ONLINE COMMODITY EXCHANGE SYSTEM

PROJECT WORK DONE AT

MICROSIGN TECHNOLOGIES Pvt. Ltd., COCHIN

PROJECT REPORT

SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENTS
FOR THE AWARD OF THE DEGREE OF
MASTER OF COMPUTER APPLICATIONS
OF BHARATHIAR UNIVERSITY, COIMBATORE.

SUBMITTED BY

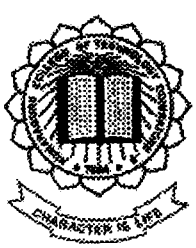
BENOY ANTONY
0038M1020

GUIDED BY

P-980

EXTERNAL GUIDE
Mr. Bejoy Varghese,
Sr. Programmer,
Microsign Technologies
Cochin.

INTERNAL GUIDE
Mr. A. Muthukumar, M.Phil.,
Assistant Professor,
Dept. of Computer Science & Engg.,
Kumaraguru College of Technology,
Coimbatore.



Department of Computer Science and Engineering
Kumaraguru College of Technology
Coimbatore – 641006

April 2003.

CERTIFICATE

Department of Computer Science and Engineering

KUMARAGURU COLLEGE OF TECHNOLOGY

COIMBATORE – 641 006.

CERTIFICATE

This is to certify that the project work titled

ONLINE COMMODITY EXCHANGE SYSTEM

Done by

BENOY ANTONY

0038M1020

Submitted in partial fulfillment of the requirements for the award of the degree of

Master of Computer Applications of Bharathiar University.

Professor and Head

Internal Guide

Submitted to University Examination held on 16/04/03.

Internal Examiner

External Examiner

16/4/03

Microsign Technologies Pvt. Ltd.
www.microsigntech.com

Regd. Off : Financial Square
Old Railway Station Road
Ernakulam North, Cochin - 682 018
Keralam, India
Tel : 91-484-2395017, 2395634
e-mail : info@microsigntech.com

Micros gn

MT/2003/246

March 30, 2003

CERTIFICATE

This is to certify that Mr. Benoy Antony, Final year MCA, Kumaraguru College of Technology has undergone project work on ONLINE COMMODITY EXCHANGE SYSTEM for Microsign Technologies Pvt.Ltd, Cochin, during the period December 2002 to March 2003.

The project has been completed successfully under my guidance and supervision and we are satisfied with his work.

Bejy Mathew

Managing Director

DECLARATION

DECLARATION

I hereby declare that this project entitled **ONLINE COMMODITY EXCHANGE SYSTEM,** submitted to the Bharathiyar University as the project of Master of Computer Application Degree, is a record of original work done by me under the supervision and guidance of **Mr. Bijoy varghese**, MCA, Senior Programmer, Microsign Technologies Pvt Ltd, Cochin and **Mr. A. Muthukumar**, M.Phil., Course Co-ordinator, MCA, Kumaraguru College Of Technology, Coimbatore. And, this project work has not formed the basis for the award of any Degree/Diploma/Associateship/Fellowship or similar title to any candidate of any university.

Place: Coimbatore

Date:

(Benoy Antony)

ACKNOWLEDGEMENT

ACKNOWLEDGEMENT

I express my profound respect and sincere gratitude towards our principal **Dr. K.K. Padmanaban Ph.D**, *Kumaraguru College of Technology, Coimbatore,* for granting me permission to do the project.

I express my sincere thanks to **Dr. S. Thangasamy Ph.D,***Head of the department, Computer Science and Engineering, Kumaraguru College of Technology,* for allowing me to take up the project at Microsign Technologies Pvt Ltd, Cochin.

I would like to express my deep and heartfelt thanks to my project guide **Mr. A. Muthukumar, M.Phil,** *Course Coordinator, MCA, Kumaraguru College Of Technology*, for his valuable advice and suggestions through out my project work.

I express my thanks to all staff members of *Computer Science Department, Kumaraguru College of Technology, Coimbatore,* for their suggestions.

My sincere thanks to **Mr.Bejy Mathew,** *Managing Director of Microsign technologies Pvt Ltd, Cochin,* for allowing me to do the project and **Mr. Rajan Varghese**, *Director and Project Leader of Microsign,* for his valuable guidance and suggestions to complete this project. I am deeply indebted to **Mr. Manu Varghese and Mr. Thomas George,** *Senior Programmers of Microsign,* for their valuable advice and encouragement.

I express my sincere and heartfelt thanks to my external guide, **Mr. Bijoy Varghese, MCA,** *senior programmer of microsign,* for giving me a guidance and support to complete the project.

I express my thanks to my parents, and my friends for their encouragement and support throughout my project.

SYNOPSIS

SYNOPSIS

Commodity exchange system is mainly developed for wholesalers and commodity brokers. Online trading of commodity is also possible. There are mainly two parts for this system. One is dealer's part and the other is central agency. Exchange in commodity exchange system means a central agency. The central agency controls all the activities of the user. There is an agreement between dealers and central agency. Online brokers and wholesale dealers can use this system. The authorized user can buy or sell the selected commodities without actual supply. This type trading will help the brokers. So they can collect the brokerage from their clients. In the case wholesale dealer's, they can find the current market place and market rate for their commodities. It will be great useful for them to find the market for their products. So they can earn more profit.

Commodity exchange provides you the fastest, most efficient order execution possible. Routes your order directly into the matching engine, you should expect to receive confirmations in seconds. However, beware of firms that offer electronic order placement that requires an individual to authorize your trade prior to execution. This process can delay your order by minutes if someone is not available to authorize those transactions. You can expect that your order exceution will be filled in much more efficient manner across markets. It will provide additional access to the markets.

In short, commodity exchange package strives to provide its dealers the ability to place, monitor and control all aspects of their trading from one interface that is available anywhere a dealer can log into. It has the ability to access the pending and executed orders quickly.

CONTENTS

TABLE OF CONTENTS

INTRODUCTION

INTRODUCTION

1.1 ABOUT THE ORGANIZATION

A company situated at the heart of kochi, the commercial capital of Kerala, for providing IT related services like Software Development, Service, Network Implementation and maintenance especially capital market related, having clients throughout India and a team of dedicated professionals. Microsign Technologies (P) Ltd is one of the leading IT firm in Kerala. Formed by a team of highly qualified and well-experienced IT professionals, in recognition of the growing need for extending information technology in India.

The firm has established its strong presence in the field of development in the state. Since its inception Quality rather than Quantity has been the benchmark by which all activities at Microsign Technologies has been carried out and evaluated. This, coupled with an approach to provide clients with Full and Maximum value for money rather than Cheapest Available Solution, has been the two guiding pillars of our operations. It is also engaged in support and service of leading capital market softwares in the state.

They are mainly focused on Capital Market. Their main product is a Capital Market BackOffice software-StockSERGE having around 50 clients throughout the country. Stock SERGE is multi-user back office software for stock brokers who deal with a large number of clients, branches, multiple stock exchanges with multiple segments etc. This is Enterprise software, which can integrate the transactions of a large enterprise with large number of branches spanned over different locations. . The system is adequate to keep accounting standards & stock exchange regulations in maintaining records. Moreover, our experience in the field and technical expertise are combined to deliver such a powerful product. Another developing one is Commodity Exchange. It will be suite for wholesale commodity dealers and commodity traders.

TECHNICAL STRENGTH

Operating System	: Windows 9x
	: Windows NT
	: Windows 2000
Databases	: Sybase SQL
	: MS SQL Server 2000
	: Oracle RDBMS
Client Server Tool	: PowerBuilder
	: Visual Basic 6.0, Visual C++
	: .NET
	: Magic
	: Top Speed (Clarion)
Web Tool	: ASP, JSP

LAN/WAN Management.

1.2 ABOUT THE PROJECT

Commodities – Commodity is any physical substance, such as food, grains and metals, which is interchangeable with another product of the same type and which investors buy or sell, usually through futures contracts. Commodity exchange is an organized market for almost all raw material or food that is produced in large quantities. Commodity exchanges are also called *boards of trade* or *commodity markets*. Commodities (goods) are nearly always traded without being seen, and it is unusual for the goods themselves to be physically exchanged. The price of the commodity is subject to supply and demand factors.

The primary goods (raw materials) sold on commodity exchanges include beef, cattle, cocoa, coffee, cotton, grains, lamb, rubber, Soya oil, and sugar. Metals traded include copper, gold, lead, silver, and tin. There are important international commodity exchanges all over the world especially in London, Australia, New Zealand, Singapore and Malaysia. The Chicago Board of Trade is the largest commodity exchange in the world.

There are three main types of people involved in commodity exchanges. The producer of the commodity is there to sell the goods. The user, or consumer needs to buy the goods. The investor, or speculator, buys the commodities, but never intends to take position of them. The investor operates in the market between the producer and the consumer, buying and selling to make a profit.

The Commodity Exchange System is developed as a client/server application. This system is meant for wholesaler dealers. The system facilitates online trading of commodities. Online brokers and wholesale dealers can use this system. The online trading facility helps brokers to manage trading between clients and they collect a brokerage from their clients. In the case of wholesale dealers they can find the current market place and market rate for the products they want to trade. It will help them to find the market for their products so that they can earn more profit.

In this system there are two modules.

1. Dealer's part.
2. Central agency.

Dealer's part:

The first module is for wholesale dealers or for online traders. Wholesale dealers can deal with their clients. In the case of online brokers or traders can place their order similar to stock exchange system. In stock exchange we are dealing with shares. Here we are dealing with commodities.

Each wholesale dealer can place his or her orders (buy order or sell order) online. Each dealer can select the commodity list that they want. This system will help him to get updated market rate in every second for each commodity. So price equalization will take place. If any other dealer is ready to accept this order they can make a contract based on price and quantity. Each dealer can get updated price information on his screen.

Central agency:

There is a central agency to handle all these activities. They will provide security to each client. They will provide separate login ID and password for each dealers. The fundamental principles that underlie commodity trading and the function of commodity exchanges are centuries old. Markets had already attained a degree of formalization in ancient civilization with in a fixed time and place for trading a marketplace, common barter and currency systems, and a practice of contracting for became the center of political and maritime power.

More About Commodity Exchange

There are two main methods of selling on commodity exchanges. These are known as the *spot market* and *futures market.*

Spot Market

Spot Market deals in commodities that are available for immediate delivery. The spot price is the current cash value of the commodity being sold. The spot market is also called the *non-contract market.*

Futures Market

Speculative trading is buying or selling futures contracts in hopes of making a profit from future price changes.

Futures Market, sometimes called the contact market, is where most commodity business is transacted. The buyer contracts to buy a certain quantity of a commodity at an agreed price for delivery at a specified later period-for example, six months later. If the buyer is the user or consumer, this method of the purchase gives him the security of knowing that if the spot price of the goods fluctuates after he has made his contract, he cannot lose money because he has bought at a fixed price. The producer selling forward also benefits if the price falls in the meantime. When buyers and sellers act to reduce the risk of losing money because of changing prices, they are said to be *hedging*. The other broad category of future trading is *speculative trading.*

International commodity agreements

International commodity agreements between users and producers have been arranged to try to reduce the sharp rises and falls in commodity prices. In 1977, the United Nations recommended that 18 commodities be brought under such agreements. Agreements on commodities usually set two prices: a *floor (lowest)* price and a *ceiling* (highest) price. If the price of a commodity falls below the floor price a central organization buys enough of the commodity to bring the price up to the required level. When the price rises above the ceiling, the commodity is released from stock onto the market to force the price down. Such agreements have not been successful, however. In some cases a major producer or user refuses to be included in the agreement. In other cases, either the producers or the consumers honor the agreement.

SYSTEM STUDY AND PROBLEM FORMATION

SYSTEM STUDY AND ANALYSIS

2.1 INTRODUCTION

System analysis is the process of gathering and interpreting facts, diagnosing problems and using the information to recommend improvements on the system. System analysis is a problem solving activity that requires intensive communication between the system users and system developers.

System analysis or study is an important phase of any system development process. The system is studied to the minutest detail and analyzed. The system analyst plays the role of an interrogator and dwells deep into the working of the present system. The system is viewed as a whole and the inputs to the system are identified. The outputs from the organization are traced through the various processing that the inputs phase through in the organization.

A detailed study of these processes must be made by various techniques like Interviews, Questionnaires etc. The data collected by these sources must be scrutinized to arrive to a conclusion. The conclusion is an understanding of how the system functions. This system is called the existing system. Now, the existing system is subjected to close study and the problem areas are identified. The designer now functions as a problem solver and tries to sort out the difficulties that the enterprise faces.

The solutions are given as a proposal. The proposal is then weighed with the existing system analytically and the best one is selected. The proposal is presented to the user for an endorsement by the user. The proposal is reviewed on user request and suitable changes are made. This is a loop that ends as soon as the user is satisfied with the proposal.

2.2 PRELIMINARY STUDY

In the preliminary study we found that the commodity market is greatly enhanced the ease and scope of trade in all types of commodities - food and foodstuffs, textiles, hides, metals, and lumber. However, the practices of spot trading and forward contracting were not adequate to meet the problems of the sudden shifts in supply, demand and consequently, price that had always vexed producers of basic commodities.

Preliminary Study is the process of gathering and interpreting facts, using the information for further studies on the system. Preliminary Study is a problem solving activity that requires intensive communication between the system users and system developers. It does various feasibility studies. In these studies a rough figure of the system activities can be obtained, from which the decisions about the strategies to be followed for Effective System Study and Analysis can be taken. Preliminary study also identifies the methods of data collection to be followed.

At the preliminary studies an initial picture about the system working was got. From the information got from this study, the Data Collection methods are identified. Even in the First Investigation about the system many existing drawbacks of the system could be identified, which helped a lot in the later stages of more rigorous study and analysis of the manual system.

2.3 METHODS OF DATA COLLECTION

Data Collection is the technique of getting information in the forms of feedback and various materials about the working of the current system. There are a lot of methods of data collection, in which identifying an appropriate method is a difficult task. The method depends on what type of data the analyst wants to extract from the Existing System. The main three methods used for collecting Data in the development of this project

- Review Of Written Documents
- On-Site Observation
- Interviews

When available, all documentation on data carriers is organized and evaluated include in the procedures manuals are the requirement of the system, which helps in determining in what extent they are met by the present system. Regarding existing forms, the analyst need to find out how they are filled out, how useful they are to the user, what changes need to be made, and how easy they are read.

At the initial stages of system study the main technique used was On-Site Observation where the working of the system and various transactions are observed and noted the all the relevant information. The analyst's role is that of an information seeker. One purpose of onsite observation is to get as close as possible to the real system being studied. Onsite observation is the most difficult fact finding technique. It requires intrusion into the user area. The analyst observes the physical layout of the current system, the location and movement of people and the workflow.

The onsite observation is directed towards describing and understanding events and behavior as they occur. This method, however, is less effective for learning about people's perceptions, feeling and motivations. The alternative is a personal interview. In this method, heavy reliance is place on the interviewees report for the information about the present system or experience. The quality of the response is judged in terms of its reliability and validity. Reliability of a response means that the information gathered is dependable enough to be used for making decisions about the system being studied.

2.4 EXISTING SYSTEM

Commodity markets:

The commodity markets existed primarily for cash transactions with immediate delivery. They greatly enhanced the ease and scope of trade in all types of commodities - food and foodstuffs, textiles, hides, metals, and lumber. However, the practices of spot trading and forward contracting were not adequate to meet the problems of the sudden shifts in supply, demand and consequently, price that had always vexed producers of basic commodities. Now there is no central agency to control the commodity market. There is no way to get updated market rate and find immediate market for commodities. Now there no option for sign a future contract online. Now it is very difficult to find a good market for commodities.

Commodity trading:

Now commodity trading is not popular. It is because we need more staffs to operate it. Also it will take time. Now day square trading is not possible for commodities and there is no way to get updated information online. Now in commodity trading we need the following people for trading.

Broker: licensed representative with the ability to advise and place orders.
Order Desk Clerk: individual assigned with the duties of receiving and placing customer and broker orders and directing those trades to the trading floor.
Trading Floor Clerk: individual that receives verbal orders or distributes printed copies of electronically transmitted orders and directs those orders to the trading pit.
Filling Broker: member of the exchange that handles the execution of orders.
Deck-Holder: Employee of the filling broker; accepts and maintains written and hand signaled orders for the filling broker; reports fill prices back to trading floor clerks.
Runner: individual employed by a clearing firm that delivers new orders and retrieves filled orders from the trading pits.

2.4.1 DRAWBACKS OF EXISTING SYSTEM

Supply/Demand Chaos:

It was common for farmers to bring grain and livestock to regional markets at a given time each year. They often found that the supply of meat and grain far exceeded the immediate short-term needs of packers and millers. These processors, seeing more than adequate supplies, would bid the lowest price. Often, the short-term demand could not absorb the glut of commodities at any price, however low, and goods were dumped in the street for lack of buyers.

Once the commodities reached the city, buyers were faced with the problem of inadequate storage space. Underdeveloped harbor facilities impeded the shipment of grain to eastern markets and the return movement of needed manufactured goods to western cities. The commodity exchanges, when they were organized, recognized the great need for improved transportation and storage and were a major force behind legislative efforts to improve rural roads, build inland waterways, and expand storage and harbor facilities. The exchanges made a particular contribution in leading the way to the establishment of accepted standards of grades and measures.

These efforts often bogged down in financial and legislative failure and the dismal marketing situation continued. The glut of commodities at harvest time was only part of the problem. Inevitably, there were years of crop failure and extreme shortages. Even in years of abundant yield, supplies were exhausted, prices soared, and people went hungry several months after the fall harvest and marketing of grain and livestock. Businessmen could be faced with bankruptcy because they lacked raw materials to keep their operations going. In this situation, the rural population, though having sufficient food for themselves, had crops they couldn't sell, and therefore, did not have the income to pay for needed manufactured products - tools, building materials, textiles. Due to this reasons inventory of these commodities is also very difficult. Here one another problem is also arise. There is no price equalization for commodities. The reason is unavailability of online information. Due to these reasons commodity market is very slow now.

Existing Order Placement in commodity trading:

The existing system should keep all the details manually. Thus the existing system has a lot of drawbacks. When a customer decides he wants to place an order the following steps will ensue:

1. Customer contacts his broker or order desk to place an order. The order is hand written then verbally repeated to the customer for verification and finally time-stamped.

2. The broker or order desk clerk then calls the exchange floor where a trading floor clerk receives the order, writes the information on an order ticket, repeats the order back for verification then time stamps the order.

3. At this point, the order is either hand delivered by a runner to the trading pit for execution or hand-signaled to a deck-holder.

4. The deck-holder now has your order. It is a market order so he informs the filling broker to buy one contract at the best offering price available.

5. Filling broker executes your order. He then must endorse the order with the exact price, time bracket, quantity, and include the information on the broker who took the opposite side of the trade. He must also inform the pit reporters of the trade.

6. Once the information is complete, the order ticket is handed back to the deck holder. The deck holders will either hand signal the fill price back to the clerk or wait for a runner to retrieve the filled ticket.

7. The fill information is now backing to the trading floor clerk. The order is time stamped and fill information is phoned back to the broker or order desk clerk.

8. The broker/order desk clerk now receives the fill information. The transaction price is written down and the order is time stamped. The customer is contacted, the fill price is reported and the order is timed - stamped one final time. The entire process can be as quick as 20 to 30 seconds in markets that are hand-signaled (flashed) as lengthy as 20 to 30 minutes where orders must be manually run into the pits. In summary, with traditional order placement there are eight interactions that are necessary.

2.5 REQUIREMENT ANALYSIS

Requirement analysis involves studying the current system to find out how it works and what improvements could be made. System studies result in an evaluation of how methods are working and whether adjustments are necessary or possible. A requirement is a feature that must be included in the new system. It may include ways of capturing or processing data, producing information, controlling some activities to support management. A thorough understanding of all the software requirements for the project is quite essential to the success of the software development effort. A program will disappoint the user if it is not analyzed, specified, designed or well coded.

Requirement planning is a software-engineering task that bridges the gap between the system level software allocation and software design. Requirement planning enables a system engineer to specify software functionality and performances indicate with other system elements and design constraints that the software should meet.

Goods can't flow without information: information allowing, forbidding, directing physical flows; information checking and confirming; information to provide proof and audit trails for taxes and billing. Because the information flows are so intimately connected with the physical movement of goods, it is sensible to deal with them as part of the same system.

Too often, too much of the wrong information moves too slowly to the wrong people. In many companies, this is the legacy of systems based on nineteenth-century technology, unchanged because by and large the right things happen in the end. Examining your information flows allows you to:

- Identify important information needs
- Discard flows that don't add value
- Speed and automate routine information transactions.

There are, of course, many more. In order to assess your current systems against your logistics needs you must establish for each information flow:

- Who originates and 'owns' the information
- Who needs to receive it
- What it is used for
- When it must be originated and received (and how much time it actually takes)
- What information must be transmitted
- How reliable is the information and the process?
- What it costs.

These headings don't address the need for management information systems and decision support systems, though such systems are a key part of any supply chain management process.

Commodity trading systems typically are computerized programs that signal members of the public when to buy and sell commodity futures and options contracts. Systems produce buys and sells signals based on mathematical formulas and are typically based on technical analysis of trading data (trading volume and prices), as opposed to fundamental analysis (analysis of economic factors such as supply and demand). Trading systems that are based on technical analysis attempt to predict future price movements based on historical prices, price relationships and price trends.

In deciding whether to purchase a particular trading system to trade commodity futures or options, members of the public should remember that no commodity trading system can guarantee profits. And, whether or not a trading system is used, commodity futures and options are typically high-risk endeavors.

There is always risk and there will always be losing trades - the key is to control those losses with smart money management. You'll have that. Become an OneDayTrader by starting your subscription today. Forget about day trading all day, forget about worrying constantly about buy and hold, and try a method that provides you with this opportunity for gain.

2.6 PROPOSED SYSTEM

This system is mainly developed for wholesalers. Online trading of commodity is also possible. Exchange means a central agency. Online brokers and wholesale dealers can use this system. The online trading facility will help brokers. So they can collect brokerage from their clients. In the case of wholesale dealers they can find the current market place and market rate. It will be great useful for them to find the market for their products. So they can earn more profit.

The proposed system is a user-friendly system, for providing payments to the clients. The system should support editorial staffs to handle the payments for their clients and companies concerned. Understand how online trading can provide faster access to today's futures markets. Each wholesale dealer can place his or her orders (buy order or sell order) online. Each dealer can select the commodity list that they want. This system will help him to get updated market rate in every second for each commodity. So price equalization will take place. If any other dealer is ready to accept this order they can make a contract based on price and quantity. Each dealer can get updated price information on his screen.

General Features:

This system can route orders to all of the following exchanges:

* Ability to place futures, options, futures spreads and option spreads in all markets supported by commodity exchange system.
* Quick access to filled, working, cancelled, rejected and parked orders placed into the system
* Access to net day positions by commodity.
* Monitor your account in real time with instant access to your money balances and market positions marked to the market.
* Daily access to your cleared trades whether they were placed through the system or done over the phone to any trade desk.
* Create a customized future order ticket, reducing the time it takes to build a new ticket.

* Enough flexibility to deal with the largest (and smallest) items, which will need storage in the numbers that will be needed.

- Put into known places in known order and retrieved quickly and in the right quantity rotated properly (for example, first-in, first-out)

Traditional activities have changed. These benefits can make the business behavior flexible. In traditional business behavior we got recommendation or suggestions from other people. The work in the traditional commerce is difficult to finish because they don't have opportunity to influence individual customer and make one-to-one marketing.

What would this type of convenience be worth?

- A powerful trading methodology where you can now take full advantage of those large daily swings without having to buy and sell throughout the day
- No running up large commissions, no worries about overnight exposure, especially given the current market conditions
- The ability to go long and short depending upon current market conditions
- Trading for larger profits (profits in just one trading day) that consistently add up over time
- Having more winning trades than losing trades by a comfortable margin
- Knowing what your risk is in any trade

Commodity exchange provides you the fastest, most efficient order execution possible. Routes your order directly into the matching engine, you should expect to receive confirmations in seconds. However, beware of firms that offer electronic order placement that requires an individual to authorize your trade prior to execution. This process can delay your order by minutes if someone is not available to authorize those transactions.

In short, Commodity Exchange package strives to provide its dealers the ability to place, monitor and control all aspects of their trading from one interface that is available anywhere a dealer can log into. It has the ability to quickly access filled, working, cancelled and rejected orders.

2.7 SELECTION OF SOFTWARE

Software selection is an important work in a Project Development Cycle. Software must be selected in accordance with the application and the latest technology available. The development software used is Visual Basic.NET with SQL Server 2000 as the database.

2.7.1 INTRODUCTION TO .NET

Microsoft .NET Framework Software Development Kit (SDK)

The SDK contains a wealth of resources, such as an extensive collection of tools and samples, designed to help you build powerful applications and services based on .NET Framework technology. For a brief introduction to the .NET Framework architecture and its key components, which include the common language runtime and the .NET Framework class library, see Overview of the .NET Framework.

The .NET Framework SDK also includes documentation that provides a wide range of instructive and practical information. In addition to a comprehensive class library reference, the documentation offers conceptual overviews, step-by-step procedures, information about the set of tools contained in the SDK, and tutorials that demonstrate how to create specific types of applications. As an aid to help you locate the information that interests you most, the following sections list the main areas of the documentation

Overview

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

- To provide a consistent object-oriented programming environment whether object code is stored and executed locally, executed locally but Internet-distributed, or executed remotely.

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

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

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

For example, ASP.NET hosts the runtime to provide a scalable, server-side environment for managed code. ASP.NET works directly with the runtime to enable Web Forms applications and XML Web services, both of which are discussed later in this topic.

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

VISUAL BASIC.NET

Visual Basic has been updated to include many new and improved language features that make it a powerful object-oriented programming language. These features include inheritance, interfaces, and overloading, among others. Visual Basic also now supports structured exception handling, and custom attributes. In addition,Visual Basic supports multithreading. Multithreading is the ability to assign individual tasks to separate processing threads. For more information, see what's New in the Visual Basic Language and Introduction to Objects in Visual Basic.

Introduction to objects in VB

Almost everything you do in Visual Basic is associated with objects. If you are new to object-oriented programming, the following terms and concepts will help you get started.

Classes and Objects

The words "class" and "object" are used so much in object-oriented programming that it is easy to get the terms mixed up. Generally speaking, a *class* is an abstract representation of something, whereas an *object* is a usable example of the thing the class represents. The one exception to this rule is shared class members, which are usable in both instances of a class and object variables declared as the type of the class.

What is new in VISUAL BASIC.NET

Visual Basic .NET has many new and improved language features — such as inheritance, interfaces, and overloading — that make it a powerful object-oriented programming language. As a Visual Basic developer, you can now create multithreaded, scalable applications using explicit multithreading. Other new language features in Visual Basic .NET include structured exception handling, custom attributes, and common language specification (CLS) compliance.

The CLS is a set of rules that standardizes such things as data types and how objects are exposed and interoperate. Visual Basic .NET adds several features that take advantage of the CLS. Any CLS-compliant language can use the classes, objects, and components you create in Visual Basic .NET. And you, as a Visual Basic user, can access classes, components, and objects from other CLS-compliant programming languages without worrying about language-specific differences such as data types. CLS features used by Visual Basic .NET programs include assemblies, namespaces, and attributes.

Visual Basic .NET supports many new or improved object-oriented language features such as inheritance, overloading, the *Overrides* keyword, interfaces, shared members, and constructors.

Also included are structured exception handling, delegates, and several new data types.

Inheritance

Visual Basic .NET supports *inheritance* by allowing you to define classes that serve as the basis for derived classes. Derived classes inherit and can extend the properties and methods of the base class. They can also override inherited methods with new implementations. All classes created with Visual Basic .NET are inheritable by default. Because the forms you design are really classes, you can use inheritance to define new forms based on existing ones. For details, see Inheritance.

Exception Handling

Visual Basic .NET supports *structured exception handling*, using an enhanced version of *the Try...Catch...Finally* syntax supported by other languages such as C++.

Structured exception handling combines a modern control structure (similar to Select Case or While) with exceptions, protected blocks of code, and filters. Structured exception handling makes it easy to create and maintain programs with robust, comprehensive error handlers. For details, see Exception Handling.

Overloading

Overloading is the ability to define properties, methods, or procedures that have the same name but use different data types. Overloaded procedures allow you to provide as many implementations as necessary to handle different kinds of data, while giving the appearance of a single, versatile procedure. For details, see Overloaded Properties and Methods.

Overriding Properties and Methods

The *Overrides* keyword allows derived objects to override characteristics inherited from parent objects. Overridden members have the same arguments as the members inherited from the base class, but different implementations. A member's new implementation can call the original implementation in the parent class by preceding the member name with *MyBase*. For details, see Overriding Properties and Methods.

Constructors and Destructors

Constructors are procedures that control initialization of new instances of a class. Conversely, *destructors* are methods that free system resources when a class leaves scope or is set to *nothing*. Visual Basic .NET supports constructors and destructors using the *Sub New* and *Sub Finalize* procedures. For details, see Object Lifetime: How Objects are Created and Destroyed.

Data Types

Visual Basic .NET introduces three new data types. The *Char* data type is an unsigned 16-bit quantity used to store Unicode characters. It is equivalent to the .NET Framework *System.Char* data type. The *Short* data type, a signed 16-bit integer, was named *Integer* in earlier versions of Visual Basic. The *Decimal* data type is a 96-bit signed integer scaled by a variable power of 10. In earlier versions of Visual Basic, it was available only within a *Variant*. For details, see Data Types.

Interfaces

Interfaces describe the properties and methods of classes, but unlike classes, do not provide implementations. The *Interface* statement allows you to declare interfaces, while the *Implements* statement lets you write code that puts the items described in the interface into practice. For details, see Interfaces in Visual Basic .NET.

Delegates

Delegates — objects that can call the methods of objects on your behalf — are sometimes described as type-safe, object-oriented function pointers. You can use delegates to let procedures specify an event handler method that runs when an event occurs. You can also use delegates with multithreaded applications. For details, see Delegates and the AddressOf Operator.

Shared Members

Shared members are properties, procedures, and fields that are shared by all instances of a class. Shared data members are useful when multiple objects need to use information that is common to all. Shared class methods can be used without first creating an object from a class. For details, see Shared Members.

References

References allow you to use objects defined in other assemblies. In Visual Basic .NET, references point to assemblies instead of type libraries. For details, see References and the Imports Statement.

Namespaces

Namespaces prevent naming conflicts by organizing classes, interfaces, and methods into hierarchies. For details, see Namespaces.

Assemblies

Assemblies replace and extend the capabilities of type libraries by, describing all the required files for a particular component or application. An assembly can contain one or more namespaces. For details, see Assemblies.

Attributes

Attributes enable you to provide additional information about program elements. For example, you can use an attribute to specify which methods in a class should be exposed when the class is used as a XML Web service. For details, see Attributes.

Multithreading

Visual Basic .NET allows you to write applications that can perform multiple tasks independently. A task that has the potential of holding up other tasks can execute on a separate thread, a process known as *multithreading*. By causing complicated tasks to run on threads that are separate from your user interface, multithreading makes your applications more responsive to user input. For details, see Multithreaded Applications.

CRYSTAL REPORTS

Crystal Reports for Visual Studio .NET provides two report Viewers with which you can view reports in applications: use the Web Forms Viewer for Web applications and the Windows Forms Viewer for Windows applications. All the reports that are generated in the proposed system are crystal reports.

Both Viewers can host Report Web Services.

Web form Viewers

- Host a Crystal Report in HTML format on the Web.
- Dynamically update the report they hosting.
- Interact with controls within a Web application.

Windows form Viewers

- Host and view a Crystal Report in a Windows application.
- Dynamically update the report they hosting.
- Interact with controls within a Windows application

2.7.2 ADVANTAGES OF VISUAL PROGRAMMING

- The programmer need not write code to display the required components.
- The visual programming environment displays al list of available components.
- The programmer picks up the required components from the list.
- The component can be moved, resized and even deleted, if required.
- There are no restrictions in the number of controls.
- The inter face components have some code built into them.

2.8 RELATIONAL DATABASE MANAGEMENT SYSTEM

The RDBMS is responsible for the following functions

- Managing the relationship between the data in the database.
- Ensured that data is stored correctly that the rules defining the relationship between data are violated.
- Recovering all data to a point of known consistency in the event of a system failure.

2.8.1 SQL SERVER 2000

Microsoft SQL Server 2000 features include:

- **Internet Integration.**

  The SQL Server 2000 database engine includes integrated XML support. It also has the scalability, availability, and security features required to operate as the data storage component of the largest Web sites. The SQL Server 2000 programming model is integrated with the Windows DNA architecture for developing Web

applications, and SQL Server 2000 supports features such as English Query and the Microsoft Search Service to incorporate user-friendly queries and powerful search capabilities in Web applications.

- **Scalability and Availability.**

  The same database engine can be used across platforms ranging from laptop computers running Microsoft Windows® 98 through large, multiprocessor servers running Microsoft Windows 2000 Data Center Edition. SQL Server 2000 Enterprise Edition supports features such as federated servers, indexed views, and large memory support that allow it to scale to the performance levels required by the largest Web sites.

- **Enterprise-Level Database Features.**

  The SQL Server 2000 relational database engine supports the features required to support demanding data processing environments. The database engine protects data integrity while minimizing the overhead of managing thousands of users concurrently modifying the database. SQL Server 2000 distributed queries allow you to reference data from multiple sources as if it were a part of a SQL Server 2000 database, while at the same time, the distributed transaction support protects the integrity of any updates of the distributed data. Replication allows you to also maintain multiple copies of data, while ensuring that the separate copies remain synchronized. You can replicate a set of data to multiple, mobile, disconnected users, have them work autonomously, and then merge their modifications back to the publisher.

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

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

sites. SQL Server 2000 also supports a standards-based programming model integrated with the Windows DNA, making the use of SQL Server databases and data warehouses a seamless part of building powerful and scalable systems. These features allow you to rapidly deliver SQL Server applications that customers can implement with a minimum of installation and administrative overhead.

CONNECTING TO A SQL SERVER DATA SOURCE

ADO can use any OLE DB provider to establish a connection. The provider is specified through the Provider property of the Connection object. Microsoft SQL Server 2000 applications use SQLOLEDB to connect to an instance of SQL Server, although existing applications can also use MSDASQL to maintain backward compatibility. Using the Execute method of the Connection object is one way to execute an SQL statement against a SQL Server data source.

The **Connection** object allows you to:

- Configure a connection.
- Establish and terminate sessions with data sources.
- Identify an OLE DB provider.
- Execute a query.
- Manage transactions on the open connection.
- Choose a cursor library available to the data provider.

RELATIONAL DATABASE ENHANCEMENTS

Microsoft SQL server 2000 introduces several server improvements and new features:

XML Support

The relational database engine can return data as Extensible Markup Language (XML) documents. Additionally, XML can also be used to insert, update, and delete values in the database.

Federated Database Server

SQL Server 2000 supports enhancements to distributed partitioned views that allow you to partition tables horizontally across multiple servers. This allows you to scale out one database server to a group of database servers that cooperate to provide the same performance levels as a cluster of database servers.

Indexed views

Indexed views can significantly improve the performance of an application where queries frequently perform certain joins or aggregations. An indexed view allows indexes to be created on views, where the result set of the view is stored and indexed in the database. Existing applications do not need to be modified to take advantage of the performance improvements with indexed views.

New data types

SQL Server 2000 introduces three new data types. *bigint* is an 8-byte integer type. *sql_variant* is a type that allows the storage of data values of different data types. *Table* is a type that allows applications to store results temporarily for later use. It is supported for variables, and as the return type for user-defined functions.

Distributed Query Enhancements

SQL Server 2000 introduces a new OPENDATASOURCE function, which you can use to specify ad hoc connection information in a distributed query. SQL Server 2000 also specifies methods that OLE DB providers can use to report the level of SQL syntax supported by the provider and statistics on the distribution of key values in the data source. The distributed query optimizer can then use this information to reduce the amount of data that has to be sent from the OLE DB data source. SQL Server 2000 delegates more SQL operations to OLE DB data sources than earlier versions of SQL Server. Distributed queries also support the other functions introduced in SQL Server 2000, such as multiple instances, mixing columns with different collations in result sets, and the new *bigint* and *sql_variant* data types.

Updatable Distributed Partitioned Views

SQL Server 2000 introduces enhancements to distributed partitioned views. You can partition tables horizontally across several servers, and define a distributed partitioned view on each member server that makes it appear as if a full copy of the original table is stored on each server. Groups of servers running SQL Server that cooperate in this type of partitioning are called federations of servers. A database federation built using SQL Server 2000 databases is capable of supporting the processing requirements of the largest Web sites or enterprise-level databases.

Backup and Restore Enhancements

SQL Server 2000 introduces a new, more easily understood model for specifying backup and restore options. The new model makes it clearer that you are balancing increased or decreased exposure to losing work against the performance and log space requirements of different plans. SQL Server 2000 introduces support for recovery to specific points of work using named log marks in the transaction log, and the ability to do partial database restores

2.8.2 Client/Server Architecture:

Distributed processing uses more than one processor to divide the processing for a set of related jobs. Distributed processing reduces the processing load on a single processor by allowing different processors to concentrate on a subset of related tasks, thus improving the performance and capabilities of the system as a whole. An SQL Server DB system can easily take advantage of distributed processing by using its client/server architecture. In this architecture, the database system is divided into two parts: a front-end or a client portion and a back-end or a server portion.

The Client

The client portion is the front-end database application and interacts with a user through the keyboard, display, and pointing device such as a mouse. The client portion has no data access responsibilities; it concentrates on requesting, processing, and presenting data managed by the server portion. The client workstation can be optimized for its job. For example, it might not need large disk capacity or it might benefit from graphic capabilities.

The Server

The server portion runs SQL Server software and handles the functions required for concurrent, shared data access. The server portion receives and processes the SQL and PL/SQL statements that originate from client applications. The computer that manages the server portion can be optimized for its duties. For example, it can have large disk capacity and fast processors.

HARDWARE & SOFTWARE SPECIFICATION

HARDWARE & SOFTWARE SPECIFICATION

3.1 HARDWARE REQUIREMENTS

To run in standalone mode, processor should be Pentium (any series) or AMD K6 or above. Also a RAM of minimum 32MB and a Hard Disk with minimum 50MB of free space at any point of time are required.

To run in networked mode, the server system should be with a processor of Pentium II or higher or AMD K6 or higher and minimum of 64MB RAM, 1GB of free space in the Hard Disk at any point of time. The workstation may run on terminals with Pentium (any series), AMD K5 or higher. The minimum memory requirement is 32MB. There should be at least 500MB of free space in the local Hard Disk.Network card should be of at least 10/100 kbps of transfer rate.

3.2 SOFTWARE REQUIREMENTS

In the networked mode, the server should run on Windows 2000/Windows XP and the clients on Windows 98/ Windows 2000.

The front end is Visual Basic.NET and back end is SQL Server 2000. To work with .NET, the operating system should be windows 2000 server/windows XP/windows 98. And we need high-speed processors and 128MB of memory to work efficiently.

Recommendations for usage:

- Uninterrupted power supply should be ensured for the server to prevent corruption of data. It is better if all the systems in the network are backed up with UPS.
- In the server part, there are provisions to change all the settings of the application. So, always ensure that a designated & experienced hand operates it.
- Back up the data on a daily basis, to prevent data loss.
- While running Commodity Exchange software, avoid running options applications that consume more memory.

SYSTEM DESIGN AND DEVELOPMENT

SYSTEM DESIGING AND DEVELOPMENT

4.1 SYSTEM DESIGN

Design is the first step in the development phase for every engineered product or system. Computer Software desiring, like engineering design approaches in other disciplines, changes continuously as new methods, better analysis and broader understanding evolve.

System design involves translating information requirements and conceptual design into technical specification and general flow of processing. After the user requirements are identified, related information is gathered to verify the problem and after evaluating the existing system, a new system is proposed. The proposed system consists of various tables, their maintenance and report generation.

It has been assured that the system will have the functions and promises of the proposed system. In the design the various techniques are used to present a simple efficient system. The design phase focuses on the detailed implementation of the system recommended in the feasibility study. The design phase is a transition from a user-oriented document to a document oriented to the programmers or database personnel. Systems design goes through two phases of development:

- Logical Design
- Physical Design

The data flow diagram shows the logical flow of a system and defines the boundaries of the system. For a candidate system, it describes the inputs (source), outputs (destination), database (files) and procedures (data flow), all in a format that meet the user's requirements. In logical design we specifies the users needs at a level of detail that virtually determines the information flow into and out of the system and the required data resources.

Following logical design is physical design. This produces the working system by defining the design specifications that tell programmers exactly what the candidate system must do. In turn we write the necessary programs or modify the software packages that accept input from the user. Then perform the necessary operations through the existing file and produce reports.

4.2 MODULE DESIGN

The project titled "Commodity Exchange System" is mainly divided into two modules. These are the following

- **Wholesale dealers or brokers part** (client module)
- **Central agency** (server module)

The Wholesale dealers or brokers part (Client Module) **has the following parts.**

Each user has separate login id provided by the central agency. User can placing buy order and sell order. They should have update or cancel the pending orders. There are also facilities to change password and view pending orders and executed orders. There is also a facility to see the best orders, which others place and get confirmation from the central agency at the time execution of his orders.

The Central agency (Server Module) **has the following parts.**

Logging part is used for validating the user and creation of different users. When a user login into a client then the server monitors the login details of that particular user. Resource accessing part helps a client to access a resource from another client. The server monitors the accessing process when a user access orders from any client in the network. Central agency has the full control to delete a user, publishing the commodity list. Here server will act as a matching engine of the orders place by the clients.

4.3 DATA FLOW DIAGRAMS

Data Flow Diagram is used to define the flow of the system and its resources such as information. Data Flow Diagrams are a way of expressing system requirements in a graphical manner. Data Flow Diagrams represent one of the most ingenious tools used for structured analysis. A Data Flow Diagram or DFD, as it is shortly called is also known as a bubble chart. It has the purpose of clarifying system requirements and identifying major transformations that will become programs in System design. It is the major starting point in the design phase that functionally decomposes the requirement specifications down to the lowest level of detail.

A DFD consists of a series of bubbles joined by lines. The bubble represents data transformation and lines represent flow in the system. In the normal convention, a DFD has four major symbols.

- Square, which defines source or destination of data
- Arrow, which shows data flow.
- Circle, which represents a process that transforms incoming data into outgoing flow.
- Open rectangle, which shows data store

The DFD at the simplest level is referred in simple words a "CONTEXT ANALYSIS DIAGRAM'. These are expanded by level, each explaining its process in detail. Processes are numbered for easy identification and are normally labeled in block letters. Each data flow is labeled for easy understanding.

4.4 DATABASE DESIGN

The general theme behind a database is to handle information as an integrated whole. There is none of the artificiality that is normally embedded in separate files or applications. A database is collection of interrelated data stored with minimum redundancy to serve many users quickly and efficiently. The general objective is to make information access easy, quick, inexpensive and flexible for the user.

In a database environment, common data are available which several authorized users can use. The concept behind a database is an integrated collection of data and provides a centralized access to the data from the program. It makes possible to treat data as a separate resource. While designing database, several objectives must be considered:

- Controlled redundancy
- Data Independence
- More information at low cost
- Accuracy and Integrity
- Recovery from failure
- Privacy and security
- Performance

4.4.1 DATABASE DESCRIPTION

The tables used for developing the project are given below:

COMMODITY MASTER TABLE:

The trading of commodity takes place only for the commodities inside this table. This is a master table. Each wholesaler can sell or buy these commodities. The changes in this table are restricted from the dealers or from the traders. The central agency will decide the list of commodities. Here comcode (commodity code) is the primary key. Here one trigger is attached with this table. Because of that at the time of cancellation of each row from this table will cancel all orders from buy order and sell order tables which have the same commodity code.

LOGIN MASTER TABLE:

This table is used for creating user id for wholesalers. This is a master table. Each wholesaler has separate login id and password. The password can change by the wholesale dealers. The central agency will provide user id. The remaining changes in this table are restricted from the dealers. Here user id is the primary key. Here one trigger is attached with this table. Because of that at the time of cancellation of each row from this table will cancel all orders from buy order and sell order tables which have the same user id.

CENTER MASTER TABLE:

The trading of commodity takes place only in these centers. This is a master table. Each wholesaler can sell or buy in these centers. The central agency will decide centers. The remaining changes in this table are restricted from the dealers. Here place is the primary key.

BUYORDER TABLE:

This table is used for enter the buy order from the wholesalers. Best buy order for each commodity will show in the main screen. The selection of best buy order can select from this table. At the end of the day the central agency cancel all these buy order. Here ordno (order id) is the primary key. This table will refer the commodity master table for commodity code entry and it will refer user table for user id entry and it will refer center table for place entry.

SELLORDER TABLE:

This table is used for enter the sell order from the wholesalers. Best sell order for each commodity will show in the main screen. The selection of best sell order can select from this table. At the end of the day the central agency cancel all these sell orders. Here sordno (order id) is the primary key. This table will refer the commodity master table for commodity code entry and it will refer user table for user id entry and it will refer center table for place entry.

MATCH TABLE:

This table is used for enter the matching entries. From this table we get the executed orders. Here match no (match number) is the primary key. This table is the main part of the matching engine. This table will refer the commodity master table for commodity code entry and it will refer user table for sell user id and buy user id entry and it will refer center table for sell place and buy place entry. There are two fields msgstatus1 and msgstatus2, which help to send the messages to the clients.

TEMPREPORT TABLE:

This table is used for enter the details of server side report. From this table we will get the user vice details of executed orders. This is a temporary table. It helps to view the buy and sell orders separately.

CURRENTUSERS TABLE:

This table is used to get details of the current users of the system. The field status helps to disable the client from using the system at run time.

COMMODITY.TXT:

This text file is used to store the names of commodities, which the wholesaler used to deal. The wholesaler can change the details of this file during run time. This file is situated in client side only.

4.5 CODE DESIGN

The purpose of code is to facilitate the identification and retrieval of items of information. A code is an ordered collection of symbols designed to provide unique identification of an entity or an attribute. Codes are built with mutually exclusive features. Codes in all cases specify objects physical or on performance characteristics. They are used to give operational distractions and other information. Codes also show interrelationship among different items. Codes are used for identifying, accessing, sorting and matching records. The codes ensure that only one value of code with a single meaning is correctly applied to give entity or attribute as described in various ways. Codes can also be designed in a manner easily understood and applied by the user.

4.6 VALIDATION RULES AND CHECKS

The major decisions of a validation stage are concerned with handling errors and distribution of data. The data relevant to the system enters it through a set of validation procedures. Often they are caused for by a generalized input validation package tailored for the needs of a particular system.

There are various ways of handling errors open to the designer, which includes rejection of the item of input or processing the next item, writing error record and signaling the appropriate message to the user. Error procedures must be specified in detail showing decisions, actions and exceptions.

The program developed is checked thoroughly for errors by testing it with data and the errors that are raised during this is clarified. The program may halt during an unpredictable error.

The major constraints used in the current system falls under the fore said categories. One of the major constraints imposed is the NOT NULL constraint. When such a constraint is imposed on a column or set of columns in a table, it will not accept null values. Another constraint is the UNIQUE constraint. Its main job is to prevent duplication of values within rows of a specified column or set of columns in a table.

PRIMARY KEY constraint avoids duplication of rows and does not allow NULL values when enforced in a column or set of columns. As a result, it is used to identify a row. Referential integrity constraints are used to establish Parent-Child or a Master-Child relationship between two tables having common columns. To implement this, we should define the column in the parent table as Primary key and the same column in the child table as a Foreign Key in the child table referring to the corresponding parent entry

4.7 SECURITY ASPECTS

A computer system is secure if neither its ability to attain its objectives nor its availability to survive can be adversely affected by an unwanted event. A computer-based security is a combination of many assets or resources designed to perform some function or to provide service.

In this system, several measures have been taken to provide some security. Loss of confidentiality is reduced to a great extent. The facility to impose strict authorization is completely vested in the hands of the central agency. They have the full authority to add or delete user to and from the system respectively. Only valid users can enter the system. They have to provide a valid username and password, to prove that they are valid users. If any one of this is wrong, access is denied to the system. The client can do change of password for the security purpose.

4.8 INPUT/OUT FORM DESIGNS

4.8.1 INPUT DESIGN

Input design is the process of converting user-oriented description of the inputs to a computer-based business system into a programmer-oriented specification. Input data are collected and organized into a group of similar data. Inaccurate input data is the most common cause of data processing errors. If poor input design permits bad data to enter a computer system, the outputs produced are of little value. The input design process initiated

in the study Phase, as a part of feasibility study. Effective input design minimizes errors made by data entry operators. The goal of designing input data is to make data entry as easy, logical and free from errors as possible.

In addition to the general form considerations such as collecting only required data, grouping similar or related data, input design requires consideration of the needs of the data entry operator. In entering data, operators needs to know the following:

- The allocated space for each field. The field length must known to the data entry operator, so that the data entered will not exceed the allocated space and/or numeric data may be right justified where appropriate.
- Field sequence of fields must match the sequence of the fields on the source document. The data entry operator must able to scan the source document in a logical sequence.
- The format must be identified to the data entry operator. That is, a data field that is to be entered in an edited format must be documented.

Wholesale dealers or brokers input screens:

There is Login screen, which handle login process of a particular user. In this screen user enters the user name and password as inputs. Then the server validates these inputs and the result is given to the user. There should be forms for change the password.

There is buy order and sell order screens. This helps the wholesale dealers to place order. They can place buy order and sell order. In this form system ask for the price, place, quantity, minimum quantity and the retailer code for each commodity that the user wish to sell or buy.

There is also a screen for updating or cancel pending orders. In input design, data is accepted for computer processing and input into the system is done through valid users.

Central agencies input screens:

In server side we have registration screens like register a user, here the system ask for the details of the user like name, address, password and user ID. For registering a commodity it will ask for the details of that commodity like commodity code, commodity description and group of that commodity.

There are also screens for deleting a user and deleting a commodity. And also have screens for viewing the executed orders and pending orders.

4.8.2 OUTPUT DESIGNS

Output means reports. These reports are generated from stored or calculated values. Reports are displayed either screen preview or printed form. Various reports can be generated. Outputs are the most important and direct source of information to the management. Intelligent output design will improve much in decision-making. Outputs are also used to provide a permanent hard copy of the results for later uses. Reports also form one of the major concepts of output. Outputs can be meant for users as well as management.

Client side output design:

In client side various reports like executed orders report, pending orders report, list of commodities can be generated. The updated information about the commodity market can be scene in the client side screen. There is facility to see the pending orders and cancel orders. If click on the order which scene in the main screen we get the best five orders trade.

Server side output design:

In server side various reports like executed orders reports, pending orders reports, list of commodities can be generated. It is available for each user and shows separately the buy and sell list of orders. The updated information about the commodity market can be scene in the client side screen.

TESTING AND IMPLEMENTATION

TESTING AND IMPLEMENTATION

5.1 TESTING

Software testing is a critical element of software quality Assurance and represents the ultimate view of specification, design and coding. A good test has a high probability of finding an error. Tests should be planned before actual testing begins. There are a number of tests that need to be conducted so as to ensure the accuracy of the software developed. If testing is conducted successfully it will uncover errors in the software. As a secondary Benefit, testing demonstrates that software functions appear to be working according to specification and that performance requirements appear to have been met. In addition data collected as testing is conducted provides a good indication of software reliability and some indication of software quality as a whole.

Testing Objectives

There are several rules that can serve as testing objectives. They are

- Testing is a process of executing a program with the intent of finding an error.
- A good test case is one that has high probability of finding an undiscovered error.
- A successful test is one that uncovers an undiscovered error.

If testing is conducted successfully according to the objectives as stated above, it would uncover errors in the software. Also testing demonstrates that software functions appear to the working according to the specification, that performance requirements appear to have been met.

There are three ways to test a program

- For Correctness
- For Implementation efficiency
- For Computational Complexity.

Tests for correctness are supposed to verify that a program does exactly what it was designed to do. This is much more difficult than it may at first appear, especially for large programs.

Tests for implementation efficiency attempt to find ways to make a correct program faster or use less storage. It is a code-refining process, which reexamines the implementation phase of algorithm development.

Tests for computational complexity amount to an experimental analysis of the complexity of an algorithm or an experimental comparison of two or more algorithms, which solve the same problem.

Testing Correctness

A software product can be tested in two ways

- White Box Testing
- Black Box Testing

White box testing for Commodity Exchange System

White box testing has been done for online commodity trading system and the result was found to be successful. Testing ensured that all independent paths with each module of the project has been exercised at least once and that all logical conditions work correctly on their true and false sides. Testing has checked all loops at their boundaries and also within their operational limits.

All modules in the project were tested using white box testing technique and were found to work properly according to the predefined standards.

Black box testing for Commodity Exchange System

Black box testing has been done for online commodity trading system. The system was tested with various types of input data and condition. The 2 modules of the project were tested separately using Black box testing strategies and the result was noted to be successful the system was tested with varying inputs and conditions. The system responded appropriately for each category of input data. The test cases and input types were set and expected output was also recorded before the actual test was done. On testing these were found to be accurate and black box testing was completed successfully for the project.

Testing of client – server Architecture

Online commodity trading system being developed according to client – server architecture, was tested according to client server testing strategies. The distributed nature of client/server environments, the performance issues associated with transaction processing the potential presence of a number of different hardware platforms, the complexities of network communication, the need to service multiple clients from a centralized/distributed data bases and the coordination requirements imposed on the server all combine to make testing of client/server architectures and the software that reside within them considerably more difficult than testing stand alone applications. The testing of the system according to client server architectural strategies was found to be successful.

5.2 Unit testing

Unit testing focuses on verification effort on the smallest unit of software design – module. The module interface was tested to ensure that information properly flows into and out of the program unit under test. The local data structure was examined to ensure that data stored temporarily maintains its integrity during all steps in the execution of the algorithms execution Boundary condition were tested to ensure that the module operates
Properly at boundaries established to limit or restrict processing. Each and every form in online Commodity Exchange package was tested. Each input field was tested with varying inputs to see how it responds. System generated appropriate messages as the inputs were provided. It worked well with valid inputs and provided error messages in case invalid inputs were provided. Thus unit testing conducted for Commodity Exchange package proved successful.

5.3 Integration testing

Integration testing is s systematic technique for constructing the program structure while constructing tests to uncover errors associated with interfacing. The objective is to take unit tested modules and build a program structure that has been dictated by design.

Integration testing for Commodity Exchange package

The various modules in online commodity trading system were unit tested. Those modules after unit testing was grouped together and tested using integration testing techniques.

Top down integration is an incremental approach to construction of program structure. Modules are integrated by moving down ward through the control hierarchy, beginning with the main control module.

The main form in Commodity Exchange package is the home page. It is from the home page that the branches to various sub modules within the project. The sub modules are attached to the home page in a depth first or breadth first manner

5.4 Validation Testing

At the culmination of integration testing, software is completely assembled as a package. Interfacing errors have been uncovered and corrected and a final series of software test-validation testing begins. Validation testing can be defined in many ways, but a simple definition is that validation succeeds when the software functions in manner that is reasonably expected by the customer. Software validation is achieved through a series of black box tests that demonstrate conformity with requirement. After validation test has been conducted, one of two conditions exists.

- The function or performance characteristics confirm to specifications and are accepted.
- A validation from specification is uncovered and a deficiency created.

Deviation or errors discovered at this step in this project is corrected prior to completion of the project with the help of the user by negotiating to establish a method for resolving deficiencies. Thus the proposed system under consideration has been tested by using validation testing and found to be working satisfactorily.

5.5 Output Testing

After performing the validation testing, the next step is output testing of the proposed system, since no system could be useful if it does not produce the required output in the specific format. The output generator or displayed by the system under consideration is tested by asking the users about the format required by them. Here the output is considered in two ways: One is on screen and the other is printed format. The output format on the screen is found to be correct as the format was designed in the system design phase according to the user needs. Hence output testing does not result any correction in the system.

5.6 User Acceptance Testing

User acceptance of the system is key factor for the success of any system. The system under consideration is tested for user acceptance by constantly keeping in touch with prospective system and user at the time of developing and making changes whenever required. This is done in regarding to the following points.

* Input screen design
* Output screen design
* On-line message to guide the user
* Menu driven system
* Format of ad-hoc reports and other reports

5.7 IMPLEMENTATION

The implementation phase is less creative than system design. It is primarily concerned with user training, site preparation, and file conversion. When the candidate system is linked to terminals or remote sites, the telecommunication network and tests of the network along with the system are also included under implementation.

During the final testing, user acceptance is tested, followed by user training. Depending on the nature of the system, extensive user training may be required. Conversion usually takes place at about the same time the user is being trained or later.

System testing check the readiness and accuracy of the system access update and retrieve data from new files. Once the program becomes available the test data are read into the computer and processed. In most conventions Parallel Run was conducted to establish the efficiency of the system.

At the beginning of the development phase a preliminary implementation plan is created to schedule and manage the many different activities that must be integrated into plan. The implementation plan is updated throughout the development phase, culminating in a changeover plan for the operation phase. The major elements of implementation plan are test plan training plan, equipment installation plan and a conversion plan.

There are three types of implementation:

* Implementation of a computer system to replace a manual system.
* Implementation of a new computer system to replace an existing one.
* Implementation of a modified application to replace an existing one, using the same computer.

5.9 Maintenance

After the installation phase is completed and the user staff is adjusted to the changes created by the candidate system, evaluation and maintenance begin. Like any system, there is an aging process that requires periodic maintenance of hardware and software. If the new information is inconsistent with the design specifications, then changes have to be made. Hardware also requires periodic maintenance to keep in tune with design specifications. The importance of maintenance is to continue to bring the new system to standards. The definition of software maintenance can be given by describing four activities that are undertaken after the program is released for use.

The first maintenance activity occurs since it is unreasonable to assume that software testing will uncover all errors in a large software system. The process of including the diagnosis and correction of one or more errors is called corrective maintenance.

The second activity that contributes to a definition of maintenance occurs since rapid change is encountered in every aspect of computing. Therefore adaptive maintenance modifies software to properly interface with a changing environment.

The third activity involves recommendations for new capabilities, modifications to the existing functions and general enhancements when the software is used. To satisfy requests perceptive maintenance is performed.

The fourth maintenance activity occurs when software is changed to improve future maintainability or reliability. This is called preventive maintenance. User priorities, changes in organizational requirements, or environmental factors also call for system enhancements.

SCOPE FOR FUTURE DEVELOPMENT

SCOPE FOR FUTURE DEVELOPMENT

Commodity Exchange System is designed with future in mind. Due care has been taken to assimilate the needs for future development. The Software was constructed along the lines suggested by Subject experts and hence all required data is stored. Care has also been taken for providing permission for users for access of resource. As the security information is maintained with a database any enhancements required can be made with subsequent updates within the database.

Order entry and execution has taken many important steps toward providing its Client's efficient executions. From the introduction of the commodity trading system to the many front-end solutions that firms offer their clients today, trading in the futures markets is quickly approaching the standards for online trading that have been established by their equity trading counterparts. In the coming months you should expect that both the exchanges and individual firms would offer additional technology to improve online trading access and service.

In future there is a chance to develop a multi-user back office software associates with commodity exchange for commodity brokers who deal with a large number of clients, branches with multiple segments etc. we can develop a tool to automate the transactions associated with commodity-broking and to perform accounting entries. The scope of this software can extend from downloading the daily trade to performing settlement billing, from creating an account to preparing Profit & Loss account and Balance Sheet. It is possible to make an automated solution for almost all the problems encountered while manual maintenance of commodity-broking records and deficiencies.

CONCLUSION

CONCLUSION

Commodity exchange provides you the fastest, most efficient order execution possible. Routes your order directly into the matching engine, you should expect to receive confirmations in seconds. Commodity Exchange package strives to provide its dealers the ability to place, monitor and control all aspects of their trading from one interface that is available anywhere a dealer can log into. It has the ability to quickly access filled, working, cancelled and rejected orders.

The system has been developed for the given conditions and is found working effectively. The developed system is flexible and changes whenever can be made easy. The software is neat and simple in manner and help to reducing the operator's work. It has developed using the facilities and functionalities of. Visual Basic. NET.

The user-friendly software, Commodity Exchange System, successfully overcome strict and severe validation checks performed using the test data. The results attained were fully satisfactory. An attempt was made to obtain maximum perfection in documenting the software in a simple, precise and self-explanatory manner.

The system was verified with valid as well as invalid data in each manner. The system is done with an insight into the necessary modifications that may require in the future. Hence the system can be maintained successfully without much rework.

APPENDIX1-TABLES

APPENDIX 1-TABLES

1.COMMODITY MASTER TABLE

The trading of commodity takes place only for the commodities inside this table. This is a master table. Here Comcode (commodity code) is the primary key.

Field Name	Key Relationship	Field Type
Comcode	Primary Key	Int
Comdesc	Not Null	Varchar(50)
Comgroup	Not Null	Char

2.LOGIN MASTER TABLE

This table is used for creating user id for wholesalers. This is a master table. Each wholesaler has separate login id and password. Here user id is the primary key.

Field Name	Key Relationship	Field Type
Userid	Primary Key	Varchar(20)
Password	Not Null	Varchar(20)
Uname	Not Null	Varchar(20)
Uaddr	Not Null	Varchar(50)
User_Level	Allow Null	Varchar(20)

3.CENTER MASTER TABLE

The trading of commodity takes place only in these places. This is a master table. Each wholesaler can sell or buy in these places. Here Center id is the primary key.

Field Name	Key Relationship	Field Type
Centerid	Primary Key	Varchar(10)
Centerdesc	Not Null	Varchar(50)

4.CURRENTUSER TABLE

The list of current users can be seen through this table. It is a temporary table. With the help of status field we can disable the activities of the user.

Field Name	Key Relationship	Field Type
Cuserid	Not Null	Varchar(20)
Status	Not Null	Char

5.BUYORDER TABLE

This table is used for enter the buy order from the wholesalers. Here Ordno (order id) is the primary key. This table will refer the commodity master table for commodity code entry and it will refer user table for user id entry and it will refer center table for place entry.

Field Name	Key Relationship	Field Type
Ordno	Primary Key	Int
Orddate	Not Null	Varchar(10)
Ordtime	Not Null	Varchar(15)
Comcode	Foreign Key	Int
Comdesc	Not Null	Varchar(50)
Buyqty	Not Null	Bigint
Buymin	Not Null	Bigint
Buyrate	Not Null	Bigint
Userid	Foreign Key	Varchar(20)
Centerid	Foreign Key	Varchar(10

6.SELLORDER TABLE

This table is used for enter the sell order from the wholesalers. Here Sordno (order id) is the primary key. This table will refer the commodity master table for commodity code entry and it will refer user table for user id entry and it will refer center table for place entry.

Field Name	Key Relationship	Field Type
Sordno	Primary Key	Int
Sorddate	Not Null	Varchar(10)
Sorddate	Not Null	Varchar(15)
Scomcode	Foreign Key	Int
Scomdesc	Not Null	Varchar(50)
Sellqty	Not Null	Bigint
Sellmin	Not Null	Bigint
Sellrate	Not Null	Bigint
Suserid	Foreign Key	Varchar(20)
Sceneterid	Foreign Key	Varchar(10

7.MATCH TABLE

This table is used for enter the matching entries. From this table we get the executed orders. Here Matchno (Match number) is the primary key. This table will refer the commodity master table for commodity code entry and it will refer user table for sell user id and buy user id entry and it will refer center table for sell place and buy place entry.

Field Name	Key Relationship	Field Type
Matchno	Primary Key	Int
Morddate	Not Null	Varchar(10)
Mcomcode	Foreign Key	Int
Mcomdesc	Not Null	Varchar(50)
Mqty	Not null	Bigint
Mmin	Not Null	Bigint
Mrate	Not Null	Bigint
Buserid	Foreign Key	Varchar(20)
Suserid	Foreign Key	Varchar(20)
Centerid	Foreign Key	Varchar(10)
Msgstatus1	Allow Null	char
Msgstatus2	Allow Null	char

8.TEMPREPORT TABLE

This table is used for enter the details of server side report. From this table we will get the User vice details of executed orders. This is a temporary table.

Field Name	Key Relationship	Field Type
Comcode	Not Null	Int
Comdesc	Not Null	Varchar(50)
Buyer	Not Null	Int
Seller	Not Null	Int
Quantity	Not Null	Bigint
Rate	Not Null	Bigint
Tradetype	Not Null	char

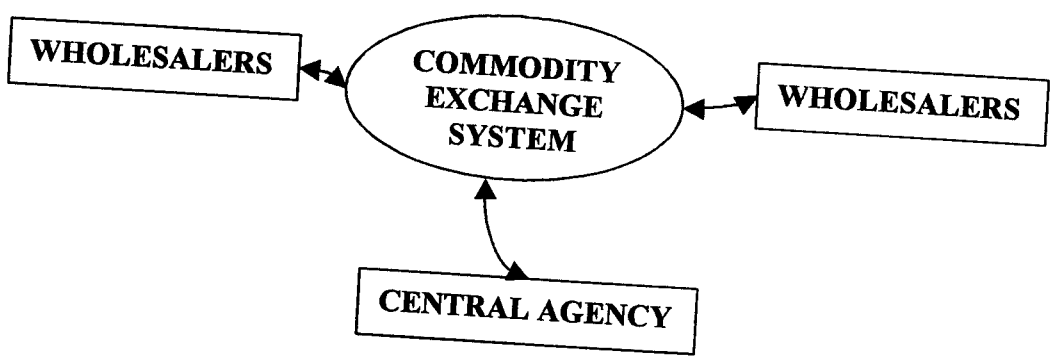
9. COMMODITY.TXT

This text file is used to store the names of commodities, which the wholesaler used to deal. The wholesaler can change the details of this file during run time. This file is situated in client side only.

APPENDIX2-DATA FLOW DIAGRAMS

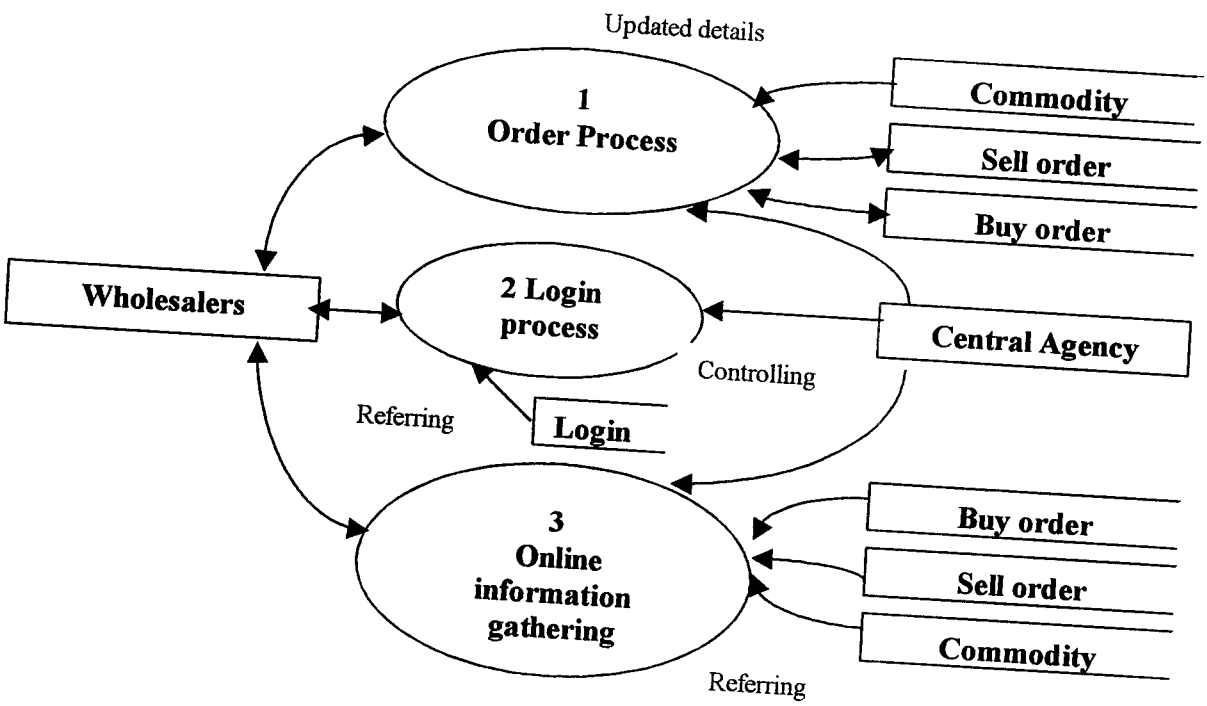
APPENDIX 2-DATAFLOW DIAGRAMS

DFD - LEVEL-0: COMMODITY EXCHANGE

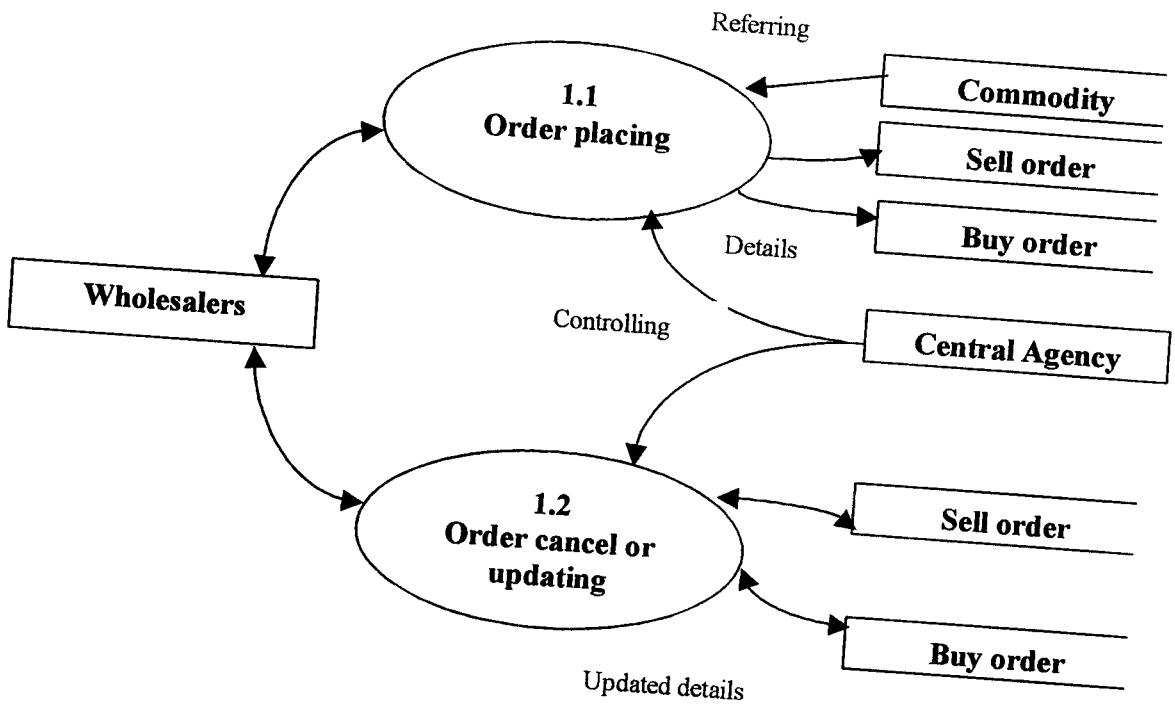


CLIENT-SIDE DATA FLOW DIAGRAMS

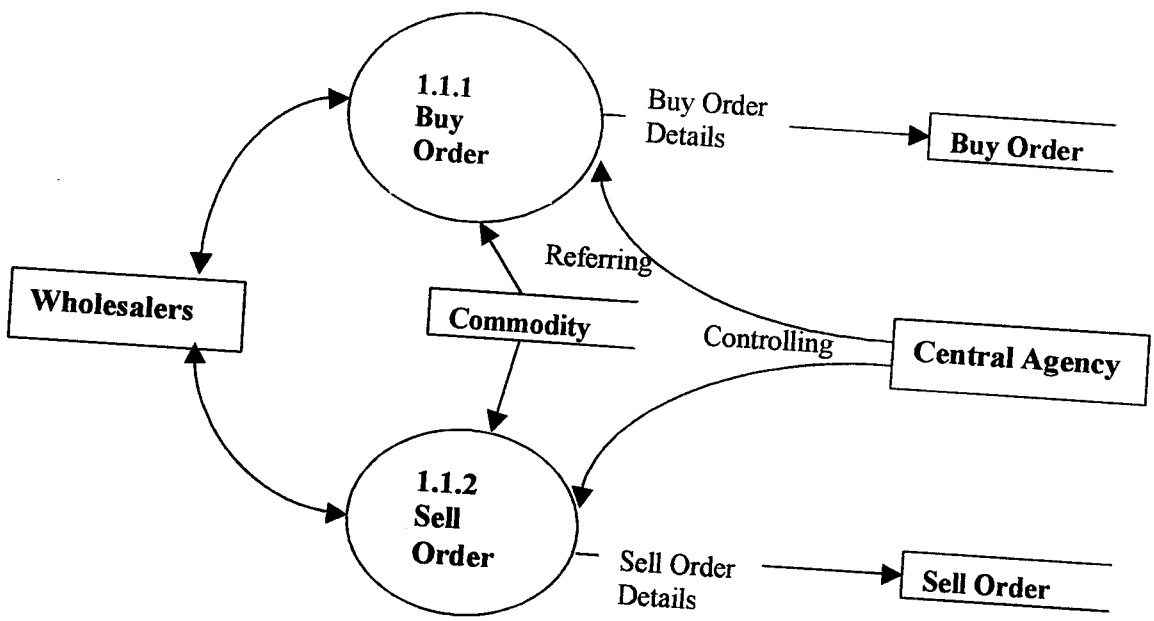
DFD LEVEL-1: DEALER INTERACTION



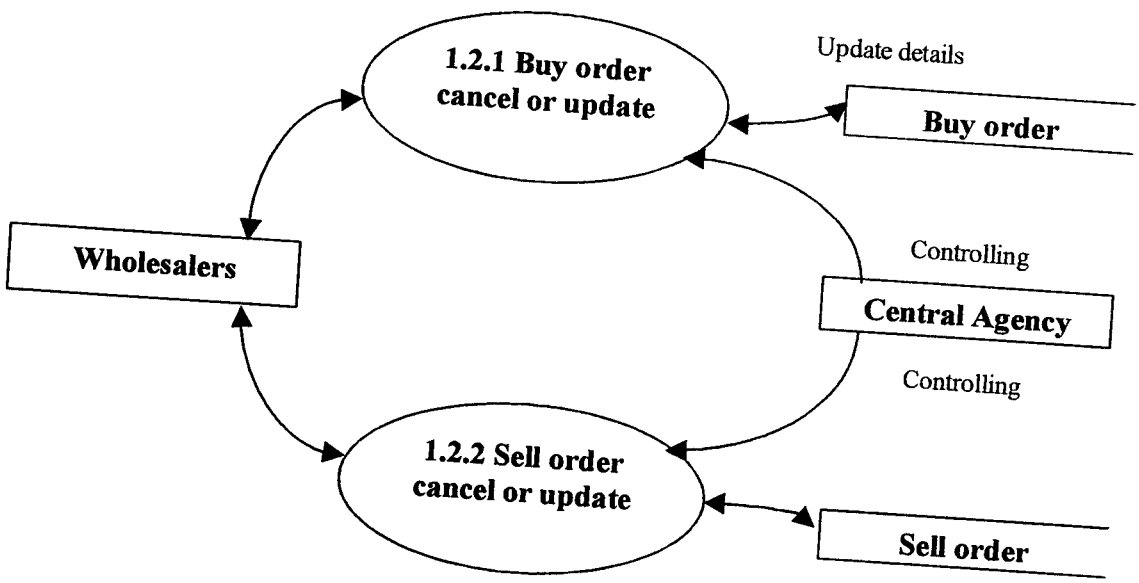
DFD LEVEL-2: ORDER PROCESSING



DFD LEVEL-3: ORDER PLACING

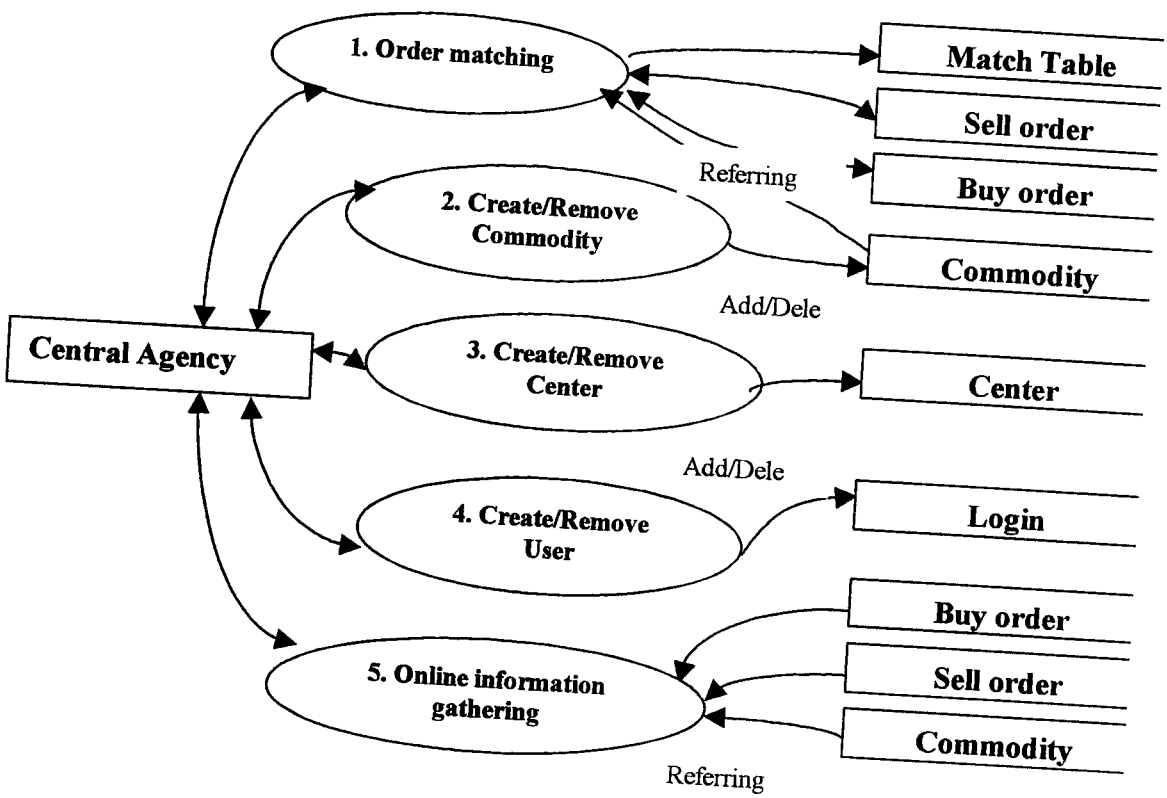


DFD LEVEL-3: ORDER UPDATION OR CANCEL

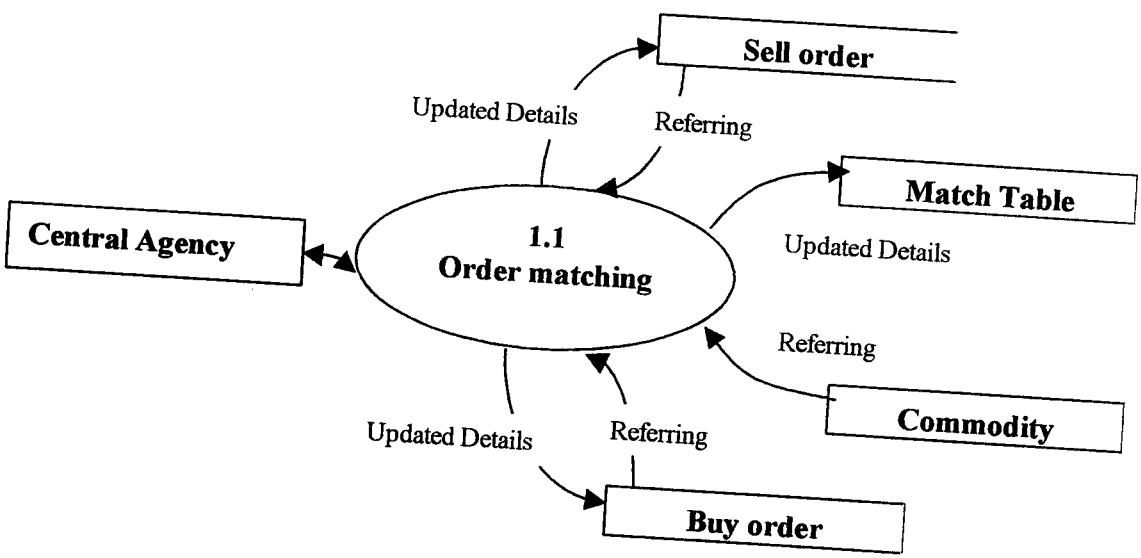


SERVER-SIDE DATA FLOW DIAGRAMS

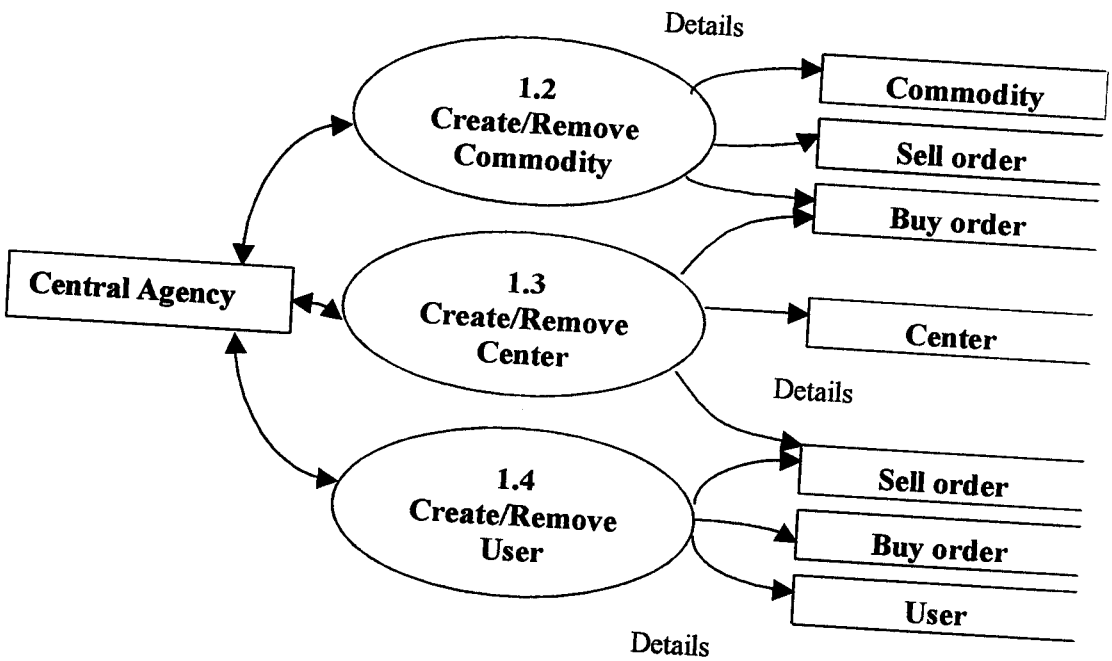
DFD LEVEL-1: CONTROL AGENCY



DFD LEVEL-2: ORDER MATCHING



DFD LEVEL-2: CREATION BY CONTROL AGENCY

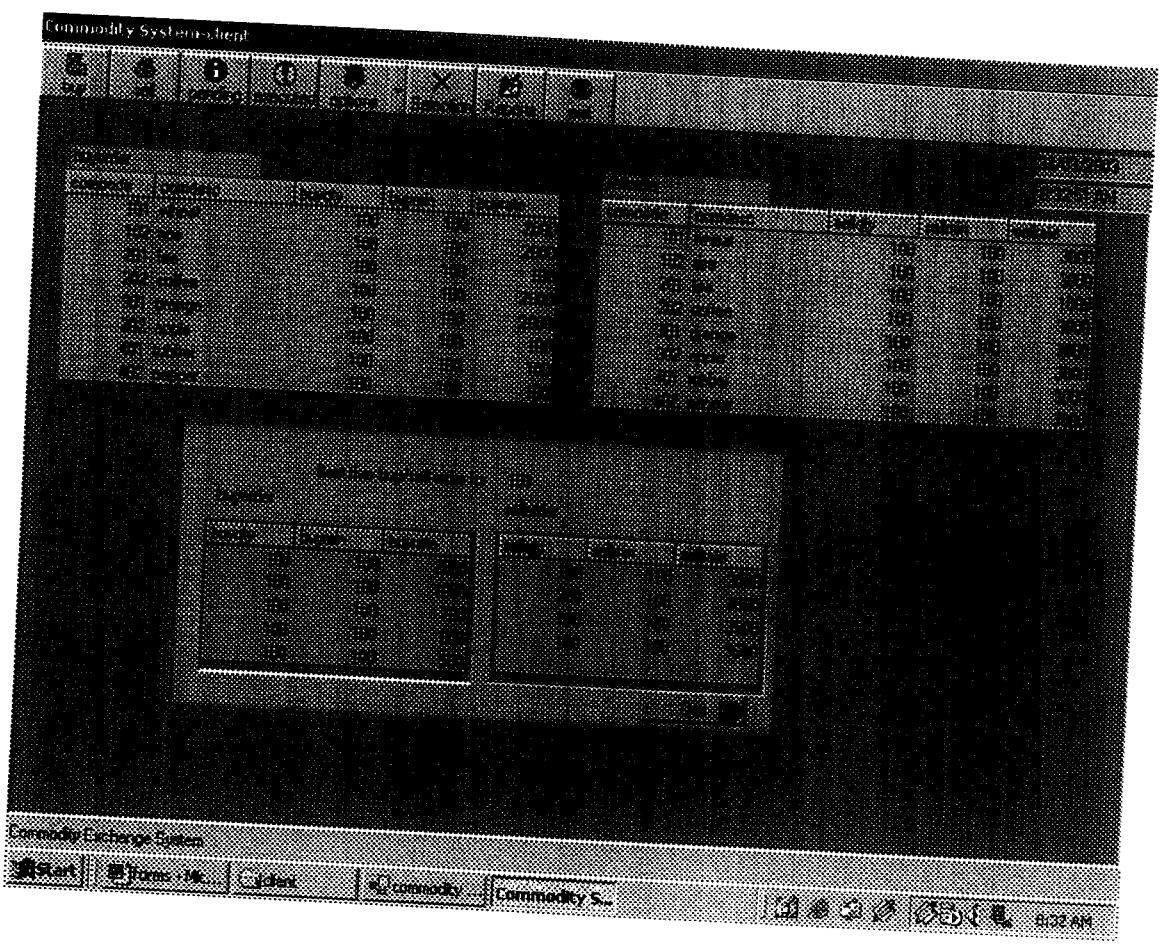


APPENDIX3- SAMPLE FORMS

CLIENT-SIDE FORMS

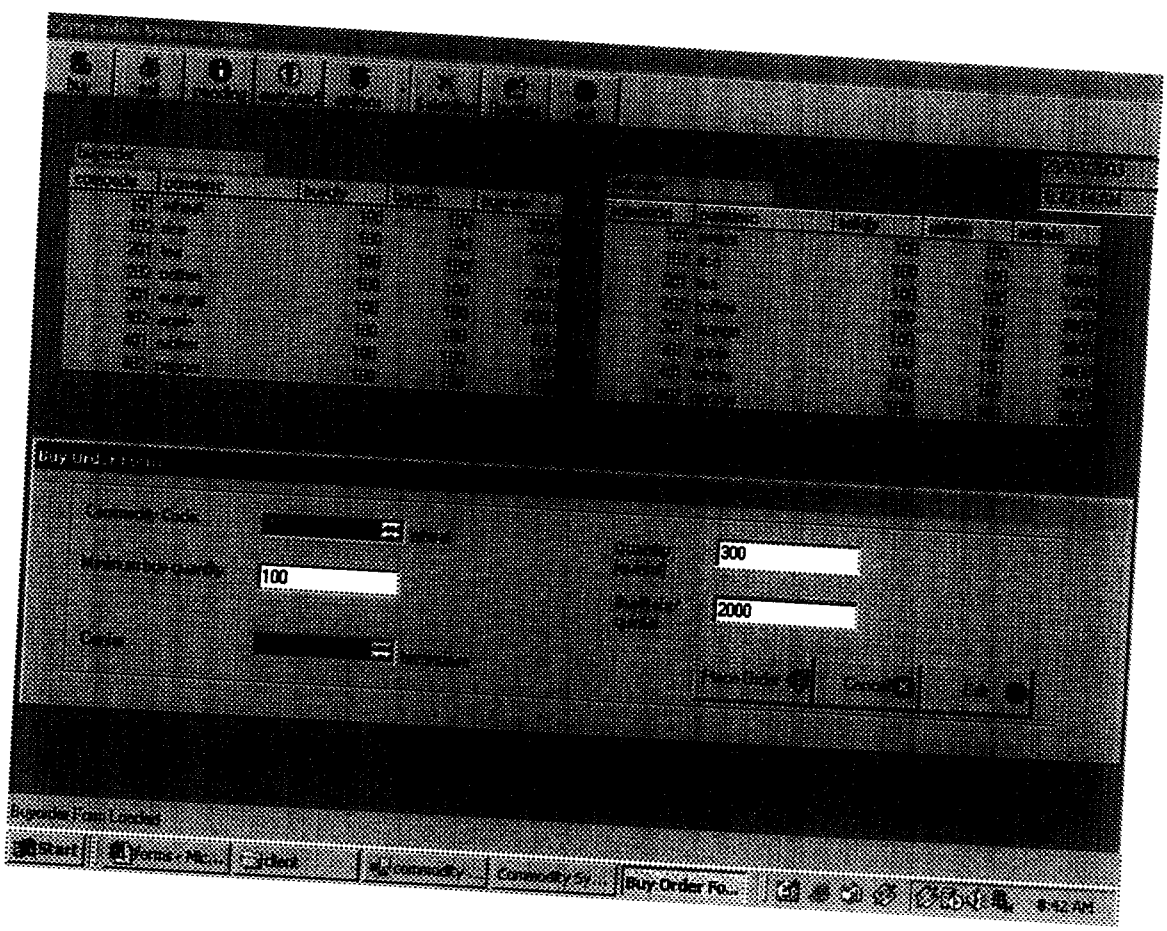
MAIN FORM

This is the Main Form (client-side). This form is used to access the online information's from the central agency. We can see the updated information's in every second through this screen. If we click on any pending order displayed in the screen, a popup window will open. In that we can see the best five buy order and best five sell order.



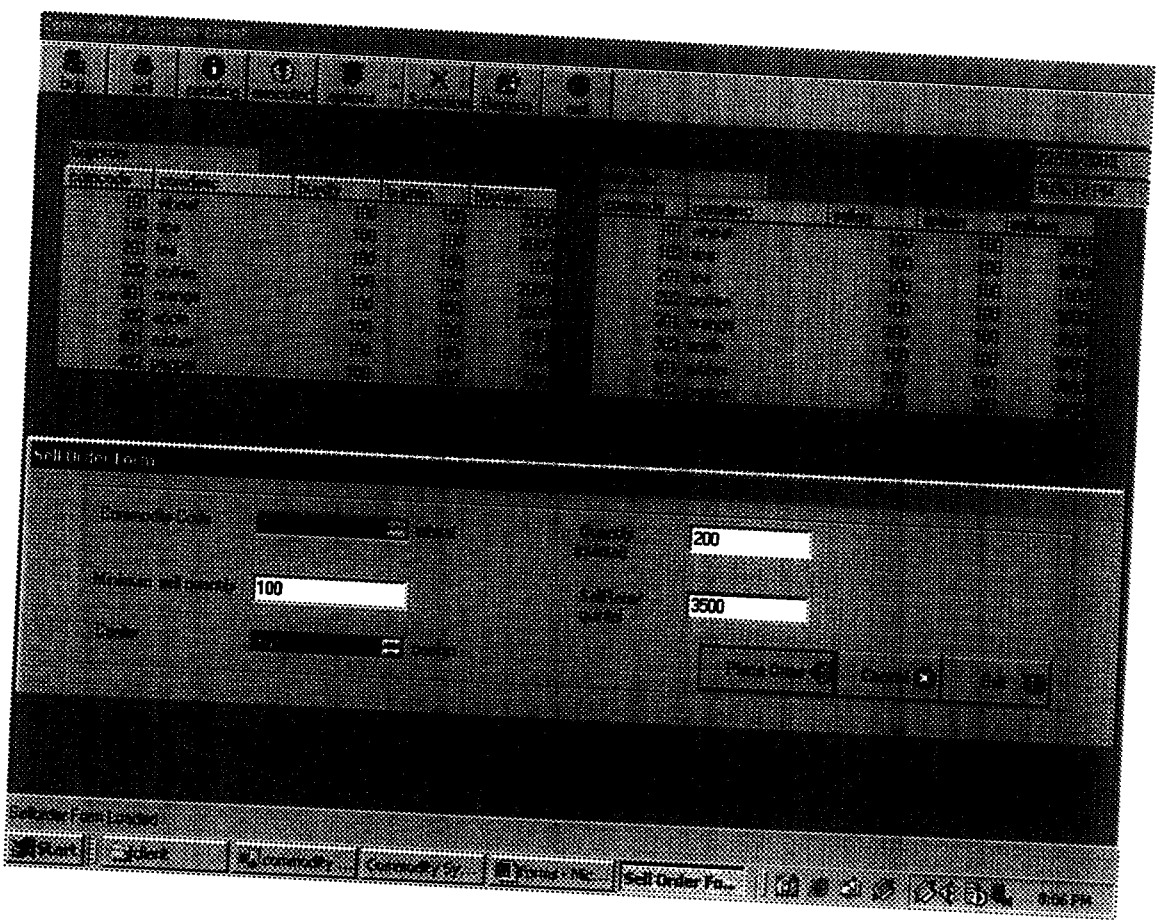
BUY ORDER FORM

This form is used in the client side to get information's from the server. Here we have buttons for placing buy order and sell order. If we click on any of these buttons new forms will load and display.



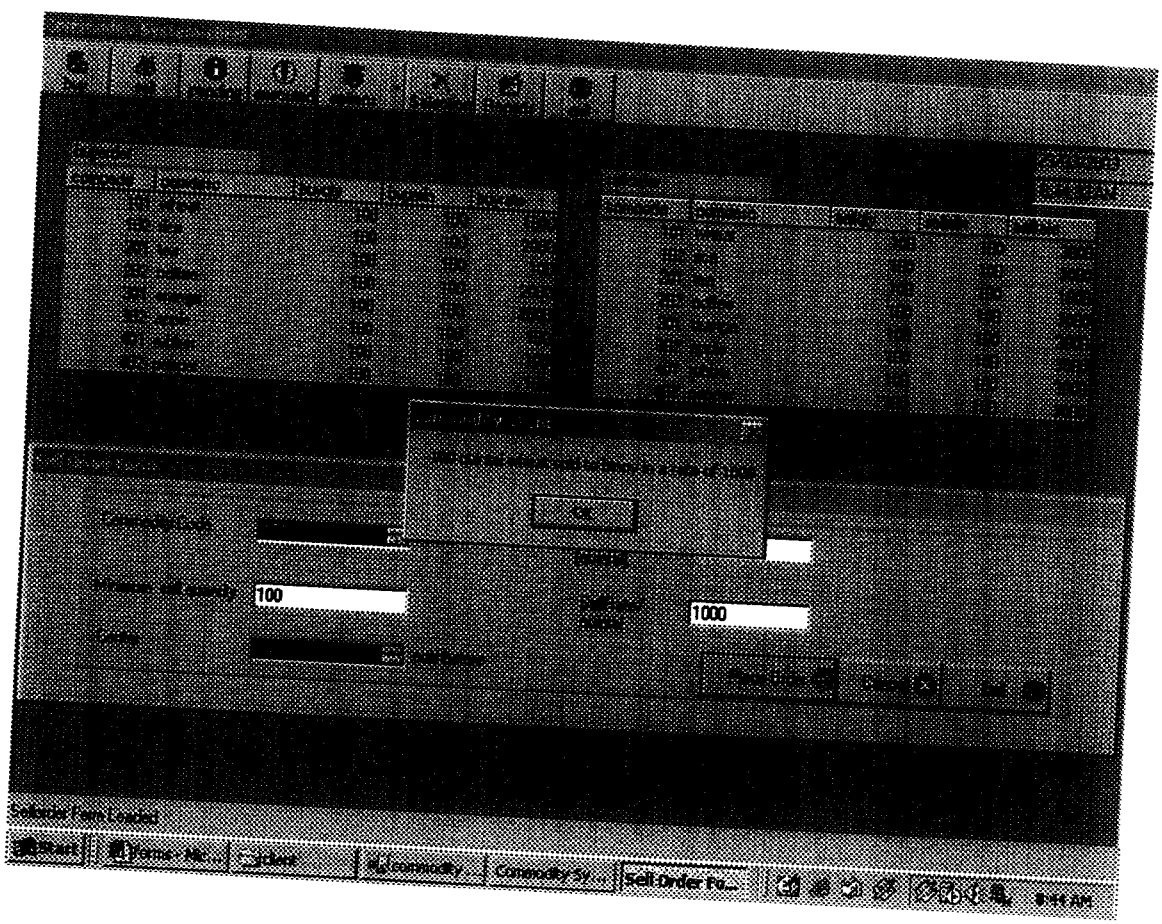
SELL ORDER FORM

If we click on the button sell, sell order form will be loaded. The sell order can be placed with the help of this form. The buttons cancel help to clear the fields in the form.



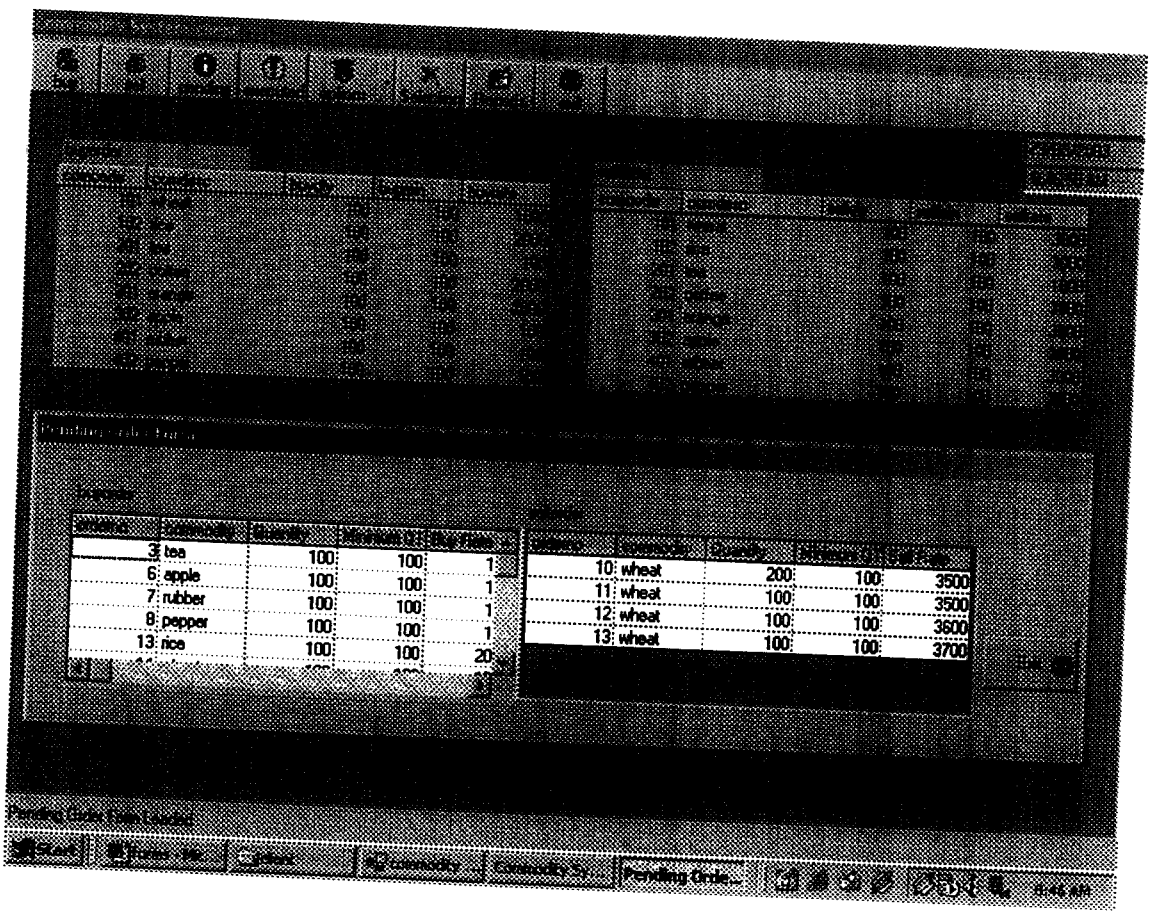
EXECUTED ORDER MESSAGE

This is a message from the server, which shows an order is executed. When, an order is placed, the server checks the matching orders. If the matching occurs, a message will send to the each client. That message is displayed below.



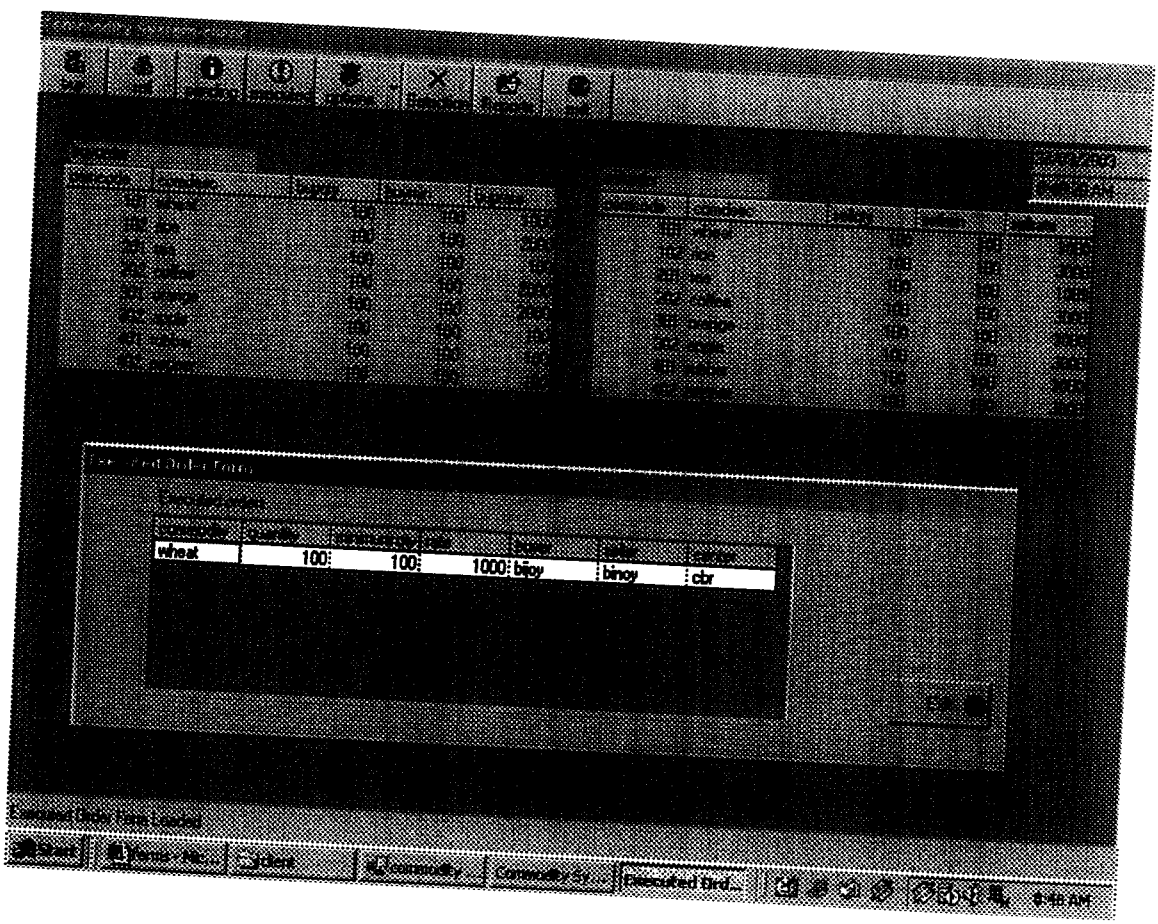
PENDING ORDER FORM

This form is used to see the pending orders of a client. If we click the pending button on the main form, pending order form will be loaded.



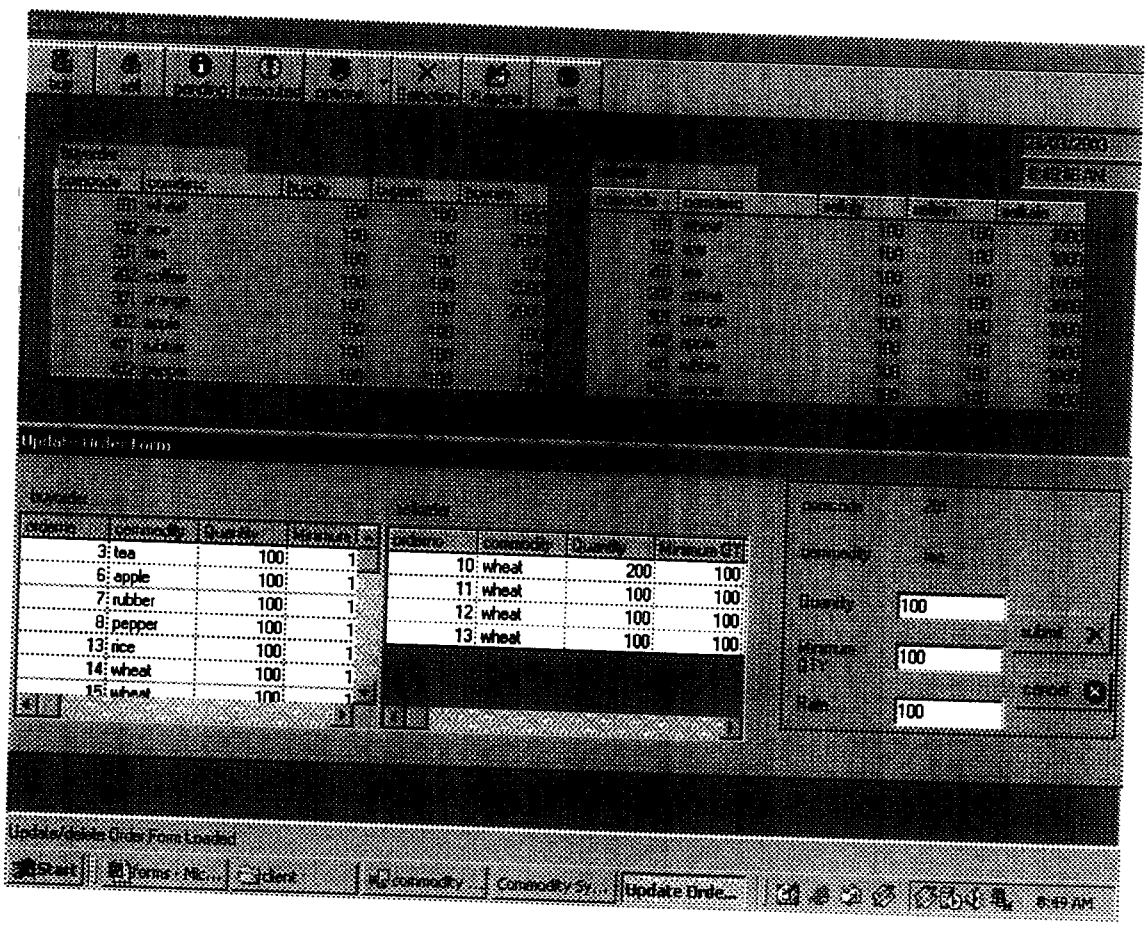
EXECUTED ORDER FORM

This form is used to see the executed orders of a client. If we click the executed button on the main form, executed order form will be loaded.



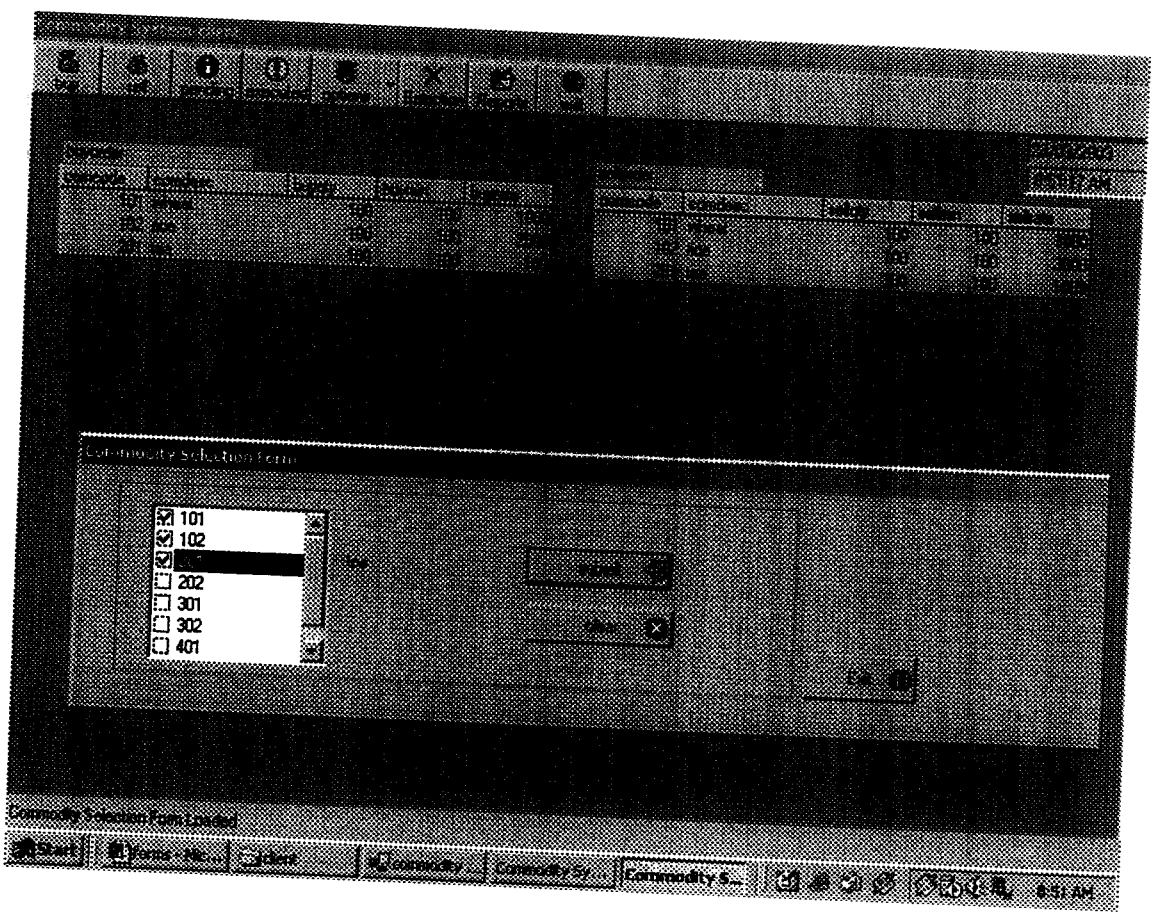
UPDATE/DELETE ORDER FORM

If we click on the button option update/delete order, update/delete order form will be loaded. This form helps to update /delete the pending orders. If we click on the ordno field, update button and delete buttons enabled.



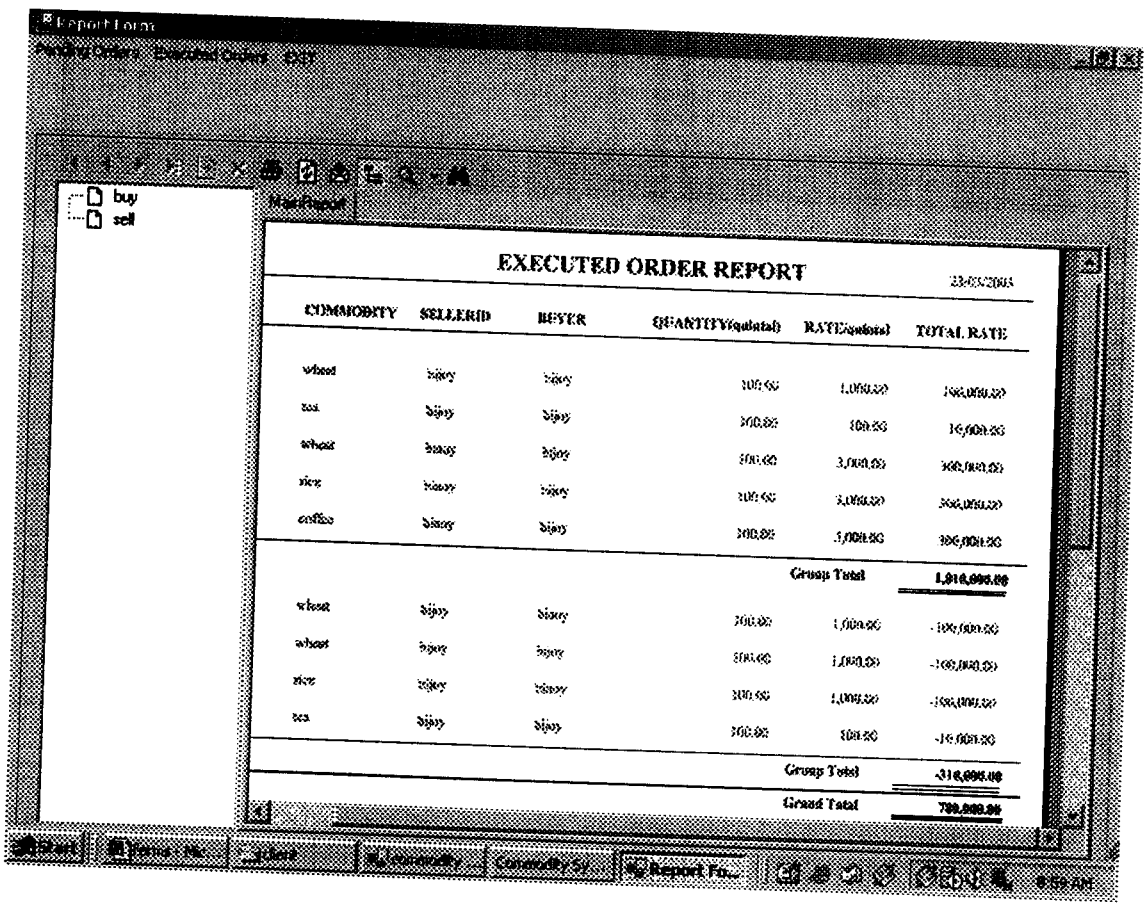
COMMODITY SELETION FORM

If the client wants to select the list of commodities that they want to trade, they can use this form. Select the list from the whole list.



REPORT FORM

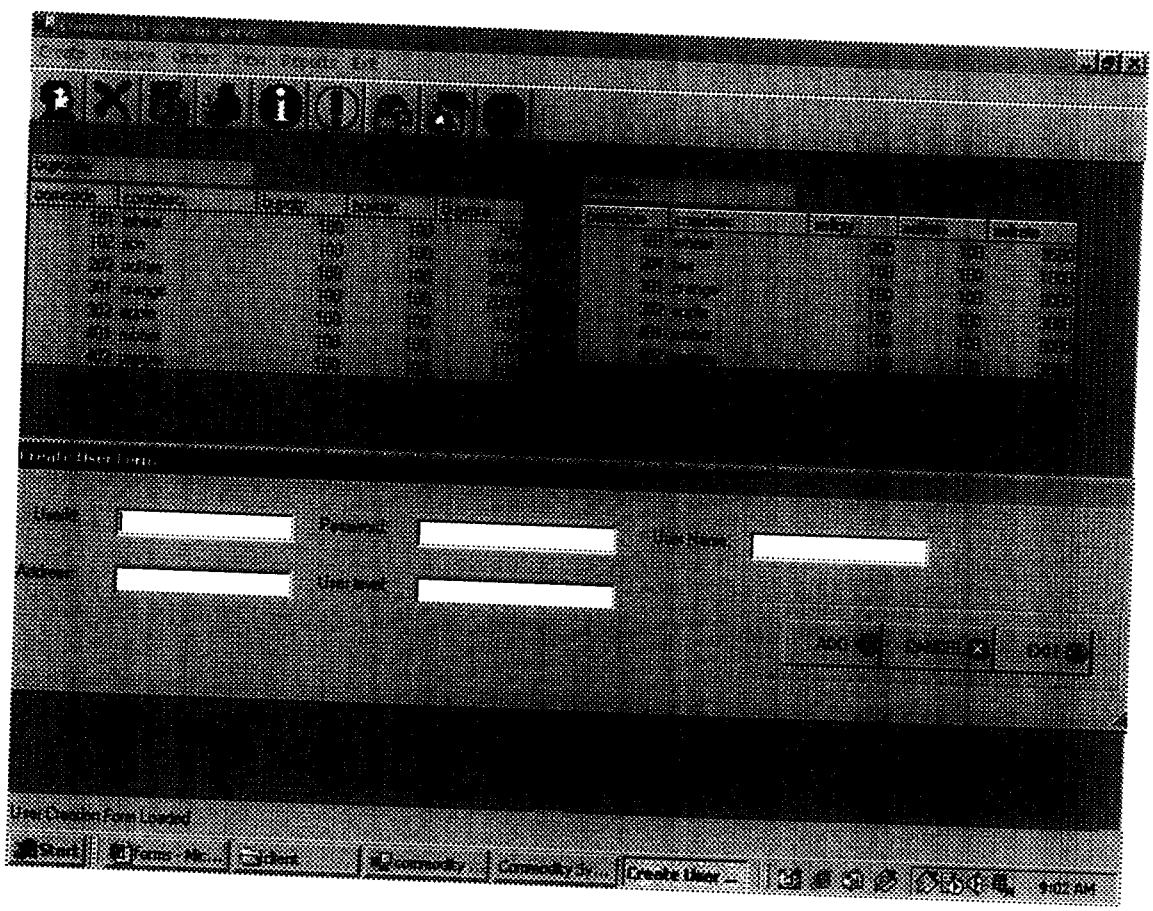
This report form help to display various kinds of reports like pending buy order report, executed buy order report etc. There is an executed order report is shown below. This report shows the group wise details of the executed orders.



SERVER-SIDE FORMS

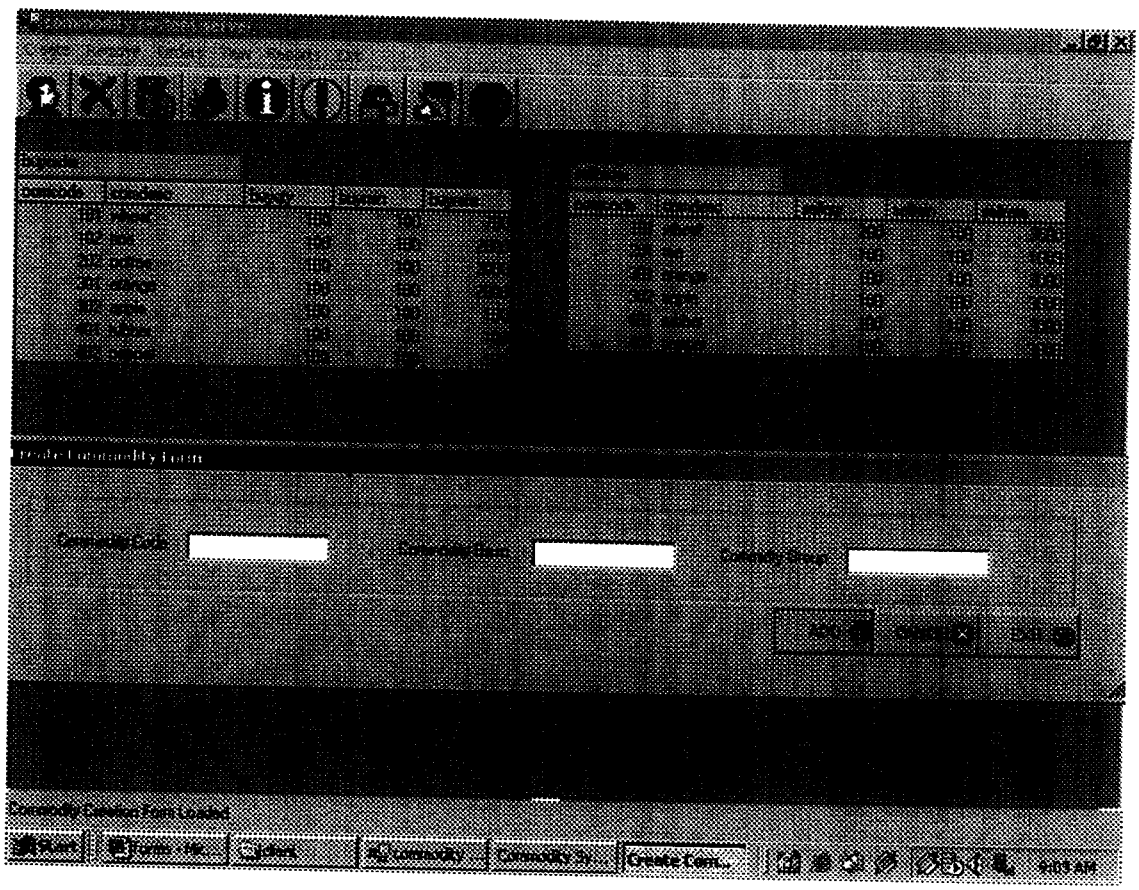
USER CREATION FORM

This is the Main Form (server-side). The screen displayed here also similar to the client-side. There is a menu in this form that help to do operations on the server-side. If we click on the menu item create user, a form will be loaded and which help to create new users.



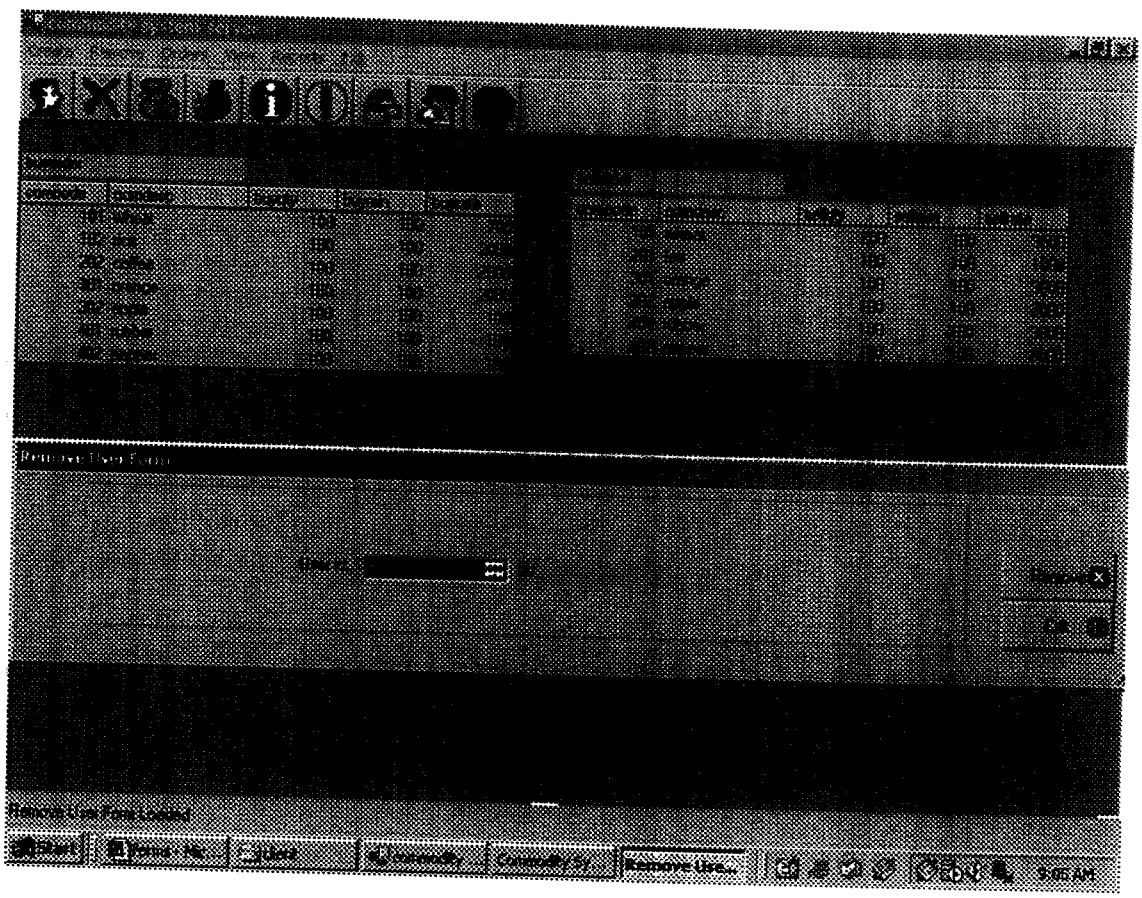
COMMODITY CREATION FORM

If we click on the menu item create commodity, a form will be loaded and which help to create new commodities.



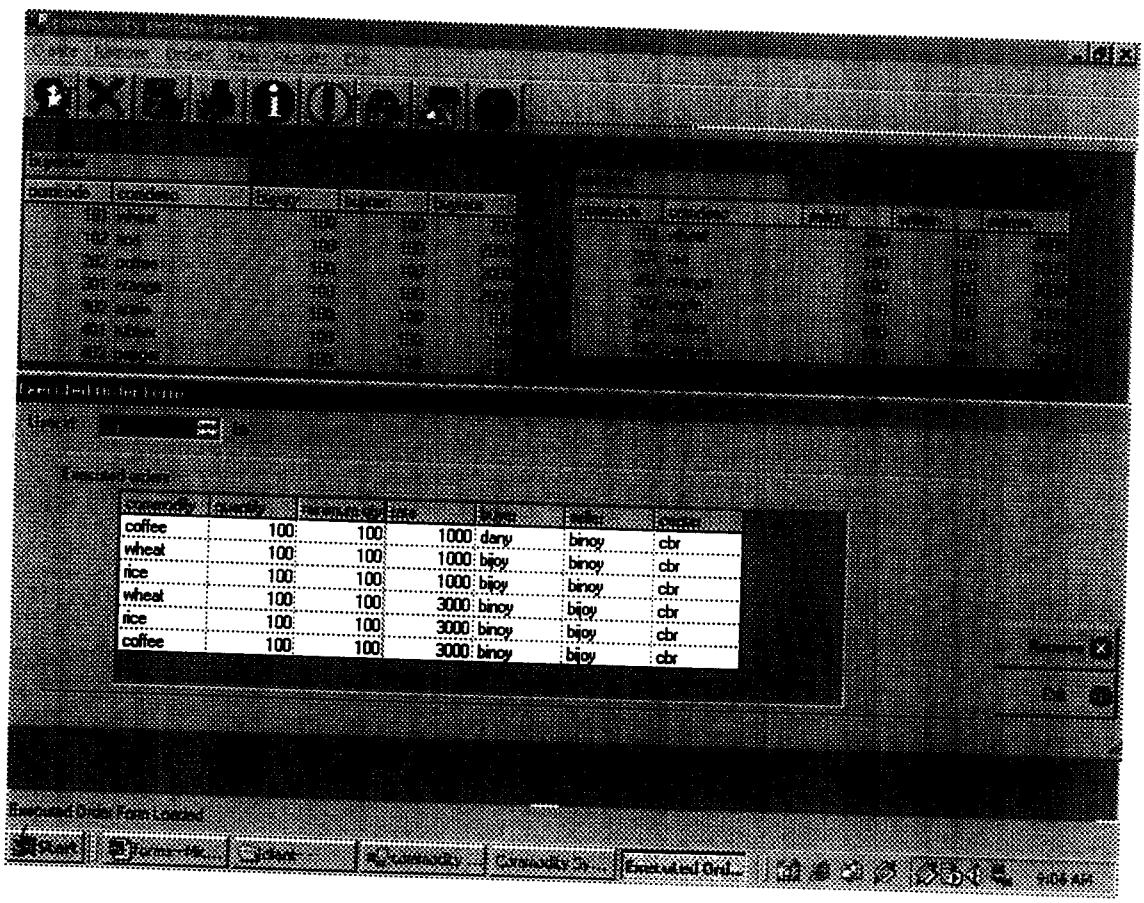
REMOVE USER FORM

If we click on the menu item Remove user, a form will be loaded and which help to remove users.



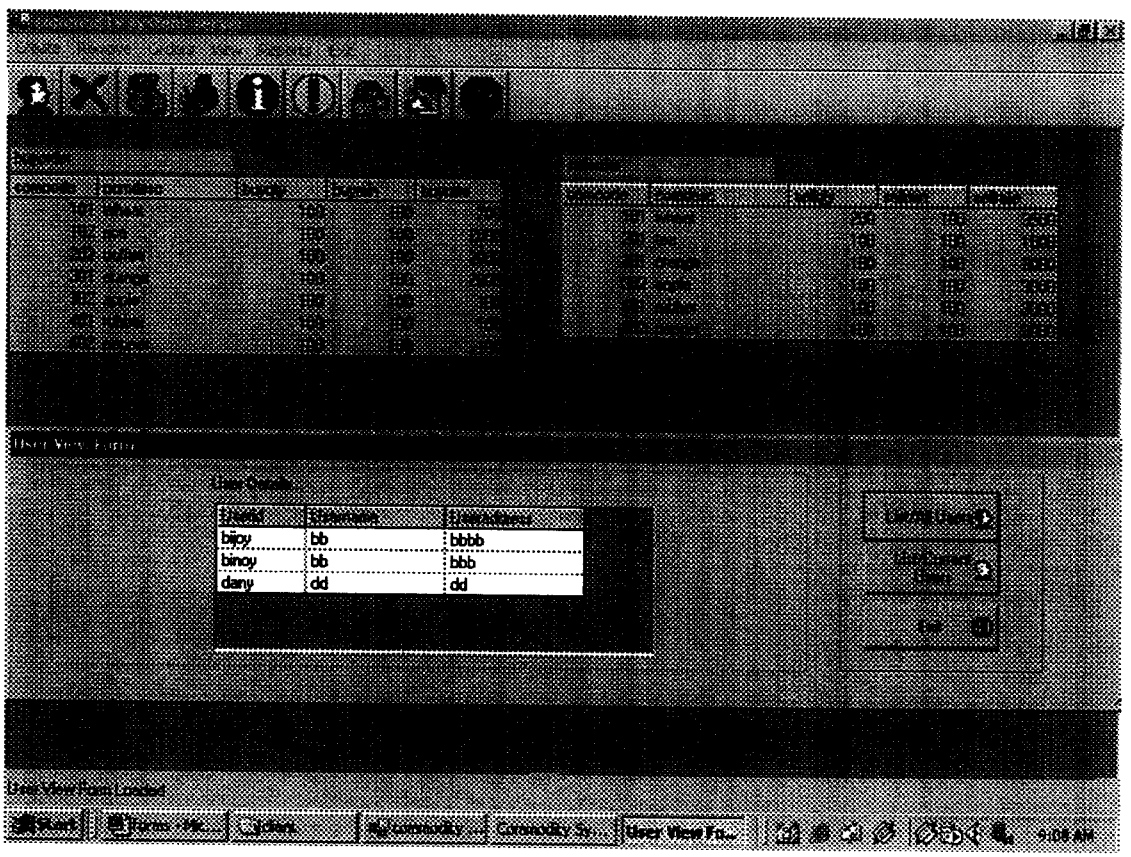
EXECUTED ORDER FORM

This form is used to see the executed orders of each client. If we click the executed button on the main form, executed order form will be loaded.



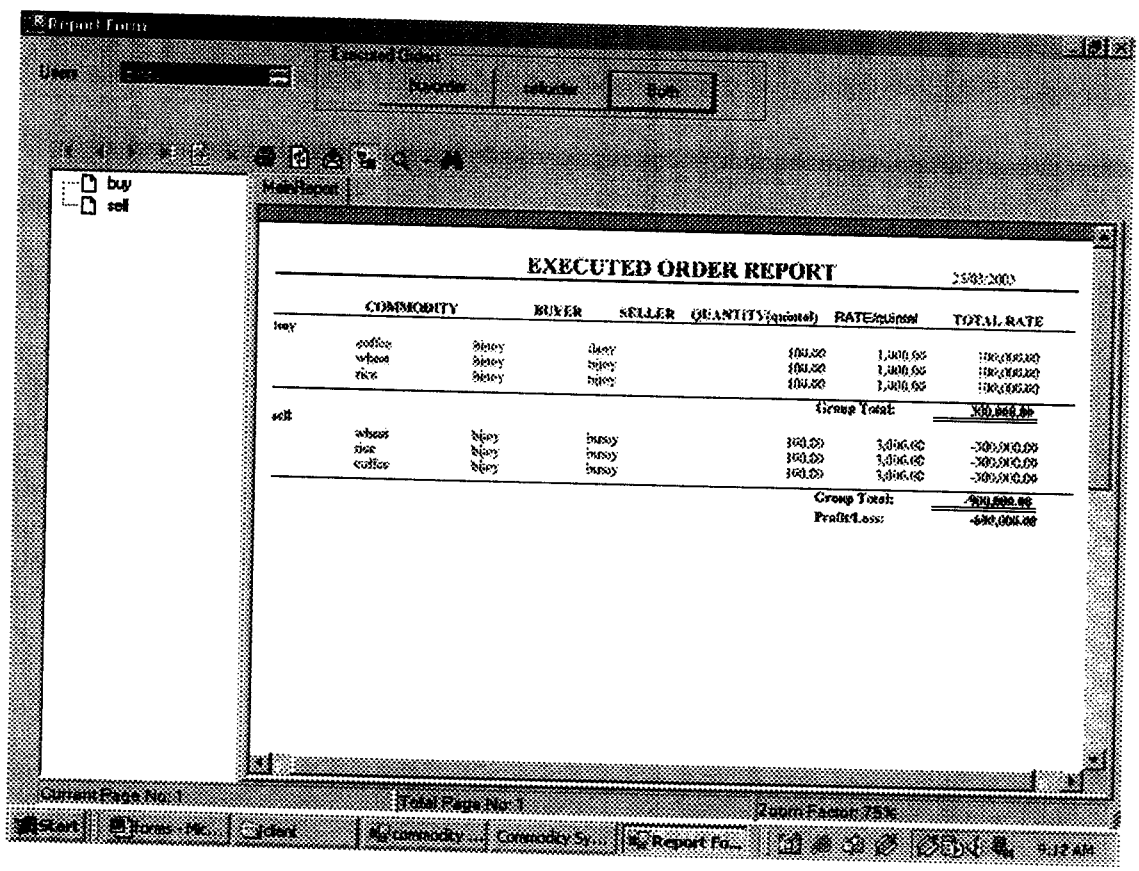
USER VIEW FORM

This form is used to View the list of all users/current users. If we click the View button on the main form, User view form will be loaded.



REPORT FORM

This report form help to display various kinds of reports like executed buy order report, executed sell order report etc. There is an executed order report of a user is shown below. This report shows the group wise details of the executed orders.



BIBLIOGRAPHY

i. James A. Senn, Analysis & Design Of Information Systems, McGraw Hill International, 1989.

ii. Roger S. Pressman, Software Engineering, McGraw Hill International, 1997.

iii. Ron Soukup and Kalen Delaney, Microsoft SQL Server 7.0, WP Publishers and Distributors, 2001.

iv. David S. Platt, Microsoft .NET, WP Publishers and Distributors, 2001.

v. William Perry, Effective Methods for Software Testing, Wiley –QED Publication, 1995.

vi. Evangelos Petroutsos, Visual Basic 6.0, BPB Publications, 1998.