Abstract—This project ‘Streaming File Sharing Media Communication and Intranet Chatting System’ (Social Networking Site) yields Server-Client architecture based chatting system that varies from the conventional chatting system. Users can interact publicly with everyone else on the server, or they can communicate privately using advance java Chat’s “whisper mode”. Users can create and manage chat rooms, which can be either "public" or "private". The server can also manage offline messages and files for user. Its good quality and security enhances concept of chatting to a feel of real meeting, sharing contents, portals for job seekers, news and updates corners. Advance Java Tools usage for betterment of all these is point to be focused.

I. INTRODUCTION

This Project ‘Streaming File Sharing Media Communication and Intranet Chatting System (Social Networking Site)’ is used for chatting purpose with the remote clients or users on Internet or local networks. Here in this project a java based client / server architecture is used to chat with end to end remote users. When a Client wants to chat with a user on a remote host, he sends a request to the Server with an identification name like chat-id, the server responds to the request by identifying the client-id which is already registered in the server domain and when matched his request is granted and the client can begin to chat with the remote users present on the internet or local network. The power of Internet is such that it integrates together LANs located across diverse software and hardware forms into a single large communication network that spans to globe. The user needs to have client software such as Netscape Navigator or Internet Explorer to retrieve information as well as to chat on the www. WWW is referred to as Netsurfing. And it supports chatting, text, video and audio. The plus of developing “Intranet Chatting” over other chatting tools is that, with the help of java, the programmer can create an applet application which can be used over the internet as a server.[2] Applets are machine independent and so java programs can run on any computer. The term client/server is used in the context of networking, what it actually means. It is important to understand the terms client/server because the Intranet Chatting System is supported completely by client/server model. A server is anything that has some resource that can be shared. There are compute servers, which provide computing power, web servers, which store web pages. A client is simply any other entity that wants to gain access to a particular server. The interaction between client/server is like an interaction between an electrical socket and a lamp. The server is a permanently available resource while client is free to “unplug” after it has been served.

A. WHY CHATTING SYSTEM?

Chatting system is a comfortable style of conversation and less costly services. People prefer to do chat via internet. In local networks chatting system is a good practice. When chatting is being on same network within a limited area this is called intranet chatting. When chatting are being on different networks or on internet then it is called online or internet chatting. But on internet chatting only need server. The server will responsible for that chatting.

II. PROBLEM ANALYSIS

Have you ever thought that we can chat with people residing at far of places i.e., remote chatting around the world now we can do this, that’s what technology is all about and you will be surprised that you can chat with people all over the globe just sitting at your PC, sounds interesting yes!, just at the click of a button [3] you can communicate and share information between different users who are present on their terminals at the time you logged in. All this can be done through a program called ‘Chat’ and the project ‘Intranet Chatting’ is all about chatting [4]. The program is called Chat, when you are Chatting, everything you type will instantly be transmitted around the world to other users that might be watching their terminals at the time. They can then type something and respond to your messages, and vice versa. Chatting is based on a client-server model. Clients are programs that connect to a server; a server is a program that transports data, (messages), from a user client to another. There are clients running on many different systems that allow you to connect to a Chat server. Currently most of the current Chat applications are text based and few are capable of transferring tiny images, but there lacks a Chat system for the technical people who can chat as well as they can represent their ideas in the form of drawing the pictures online. Many vendors distribute even these technologies separately but to obtain these features at one system will be the haunting task. So what should we do and how should we solve the problem that even the technical people are
benefited by this chatting technology. Through current chatting technologies we are able to send only text based messages to people and tiny images, but this type of chatting is not helping the technical people to work efficiently when the question of sending big pictures like say business plans to the customers that is the business clients to approve of the plan or the project client, imagine big business plans and deals getting approved through chatting and large business projects started, how beneficial it will be to the technical people as well the client on the other side. Let’s see how to solve this problem.

To solve the inconveniences as mentioned above, an “Streaming File Sharing Media Communication and Intranet Chatting System” is proposed. The proposed system’s interesting feature is its white board drawing utility. You can draw freehand, do circles, squares, lines, text, or paste image files to the canvas. [4] This is ideal when users want to “sketch” concepts for one another. This feature of “Intranet Chatting” can be a boon for the technical people who want to share their ideas or concepts in the pictorial form. With the help of the ‘White Board’ drawing utility now the technical people can carry out their tasks easily and can share their big picture plans regarding their business to the clients, exchange ideas and concepts and many more things, basically exchange as well as share the information along with the using the drawing utility even long conversations can be made between two users which may be important business meetings or deals to be sanctioned and all this is carried out with the support of applets with the help of image based web menu images can be transferred [2].

A. SCOPE AND STUDY OF THE PROJECT
This application is a java client/server combination, which can be used to chat over the internet or local networks with these features and with the advent of www, web browsers and with “intranet chatting”, internet has become the media of applications. This project “Intranet Chatting” is mainly depended on client/server model. The client requests the server and server responses by granting the clients request.

1) Server
The server should be able to perform the following features: The first and foremost problem is to find the server. We should identify the program in the server which processes the client’s request.

2) Administrator user
Creating of private room with the password facility to enable private chats with the users online. The server is always waiting for clients requests [1]. The clients come and go down but the server remains the same.

3) Client
The client should be able to perform the following features: Should be able to send message to anybody in the room with clients unique chat name created in the server for chatting purpose. In all the network applications, we find two sort program where the first i.e., [2] server sends the information and the second i.e., client receives the information.

4) Input and output
The user has to provide to which server it has to connect. In this, the server name to be provided is local host. The user has to provide the username and the password for proceeding with the server for chatting purpose [6].

III. REQUIREMENT GATHERING

Requirement list for the client/server side, for streaming and media communication is given in Table 1 below.

<table>
<thead>
<tr>
<th>SR: #</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SRS-01</td>
<td>Server’s TCP/IP Stack should be configured</td>
</tr>
<tr>
<td>SRS-02</td>
<td>Server should be connected to network</td>
</tr>
<tr>
<td>SRS-03</td>
<td>Server should registered the clients</td>
</tr>
<tr>
<td>SRS-04</td>
<td>Server should be configured the Listening port</td>
</tr>
<tr>
<td>SRS-05</td>
<td>Admin user should be able to control the server</td>
</tr>
<tr>
<td>SRS-06</td>
<td>Server Admin can create more admin</td>
</tr>
<tr>
<td>SRS-07</td>
<td>Server should be able to disconnect any one user</td>
</tr>
<tr>
<td>SRS-08</td>
<td>Server should be able to disconnect all users</td>
</tr>
<tr>
<td>SRS-09</td>
<td>Server should inform all users about its status before going to shutdown</td>
</tr>
<tr>
<td>SRS-10</td>
<td>Server should be able to shutdown</td>
</tr>
<tr>
<td>SRS-11</td>
<td>Client should be connected to server</td>
</tr>
<tr>
<td>SRS-12</td>
<td>Client could change the connection setting</td>
</tr>
<tr>
<td>SRS-13</td>
<td>Client is able to chat</td>
</tr>
<tr>
<td>SRS-14</td>
<td>Client is able to privately chat with selected users</td>
</tr>
<tr>
<td>SRS-15</td>
<td>Client should be able to create chat rooms</td>
</tr>
<tr>
<td>SRS-16</td>
<td>Client should be able to make private chat rooms</td>
</tr>
<tr>
<td>SRS-17</td>
<td>Client should be able to leave the chat room</td>
</tr>
<tr>
<td>SRS-18</td>
<td>Client can Copy the text</td>
</tr>
<tr>
<td>SRS-19</td>
<td>Client can Paste the text</td>
</tr>
<tr>
<td>SRS-20</td>
<td>Client can import picture</td>
</tr>
<tr>
<td>SRS-21</td>
<td>Client can export Canvas / whiteboard</td>
</tr>
<tr>
<td>SRS-22</td>
<td>Client can clear Canvas</td>
</tr>
<tr>
<td>SRS-23</td>
<td>Client should be able to page the user</td>
</tr>
<tr>
<td>SRS-24</td>
<td>Client should be able to boot the user</td>
</tr>
<tr>
<td>SRS-25</td>
<td>Client can change the boot setting</td>
</tr>
<tr>
<td>SRS-26</td>
<td>Client can invite the user</td>
</tr>
<tr>
<td>SRS-27</td>
<td>Client can modify the chat room setting</td>
</tr>
<tr>
<td>SRS-28</td>
<td>Client can see the another client’s information</td>
</tr>
<tr>
<td>SRS-29</td>
<td>Client can show/hide the Canvas</td>
</tr>
<tr>
<td>SRS-30</td>
<td>Client should able to draw different shapes in Canvas</td>
</tr>
<tr>
<td>SRS-31</td>
<td>Client can write text in Canvas</td>
</tr>
<tr>
<td>SRS-32</td>
<td>Client can leave chat room</td>
</tr>
<tr>
<td>SRS-33</td>
<td>Client can save messages log.</td>
</tr>
<tr>
<td>SRS-30</td>
<td>Client can read saved message</td>
</tr>
<tr>
<td>SRS-31</td>
<td>Client can see the status update of user.</td>
</tr>
</tbody>
</table>

A. CONSTRAINTS & BUSINESS REQUIREMENTS
This document contains the business requirements i.e. the needs of the organization with regards to the solution. Business requirements are necessary to persuade the solution that is offered and that must be delivered to capitalize on a certain business term opportunity. Following are the key sections in this regard [6,7].

B. THE COST BENEFIT ANALYSIS
This section discusses how productive and cost effective the solution is, the expectations from the user are none other than practicing their goals (the business goals) through robust and efficient solution. Moreover, this section
estimates the cost of the solution so that the customer is entertained with a proof of its decision [2].

C. AVAILABILITY / RELIABILITY & SECURITY
Access to Services and Solution section defines the business sponsor’s expectations of the solution’s availability. It describes the tolerance for availability versus cost. This section specifies the times periods that are acceptable for downtime and those where availability must exist [6]. This section describes the overall solution’s security vision and provides explanation of certain issues related to security matters

D. MULTIPLE DEVICE PERFORMANCE
This section defines the need to support the solution across many possible devices/form factors. It also identifies the acceptable and unacceptable constraints for feature degradation. This section defines the performance metrics the solution must meet across all user groups and deployments (i.e., screen load on a PDA must be under 10 seconds, complete with data population). This information should identify metrics for all key elements of the solution [10].

E. COST BENEFIT ANALYSIS - ROI
The Sharp X couriers need not only to manage their products and services for customers or outside world, also there are institutional matters. In fact the solution is meant to entertain shipments and related things by providing a solution for how well the courier company is internally able/strong to fulfill these outer needs. Sharp X couriers have different number and different kind of employees that are intended to use the solution. So every business rule is like a duty to perform and those which are required to be fulfilled with the help of the online application are well identified and should provide best possible solution in a certain given scenario. For example: Logging into the system is categorized by the branch. Hence the employees can use their respective legal domain in the big picture of the solution, right from the start [8]. Issuing discount in the solution needs verification by BDM. There are different options for issuing discount to the customer. Consignment number is most vital entity in the solution hence is generated for the general couriers. The solution maintains a check on number of records that are older or no more needed hence fulfilling the goal of saving necessary data, and auto or manual deletion. The user is not supposed to take headache of remembering rules and regulations because the solution is well compiled not allowing illegal moves as well as informative. Consignment number is vital here. The system records and processes shipment booking, billing details, discount, delayed shipments, online tracking information, quotations, online pick up requests, employee accounts, customer accounts, fixing accounts, sales record, packages, printing delivery sheets, and other necessary.

F. SECURITY FOR SOFTWARE ERA & DATABASES
Here is discussing [8] the three main areas according to security prospective which have its serious impact on software quality which are listing below

- Desktop application software
- Web site application software
- Network devices and real time system

Every Software development process has different phases. Best way is to start security testing from the start phase and remove it at that time so that it may not transfer to [9] the next phase as a bug which is open door for the hacker and then software leads to failure and quality touches the bottom of graph. Both developer and testing if anti-minded to each other then software must approaches almost quality target. For testing the security testing must be think like a hacker and try to crash the software it terms of data steeling, virus embedding in above three types as the hackers do, to know the mechanism that he follows and destroy the internal structure [10].

There are many approaches that hacker follows i.e. heavy processing software that can hang computer, SQL injection, anti-DDOS and special request to web sites etc. SQL injection is actually the following query that can display all the data in database base without any authorization (user name and password). So this is the violation of the standards and data privacy which looses the software confidence.

Query = “select * from USER_DATA_TABLE where 1=1”

This can be done by attaining the pattern of SQL line in the code then placed instead of them. In hackers language its mean everything is fine just display the data. Anti-DDOS system to destroy IP is used to hack the network and take advantages of the network devices. Firewall is safe too but sometimes its nothing in front of hacker, it means just relay on firewall is not enough. Now a day’s web sites are build in such a way that a big hole for hacker placed unintentionally which is due to inexperience developers and testers and no record keeping the tracing path and missing functionality. Databases are the central part of any software or website, it contains a lot of data which handles privacy of data and restricts the unauthorized access to prevent the system from failure and achieve the quality product. It must be maintained safe and sound from such kind of threat factors. The applications used by all users in wireless networks degrade the quality in contrast to the wired networks. More simultaneously active access points cause networks degrade the quality in contrast to the wired networks. More simultaneously active access points cause networks degrade the quality in contrast to the wired networks. Delay & jitters are perhaps the most focal for Voice application. Transmission of voice data packets must be guarantee. Source of the viruses are sexual media and untested software of local companies. In Pakistan “www.youtube.com” and all type of nudity is blocked by PTA but still free unblock proxy servers and software are available. Software security provided to software from 16 years is based on honey pots (these are the well arranged portion of the software left for the hackers with some deficiencies to note their activities. It reduces cost and complexity of both tester and developer but risk and time is the factor to be discussed. Code may be optimized and have considered the core functionality based testing.
IV. DESIGN

Fig. 1 Structural Class Diagram

V. IMPLEMENTATION

A. CODE EXPLANATION

The intranet chatting application is developed using awt (abstract window toolkit) and swing package, and the Structural Class Diagram is shown in Fig 1. The java.awt and javax.swing package is much useful for creating user interfaces and for painting graphics and images.[11] A user interface object such as a button or a scrollbar is called, in awt terminology, a component. The component class is the root of all awt components. Some components fire events when a user interacts with the components. A container is a component that can obtain components and other containers. A container can also have a layout manager that controls the visual placement of components in the container. The java.awt package is an old GUI package, the new [1,2] package is now javax.swing, and advance GUI components are given in it. We can change the look and feel of those components.

This package provides advanced features and advanced visualization components. The java.awt package implements different interfaces like LayoutManager, which defines the interface for classes that know how to layout containers. Paint interface defines how color patterns can be generated for graphics2d operations. A class implementing the paint interface is added to the graphics2d context in order to define the color pattern used by the draw and fill methods.

The java.net package provides the classes for implementing networking applications. Using the socket classes, one can communicate with any server on the internet or implement their own internet server. A number of classes are provided to make it convenient to use universal resource locators (URL) to retrieve data on the internet. The java.net package implements datagramsocketimplfactory for implementing data gram socket implementations. Classes datagramsocket to create actual socket implementation use it [5,9]. socketimplfactory interface defines a factory for socket implementations. It is used by the classes socket and
serversocket to create actual socket implementations. socketoptions interface of methods to get/set socket options.

B. OBJECT ORIENTED PROGRAMMING AND JAVA
Object-oriented programming was developed because of limitations found in earlier approaches of programming. To appreciate what OOP does, we need to understand what these limitations are and how they arose from traditional programming.

C. PROCEDURAL LANGUAGES
Pascal, c, basic, FORTRAN, and similar languages are procedural languages. That is, each statement in the language tells the computer to do something: get some input, add these numbers divide by 6, and display the output. A program in a procedural language is a list of instructions [10]. For very small programs no other organizing principle (often called a paradigm) is needed. The programmer creates the list of instructions, and the computer carries them out.

D. DIVISION INTO FUNCTIONS
When programs become larger, a single list of instructions becomes unwieldy. Few programmers can comprehend a program of more than a few hundred statements unless it is broken down into smaller units. For this reason the function was adopted as a way to make programs more comprehensible to their human creators. (The term functions are used in C++ and C. In other languages the same concept may be referred to as a subroutine, a subprogram, or a procedure.) A program is divided into functions, and (ideally, at least) each function has a clearly defined purpose and a clearly defined interface to the other functions in the program. The idea of breaking a program into functions can be further extended by grouping a number of functions together into a larger entity called a module, but the principle is similar:[12] grouping a number of components that carry out specific tasks. Dividing a program into functions and modules is one of the cornerstones of structured programming, the somewhat loosely defined discipline that has influenced programming organization for more than a decade.

E. PROBLEMS OF STRUCTURED PROGRAMMING
As programs grow ever larger and more complex, even the [15] structured programming approach begins to show signs of strain. You may have heard about, or been involved in, horror stories of program development. The project is too complex, the schedule slips, more programmers are added, complexity increases, costs skyrocket, the schedule slips further, and disaster ensures. Analyzing the reasons for these failures reveals that there are weaknesses in the procedural paradigm itself. No matter how well the structured programming approach is implemented, large programs become excessively complex. What are the reasons for this failure of procedural languages? One of the most crucial is the role played by data.

F. DATA UNDervalued
In a procedural language, the emphasis is on doing things—read the keyboard, invert the vector, check for errors, and so on. The subdivision of a program into functions continues this emphasis. Functions do things just as single program statements do. What they do may be more complex or abstract, but the emphasis is still on the action as the implication of structural class diagram is shown in Fig 1. What happens to the data in this paradigm? Data is, after all, the reason for a program's existence. The important part of an inventory program isn’t a function that displays the data, or a function that checks for correct input; it's the inventory data itself. Yet data is given second-class status in the organization of procedural languages. For example, in an inventory program, the data that makes up the inventory is probably read from a disk file into memory, where it is treated as a global variable. By global we mean that the variables that constitute the data are declared outside of any function, so they are accessible to all functions. [14] These functions perform various operations on the data. They read it, analyze it, update it, rearrange it, display it, and write it back to the disk, and so on.

We should note that most languages, such as paschal and c, also support local variables, which are hidden within a single function. But local variables are not useful for important data that must be accessed by many different functions. Now suppose a new programmer is hired to write a function to analyze this inventory data in a certain way. Unfamiliar with the subtleties of the program, the programmer creates a function that accidentally corrupts the program. [13] This is easy to do, because every function has complete access to the data. It's like leaving your personal papers in the lobby of your apartment building: anyone can change or destroy them. In the same way, global data can be corrupted by functions that have no business changing it. Another problem is that, since many functions access the same data, the way the data is stored becomes critical. The arrangement of the data can't be changed without modifying all the functions that access it. If you add new data items, for example, you'll need to modify all the functions that access the data so that they can also access these new items. It will be hard to find all such functions, and even harder to modify all of them correctly. It's similar to what happens when your local supermarket moves the bread from aisle 4 to aisle 12. Everyone who [12] patronizes the supermarket must figure out where the bread has gone, and adjust their shopping habits accordingly.

G. THE OBJECT ORIENTED APPROACH
The fundamental idea behind object-oriented languages is to combine into a single unit both data and the functions that operate on that data. Such a unit is called an object. An object’s functions, called member methods in Java, typically provide the only way to access its data. If you want to read the item and return the value to you, you call a member function in the object.[8] It will read the item and return the value to you. You can’t access the data directly. The data is hidden, so it is safe from accidental modification. Data and its functions are said to be encapsulated into a single entity.
Data encapsulation and data hiding are key terms in the description of object-oriented languages. If you want to modify the data in an object, you know exactly what functions interact with it: the member functions in the object. [6] No other functions can access the data. This simplifies writing, debugging, and maintaining the program.

H. ABSTRACTION
An essential element of object-oriented programming is abstraction. Humans manage complexity through abstraction. For example, people do not think of a car as a set of tens of thousands of individual parts. They think of it as a well-defined object with its own unique behavior. This abstraction allows people to use a car to drive to the grocery store without being overwhelmed by the complexity of the parts that form the car [4]. They can ignore the details of how the engine, transmission, and braking systems work. Instead, they are free to utilize the object as a whole. Object-oriented concepts form the [10] heart of Java just as they form the basis for human understanding. It is important that you understand how these concepts translate into programs. As you will see, object-oriented programming is a powerful and natural paradigm for creating programs that survive the inevitable changes accompanying the life cycle of any major software project, including conception, growth, and aging.

I. ENCAPSULATION AND INHERITANCE
Encapsulation is the mechanism that ties together code and the data it manipulates, and keeps both safe from outside interference and misuse. One way to think about encapsulation is as a protective wrapper that prevents the code and data from being arbitrarily accessed by other code outside the wrapper. Access to the code and data inside the wrapper is tightly controlled through a well-defined interface. To relate this to the real world, consider the automatic transmission on an automobile. Since the inner workings of a class.

Polymorphism (from the Greek, meaning “many forms”) is a feature that allows one interface to be used for a general class of actions. The specific action is determined by the exact nature of the situation. Consider a stack (which is a last-in, first-out list). The algorithm that implements each stack is the same, even though the data being stored differs. In a non-object-oriented language, you would be required to create three difference sets of stack routines, with each set using different names. However, because of polymorphism, in Java you can specify a general set of stack routines that all share the same names [3]. This helps reduce complexity by allowing the same interface to be used to specify a general class of action. It is the compiler’s job to select the specific action (that is, method) as it applies to each situation. You, the programmer, do not need to make this selection manually. You need only remember and utilize the general interface [1,2,3]. People interface with encapsulated features on cars all the time [4]. The brake and gas pedals hide an incredible array of complexity with an interface so simple you can operate them with your feet.

VI. TESTING AND MAINTENANCE
1) WHITE BOX TESTING AND BLACK BOX TESTING
White-box testing (also known as clear box testing, glass box testing, transparent box testing, and structural testing) is a method of testing software that tests internal structures or workings of an application, as opposed to its functionality (i.e. black-box testing). In white-box testing an internal perspective of the system, as well as programming skills, are used to design test cases. The tester chooses inputs to exercise paths through the code and determine the appropriate outputs. This is analogous to testing nodes in a circuit, e.g. in-circuit testing (ICT) [2]. While white-box testing can be applied at the unit, integration and system levels of the software testing process, it is usually done at the unit level. It can test paths within a unit, paths between units during integration, and between subsystems during a system–level test. Though this method of test design can uncover many errors or problems, it might not detect unimplemented parts of the specification or missing requirements.

White-box test design techniques include:

- Control flow testing
- Data flow testing
- Branch testing
- Path testing
- Statement Coverage
- Decision Coverage

It was a good idea to do our stress testing early on, because it gave us time to fix some of the unexpected deadlocks and stability problems that only occurred when components were exposed to very high transaction volumes.
Black-box testing is a method of software testing that tests the functionality of an application as opposed to its internal structures or workings. This method of test can be applied to all levels of software testing: unit, integration, system and acceptance. It can also dominate unit testing as well.

1) Test Procedures
The tester is only aware of what the software is supposed to do, but not how i.e. when a certain input is entered, a certain output is returned; without being aware of how the output was produced in the first place.

2) Test Cases
Test cases are built around specifications and requirements, i.e., what the application is supposed to do. It uses external descriptions of the software, including specifications, requirements, and designs to derive test cases. These tests can be functional or non-functional, though usually functional. The test designer selects valid and invalid inputs and determines the correct output. There is no knowledge of the test object’s internal structure.

- Test design techniques
- Typical black-box test design techniques include:
  - Decision table testing
  - All-pairs testing
  - State transition tables
  - Equivalence partitioning
  - Boundary value analysis

This software is fully qualifying the Rules and Regulation of Black Box Testing. We have qualified it according to Black Box Testing technique.

VII. CONCLUSION

In this paper social networking websites development idea in capacity of Java advance technology along with high security prospective and with much good quality aspects is discussed. Resultant consists on authorize user can share data & communicate with each other via SMS, showing daily news, updates & job portal for job seekers, fully dynamic site having data about professional market technology & subjects, codes, entertainment & many facilities provided by admin. In this project, we used Advance JAVA tools like hibernate and Spring3.0 as front-end & XML as back-end with point of security and quality.

VIII. REFERENCES


