**Share Market System**

**Objective and scope of the project:**

Share Market System is designed and developed to maintain all records of the company, admin, broker and client. In this software, there are lots of calculations regarding to shares, current rate and buying rate etc. These calculations are done previously manually but now all are handled using this software.

This software is designed with the help of C#.net as front-end and SQL server 2005 as back-end.

The Share market system is desktop application software. The purpose of this software is to transform the existing system into well organized computer system which would be able to handle the large amount of data and calculations.

* Records of the company are maintained in COMPANY\_DETAILS such as company name, number of shares it contains and date.
* Records of the admin who is going sell the shares of the company are maintained in ADMIN\_DETAILS such admin name, type of the admin, number of shares and list of clients under the particular type of the admin.
* Information of all the shares of the particular admin are going to maintain in the SHARE\_DETAILS such as admin name, number of shares, date, opening rate, current rate, company name and broker name to which admin is selling his shares of a particular company.
* Records of the broker who is buying the shares from the company and selling them to the clients are also maintained in the BROKER-DETAILS such as his name, address, broker-code and pan no. etc.
* And all the transaction of shares is maintained in the different form i.e. BROKER\_SHARES. Mainly the respective customers and commission that he would be getting from that customer, buying and selling rate and the current rate that is provided by the admin is also maintained in this form.
* The details of the customer are maintained in the CLIENT\_DETAILS form such as client name, number of shares it contains and the admin type.
* And the details of the client’s shares are maintained in the CLIENT\_SHARES form and transaction of shares from client to admin is also maintained in the same form.

**Theoretical Background and Problem Definition:**

Understanding the problem in the existing system & finding requested solution is the most important activity while planning the project. Hence the developing a new system we must get through problem associated with the current system.

* In share market, there are lots of traders and investors who invest to earn a lot of returns from their share of investment. For each customer, there should be proper data maintenance with proper information for security purpose.
* For the number of clients, it should be difficult to maintain the data in no. of registers and handling the transaction of shares manually.
* Also the calculations part that is for selling and buying the shares is also handled manually in previous system.
* The manpower required for this kind of transaction and maintenance of data is higher than the actual requirement.

**System Analysis:**

**3.1 REQUIREMENT ANALYSIS**

**3.1.1 FUNCTIONAL REQUIREMENTS**

**a) Strong Data Validation:**

There is possibility that user might enter wrong data and wrong data may cause inconsistency to the database and hence to the system. To avoid this, data should be validated whenever entered.

**b) Automatic updating of the database:**

After any transaction is performed, it is necessary that the updating should be reflected in the database without any inconsistency.

**c) Provide efficiency querying based on user requests:**

The major purpose is to generate efficient result on any user request. This will be done by our query processing system, which should be able to process any combination of queries will be done dynamically at run time depending on the user.

**3.1.2 EXTERNAL INTERFACE REQUIREMENTS**

**a) User friendly interface:**

The interface should be developed in such a manner that it is very user friendly, this not only improve interaction but also saves data entry time.

**b) Making well designed forms for capturing data:**

The forms for capturing the data should be well-designed using pop-down menus and drag & drop facilities, which reduce the data entry effort on the part of the user.

**3.1.3 PERFORMANCE REQUIREMENTS**

**a) Security:**

All users are not allowed to access the database. Hence there is a need to check authority of every user. Username and Password validation helps to deny unauthorized access to the system.

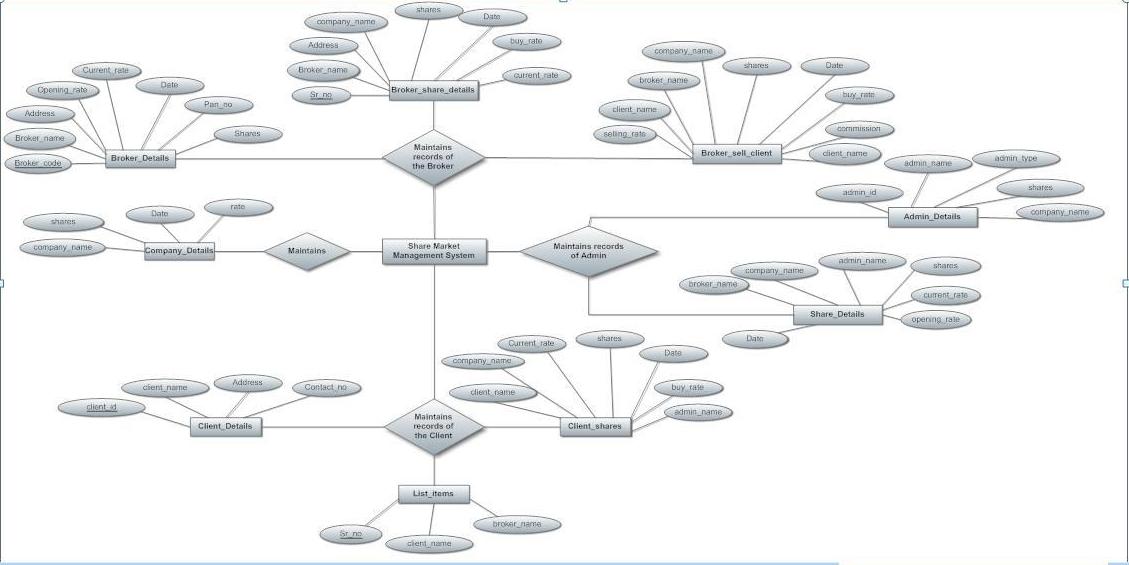
There are 3 main types of users who will be using the software

They are:-

1. Admin
2. Broker
3. Client

Each user is given the specific rights to access the data in Read only, Read Write, Delete.

**3.2 ENTITY RELATIONSHIP DIAGRAM:-**

****

**3.3: CONTEXT LEVEL DIAGRAMS:-**



Share\_details

Broker\_shares

Broker\_details

1.3

Admin share details

1.2

Sell shares to broker

Admin

1.1

Select Broker

2.1

Select Client

Client\_details

2.3

Broker share details

Broker

2.2

Sell shares to client

Broker\_shares

Client\_shares

3.2

Sell shares to Admin

Client\_shares

Share\_details

Admin\_details

3.3

Share details

Client

3.1

Select Admin

**3.4 Methodology adopted, system implementation and details of hardware and software used**

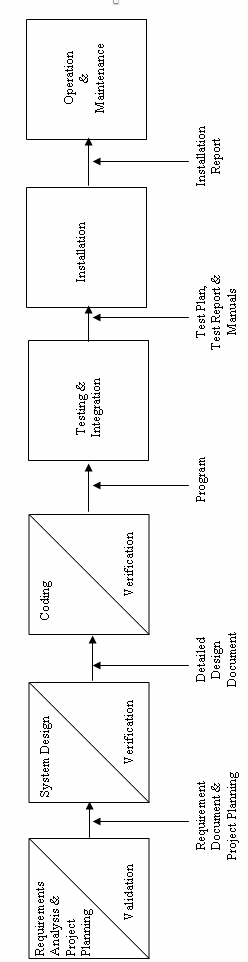
**3.4.1 PROCESS MODEL – SOFTWARE ENGINEERING**

In the development of software we have used **Waterfall Model**, the linear sequential mode. This model encompasses the following activities:

**a) Analysis Phase:**

System Analysis:-

This refers to the gathering of system requirements, with the goal of determining how these requirements will be accommodated in the system.



**b) System Design Phase:**

This is actually a multistep process. In this we tried to focus on some distinct attributes of a program like data structure, software architecture, interface representations and algorithmic detail. In this we tried to translate requirements into representation of the software which can be assessed for quality before coding begins. In the verifications, I have tried to ensure that the design is satisfying the requirements and is of good quality. I have tried to find out if there is any misinterpretation of specified any requirements.

**c) Code Generation Phase:**

In this phase, we translated design of a system into code which can be compiled and executed. In this phase we have done actual coding for all forms. In this we tried to produce simple program which are clear to understanding and modify.

We have used dynamic method to verify the code. We have executed program on some test data and output of the program examined to determine if there are any error present. I have read the code carefully to detect any discrepancies between the design specification and the actual implementation.

**d) Testing:**

Testing plays a critical role in quality assurance for software. Due to limitations of the verification methods for the previous phase, design and requirement faults also appear in the code. Testing is used to detect these errors, in addition to the errors introduced during the coding phase.

# 3.4.2 TOOLS/ENVIRONMENT USED

# SOFTWARE / HARDWARE REQUIREMENTS SPECIFICATION

**PLATFORM:**  Windows XP Professional

**FRONT END:** Visual Studio 2005.

**BACK END:** SQL Server 2005

**HARDWARE REQUIREMENTS:**

Intel Pentium III 733 MHz or Higher.

256 MB RAM or Higher.

**3.4.3 SYSTEM DESIGN**

# Data Structure

1. Table\_name: LOGIN\_DETAILS

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELD** | **SIZE** | **TYPE** | **DESCRIPTION** |
| Sr\_no | 4 | Int | Sr\_no |
| User\_id | 50 | Varchar | User\_id |
| Password | 50 | Varchar | Password |
| Role | 50 | Varchar | role |

1. Table\_name: ADMIN\_DETAILS

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELD** | **SIZE** | **TYPE** | **DESCRIPTION** |
| ADMIN\_ID | 4 | Int | Unique ID for Admin |
| ADMIN\_NAME | 50 | Varchar | Name of the admin |
| ADMIN\_TYPE | 50 | Varchar | Type of the admin |
| SHARES | 4 | int | No. of shares admin contains |
| COMPANY\_NAME | 200 | Varchar | Company name of which admin is buying shares |

1. Table\_name: SHARE\_DETAILS

(Admin Share Details)

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELD** | **SIZE** | **TYPE** | **DESCRIPTION** |
| ADMIN\_ID | 4 | Int | Unique admin ID |
| ADMIN\_NAME | 50 | varchar | Name of the admin |
| ADMIN\_TYPE | 50 | varchar | Type of admin |
| SHARES | 4 | int | Number of share admin contains |
| DATE | 10 | smalldatetime | date at which admin is selling his shares |
| OPENING\_RATE | 4 | int | The actual rate per share |
| CURRENT\_RATE | 4 | int | Rate at which admin is selling his shares |
| COMPANY\_NAME | 50 | varchar | Name of the company whose shares are sold by the admin |
| BROKER\_NAME | 50 | varchar | Name of the broker to which admin is selling his shares |

1. Table\_name:COMPANY\_DETAILS

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **FIELD** | **SIZE** | | **TYPE** | | **DESCRIPTION** | |
| COMPANY\_NAME | 50 | | Varchar | | Name of the company | |
| SHARES | | 4 | | int | | Number of share companycontains | |
| DATE | 10 | | smalldatetime | | date at which company having no. of shares | |
| RATE | 4 | | int | | Rate per share | |

1. Table\_name: BROKER\_DETAILS

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELD** | **SIZE** | **TYPE** | **DESCRIPTION** |
| BROKER\_NAME | 50 | Varchar | Name of the broker |
| BROKER\_CODE | 50 | Varchar | Unique code of the broker |
| PAN\_NO | 50 | Varchar | Pan No. of the broker |
| ADDRESS | 200 | Varchar | Address of the broker |
| COMPANY\_NAME | 200 | Varchar | Name of the company of which broker is buying shares |
| SHARES | 4 | Int | No of shares |
| OPENING\_RATE | 50 | Varchar | The actual rate per share |
| CURRENT\_RATE | 50 | Varchar | Rate at which broker is buying his shares |
| DATE | 10 | Smalldatetime | Date at which broker is buying shares |

1. Table\_name: BROKER\_SHARE\_DETAILS

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **FIELD** | | **SIZE** | **TYPE** | **DESCRIPTION** |
| BROKER\_NAME | | 50 | Varchar | Name of the broker |
| COMPANY\_NAME | | 50 | Varchar | Name of the company whose shares are sold by the broker |
| SHARES | | 4 | int | Number of share broker contains |
| DATE | 10 | | smalldatetime | date at which broker is selling his shares to client |
| BUY\_RATE | 4 | | int | The buying rate per share |
| CURRENT\_RATE | 4 | | int | Rate at which broker is selling his shares |

1. Table\_name: BROKER\_SELL\_CLIENT

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **FIELD** | | **SIZE** | **TYPE** | **DESCRIPTION** |
| BROKER\_CODE | | 4 | varchar | Unique code of the broker |
| BROKER\_NAME | | 50 | Varchar | Name of the broker |
| COMPANY\_NAME | | 50 | Varchar | Name of the company whose shares are sold by the broker |
| SHARES | | 4 | int | Number of share broker contains |
| DATE | 10 | | smalldatetime | date at which broker is selling his shares to client |
| BUY\_RATE | 4 | | int | The buying rate per share |
| CURRENT\_RATE | 4 | | int | Rate at which broker is selling his shares |
| CLIENT\_NAME | 50 | | Varchar | Name of the client to which broker is selling his shares |
| COMMISSION | 50 | | Varchar | Commission taken by broker from the client |
| SELLING\_RATE | 4 | | Int | It is current rate including commission of the broker |

1. Table\_name:LIST\_ITEMS

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELD** | **SIZE** | **TYPE** | **DESCRIPTION** |
| Sr\_no | 4 | Int | Sr\_no |
| BROKER\_NAME | 50 | Varchar | Name of the broker |
| CLIENT\_NAME | 50 | Varchar | Name of the client |

1. Table\_name:CLIENT\_DEATILS

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELD** | **SIZE** | **TYPE** | **DESCRIPTION** |
| CLIENT\_ID | 4 | Int | Unique client ID |
| CLIENT\_NAME | 50 | Varchar | Name of the client |
| ADDRESS | 200 | varchar | Address of the client |
| CONATCT\_NO | 50 | Varchar | Contact no of the client |
| SHARES | 4 | int | No. of shares client contains |
| COMPANY\_NAME | 100 | Varchar | Name of the company of which client is buying the sahres |
| OPENING\_RATE | 50 | Varchar | Actual rate of shares |
| CURRENT\_RATE | 50 | Varchar | Current rate of shares |

1. Table\_name:CLIENT\_SHARES

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELD** | **SIZE** | **TYPE** | **DESCRIPTION** |
| CLIENT\_NAME | 50 | Varchar | Name of the client |
| CLIENT\_ID | 4 | Varchar | Unique ID of the client |
| COMPANY\_NAME | 50 | Varchar | Name of the company |
| SHARES | 4 | int | No. of shares client contains |
| DATE | 10 | smalldatetime | date at which client is selling his shares to admin |
| BUY\_RATE | 4 | int | The buying rate per share |
| CURRENT\_RATE | 50 | Varchar | Current rate of the shares |
| ADMIN\_NAME | 50 | varchar | Name of the admin |

**3.4.4 TESTING**

Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design, coding.

**Testing objectives**

1. Testing is a process of executing a program with the intent of finding an error.

2. A good test case is the one that has high portability of finding an as-yet undiscovered error.

3. A successful test is one that uncovers an as-yet discovered error.

The main objective here will be to design test cases to uncover different classes of errors and to do so with minimum amount of time and efforts. If testing is conducted successfully it will uncover errors in the software. Another advantage is that it demonstrates that software functions appear to be working according to the specifications and performance requirement have been met.

But Testing cannot show the absence of defects it can show only that software errors are present.

**Strategies used for software testing**

The software engineering process is viewed as spiral as shown:

**Unit testing** begins at the vortex of the spiral and concentrates at the each unit of the software as implemented in the source code. Unit testing assures each module tested individually functions properly as a unit.

**Integration testing** focuses on the design and construction of the software architecture. It is a systematic technique for constructing a program structure while conducting tests to uncover errors associated with interfacing. The objective is to take unit-tested module and build a program structure that has been specified as design.

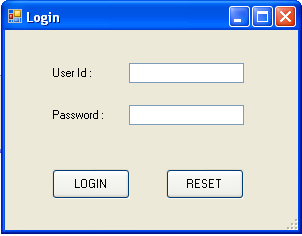
**Validation testing** takes care of the requirements established as part of the software requirements analysis are validated against the software that has been constructed. It is said to be successful when the software functions in a manner that can be reasonably expected by the customer.

**System testing** tests software and other system elements as a whole. These tests fall outside the scope of software engineering process and are not conducted solely by the software developer.

**3.4.5 DATA MODULES AND THEIR DESCRIPTION**

**Login:-**

* Enter User ID and password to login.
* Reset clears the fields so that the user can enter the username and password again.



**Master:-**

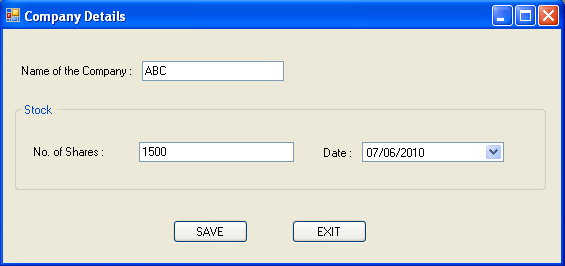
* Master form is the MDI form
* It contains 4 child forms as following:

1. Company
2. Broker
3. Client
4. Admin



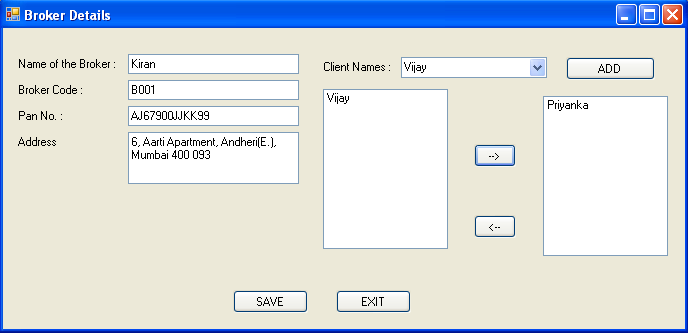
**Company details:-**

* This form contains details of the company
* And also number of shares it contains at particular date.



**Broker details:-**

* This form contains details of the broker such as broker name, broker code, pan no., address and list of customers.
* In this form, facility is provided to the broker to select the particular clients from the list and also to deselect the client from the list.



**Broker Shares:-**

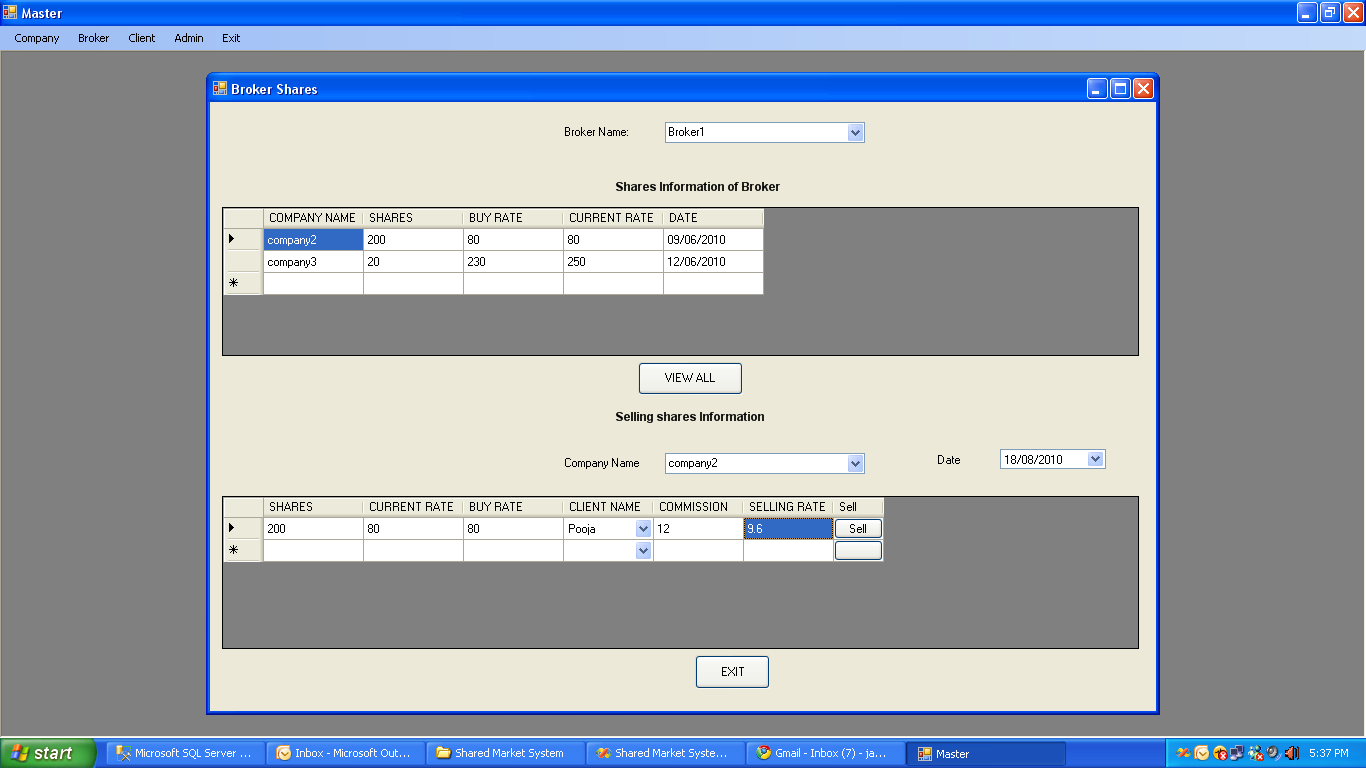
* In this form, details of the broker’s shares are maintained as well as detaisl of the transaction of shares from broker to client are maintained.
* In this form there are 2 main buttons, those are

1. VIEW ALL-

Using this button, all the information of broker’s share can be displayed with no. of share it contains, buying and current rate of the share, and which date all this transaction occurred.

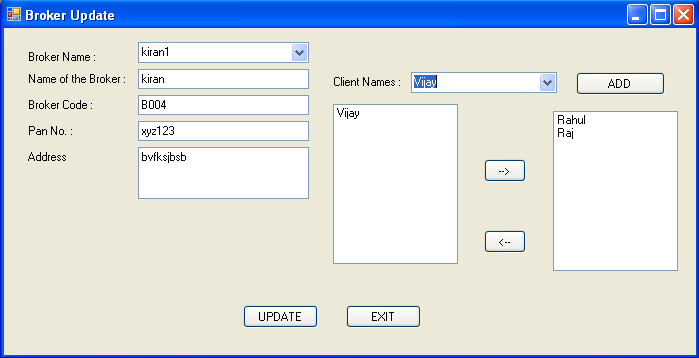
1. SAVE-

This button to save the data entered and all the calculation that is handled part that is in the datagridview.

* In the datagridview SELL button is given to sell the shares to the client including current rate and commission per share. 

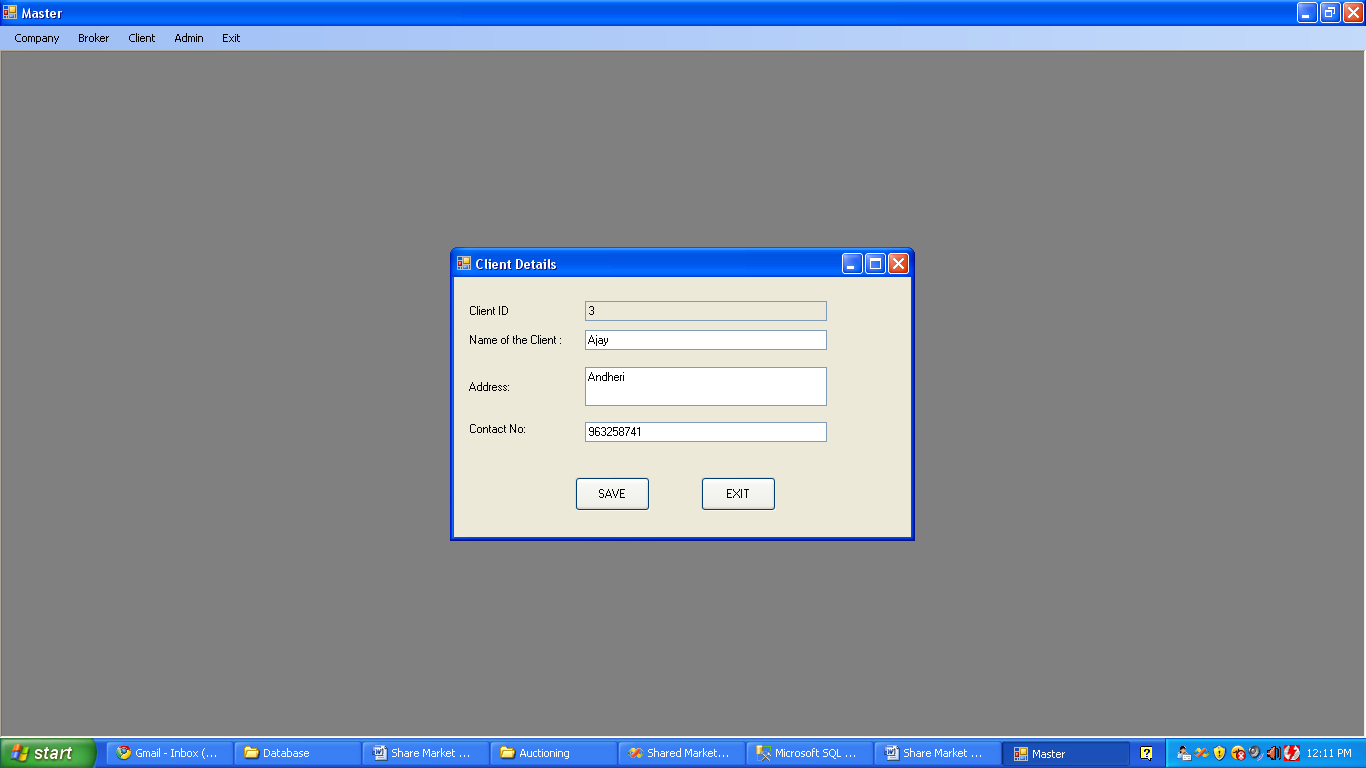
**Broker update:-**

* In this form, facility to update the details of the broker is provided as per the need of the broker.
* Broker can also update the client list already it contains and can also add new client to the selected list.



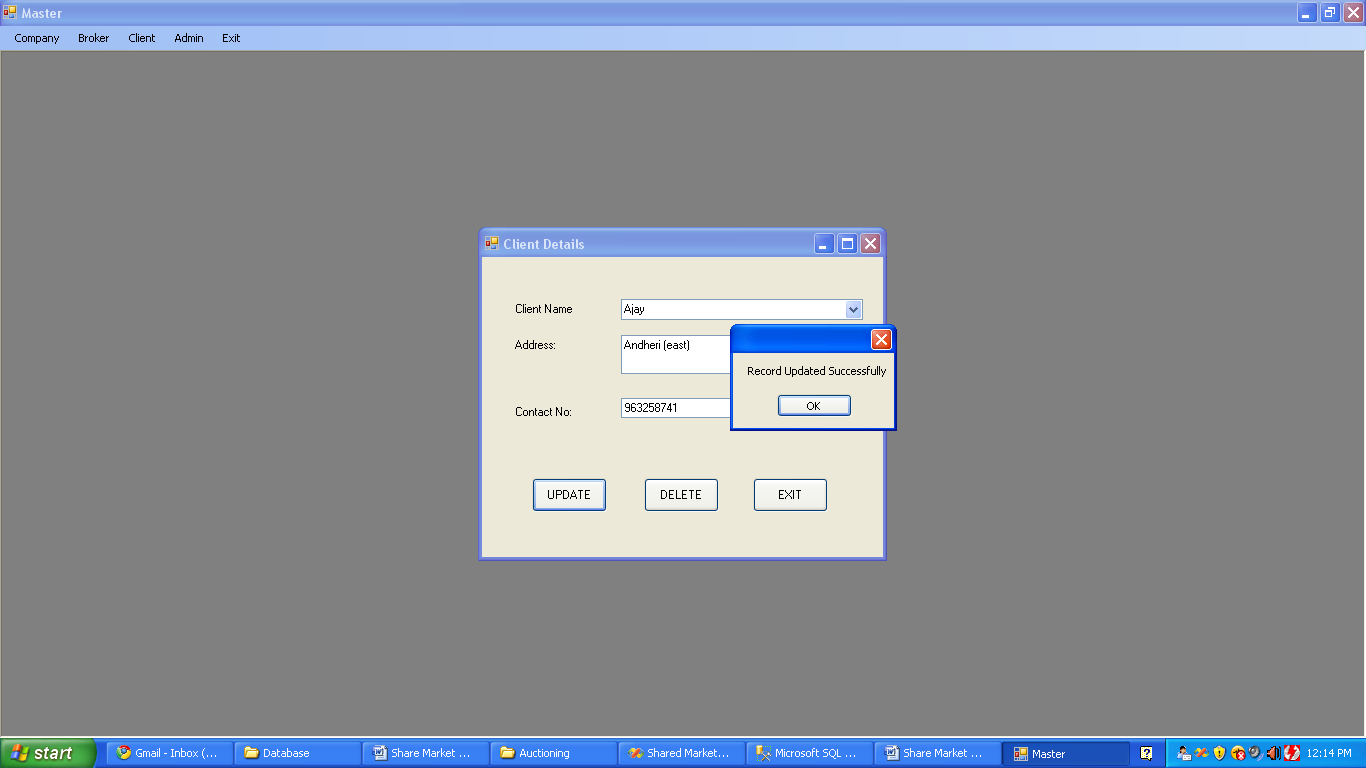
**Client Details:-**

* In this form, details of the client are maintained such as name of the client, address, contact no, etc.
* Client ID is auto generated.



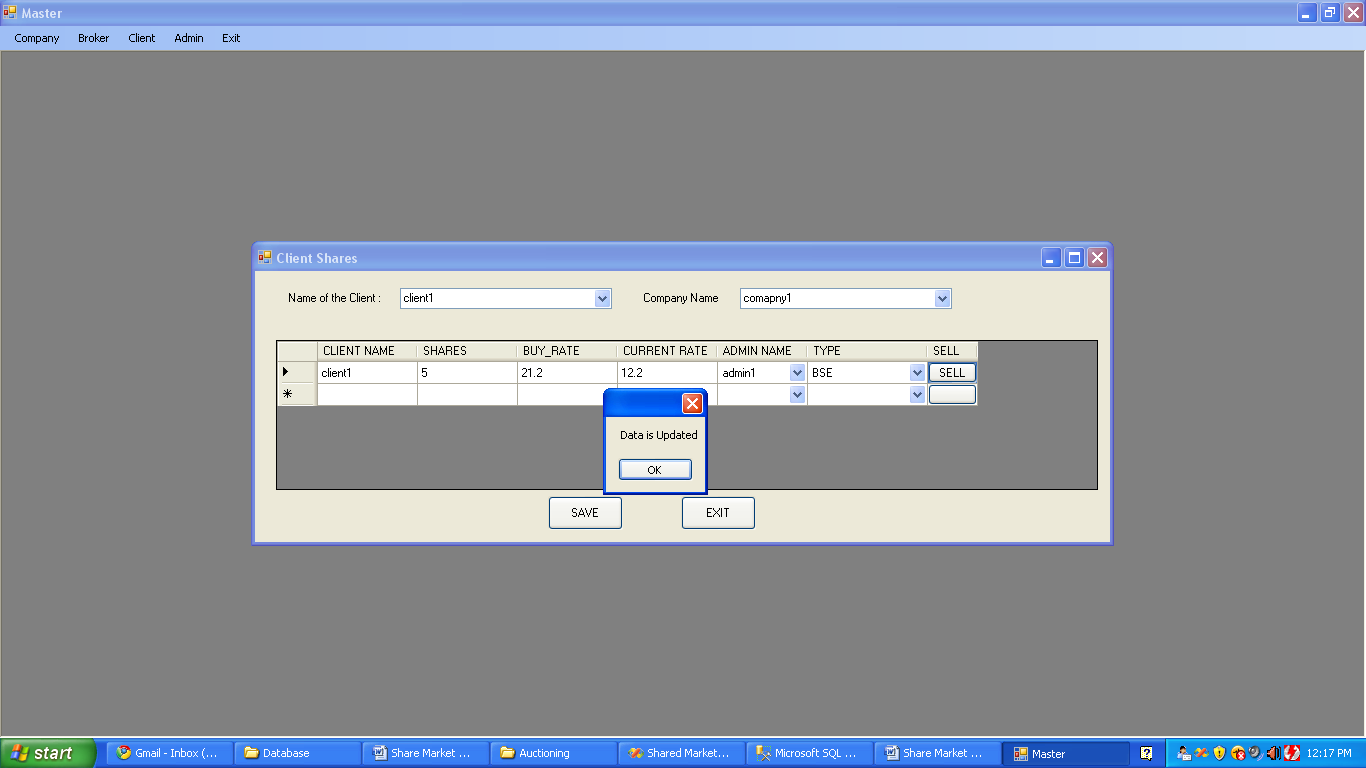
**Client Update:-**

* In this form, facility to update the details of the client is provided as per the need of the client.
* On click of Delete button the particular record would be deleted from the database.



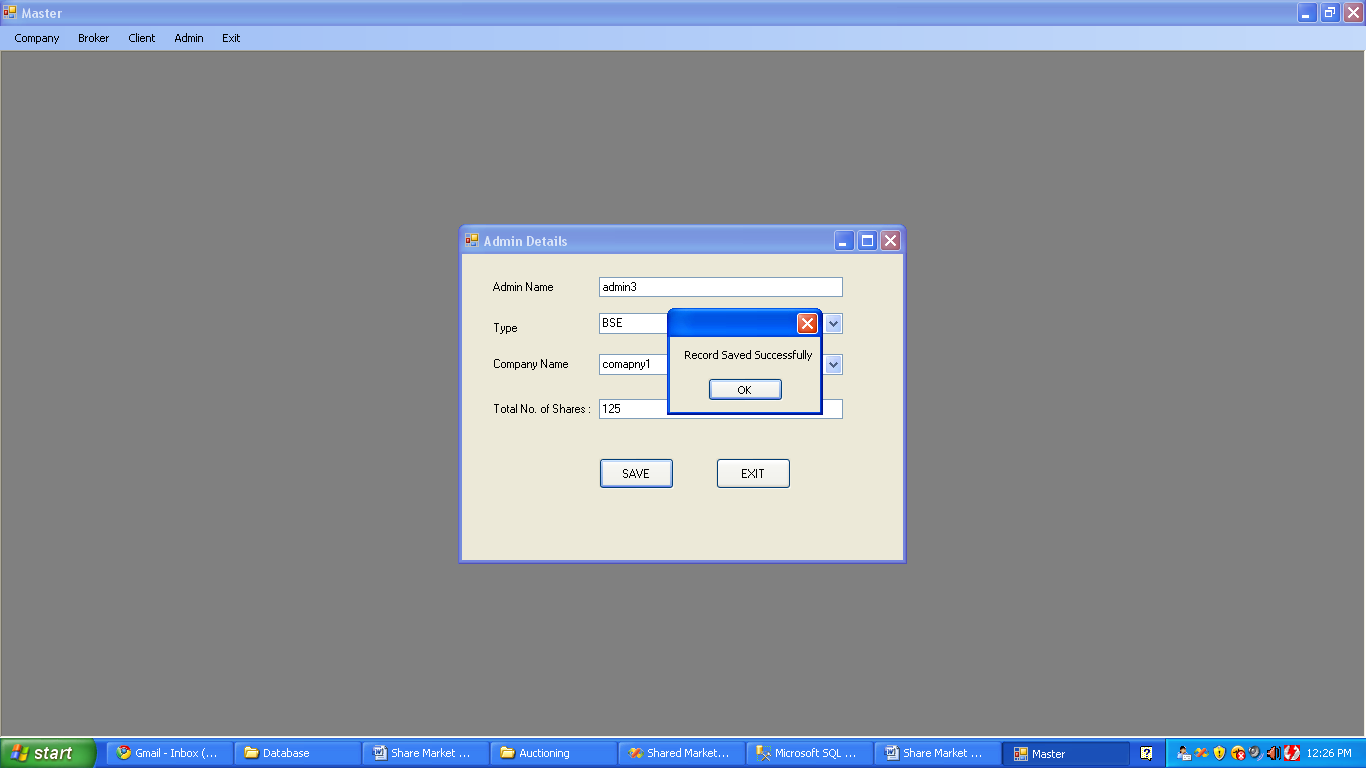
**Client shares:-**

* The shares details of the client are maintained here.
* On selecting the client name in first combobox the name of the companies of which that client is having shares are pop up in the second combobox.
* After selecting the company name the details such as no of shares, buy rate, current rate would be displayed in the datagridview below.
* The user has to select admin name and type from the datagridview.
* On click of sell button the selected shares would be sold to the selected admin and all details would be updated.



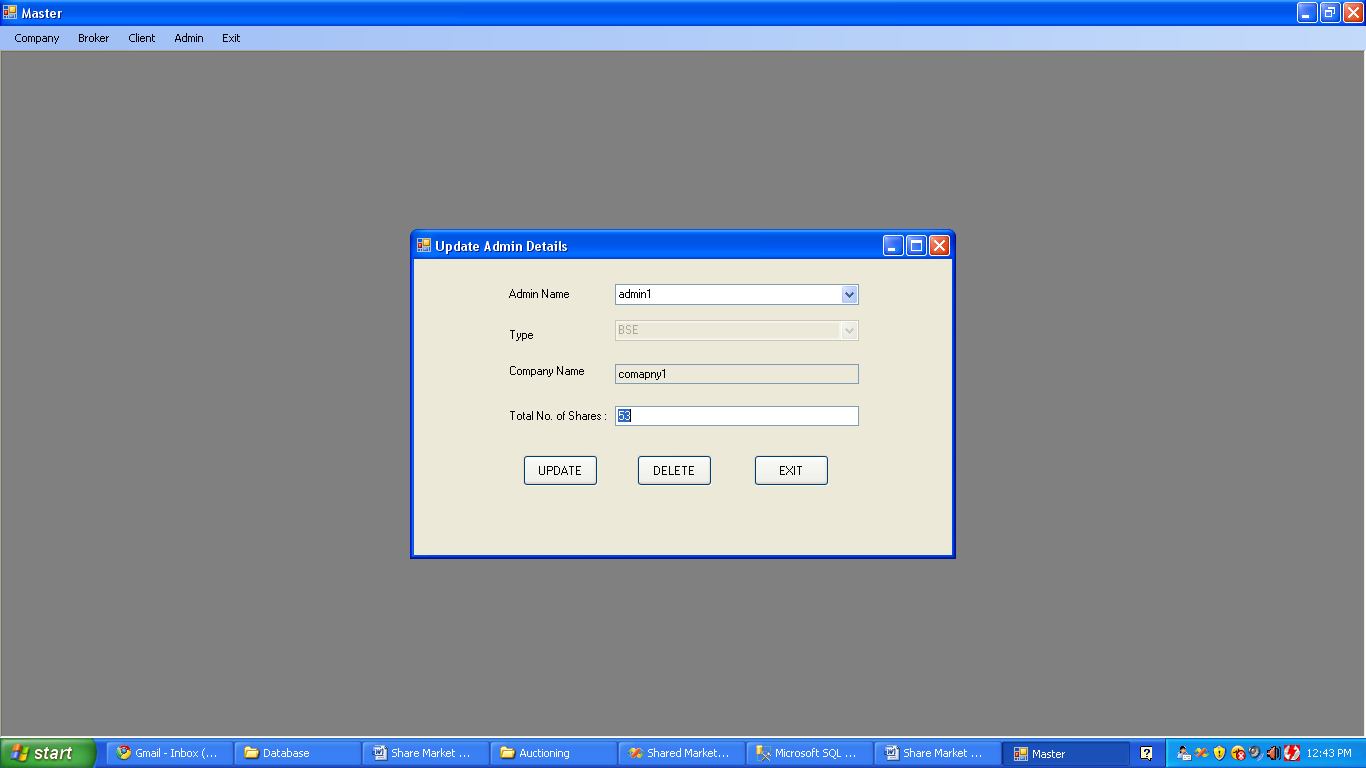
**Admin Details:-**

* Admin Details such as name, admin type, company name and no of shares are maintained here.
* On click of save button the record would be saved to the database.



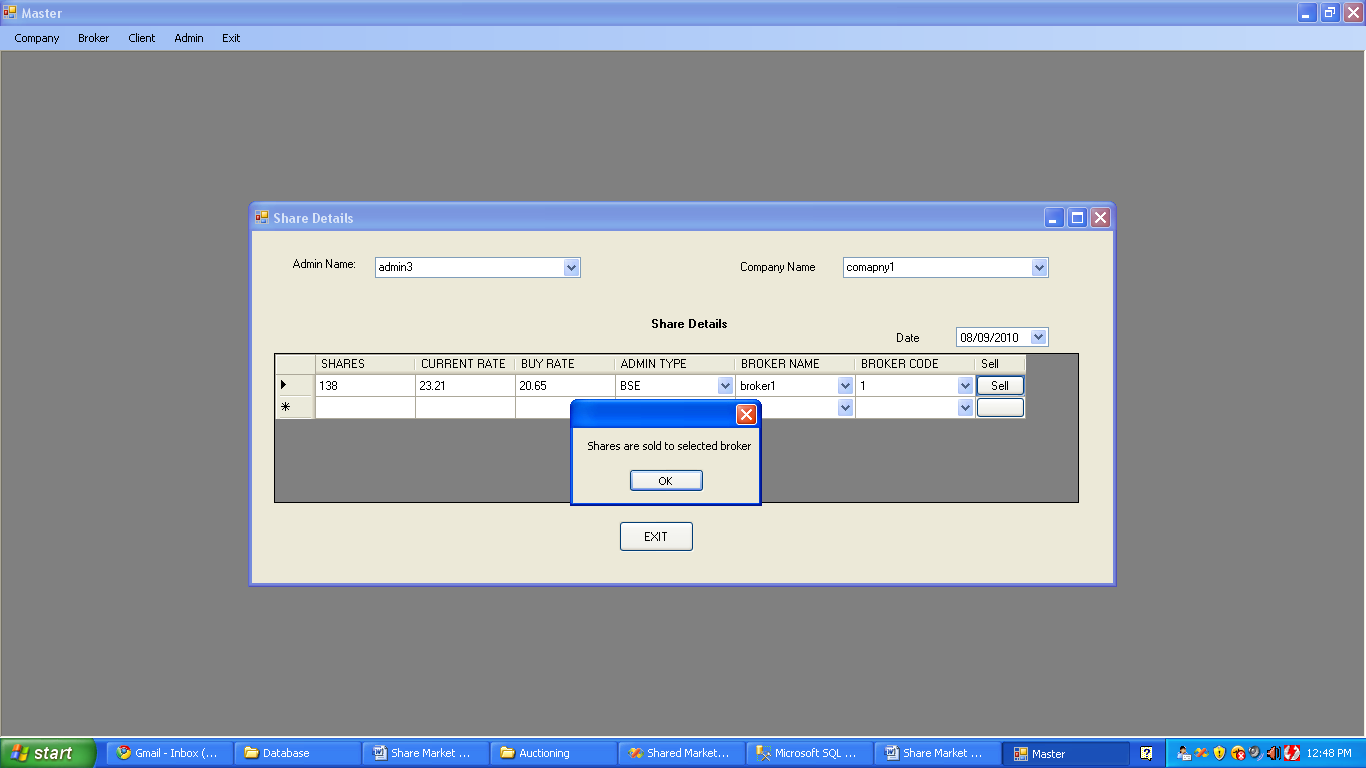
**Update Admin Details:-**

* Admin Details would be updated here.
* On click of update button the particular record would be updated and saved to the database.
* On click of Delete button the particular record would be deleted from the database.

****

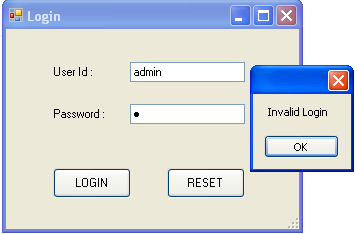
**Admin Shares Details:-**

* Admin shares details are maintained here.
* After selecting admin name from the first combobox company names of which the admin is having shares would be pop up in the second combobox.
* After selecting the particular company name the details such as no of shares, current rate, opening rate, admin type, etc would be displayed in the second datagridview below.
* The user has to select broker name and the code from the comboboxes.
* On click of sell button the shares would be sold to particular broker and the record would be updated and saved to the database.

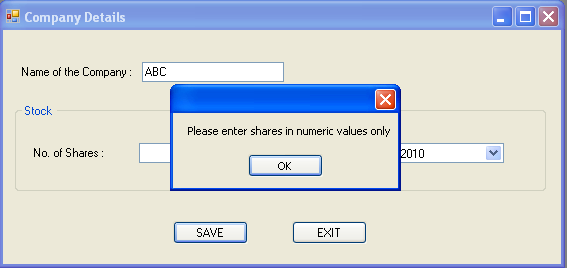
****

**Validation screens:-**

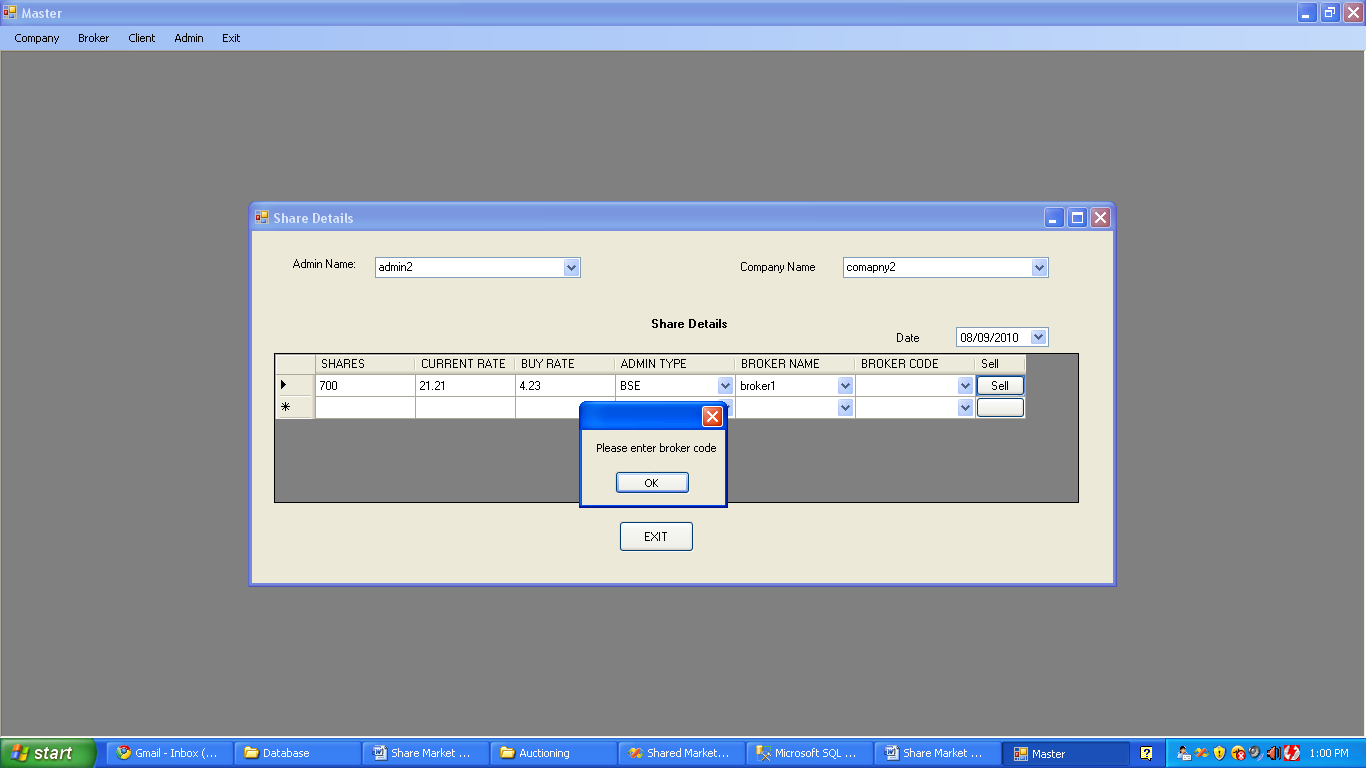
Login form validation:

****

Validation for numeric values in company\_details form

****

Validation to enter required field:

****

