.addBatch("INSERT INTO departments VALUES (260, 'Book')");
stmt.addBatch("INSERT INTO emp_dept VALUES (1000, 260)");
// Here we submit a batch of update commands for execution
int[] updateCounts = stmt.executeBatch();
```

In the example, autocommit mode is disabled to prevent the driver from committing the transaction when Statement.executeBatch() is called. Disabling

autocommit allows an application to decide whether or not to commit the transaction in the event that an error occurs and some of the commands in a batch cannot be processed successfully.

So we need to turn off auto-commit in case of batch updates.

4.10.7 Execution of batch updates

The `Statement.executeBatch()` method submits a statement's batch to the underlying data source for execution. Batch elements are executed serially in the order in which they are added to the batch. When all the elements in a batch execute successfully, `executeBatch()` returns an integer array. It contains one entry for each element in the batch. The entries in the array are ordered according to the order in which the elements were processed. An entry in the array may have the following values:

1. If the value of an array entry is greater than or equal to zero, then the batch element was processed successfully. The value is an update count indicating the number of rows in the database that were affected by the element's execution.
2. A value of -2 indicates that an element was processed successfully, but that the number of effected rows is unknown. Calling `executeBatch()` closes the calling Statement object's current result set if one is open.

The statement's internal list of batch elements is reset to empty once `ExecuteBatch()` returns. `ExecuteBatch()` throws a `BatchUpdateException` if any of

the elements in the batch fail to execute properly, or if an element attempts to return a result set. Only DDL and DML commands that return a simple update count may be executed as part of a batch. When a `BatchUpdateException` is thrown, the `BatchUpdateException.getUpdate-Counts()` method can be called to obtain an integer array of update counts that describes the outcome of the batch execution.

Chapter 5

Implementation of Web Auction system

5.1 Web auction system architecture

The implemented web auction system follows a three-tier architecture. It has client component (front end application), middle tier component, and backend tier component. The three tier architecture is as shown in the figure below.

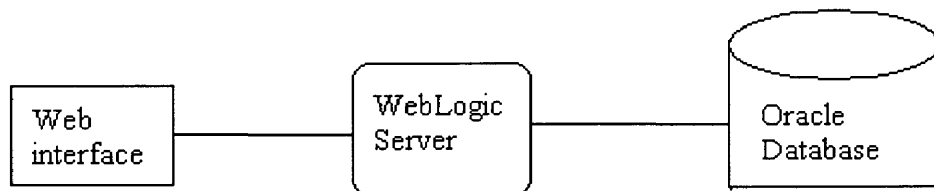


Figure 14 Web auction system architecture

5.1.1 Client component

The client component usually contains programs executed by users, including Web browsers and network-capable application programs. For the current system's client interface HTML pages act as the front end. For the user interface we have deployed servlets and JSP pages in WebLogic Server.

We can also make a J2EE application client to run on client machine which can provide a richer user interface than can be provided by a markup language. Application clients directly access EJBs running in the server and can, as appropriate, communicate through HTTP with servlets running in the Web server.

5.1.2 Middle Tier component

The middle tier of the web auction system consists of WebLogic Server. For the implementation of the project only a single WebLogic server has been chosen. But for the purpose of scalability, additional WebLogic servers can be added. By having the WebLogic Server cluster option we can distribute client requests and back-end services among multiple WebLogic Servers.

Programs in the user component execute as if a single server is present. As the workload increases, we can add WebLogic Servers to the cluster to share the work. The cluster uses a selectable load-balancing algorithm to choose a WebLogic Server in the cluster that is capable of handling the request. Applications can be made easily scalable by having middle-tier server and reliability, scalability, and high performance can be easily achieved.

5.1.3 Backend tier component

Database is the backend tier component in the three tier configuration. We have chosen Oracle for the database. Oracle provides the most appropriate database connectivity for Java. Oracle enterprise version has been chosen for the web auction system. Selective components of the enterprise version have been

installed and other components such as transparency gateways, statspack, and performance tuning pack have been avoided. After the installation of the database components, auction user is created and appropriate tables that store the user, bidding, and transaction data are created.

5.2 User Interface Overview

In the web auction system for the front-end of the application HTML, JSPs are used. Servlets and EJB's communicate with the middle-tier. WebLogic Server is used for the middle-tier. Oracle database is used to handle all the DBMS requests. Multiple users have been created to show the functionality of the project. Some of the users sell the item at the auction site. Some users bid for the items. There is also an administrator login available to delete any offensive content being sold. Additionally administrator can also delete a particular category. User and administrator interface screens are shown in this section.

For any user to sell or buy items at online-auction site must be registered.

Registration

Step One - Your information

Login ID: Not case sensitive, 5 characters minimum, 20 maximum

Password: Please include one numeric character. 5 character minimum. Can not contain "Login".

Verify Password:

Primary Email:

Address: Address of all your correspondence.

Credit Card No.:

First Name:

Last Name:

Phone Number: Just Numbers, No dashes

Step Two - Address Information

Address Information:

House No.:

Street No.: Please enter full street address.

City: Full spelling, no punctuation or numbers.

State: AL

Country: United States

Zip:

Figure 15 Registration page

On the home page of auction web-site, user can find the link to register. The registration link opens up the registration form. The registration form is as shown in the figure above. User needs to enter all the required details provided in the registration form.

After registration, user can enter the auction web site through the login page. In the login page, he needs to provide his 'loginid' and 'password' in this page. This page is as shown below

NetAucs Login Page - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Home Search Favorites

Address http://localhost:7001/auction/login.htm Go Links Google

NETAUCS

[Home](#) [Login](#) [Register](#) [Sell](#) [Help](#)

Search Choose Category

User ID

Password

| [New Items](#) | [Closing Items](#) | [Sell Items](#) | [Home](#) |

Local intranet

Figure 16 Authentication

After the user is successfully logged on, he is directed to account summary page. On the account summary page, user can find details regarding his selling items, buying items and personal information. By using the third link, edit personal information, user can edit the information he has given during the registration. This page is as shown in the figure below.

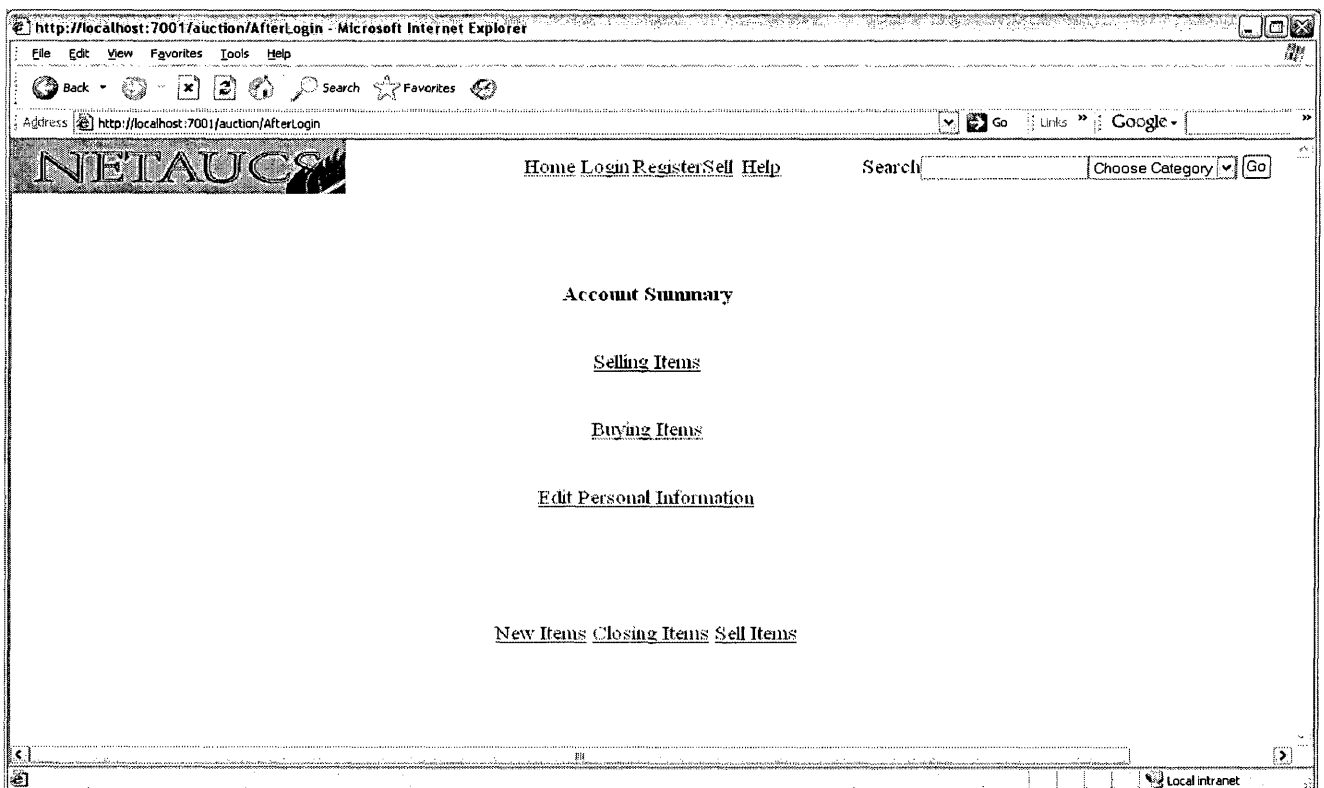


Figure 17 Account summary

Selling items link lists the items that the user has put up on the auction site for selling purpose. Here we show the selling items list for user 'Mike' when he clicks on the selling items link.

The screenshot shows a Microsoft Internet Explorer window with the address bar displaying `http://localhost:7001/auction/Selling?uname=mike`. The page header includes a navigation bar with links: [Home](#), [Login](#), [Register](#), [Sell](#), and [Help](#). There is also a search bar and a category dropdown menu. The main heading is "Items you are Selling". Below this is a table listing items for sale.

Item	Description	Current Price	Number Of Bids	Auction Starting Date
30005	published by Grosset and Dunlap; binding damages but all pages are together	20	0	2004-11-27
50003	Wilson NFL JR leather football; has a pittsburgstealers logo on it; made of genuine leather	8	0	2004-11-27
50004	Calvin Klein Crave 2.5 fl oz brand new; retail price 75 dollars; get it here for a lower price; great for christmas gift	45	0	2004-11-27
60002	Motorola A388 GSM PDA Tri band World phone ; has all protective lenses ; only pda and stylus included-nothing else	350	0	2004-11-27

Below the table, there are links: [New Items](#), [Closing Items](#), [Sell Items](#), and [Home/Search](#). The status bar at the bottom shows "Done" and "Local intranet".

Figure 18 Items for sale

When the user clicks on the buying items list on the account summary page, then the list of items that he is bidding for are shown. Here user mike is bidding for two items which are displayed. The current price of these items along with the seller's name and auction close date are displayed.

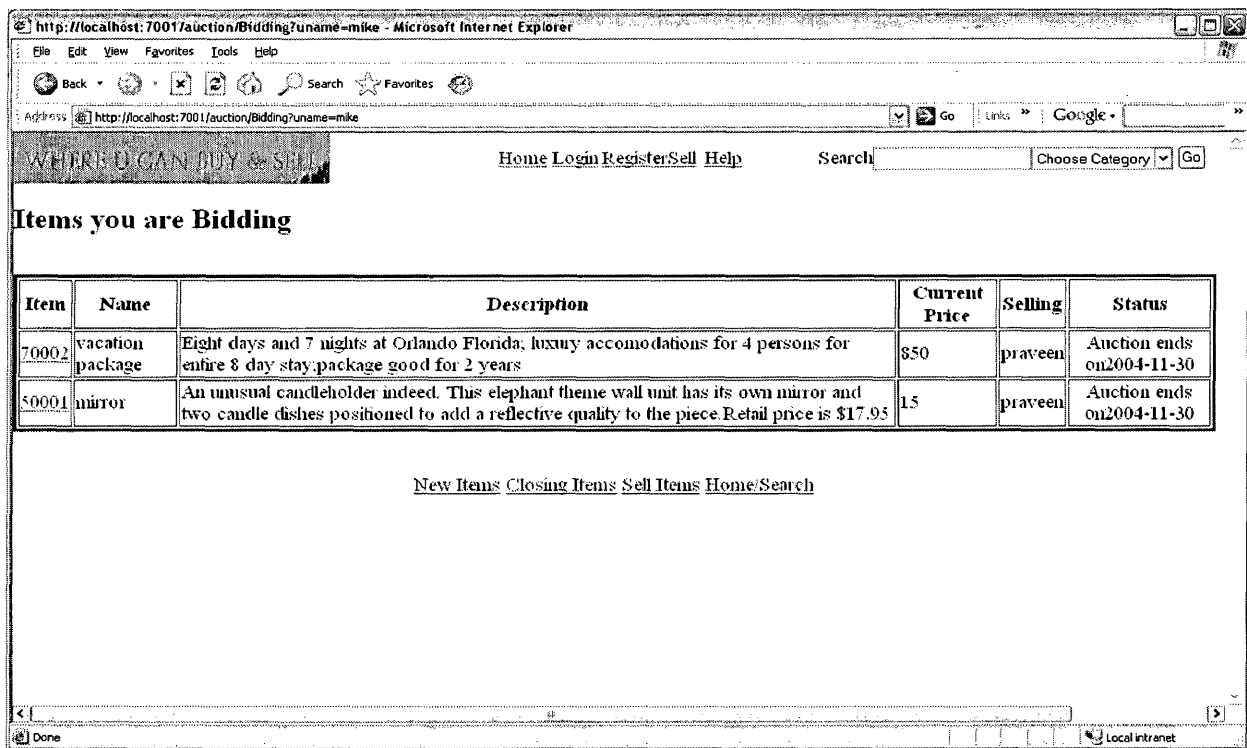


Figure 19 Items that have been bid

If the user plans to buy items, then he may use the search button located on the home page to search for items. If he finds any item that he is interested, he can click on the item number. Here user 'Mike' searched for an IPOD and found one on sale which he is interested in. He then clicks on IPOD's item number. This takes user mike to 'Item wise details' page.

Here the details of the IPOD such as bid start price, increment amount, bid start date, bid end date, seller are listed. If the user wishes to bid for the item then he should bid for atleast current bid amount and minimum increment. If not then the application throws an exception. The web page where user mike bids for IPOD is as shown below. If he bids for the IPOD then he will be the highest bidder.

The screenshot shows a web browser window with the address `http://localhost:7001/auction/ItemDetails?id=10005`. The page title is "Item Wise Details" and the item name is "ipod". A table displays the following details:

Start Price	400
Minimum Increment	15
Number of Bids	0
Auction Start Date	2004-11-27
Auction End Date	2004-11-30
Seller	bob
Highest Bidder	No Bidder Till now
Current Bid Price	400
Description	apple ipod never used brand new

Below the table, there is a section titled "Do you want to bid on this item?". It contains a form with the following fields:

- User Id:
- Password:
- Your Bid Amount:

A "Bid for Item" button is located below the form. At the bottom of the page, there are links: [New Items](#), [Closing Items](#), [Sell Items](#), and [Home](#). The browser status bar shows "Error on page." and "Local intranet".

Figure 20 Item details

User Mike can also post an item for auction. He can do so, by clicking on the sell item link. When he does click the link, he is redirected to post an item for auction page. At that page, Mike needs to enter the necessary information such as Item

name, description, bid starting price, bid increment, summary etc. In this shown example, Mike enters the details for a Tag Heuer watch. He mentions that its starting price is \$1200 and the bid increment is \$20. He also writes a small description of the watch.

The web page after Mike enters the description is as shown below.

NETAUCS Home Login Register Sell Help Search Choose Category Go

Post an Item for Auction

User Id: Mike

Password: ••••••••

Item Name: Tag Heuer Watch

Category: Jewellery

Summary: Tag Heuer Watch

Starting Price \$: 1200

Increment Price: 20

Description: Tag Heuer Watch , two years old, good condition, guarantee certificate available from manufacturer

Duration: 7

Post Item

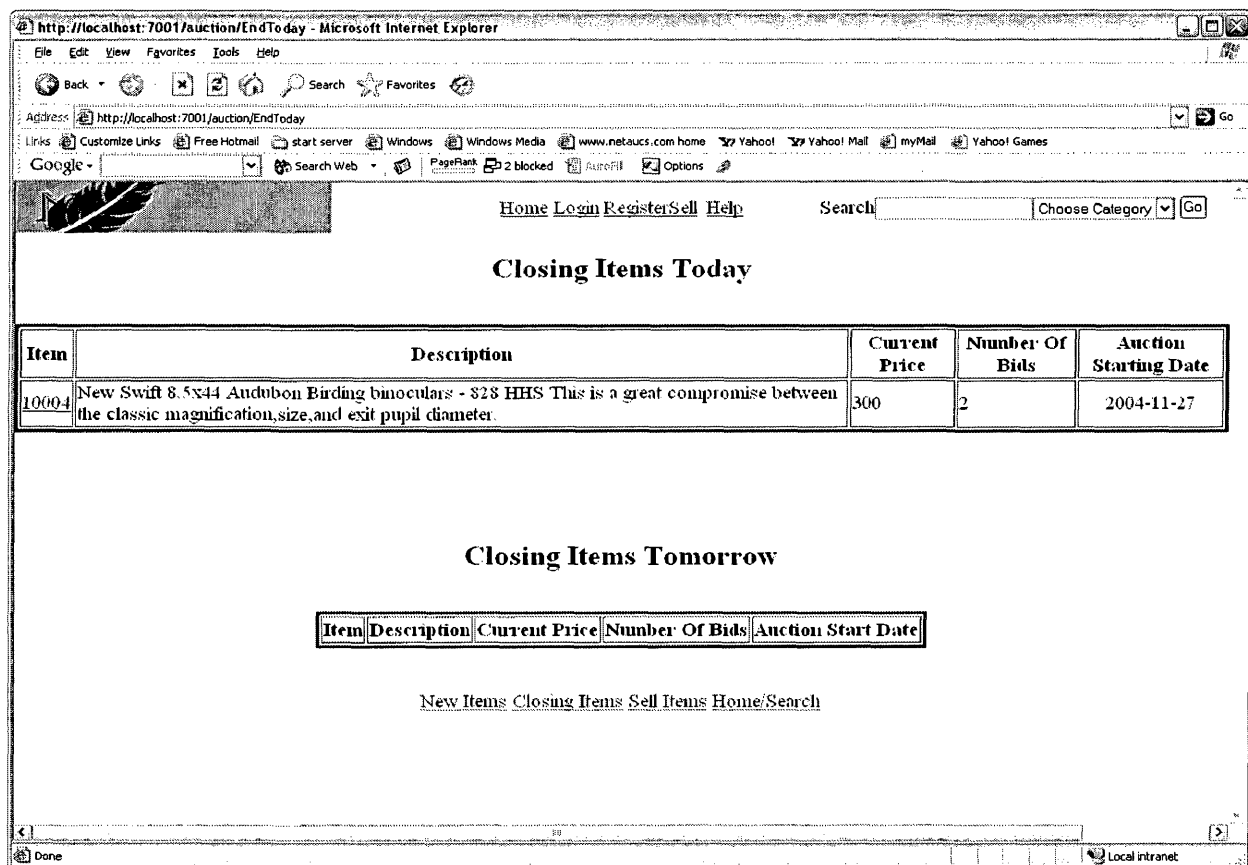
| New Items | Closing Items | Sell Items | Home |

Figure 21 Post an item for auction

Users can also view the new and closing items. New items are items that are added during that particular day and closing items are those items that are closing during that day. Having a closing items list helps bidders to know exactly

which items are closing today and for what price. So bidders can have a better chance to buy the items for that are being sold for less on that day. Here the closing items webpage is displayed. It shows items being closed today and tomorrow.

We can observe in the web page below that there is one item that is being closed today and there are no items that are being closed tomorrow.



Home Login Register Sell Help Search Choose Category Go

Closing Items Today

Item	Description	Current Price	Number Of Bids	Auction Starting Date
10004	New Swift 8.5x44 Audubon Birding binoculars - 328 HHS This is a great compromise between the classic magnification, size, and exit pupil diameter.	300	2	2004-11-27

Closing Items Tomorrow

Item	Description	Current Price	Number Of Bids	Auction Start Date
------	-------------	---------------	----------------	--------------------

New Items Closing Items Sell Items Home/Search

Figure 22 Closing items (today/tomorrow)

Web auction system has an administrator login. So that any items that are screened as offensive by the auction system can be eliminated. Administrator also has an additional feature to delete any category.

The web page below lists the items listed in all the categories. Administrator can delete the items at his discretion.

Item	Description	Current Price	Number Of Bids	Auction End Date
<input type="checkbox"/> 10005	apple ipod never used brand new	400	2	2004-11-30
<input type="checkbox"/> 30005	published by Grosset and Dunlap; binding damages but all pages are together	20	0	2004-11-30
<input type="checkbox"/> 30003	original monet imitation painting. if you are a monet fan or collector this is one piece of art you need to have	2000	0	2004-11-25
<input type="checkbox"/> 70001	samsoute hard shell vintage suitcase oxford gray color used for 2 years.	15	1	2004-11-30
<input type="checkbox"/> 40003	Real pearl ear rings made of 10k gold made in china used rarely still have the shine	75	0	2004-12-02
<input type="checkbox"/> 30006	a fine full size violin; bodysize 35x3; in good condition; polished and cleaned; no damages; back is maple and body ebony; comes with a bow	400	0	2004-11-30
<input type="checkbox"/> 50003	Wilson NFL JR. leather football; has a pittsburgsteelers logo on it; made of genuine leather	8	0	2004-12-02
<input type="checkbox"/> 30004	Original Sea Scape water painting : This is a fine watercolour depicting ships off the Rock of Gibraltar; The painting measures 6" x 14" and in its frame 11" x 19"	35	0	2004-12-04
<input type="checkbox"/> 10004	New Swift 8.5x44 Audubon Birding binoculars - 828 HHHS This is a great compromise between the classic magnification, size and exit pupil diameter.	300	2	2004-11-28
<input type="checkbox"/> 60001	Nokia Nokia 3200 phone head set free; Service compatibility: GSM 850/1800/1900 MHz	185	0	2004-12-04
<input type="checkbox"/> 70002	Eight days and 7 nights at Orlando Florida; luxury accomodations for 4 persons for entire 8 day stay; package good for 2 years	\$50	2	2004-11-30
<input type="checkbox"/> 10006	BRAND NEW Sony PlayStation 2 Game Console in a SEALED BOX, never opened. This is the newest PlayStation 2 with a compact design	175	0	2004-11-30
<input type="checkbox"/> 50004	Calvin Klein Crave 2.5 fl oz brand new; retail price 75 dollars; get it here for a lower price; great for christmas gift	45	0	2004-11-30
<input type="checkbox"/> 40004	New Kenneth Cole Ladies sub dial designer square watch; Water Resistant ; Manufactured in 2000; Condition: New	60	0	2004-12-02
<input type="checkbox"/> 60002	Motorola A388 GSM PDA Tri band World phone ; has all protective lenses ; only pda and stylus included-nothing else	350	0	2004-11-30
<input type="checkbox"/> 40005	Tag Heuer Watch , two years old, good condition, guarantee certificate available from manufacturer	1200	0	2004-12-04
<input type="checkbox"/> 20003	New machine used only for 3 months	2000	1	2004-11-25
<input type="checkbox"/> 50001	An unusual candleholder indeed. This elephant theme wall unit has its own mirror and two candle dishes positioned to add a reflective quality to the piece. Retail price is \$17.95	15	5	2004-11-30

Figure 23 List of items

Thus it can be observed that the web auction system has all the functionality for the buyer, seller and the administrator.

Chapter 6

Conclusion

With the advancement in computers and communication technologies, almost every house has a PC. Traditional buying and selling of merchandize took a new step due to this advancement. Before, people had to go to stores, retailer shops to buy stuff. To sell the stuff it used to be even more difficult as you can't find the appropriate buyer in your geographical region. With the advent of online auctioning, the process of selling and buying has become lot more easy.

Sellers can now market their new or used items to a larger number of prospective buyers. Buyers can also find new or used items for a lot cheaper price than the commercial market. Online auctioning has been particularly useful for prospective buyers looking for used merchandize. As the used merchandize is tough to find at stores and they are usually available at cheaper price.

Online auctioning has seen a tremendous growth in the recent years. With the increase in the number of issues, other issues had to be addressed. These issues include reliability, scalability of the application to handle large number of users, providing security for the online transactions. Also in the most of the online auctions, the selling merchandize is not screened to see it is appropriate to sell it online.

Most of the issues that are concerned with online auctioning are discussed in the project, appropriate solutions are also provided. Web auction system is then designed to address most of the issues that are concerned with today's online auctioning systems. Three-tier architecture is proposed for the web auction system. This design is considered as it gives a greater reliability, scalability and flexibility.

HTML, JSP are used to design the front end of the web auction system. WebLogic server is considered for the middle-tier server. WebLogic provides greater scalability, reliability and performance. The various methods that can be used to increase these mentioned factors are also discussed. Servlets and EJB's are used for the communication between the front end application and the middle-tier server.

Oracle is chosen as the database for implementation of the project. In case the number of users using the web auction system grow considerably, then Oracle handles the growth more efficiently. Additionally, packages such as stats pack, tuning pack available with Oracle help in improving the performance of the application.

The designed web auction system handles large number of user connections at any given time and is easily scalable. It also addresses lot of issues that are being faced by online-auction systems today. The implementation of web auction system shows an effective way of building a scalable, reliable and robust auctioning system.

Chapter 7

References

1. M. P. Wellman and P. R. Wurman, "Real time issues for Internet auctions"
First IEEE Workshop on Dependable and Real-Time E-Commerce
Systems June 1998.
2. Y. Amir, B. Awerbuch, and R. S. Borgstrom, The Java
Market: Transforming the Internet into a Metacomputer, Technical Report,
CNDS-98-1, Department of Computer Science, John Hopkins University,
1998.
3. www.ebay.com
4. Bill Roth, An Introduction to Enterprise JavaBeans Technology, SUN
technical articles
5. JavaServer Pages Technology, SUN whitepapers.
6. The Java Servlet API, SUN technical documentation
7. WebLogic Security Framework Technical Overview, BEA Whitepaper,
2004
8. Clustering in BEA WebLogic Server, BEA Whitepaper, 2004
9. JDBC Developer's Guide and Reference, Oracle Technical
Documentation
10. www.metalink.oracle.com
11. <http://www.nwfusion.com/intranet/0330obi.html>
12. <http://e-docs.bea.com>