1. **INTRODUCTION**
   1. **ABSTRACT**

Chatting websites are widely popular among Internet Users. Many websites give users a way to chat anonymously with strangers which can be fun as well as interesting Now-a-days, websites also offer services such aa therapy and counseling via web chat. Our project intends to give the campus community an anonymous chat client which can be used to have discussion on various topics and also share documents and and related files among students.

**1.2 GOALS**

* *Communication:* To develop an instant messaging solution to enable users to seamlessly communicate with each other.
* *User Friendliness:* The project should be very easy to use enabling even a novice person to use it.
  1. **PROBLEM STATEMENT**

To create a simple web chat client where users can login anonymously without registering. All the people within the community network will have access to it so that we have legitimate users. Users can login with a nickname and choosing an avatar. With this chat client we intend to reduce the gap between people and serve as a platform for sharing ideas and discuss on various topics.

Any number of users can be connected without any performance degradation.

**1.5 PROJECT OVERVIEW**

Different interfaces are created for both Login and Chat. The Project is divided into four modules. namely.

The Login module covers the task related to the user such as Login, Nickname and Avatar selection. The remaining modules we create an interface for the user to chat with other users and share files with other users. The Project consists for the following modules.

1. Login
2. Chat
3. Online User
4. File Sharing

**Module 1 : Login**

In this Login module, we register the details of the user like the username the user would like to select and the avatar image. In this module we create an interface for the customer in which each user will be provided with a name.

The user can only access the Chat Page only if his credentials are authenticated and are valid.

**Module 2 : Chat**

In this module the user is granted access to the Chat page where he can interact with other users and share files with them.

Since each user is granted a unique name, it is easy to identify a particular user from others.

**Module 3 : Online User**

When a user logins to the chat page, their has to be someway to known the number of users that had login, and their usernames.

So the main role of this module is to display the usernames of the number of users that had login.

**Module 4 : File Sharing**

In this module the functionality is to share files between users, and since this is a group chat application the files are shared simultaneously between users, if the file is an image the user can view the image with the inbuilt Lightbox Javascript plugin.

Else the user can download the files and view it on their computer.

**2. LITERATURE SURVEY**

**2.1 NODE.JS**

Node.js is an open-source, cross-platform JavaScript runtime environment for developing a diverse variety of tools and applications. Although Node.js is not a JavaScript framework, many of its basic modules are written in JavaScript, and developers can write new modules in JavaScript. The runtime environment interprets JavaScript using Google's V8 JavaScript engine.

Node.js has an event-driven architecture capable of asynchronous I/O. These design choices aim to optimize throughput and scalability in Web applications with many input/output operations, as well as for real-time Web applications (e.g., real-time communication programs and browser games).

**2.2 SOCKET.IO**

Socket.IO is a JavaScript library for real-time web applications. It enables real-time, bi-directional communication between web clients and servers. It has two parts: a client-side library that runs in the browser, and a server-side library for node.js. Both components have a nearly identical API. Like node.js, it is event-driven.

Socket.IO primarily uses the Web-Socket protocol with polling as a fallback option, while providing the same interface. Although it can be used as simply a wrapper for Web-Socket, it provides many more features, including broadcasting to multiple sockets, storing data associated with each client, and asynchronous I/O. Socket.IO provides the ability to implement real-time analytics, binary streaming, instant messaging, and document collaboration

**2.3 ANGULAR.JS**

Angular.js is a complete JavaScript-based open-source front-end web application framework mainly maintained by Google and by a community of individuals and corporations to address many of the challenges encountered in developing single-page applications.

The JavaScript components complement Phone-Gap, the framework used for developing cross-platform mobile apps. It aims to simplify both the development and the testing of such applications by providing a framework for client-side model–view–controller (MVC) and model–view–view-model (MVVM) architectures, along with components commonly used in rich Internet applications.

**2.4 BOOTSTRAP**

Bootstrap is a free and open-source front-end web framework for designing websites and web applications. It contains HTML- and CSS-based design templates for typography, forms, buttons, navigation and other interface components, as well as optional JavaScript extensions. Unlike many web frameworks, it concerns itself with front-end development only.

Bootstrap is the second most-starred project on GitHub, with more than 100,000 stars and 45,000 forks.

**2.5 JQUERY**

jQuery is a cross-platform JavaScript library designed to simplify the client-side scripting of HTML. jQuery is the most popular JavaScript library in use today, with installation on 65% of the top 10 million highest-trafficked sites on the Web. jQuery is free, open-source software licensed under the MIT License.

jQuery's syntax is designed to make it easier to navigate a document, select DOM elements, create animations, handle events, and develop Ajax applications. jQuery also provides capabilities for developers to create plug-ins on top of the JavaScript library. This enables developers to create abstractions for low-level interaction and animation, advanced effects and high-level, themeable widgets. The modular approach to the jQuery library allows the creation of powerful dynamic web pages and Web applications.

**3. SYSTEM ANALYSIS**

**3.1 SYSTEM OBJECTIVES**

Communication over a network is one field where this tool finds wide ranging application. This tool can be used for large scale communication and conferencing in an organization or campus of vast size, thus increasing the standard of co-operation. In addition it converts the complex concept of sockets to a user friendly environment. This web application can have further potentials, such as voice and video chatting options that can be worked upon later.

**3.2 RELATION TO EXTERNAL ENVIRONMENT**

This application helps in two major aspects –

* Anonymous Communication between users.
* Seamless transfer or exchange of text, docs, pdf’s, excel, mp3’s.

**3.3 DESIGN CONSIDERATIONS**

**Approach**:

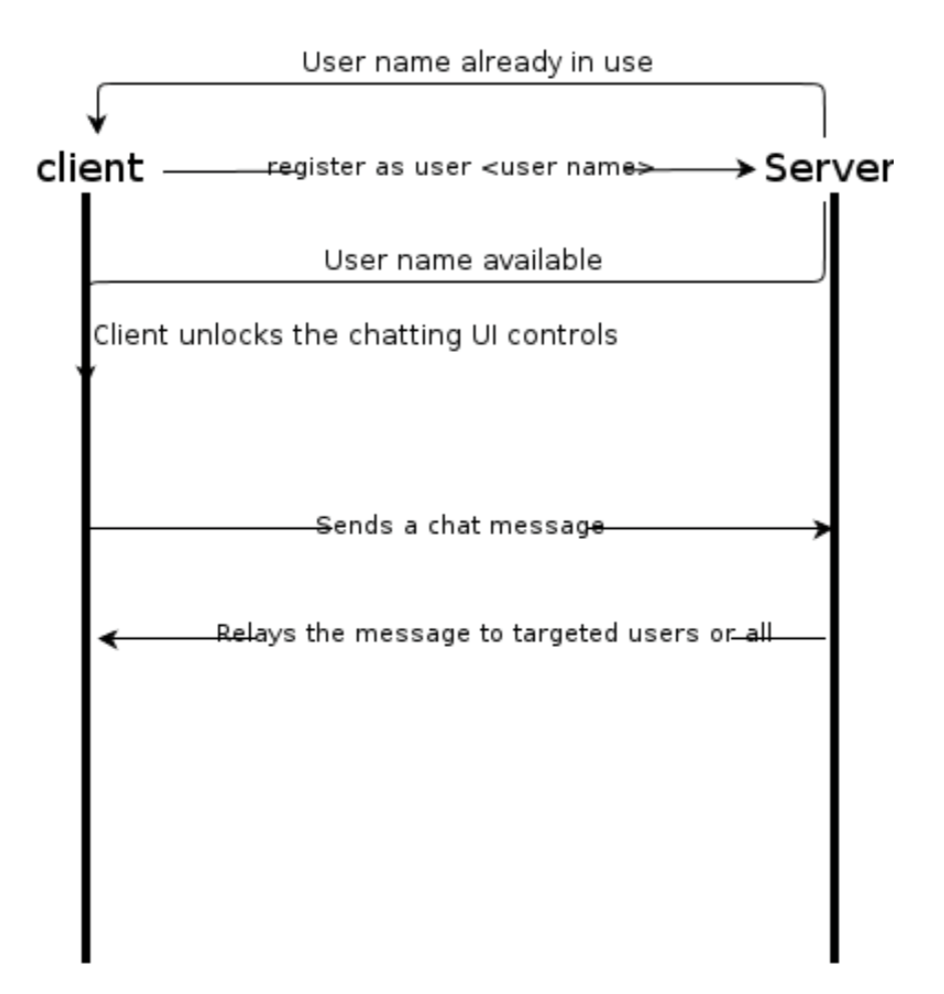
The application had been designed using JavaScript and JavaScript libraries using the text editor Sublime text.

**Methodology:**

The application has been created using node.js and socket.io. Node.js is a server side platform and its applications are written in JavaScript.

Socket.io helps to create a bi-directional and real-time communication between web clients and server. The client-side library of socket.io runs in the browser and the server side library is for node.js

The .json manifest file in the root folder describes our project. This file is used to give information to npm(node package manager which is the default package manager for node.js) that allows it to identify the project as well as handle the project’s dependencies.



**Working of the Node.JS Server:**

* It continuously loops on the code waiting for any user to connect.
* When a user connects it sends all the JavaScript files, HTML files and CSS files to the client.
* On initial connect it adds user to the Chat Room.
* The server loops for every connection made by all the clients.
* The server communicates with the client by emitting events like updating the message, sharing files like docs, pdfs, excel sheets, mp3’s.
* On disconnect the server discards that client and updates that room.

**Working of the Client:**

* On connecting to [http://ServerIP:8282](http://serverip:8282) the client get connection from the server that is continuously listening.
* The client is given a unique ID to identify itself to the sever.
* Initially the client is added by default to the Chat room.
* The client gets the HTML, CSS and JS files from the sever and displays the page for sending, receiving messages.
* The client communicates with the server by emitting events like updating the message, sharing files etc.
* The client is not allowed to make any server side changes, so the file permissions and checked and are given accordingly.

**System Architecture**

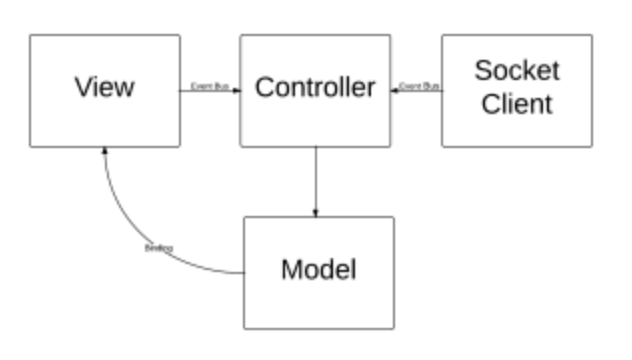
The chat application works in two views.

* **Login view:**

In this view, all the names of the systems connected to a network are enlisted. These names can later be used for communication with the help of mouse event, or in simple language: a click or a double click.

* **Chat view:**

This view is called only when an element is selected from the List view. In this form, a connection is created between the host system and the selected system with the help of a socket.



**3.4 SOFTWARE REQUIREMENT SPECIFICATION**

**3.4.1 Purpose**

This project is to create a chat application with a server and clients to enable the clients to chat with many other clients. This project is to simulate the multicast chatting. In the case of multicasting when a message is sent to a group of clients, then only a single message is sent to the router. Which means that the client will send the message only once and based on the location of the clients, the router will either pass the message to another router or if the clients are in the same local network the router will send a copy of the message to each client in that network. So this way we are reducing the number of messages being passed in the whole network.

**3.4.2 Scope**

This project is going to be a text and file communication software , it will be able to communicate between two computers using point to point communication. The limitation of this project is it does not support audio conversation. To overcome this we are concurrently working on developing better technologies. Data communication over networks is going to be most important mode of communication for the corporates. Companies have to rely on external networks which not only are unreliable but also cost inefficient. Companies would like to have a communication software wherein they can communicate instantly within their organization.

The fact that the software uses an internal network setup within the organization makes it very secure from outside attacks.

**3.4.3 Overall Description**

**Product Perspective**

While developing the new system all requirements of the end user was taken into consideration. There have been maximum efforts towards overcoming the drawbacks of the existing system, while the new system was designed & developed. When the existing system studied, it was found having some problems, existing system was very time consuming and was not efficient. The drawback of the existing system has resulted in to the development of new system, which is very user friendly and effective. Existing system was also very low in performance.

**Product Features**

* It is a centralized system.
* No database is required since it is an anonymous chatting application
* All clients are connected to the centralized server.
* This chat application can be used for group discussion.
* It allows users to find other online users.
* It allows sharing of files such as docs, mp3 audio, images, pdf’s and excel sheets.

**User Classes and Characteristics**

This system is useful for those who cannot afford to have an internet connection. For example: schools, colleges, small companies, etc.., Usually on connected networks conferencing is not possible. The proposed system allows the LAN users to create and participate in conference. This makes communications possible among number of LAN users simultaneously.

**Operating Environment**

Since it is an web application the client can use any operating system to access the application provided he has a compatible browser, but at the server end Linux operating system is preferred.

**3.4.4. System Specifications**

**Hardware Requirements**

In hardware requirements we require all those components which will provide us the platform for the development of the project. The minimum hardware required for the development of this project is aa follows –

**Ram – 1GB**

**Hard disk – 5GB**

**Processor – Dual Core**

**Compatible Ethernet/ LAN Card**

These all are the minimum hardware requirement required for the project. We want to make our project to be used in any. Type of computer therefore we have taken minimum configuration to a large extent. 1GB ram is used so that we can execute our project in a least possible RAM. 5GB hard disk is used because project takes less space to be executed or stored. Therefore minimum hard disk is used. Others enhancements are according to the needs.

**Software Requirements**

Software’s can be defined as programs which run on our computer. It acts as petrol in the vehicle. It provides the relationship between the human and a computer. It is very important to run software to function the computer. Various software’s are needed in this project for its development.

Which are as follows –

**Operating System – Linux**

**Text Editor – Sublime Text/ G-edit / E-macs**

**Others – Node.js – modules –**

**Socket.io**

**Express**

**Bootstrap**

**Light box**

We will be using sublime text as our editor because it is easier to use and provides features to the users which is used for the development of the project.

**4. SOFTWARE ARCHITECTURE**

**4.1 Socket Overview**

A socket is an object that represents a low level access point to the IP stack. This socket can be opened or closed or one of a set number of intermediate states. A socket can send and receive data down disconnection. Data is generally sent in blocks of few kilobytes at a time for efficiency; each of these block are called a *packet.*

All packets that travel on the internet must use the Internet Protocol. This means that the source IP address, destination address must be included in the packet. Most packets also contain a port number. A port is simply a number between 1 and 65,535 that is used to differentiate higher protocols. Ports are important when it comes to programming your own network applications because no two applications can use the same port.

Packets that contain port numbers come in two flavors: UDP and TCP/IP. UDP has lower latency than TCP/IP, especially on startup. Where data integrity is more important than performance; however, data sent by UDP can sometimes arrive in the wrong order and be effectively useless to the receiver. TCP/IP is more complex then UDP and has generally longer latencies, but it does guarantee that data does not become corrupted when travelling over the internet. TCP is ideal for file transfer, where a corrupt file is more unacceptable than a slow download; however, it is unsuited to internet radio, where the odd sound out of place is more acceptable then long gaps of silence.

The User Datagram Protocol is an unreliable, connectionless oriented protocol that uses an IP address for the destination host and a port number to identify destination application.

**4.2 SOFTWARE PROCESS MODEL**

The Software Process Model used is the Prototype Model. Prototyping Model is based on the idea of developing an initial implementation, exposing this to user comment and defining this through many until an adequate system had been developed.

**4.2.1 Benefits of Prototyping model**

The prototyping paradigm begins with requirements gathering. Developers and customers meet and define the overall objective for the software, identify the requirements and outline the areas where further definitions are necessary. The prototype design is often, quite different from that of the final system.

The benefits of developing a prototype early in the software process are :

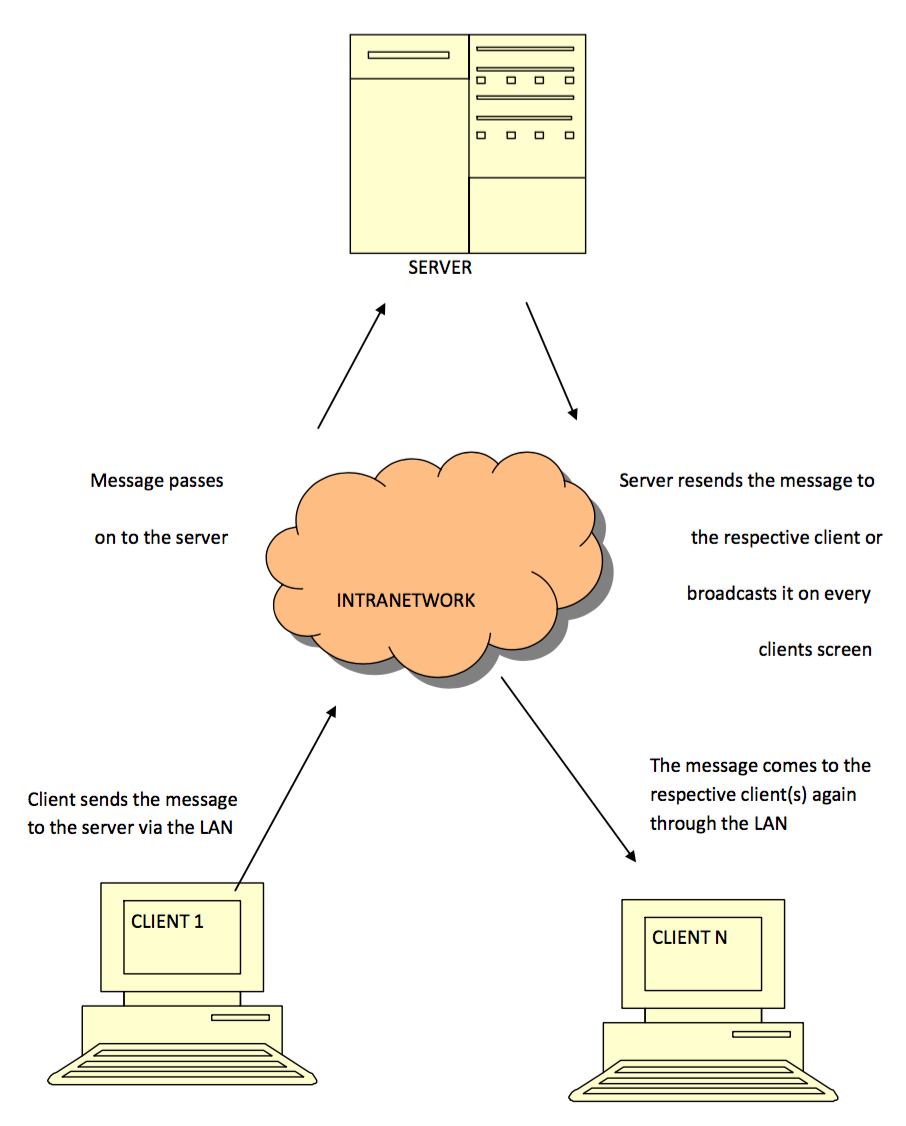
* Misunderstanding between software developers and users may be identified, as the functions are demonstrated.
* Missing user services may be detected.
* Difficult to use or confusing user services may be identified and refined.
* Software development staff may find incompleteness and inconsistency in requirement as the prototype is developed.
* A working albeit limited systems is available quickly to demonstrate the feasibility and usefulness of the application to the management.
* The prototype serves as a basis for writing the specification for a production quality system. Though the principle purpose of prototyping is to validate software requirements, software prototype also has other uses.
* A prototype system can be used for training users before the formal system has been delivered.

Prototype can run back-to-back tests. This reduces the need for tedious manual checking of test run. The same test is given to both the prototype and the system under test to look for differences in the final results and thereby making necessary changes. Thus prototype serves as a technique of risk reduction.

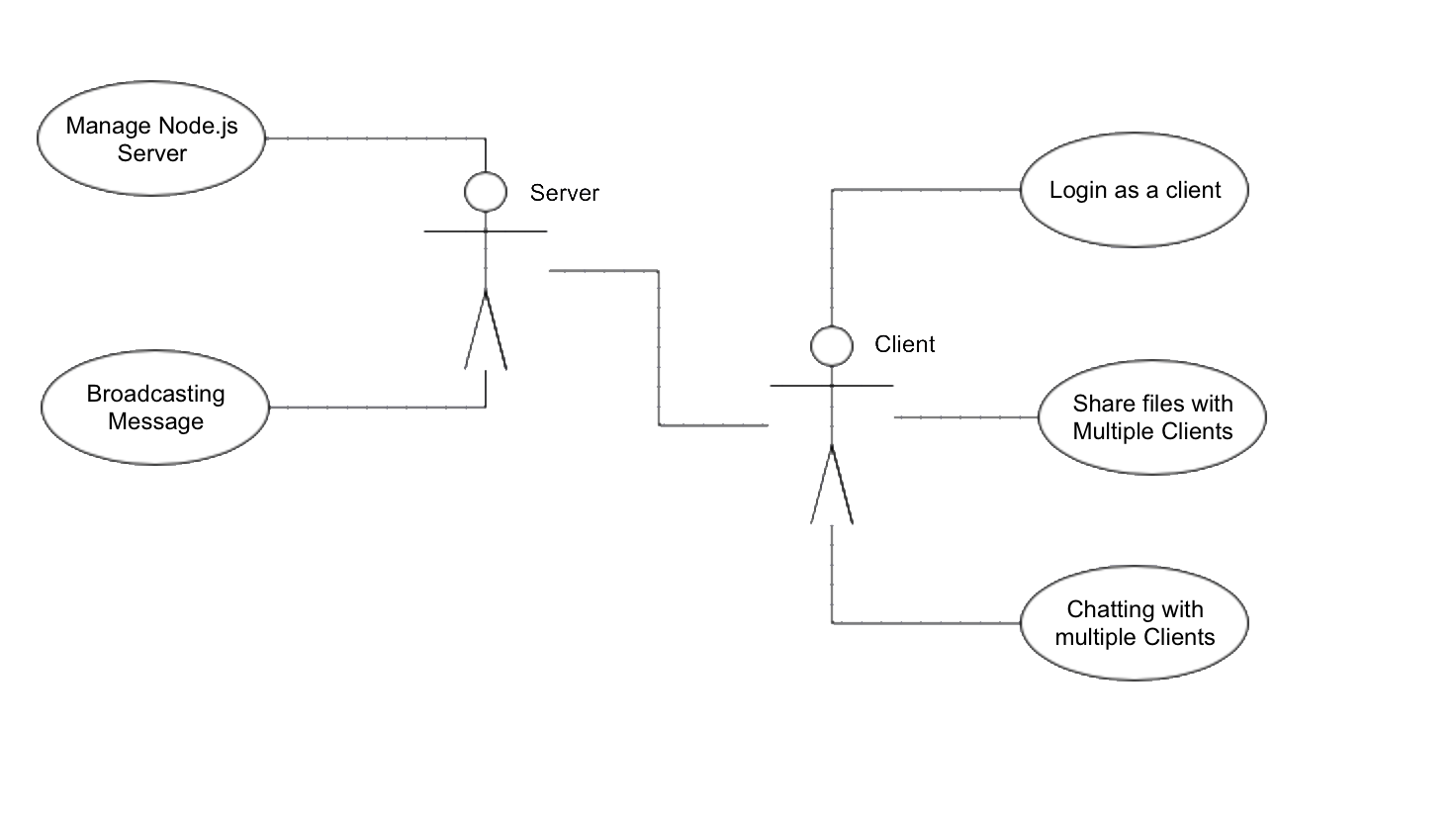
**4.2.2 Selecting the prototype approach**

The prototype paradigm can be either close-ended (throwaway prototyping) or open-ended (evolutionary prototyping), Before selecting closed or open-ended approach, it is necessary to determine whether the system to be built is suitable for prototyping or not. This is decided depending on application area, complexity, customer characteristics and projects characteristics. The throwaway is developed to understand the system requirements while the evolutionary prototype evolves through a number of versions to the final system.

**4.2.3 Entity Relationship Diagram**

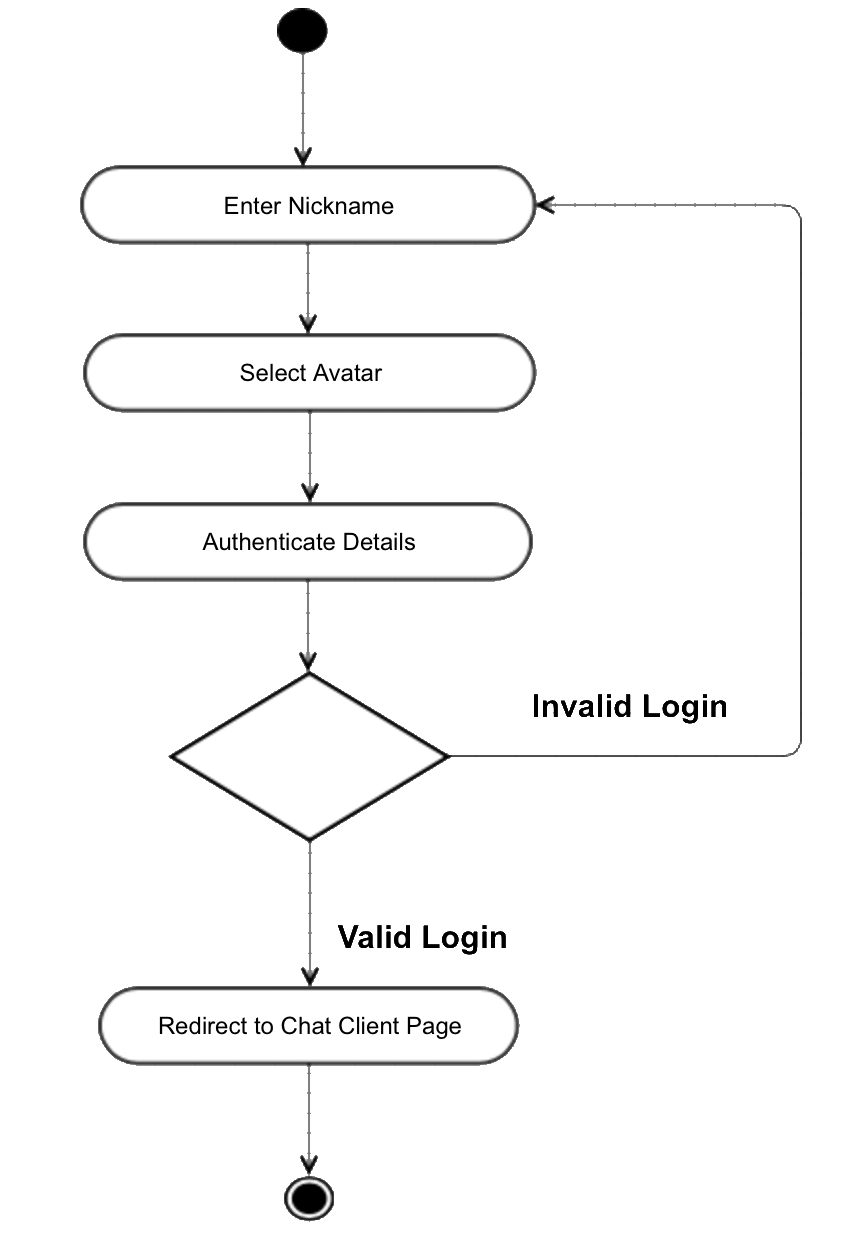


**4.2.4 USE Case Diagram**

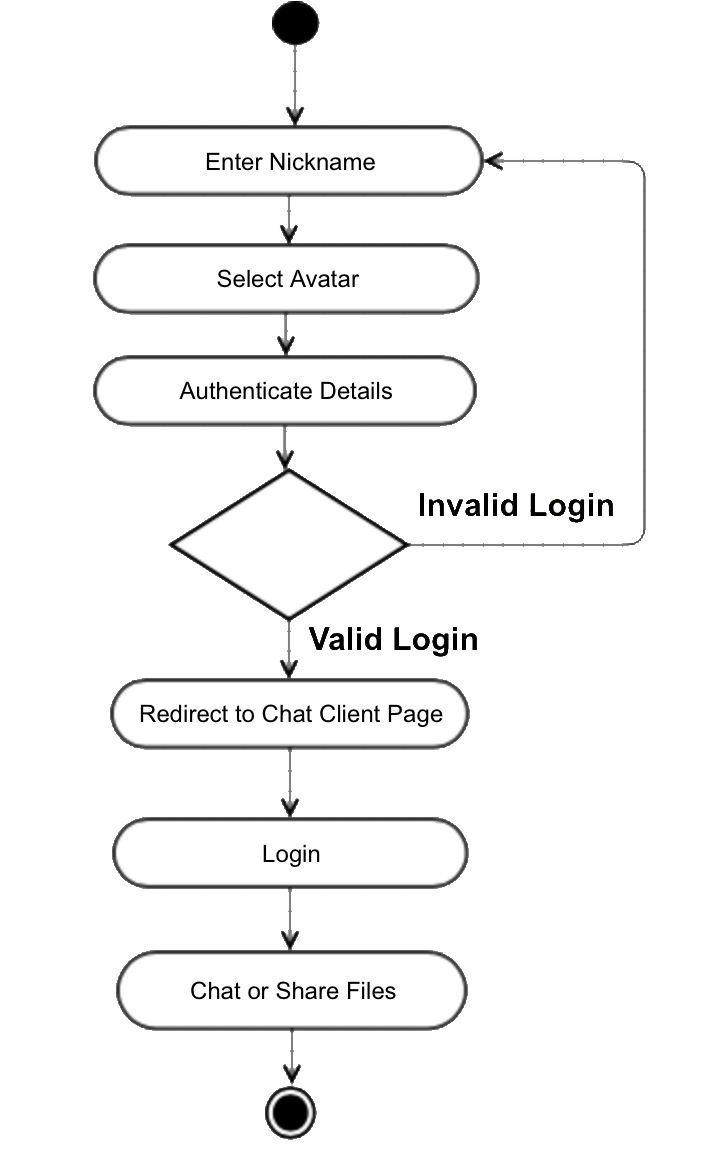
**USE Case Scripts**

* Managing Node.js Server-user inserts his personal data (login, avatar)
* Verification is taking place after submitting data.
* Verification error is signalized by error message.
* If verification doesn’t return error, user is allowed to system.
* Broadcasting Messages called by Server
* Server received the message from one client and broadcast it to other client.
* Logging as a client – inserts his personal data (login name, avatar)
* Verification takes place after submitting data.
* Verification error is signalized by error message, then user is allowed.
* Chatting with multiple client
* Client can chat with multiple client.

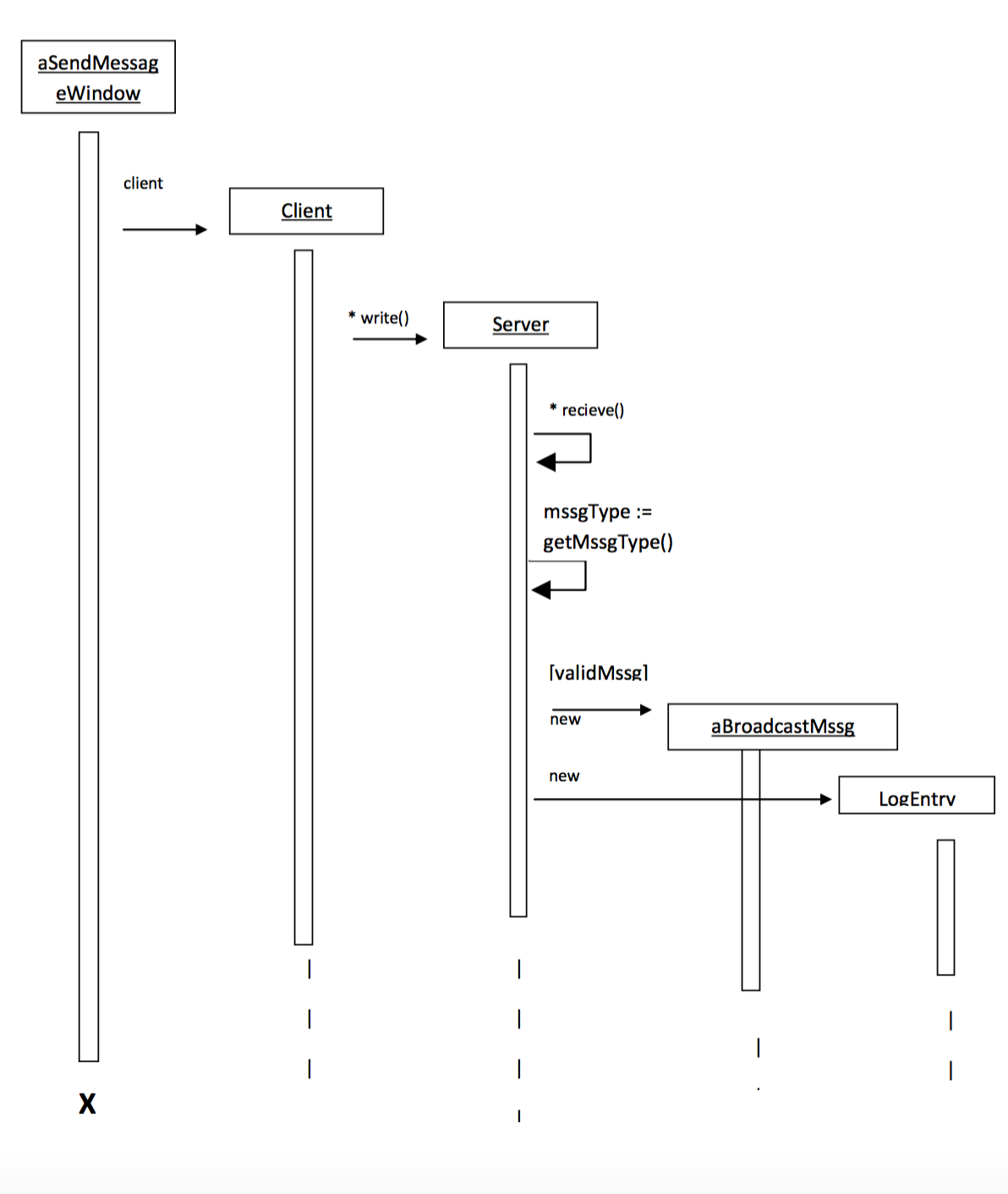
**4.2.5 Activity Diagram for Login**



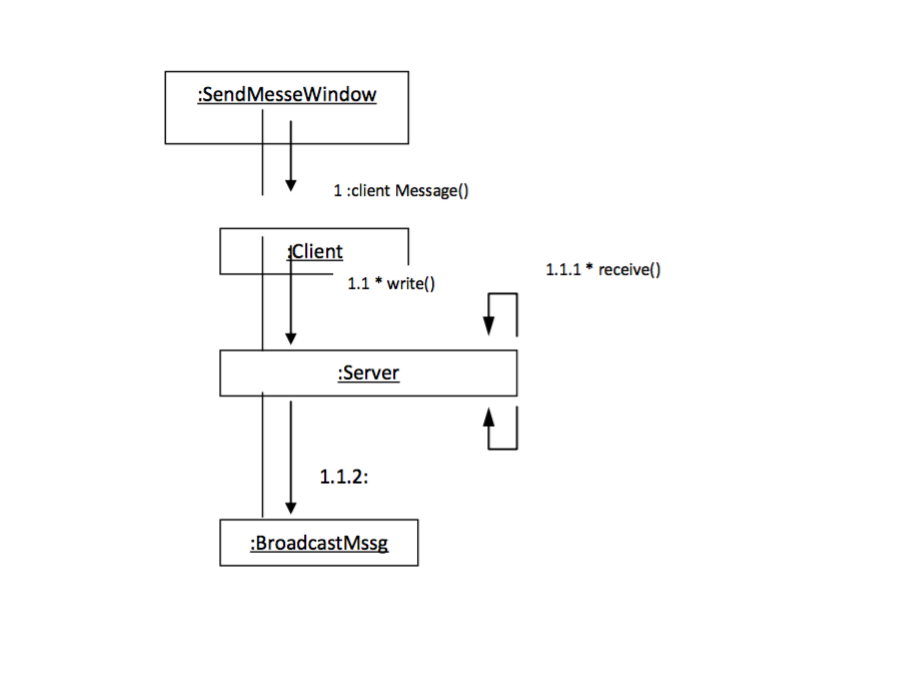
**Activity Diagram for Chat page**



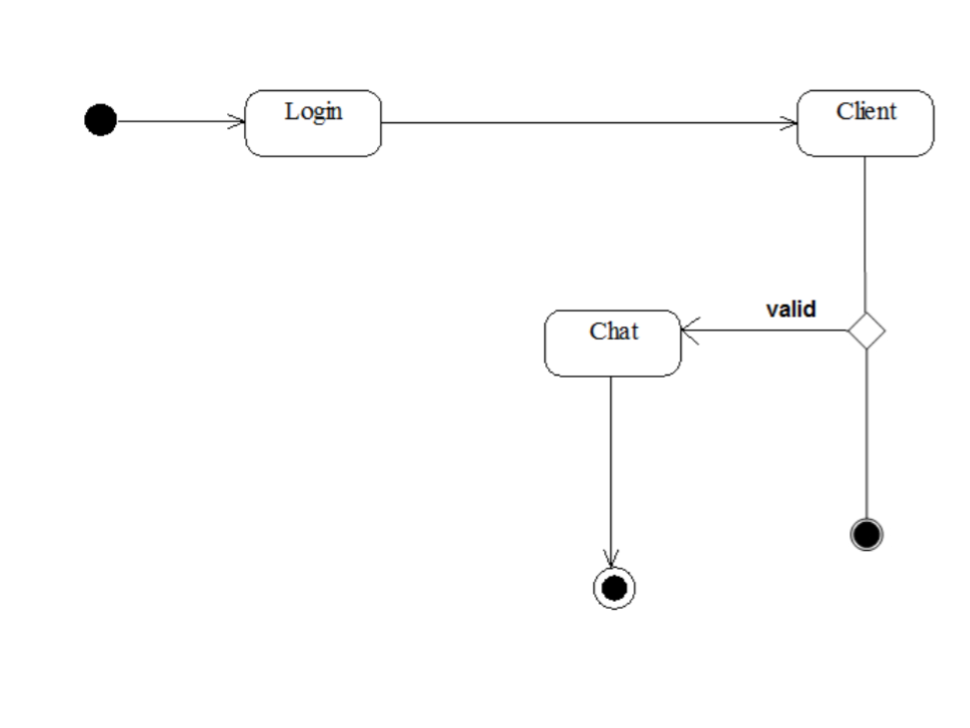
**4.2.6 Sequence Diagram**



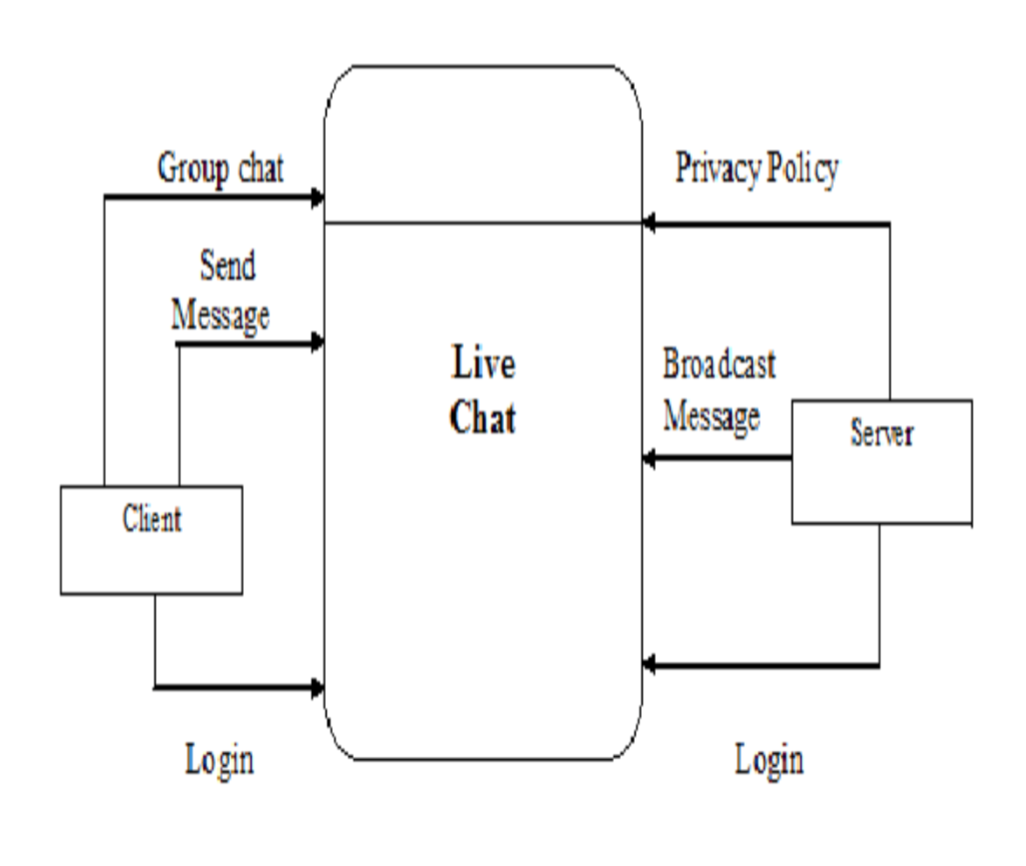
**4.2.7 Collaboration Diagram**



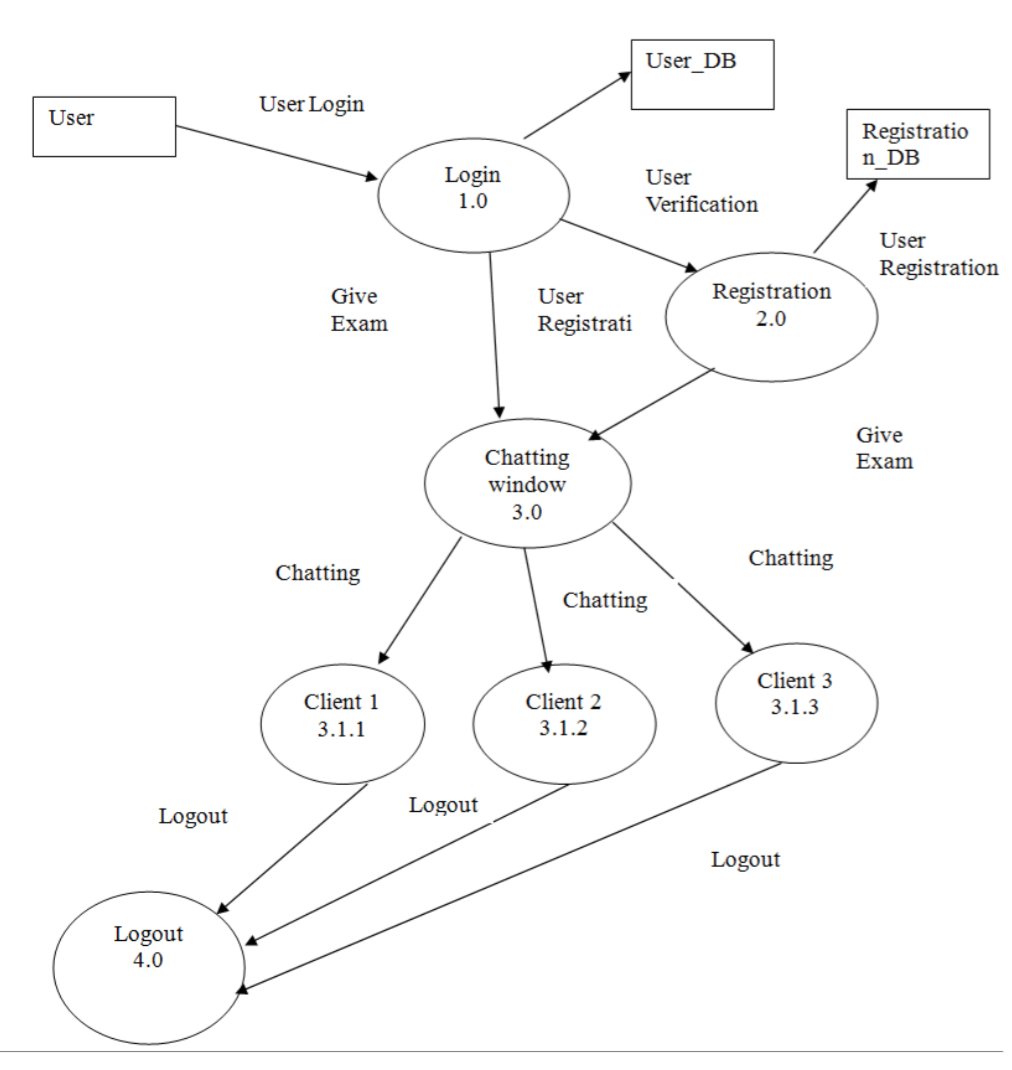
**4.2.8 State Chart Diagram**



**4.2.9 Context Level Diagram**

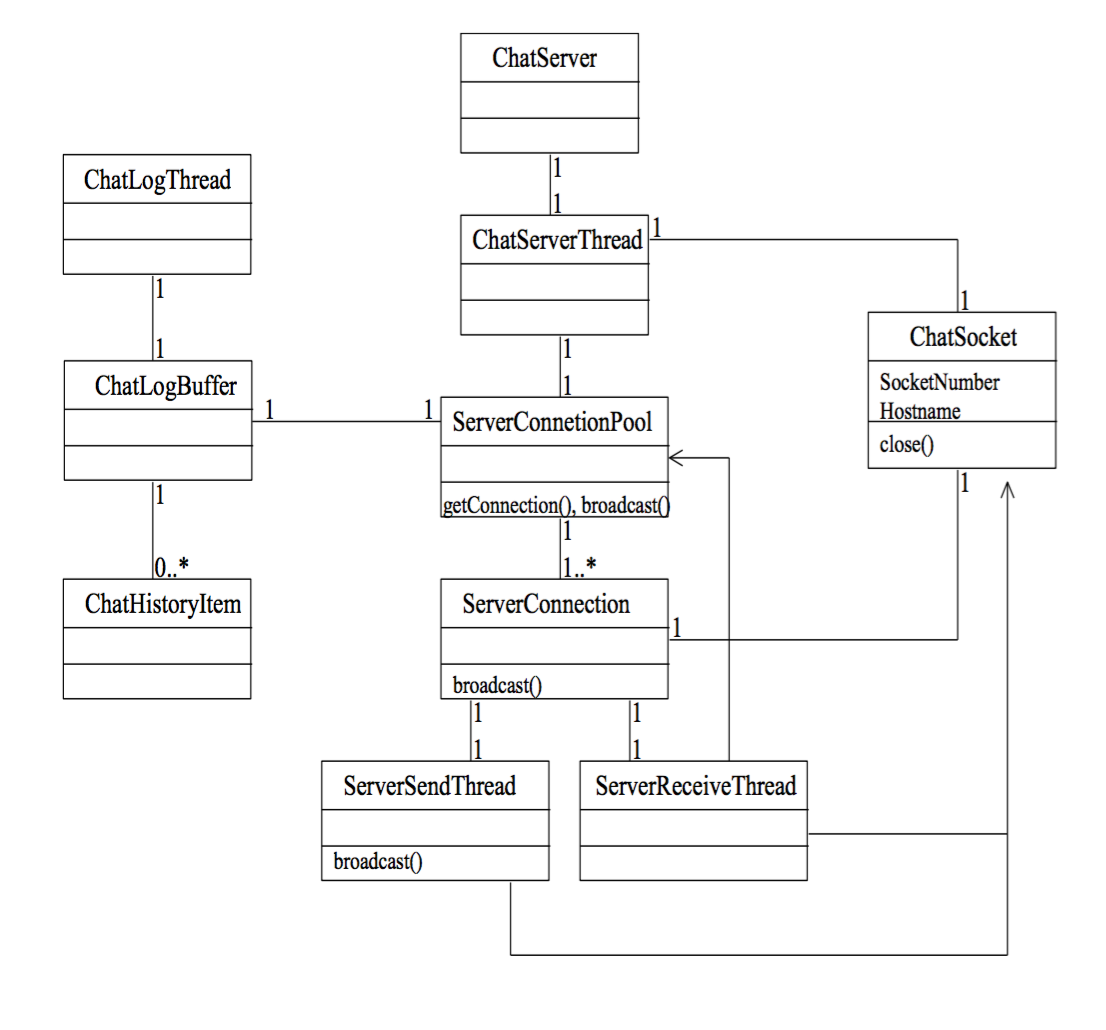


**4.2.10 Dataflow Diagram**

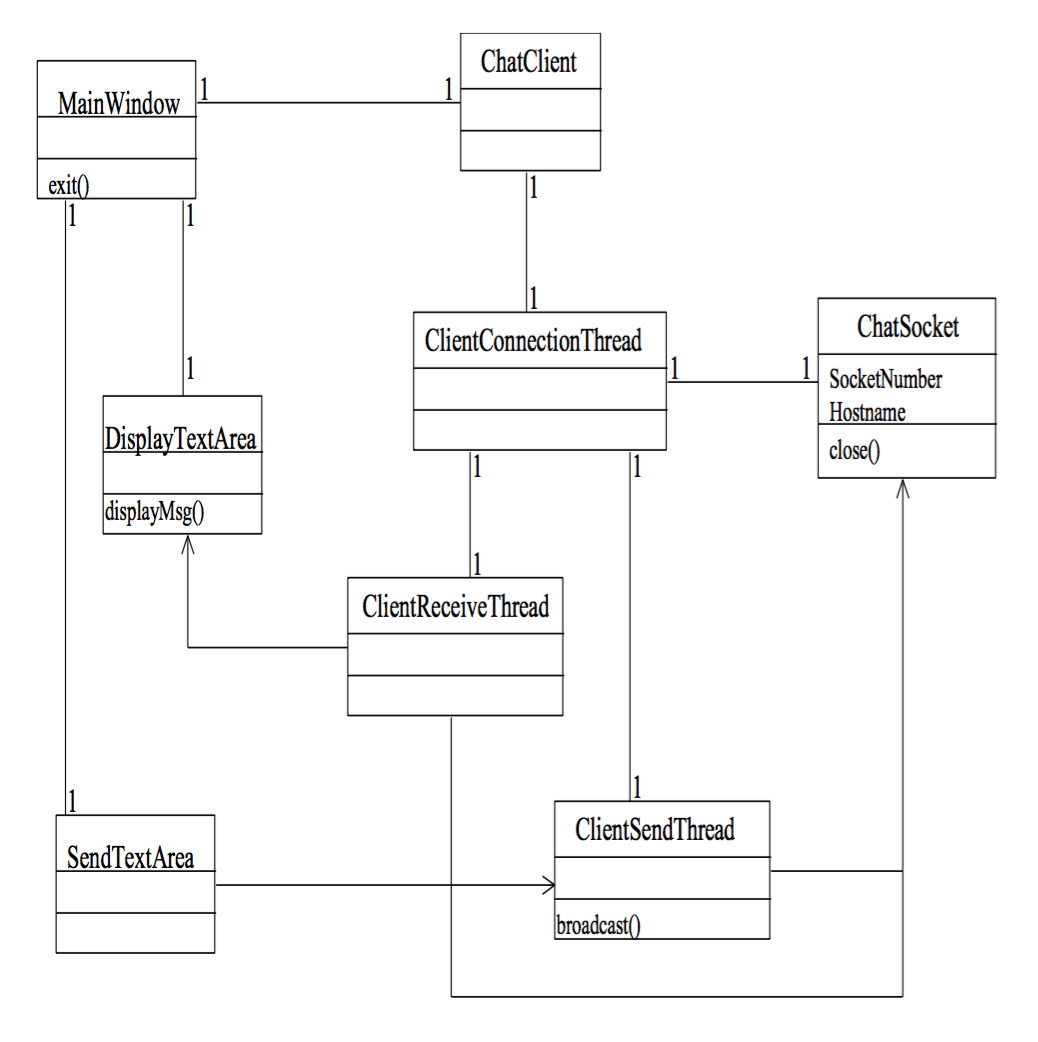


**4.2.11 Class Diagram**

**Chat Server Class Diagram**

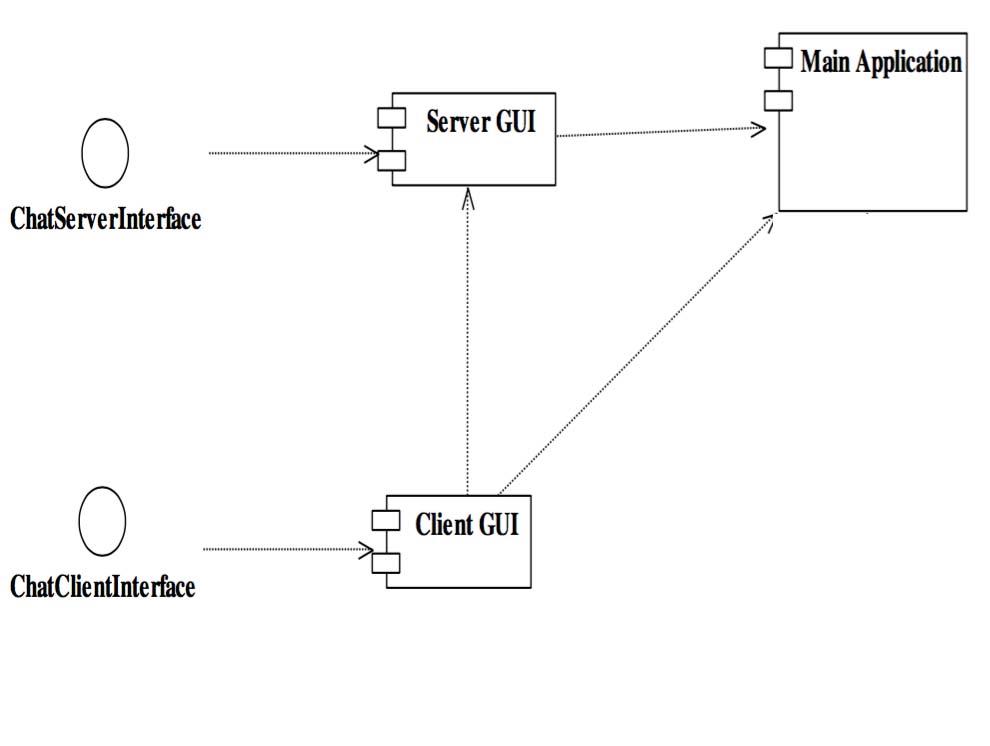


**Chat Client Class Diagram**

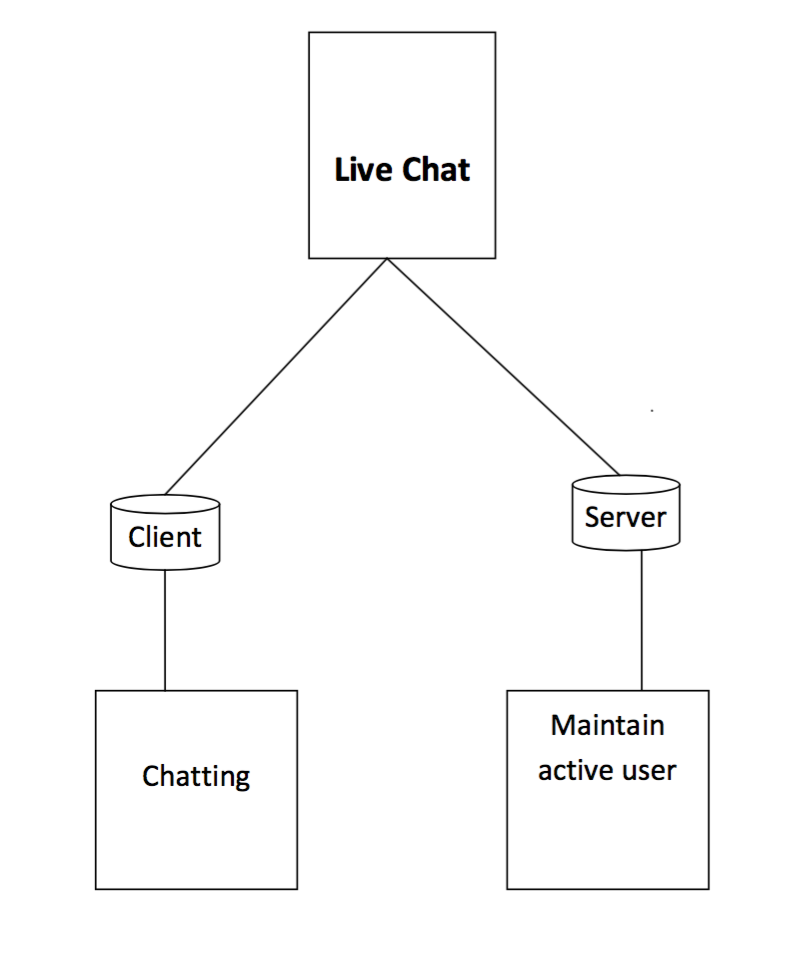


**5. SYSTEM DESIGN**

**5.1 Component diagram**



**5.2 System Flow Chart**



**6. SYSTEM CODING**

**List of tables with attributes and constraint**

**Table name :** login

**Description :** It contains information of username and avatar.

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Type** | **Size** | **Description** |
| Username | Text | 50 | Keeps the username or nickname of the User |
| Avatar | Image |  | Keeps the Avatar selected by the User |

**Table name :** Activity

**Description :** It contains the different activities in the chat area.

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Type** | **Size** | **Description** |
| Online Users | Text + Image | X | Display’s the list of users currently available online |
| Message Area | Text + Images + Files | X | Display’s the messages typed by the user and also the files and images exchanged between them. |
| Text Typing Area | Text | 500 | The Area where the user types the text to be sent. |
| Attachment Adding Button | Images + Files | X | The button area upon clicking which the user can share images and files with each other. |

**Testing and Validation**

System testing is a critical aspect of Software Quality Assurance and represents the ultimate review of specification, design and coding. Testing is a process of executing a program with the intent of finding an error.

A good test is one that has a probability of finding an yet undiscovered error. The purpose of testing is to identify and correct bugs in the developed system. Nothing is complete without testing. Testing is vital in the success of the system.

In the code testing the logic of the developed system is tested. For this, every module of the program is executed to find an error. To perform specification test, the examination of the specification stating what the program should do and how it should perform under various conditions.

Unit testing focuses first on the modules in the proposed system to locate error. This enables to detect errors in the coding and logic that are contained within that module alone. Those resulting from the interaction between modules are initially avoided. In unit testing step each module had to be checked separately.

System testing does not test the software as a whole, but rather the integration of each module in the system. The primary concern is the compatibility of individual modules. One had to find areas where modules have been designed with different specifications of data lengths, type and data element name.

Testing and validation are the most important steps of the developed system. The system testing is performed to ensure that there are no errors in the implemented system. The software must be executed several times in order to find out errors in the different modules of the system.

Validation refers to the process of using the new software for the developed system in a live environment i.e., new software inside the organization, in order to find out the errors.

The validation phase reveals the failures and the bugs in the developed system. We will come to know about the practical difficulties the system faces when operated in the true environment. By testing the code of the implemented software, the logic of the program can be examined. A specification test is conducted to check whether the specifications stating the program are performing under various conditions.

Apart from these tests, there are some special tests conducted which are given below:

**Peak Load Tests:**

This determines whether the new system will handle the volume of activities when the system when the system is at the peak of its processing demand. The test had revealed that the new software for the agency is capable of handling the demands at the peak time.

**Storage Testing:**

This determines the capacity of the new system to store transaction data on a disk or on other files. The proposed software had the required storage space available, because of the use of a number of hard disks.

**Performance Time Testing:**

This test determines the amount of the time used by the system to process transaction data.

In this phase the software developed Testing is exercising the software to uncover errors and ensure the system meets defined reqquirements. Testing may be done at 4 levels.

* Unit Level
* Module Level
* Integration & System
* Regression

**Unit Testing:**

A Unit corresponds to a screen /form in the package. Unit testing focuses on verification of the corresponding class or Screen. This testing includes testing of control paths, interfaces, local data structures, logical decisions, boundary conditions, and error handling. Unit testing may use Test Drivers, which can control programs to co-ordinate test case inputs and outputs, and Test stubs, which replace low-level modules. A stub is a 40 dummy subprogram.

**Module Level:**

Module Testing is done using the test cases prepared earlier. Module is defined during the time of design.

**Integration & System Testing:**

Integration testing is used to verify the combining of the software modules. Integration testing addresses the issues associated with the dual problems of verifications and program construction. System testing is used to verify, whether the developed system meets the requirements.

**Regression Testing:**

Each modification in software impacts unmodified areas, which results serious injuries to that software. So the process of re-testing for rectification of errors due to modification is known as regression testing.

**TESTING METHODS**

**The box approach** Software testing methods are traditionally divided into white-and black-box testing. These two approached are used to describe the point of view that a test engineer takes when designing test cases.

**White box testing:**

White box testing is when the tester has access to the internal data structures and algorithms including the code that implement these.

**Types of white box testing**

The following types of white box testing exist:

* API testing (application programming interface) – testing of the application using public and private APIs
* Code coverage – creating tests to satisfy some criteria of code coverage (e.g., the test designer can create tests to cause all statements in the program to be executed at least once)
* Fault injection methods – improving the coverage of a test by introducing faults to test code paths
* Mutation testing methods
* Static testing – White box testing includes all static testing

**Block Box Testing:**

Black-box test design treats the system as a “black-box”, so it doesn’t explicitly use knowledge of the internal structure.

Black-box test design is usually described as focusing on testing functional requirements. Synonyms for black box include: Behavioral, Functional, opaque-box and closed-box.

**A Simple Black box Specification:**

Black Box Testing is testing technique having no knowledge of the internal functionality/structure of the system. This testing technique treats the system as black box or closed box. Tester does not know how the program actually arrives at those results.

Hence tester tests the system based on the functional specifications given to him. That is the reason black box testing is also considered as functional testing.

This testing technique is also called as behavioral testing or opaque box testing or simply closed box testing. Although black box testing is a behavioral testing, Behavioral test design is slightly different from black-box test design because the use of internal knowledge is not illegal in behavioral testing.

**Test Case:**

For Login Page

* Cursor position in username field
* Validation of Username
* Case Sensitivity can be tested
* Cursor back to username field after unsuccessful attempt.

**7. SYSTEM IMPLEMENTATION**

Implementation is the stage of the project where the theoretical design is turned in to a working system. The implementation state is a system project in its own right.

It involves careful planning, investigation of the current system and its constraints on implementation, design of methods to achieve the changeover, training of staff in the change over procedure and evaluation of change over methods.

Once the planning has been completed, the major efforts are to ensure that the program in the system is working properly

At the same time concentrate on training user staff. When the staff had been trained a full system can carry out.

The various activities involved while implementing a project:

* End user education and training.
* Training on application software.
* System testing.
* Parallel run and change over to new system.
* Post implementation review.

**8. LIMITATIONS AND FUTURE ENHANCEMENT**

**8.1 LIMITATIONS**

There are mainly two limitations of the project they are:

* The firewall is to be disabled for intra network.
* It is dependent on the specific algorithm used.

**8.2 FUTURE ENHANCEMENTS**

There is always a room for improvements in any software package, however good and efficient it may be done. But the most important thing should be flexible to accept further modification. Right now we are just dealing with text communication. In future this software may be extended to include features such as:

* *Voice chat:* this will enhance the application to a higher level where communication will be possible via voice calling as in telephone.
* *Video chat:* this will enhance the feature of calling into video communication.

**11. BIBLIOGRAPHY**

**Books referred**

**Node.js Web Development.**

- By David Herron

**JavaScript Hand on Learning : Interactive front end development.**

**-** By Jie Wang

**Internet References**

**en.wikipedia.org/wiki/Node.js**

**www.tutorialspoint.com/nodejs**

**www.codeschool.com/courses/real-time-web-with-node-js**

**www.nodejs.org**