

CHAPTER THREE

METHODOLOGY

3.0 INTRODUCTION

The software engineering approach was used to develop the system in order to facilitate the development process and enhance the quality and efficiency of the software.

The development of software includes the following activities:

- Requirement analysis and definition
- Design
- Coding (using a suitable language)
- Testing

3.1 OVERVIEW OF THE SYSTEM

The system is an application for sending and receiving textual messages via IP address of any computer on a network. The system is on designing and implementing a software application that will make communication possible over TCP/IP network using the rich provision of client/server architecture. Hence with this software users on different ends of computer network can chat with each others.

The system is designed in such a way as to allow users to login(i.e., join) into the chat through login windows with their names. the moment any user logs, the server registers the name in the chat room and the user can also view the list of users in the chat and decide whether to join either the public room-where everybody in the room can be

communicated with, or private room- where you can only communicate with the selected people you want in the chat room. That is, if you choose public room your message will be visible to everybody in the room, but if you choose private room only those people you choose will be able to see your messages.

Then, the chat window will be shown where a user can send and receive messages, the window has a very large text area to display both the sent and received messages for any user and a text box where messages are typed and a button that can be clicked to send the messages. This also has menus for some operation like logging out of the room, knowing about the software, some hints on how to use the software, etc.

However, users are given the terminating the client application any point they want, they can cancel login operation; they can exit from the room if they do not wish to join either the public or private room. This makes the application more flexible than some of its counterparts, and the application is user-friendly to the extent that you do not need to be taught before using it. It also surpasses most of its counterparts in this area.

And if during chatting you feel like leaving the room, just click logout menu and the software will help the user to tell others in the room.

3.2 SYSTEM REQUIREMENT SPECIFICATION

A requirement is the operation that system must perform or description of operations the system is capable of doing in order to fulfil its purpose. The basic requirements for this system are divided into two parts; those that occur at the server side and those that occur

at the client side.

These are various operations that take place during chatting.

On the client side:

- Login to the chat room with user name
- Choose the type of chat room either public room (i.e., general room) or private room.
- Send messages (to other clients through server).
- Receive messages (from other clients through server).
- Display the received and sent messages.
- Log out (from the chat room).

On the server side:

- Listen to clients (on a certain port number).
- Accept clients (when login).
- Receive clients' messages.
- Transmit or send (or echo) received messages to all necessary clients.
- Remove client (when logged out).

The clients communicate with the server through server's IP address and port number, while the server identifies each client with its IP address.

3.3 SYSTEM DESIGN

Some UML tools, object oriented approach and tradition process decomposition were used to design this system.

The UML tools like:

- Use Case Diagram
- Class Diagram
- Activity Diagram

These diagrams were implemented as to depict the requirements pictorially so as to give the better understanding of the requirement.

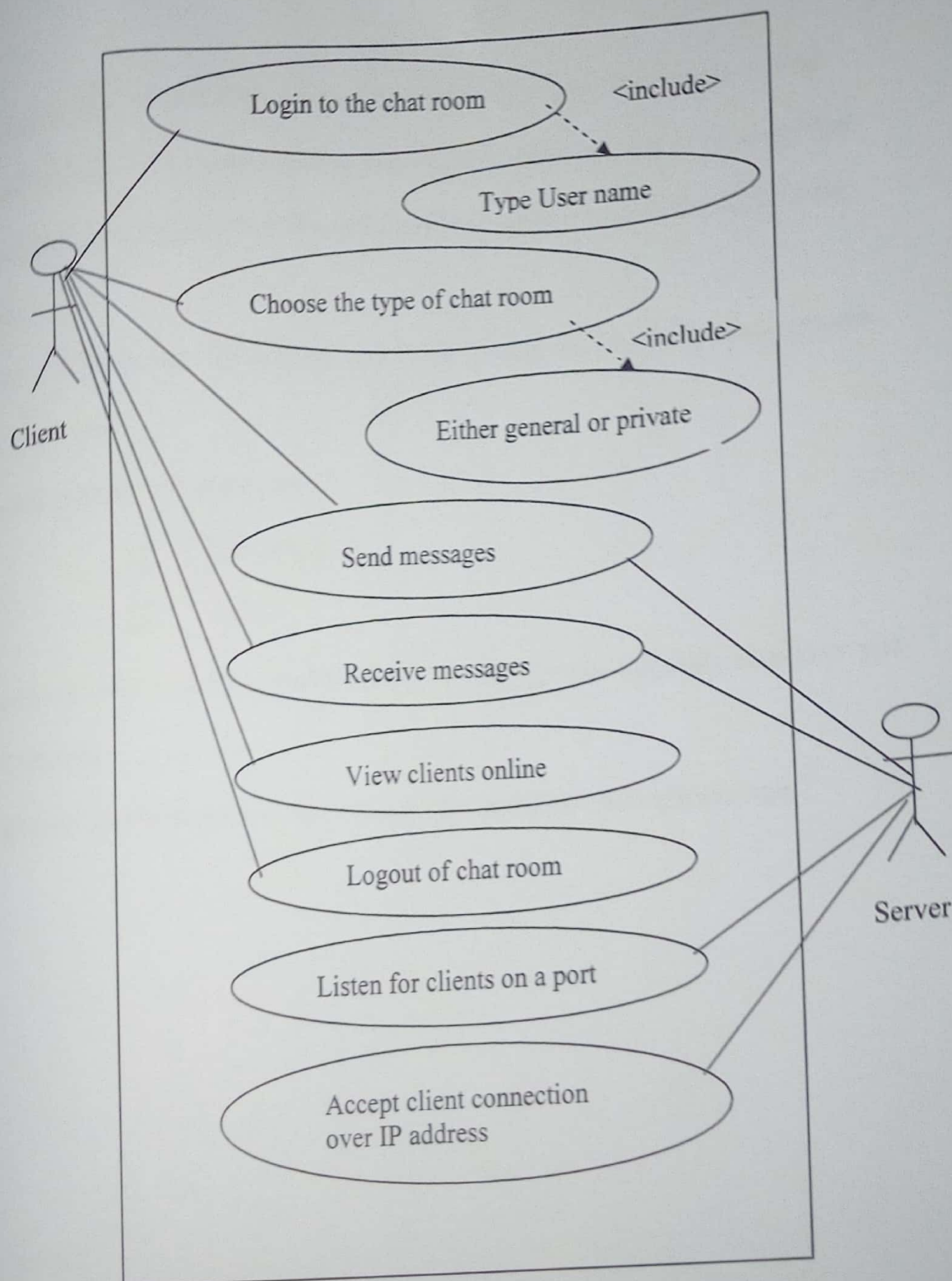


Figure 3.1 Use Case Diagram for Chat Room Design

Use Case: Chat Room Design and Implementation with Client/Server over TCP/IP

Goal in Content: To enable clients to chat with each other with the aid of server that coordinates the activities of sending and receiving messages over IP addresses of the clients.

Scope: Clients can send textual messages to one another, and receive textual messages from one another.

Primary Actors: The actors are:

- (i) Server and,
- (ii) Client

Priority: It makes communication easy, and very cheap to use especially anywhere you have network.

Frequency: Server must always be running state, and client can login any time

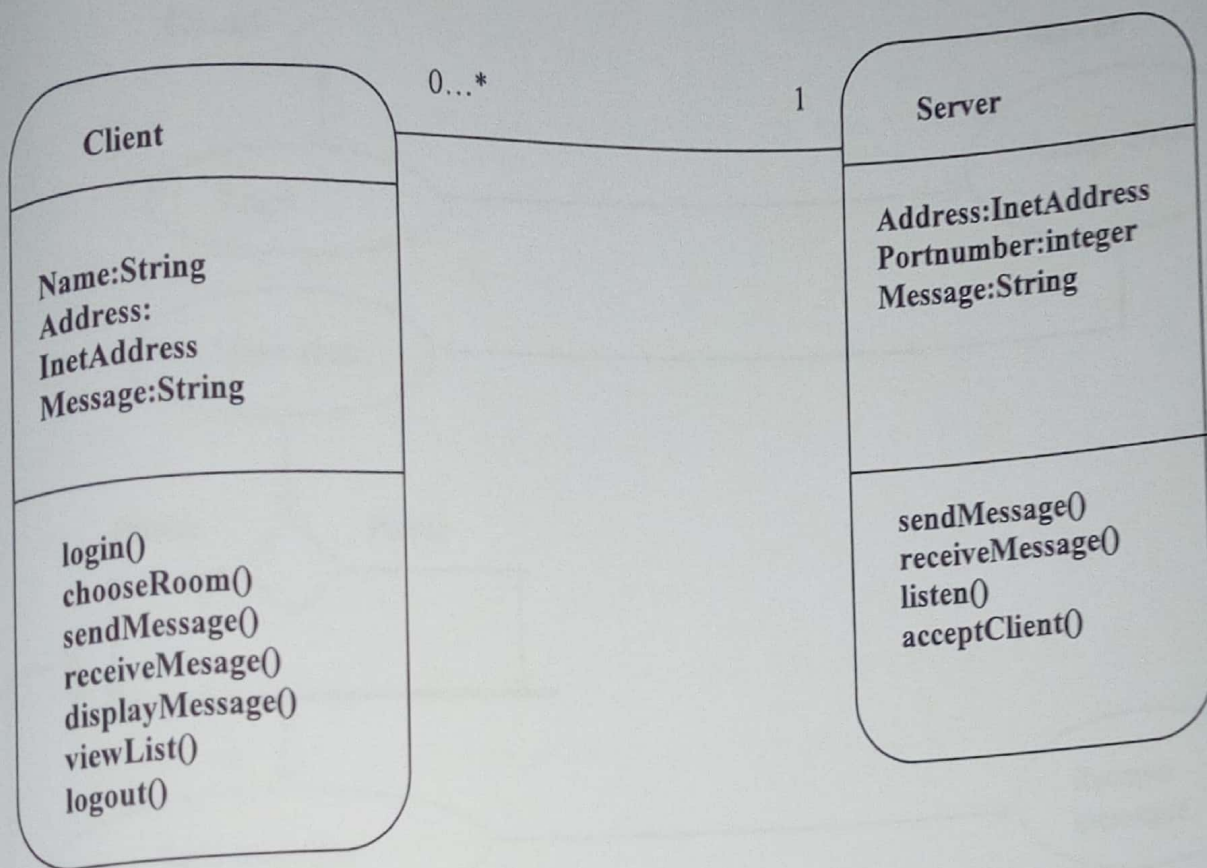
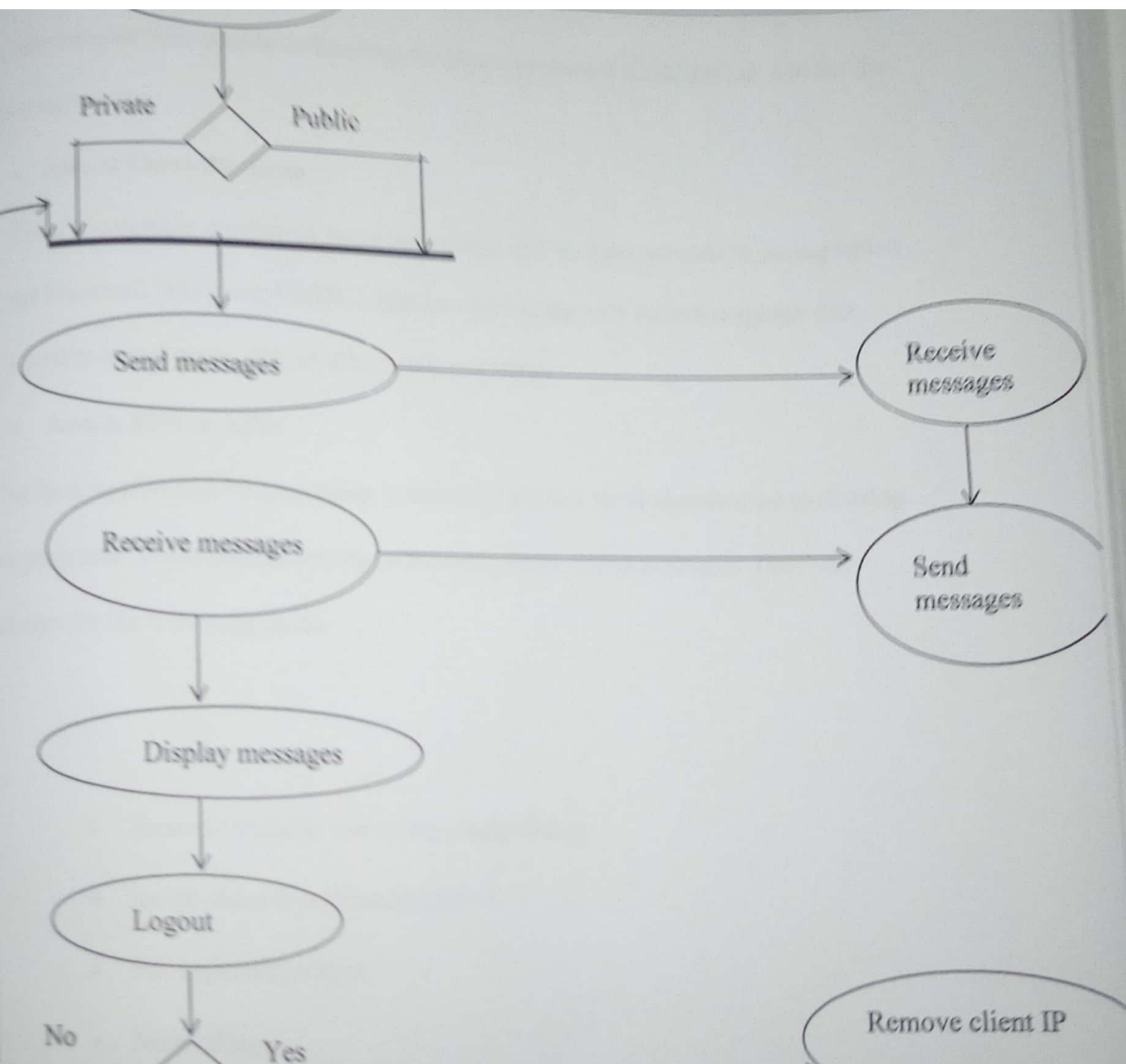


Figure 3.2 Class Diagram for Chat Room Design

The above shown diagram is called Class Diagram which is one UML diagram that is used to depict the classes that will be involved, their attributes and the operations to be performed by them.

So, we have two classes named client and server, a client must have IP address , port number and a message to be sent to other clients.



The above diagram is the activity diagram for chat room design, it shows some of activities that will occur and who carries each of those activities out.

3.5 CODING (WRITING PROGRAM)

Since the project is on network programming, making use of client/server architecture and object-oriented approach in system design; it is obvious that a programming language that supports all this operations should be used.

Java programming language was chosen as a programming language of implementation not only because it is capable of handling the afore-mentioned situations but also for the following reasons:

- **Java is Cross-Platform**

There is countless number of operating system that we have nowadays among which are Microsoft Windows, UNIX, Linux etc. Java is the only known language that operates in real sense of it on all operating systems.

- **Java is Rich in APIs**

The Java Application Programming Interface (API) is a set of classes used to develop Java programs. These classes are organized into groups called packages. There are packages for the following tasks:

- Numeric variable and string manipulation
- Image creation and manipulation
- File input and output
- Networking

- Windowing and graphical user interface design
- Applet programming
- Error handling

The API includes enough functionality to create sophisticated applications.

- **Java is Safe**

Java is based on a proven security model and offers the application developers with APIs for secure communication over the networks.