

MULTI-USER CHAT SYSTEM Project Report

***DEVELOPERS:***

***ZIRI ISSA ATHMAN - CI/00051/2012***

***OSAWA C OTTA - CI/00099/2012***

***MUNDIA JOEL MURIITHI - CI/00067/2012***

**Submitted in partial fulfillment of the requirements for the Degree of Bachelor of Science in Information Technology (BSc.IT) at Maseno University**

**Faculty of Information Technology**

**Maseno University.**



**Supervisor:Mr Saka**

Abstract

**Problem statement**

Currently, the existing technology requires Internet connectivity to enable you to chat with friends in a particular room.

The use of Multi User Chat System application enables friends to chat in the network area (LAN) by using *intranet* *mailing*. The Intranet Mailing System is applicable within an organization only. The organization has shift times and it becomes difficult for an employee of one shift to communicate to an employee of another shift. If the organization has an Intranet Mailing System (IMS) facility available to all its employees then each employee can register himself/herself and send mails to any other registered employee. This makes communication easier. This will work under any operating system.

**Methodology**

To achieve this, the Multi User Chat System uses Java’s Abstract Window Toolkit (AWT); the AWT provides the requisite elements for designing the application- scrollbars, button layouts. AWT’s child Swing is used for creating list controls, buttons, labels, tree controls, table controls. Applets are used to provide the standard interface between the applet and the browser environment.

**Modules:**

* **User Login:-** Enables the user to enter the system.
* **User Details:-**Contains 2 parts. User maintenance and password change.
* **Send and receive mails:** Design the inbox for storing received mails and maintaining sent folder for sending mails. Details are stored in a database.
* **Help:** The Help menu should display the detailed information about the system.
* **About:** The About menu should give information about this software like developed by, version and contact details.

**Proposed System**

In earlier days, communication occurred only through telephones or through letters. It was time taking and costlier way of communicating with people. This application ensures that people are chatting with friends in the network area (LAN) by using intranet mailing which works in a similar fashion as that of internet mailing system but there is no need to get an internet connection.

**Software requirements**

Technologies: core Java.

Operating system: Any Windows platform.

**Hardware requirements**

Processor: Intel Pentium based system.

Hard disk: 5 GB

RAM: 128 MB.

**Acknowledgements**

If words are considered as a symbol of approval and token of appreciation then let the words play the heralding role expressing our gratitude. The satisfaction that accompanies that the successful completion of any task would be incomplete without the mention of people whose ceaseless cooperation made it possible, whose constant guidance and encouragement crown all efforts with success. We are grateful to our project guide Mr Saka for the guidance, inspiration and constructive suggestions that helped us in the preparation of this project. We also thank the group members who have helped in successful completion of the project.

Table of contents

1………………………………………………………………………………………………………...Introduction

1.1………………………………………………………………………………………………….Scope

2. ……………………………………………………………………………………………………….System Analysis

2.1…………………………………………………………………………………………..………….Definition

2.2……………………………………………………………………………………………………….Description of Present System

2.3………………………………………………………………………………………………………..Limitation of Present System

2.4 ……………………………………………………………………………………………………….Proposed System

2.5 ……………………………………………………………………………………………………….Feasibility Study

2.6……………………………………………………………………………………… Software Engineering Paradigm Applied

3………………………………………………………………………………………………………. System Specification

3.1………………………………………………………………………………………………………. H/W Specification

3.2………………………………………………………………………………………………………. S/W Specification

4………………………………………………………………………………………………………. Organization Profile

5 ……………………………………………………………………………………………………….Specification of Language

6………………………………………………………………………………………………………. System Design

6.1………………………………………………………………………………………………………. Output Design

6.2………………………………………………………………………………………………………. Input Design

6.3………………………………………………………………………………………………………. Logical Design

6.4 ………………………………………………………………………………………………………..Data Flow Diagram

6.5…………………………………………………………………………………………………………Process Modal

7. ……………………………………………………………………………………………………….Coding

8. ……………………………………………………………………………………………………….Testing

9.……………………………………………………………………………………………………….Limitation

10. …………………………………………………………………………Future Enhancement and Conclusion

11. ………………………………………………………………………………………………………. Bibliography

1.INTRODUCTION

Chatting, is a method of using technology to bring people and ideas together despite of the geographical barriers. The technology has been available for years but the acceptance it was quit recent. Our project is an example of a chat server. It is made up of 2 applications the client application, which runs on the user’s Pc and server application, which runs on any Pc on the network. To start chatting client should get connected to server where they can practice two kinds of chatting, public one (message is broadcasted to all connected users) and private one (between any 2 users only) and during the last one security measures were taken.

2.System Analysis

2.1 Definition

System Objectives Communication over a network is one field where this tool finds wide ranging application.Chat application establishes a connection between 2 or more systems connected over an intra-net or ad-hoc. This tool can be used for large scale communication and conferencing in anorganization or campus of vast size, thus increasing the standard of co-operation. In addition itconverts the complex concept of sockets to a user friendly environment. This software can have further potentials, such as file transfer and voice chatting options that can be worked upon later.

2.2 Description of Present System

Relation to External Environment,This tool helps in two major aspects Resolving the names of all the system connected in a network and enlisting them.Used for communication between multiple systems enlisted in the resolved list.

2.3 Limitation of Present System

One of the major challenges with present chat systems is secure communication between the users. The corporate or scientific organization may have information which need be protected with intruders or hacker. The chat system should implemented such a way that the information need be secured.

2.4 Proposed System

The chat application works in two forms. List form: In this form, 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 form This form is called only when an element is selected from the List form. In this form, a connection is created between the host system and the selected system with the help of a socket.

2.5 Feasibility Study

We conducted to see if the proposed system is a feasible one with all respects. Feasibility Study is lot of the system proposal according to its workability impact of the organization, ability to meet uses need and effective use of resources. There are three main aspects in the feasibility study. The feasibility of a project can be ascertained in terms of technical factors, economic factors, or both.  A feasibility study is documented with a report showing all the ramifications of the project. Establish connection between Stop host and selected system Chat form (Text messages sent and received as packets Form).

2.6 Software Engineering Paradigm Applied

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.In this multi user chat system,a socket can send and receive data down disconnection. Data is generally sent in blocks of few kilo bytes 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 network applications because no two applications can use the same port.Packets that contain port numbers come in two flavors: UDP and TCP/IP.

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 than 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 the destination application.

3.System Specification

3.1 H/W Specification

The minimum hardware required for the development of our project is as follows:-

Ram- minimum 128 MB

Hard disk—minimum 5 GB

Processor- Pentium 3

Locaal area network.

These all are the minimum hardware requirement required for our 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.128 MB ram is used so that we can execute our project in a least possible RAM.5 GB 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.

3.2 S/W Specification

* + JRE 1.2
  + IIS Server
  + Html 3.2 and 4.0 supported Browsers
  + Any Operating System That Can Run JRE.

4. Organization Profile

The multi user chat system contains the following modules and sub-modules:

1.User Login.

2. User Details- User maintenance.

- password change.

3. Send and receive mails.

4. Add friends in the network.

5. Help- Help.

- About.

5. Specification of Language

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

To achieve this, the Multi User Chat System uses Java’s Abstract Window Toolkit (AWT); the AWT provides the requisite elements for designing the application- scrollbars, button layouts. AWT’s child Swing is used for creating list controls, buttons, labels, tree controls, table controls. Applets are used to provide the standard interface between the applet and the browser environment.

6. System Design

6.1 Output Design

We developed the user interface for the application through which the user interacts with the tool. It consists of a main window and boxes which are displayed as per the menu selections made by the user.

6.2 Input Design

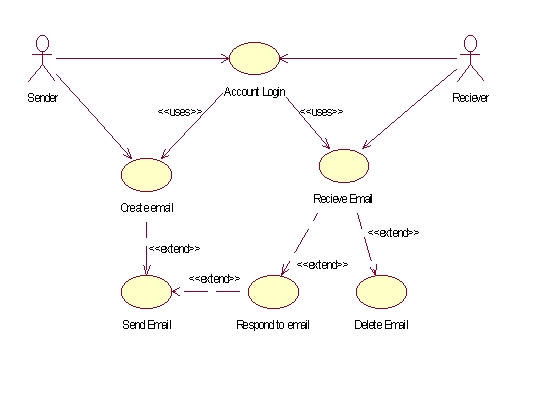
We develop the user interface for the application through which the user interacts with the tool. It consists of a main window and boxes which are displayed as per the menu selections made by the user. There are different controls such as edit controls, buttons etc which are used to get the user inputs.

Processing within modules We provide the button Connect to the user, to provide input on the basis of selection using mouse event.

6.3 Logical Design

Login module and send module.These module deals with the application’s interface with the end user. All the user inputs (connecting, chatting) are captured here.The use case diagram is used.

**Use Case Diagram:**

****

6.4 Data Flow Diagram

**Level 0 DFD**

SMTP Protocol

Mail Server

User

## Level 1 DFD

LOG IN

Authentication

Mail server

SMTP protocol

User

REGISTRATION

User file

**Level 2 DFD**

**Junk Mail**

**Delete Mail**

User

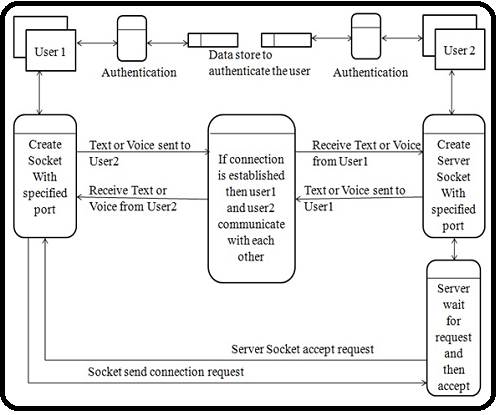
Check Inbox

**Empty Inbox**

Mail file

**Mark Mail as Favourite**

**Level 3 DFD**



6.5 Process Modal

The Software Process Model used is the Spiral Model. The choice for this model is in the light of the enhancements in the future. The enhancements would be in the area of Networks being introduced into the software.The functionalities and responsibilities of the system were partitioned and then assigned to subsystems or components. Graphical User InterfaceThe user interface that the software provides to the user is interactive. It provides two different forms,one for list of systems and the other for the actual text chatting

7. Coding

The following is the code for our interface, the code is necessary to view the different aspects of the interconnections in a system.

/\*

\* To change this template, choose Tools | Templates

\* and open the template in the editor.

\*/

public class MultiUserChatSystem extends javax.swing.JFrame {

/\*\*

\* Creates new form MultiUserChatSytem

\*/

public MultiUserChatSystem() {

initComponents();

}

/\*\*

\* This method is called from within the constructor to initialize the form.

\* WARNING: Do NOT modify this code. The content of this method is always

\* regenerated by the Form Editor.

\*/

@SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed" desc="Generated Code">

private void initComponents() {

chat = new javax.swing.JButton();

help = new javax.swing.JButton();

about = new javax.swing.JButton();

maintenance = new javax.swing.JButton();

password\_change = new javax.swing.JButton();

jTextField1 = new javax.swing.JTextField();

StartServer = new javax.swing.JButton();

jLabel1 = new javax.swing.JLabel();

setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE\_ON\_CLOSE);

setTitle("MUTI USER CHAT APPLICATION");

setBackground(new java.awt.Color(255, 255, 255));

setMinimumSize(new java.awt.Dimension(600, 450));

setPreferredSize(new java.awt.Dimension(600, 450));

getContentPane().setLayout(null);

chat.setFont(new java.awt.Font("Lucida Calligraphy", 1, 11)); // NOI18N

chat.setForeground(new java.awt.Color(255, 153, 153));

chat.setText("Chat");

chat.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

chatActionPerformed(evt);

}

});

getContentPane().add(chat);

chat.setBounds(220, 140, 134, 33);

help.setFont(new java.awt.Font("Lucida Calligraphy", 1, 11)); // NOI18N

help.setForeground(new java.awt.Color(255, 153, 153));

help.setText("Help");

help.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

helpActionPerformed(evt);

}

});

getContentPane().add(help);

help.setBounds(220, 190, 134, 31);

about.setFont(new java.awt.Font("Lucida Calligraphy", 1, 11)); // NOI18N

about.setForeground(new java.awt.Color(255, 153, 153));

about.setText("About ");

about.setBorder(javax.swing.BorderFactory.createEmptyBorder(1, 1, 1, 1));

about.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

aboutActionPerformed(evt);

}

});

getContentPane().add(about);

about.setBounds(220, 240, 134, 31);

maintenance.setBackground(new java.awt.Color(204, 204, 204));

maintenance.setFont(new java.awt.Font("Lucida Calligraphy", 1, 11)); // NOI18N

maintenance.setForeground(new java.awt.Color(255, 153, 153));

maintenance.setText("Maintenance");

maintenance.setBorder(javax.swing.BorderFactory.createEmptyBorder(1, 1, 1, 1));

maintenance.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

maintenanceActionPerformed(evt);

}

});

getContentPane().add(maintenance);

maintenance.setBounds(220, 290, 134, 30);

password\_change.setFont(new java.awt.Font("Lucida Calligraphy", 1, 11)); // NOI18N

password\_change.setForeground(new java.awt.Color(255, 153, 153));

password\_change.setText("Password Change");

getContentPane().add(password\_change);

password\_change.setBounds(220, 340, 149, 25);

password\_change.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

password\_changeActionPerformed(evt);

}

});

jTextField1.setEditable(false);

jTextField1.setBackground(new java.awt.Color(0, 0, 51));

jTextField1.setFont(new java.awt.Font("Lucida Calligraphy", 1, 11)); // NOI18N

jTextField1.setForeground(new java.awt.Color(255, 153, 153));

jTextField1.setHorizontalAlignment(javax.swing.JTextField.CENTER);

jTextField1.setText(" MULTI-USER CHAT APPLICATION");

jTextField1.setBorder(null);

getContentPane().add(jTextField1);

jTextField1.setBounds(150, 30, 280, 39);

StartServer.setFont(new java.awt.Font("Lucida Calligraphy", 1, 11)); // NOI18N

StartServer.setForeground(new java.awt.Color(255, 153, 153));

StartServer.setText("Start Server");

StartServer.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

StartServerActionPerformed(evt);

}

});

getContentPane().add(StartServer);

StartServer.setBounds(220, 90, 134, 31);

jLabel1.setIcon(new javax.swing.ImageIcon(getClass().getResource("/2013-07-14.jpg"))); // NOI18N

jLabel1.setPreferredSize(new java.awt.Dimension(600, 450));

getContentPane().add(jLabel1);

jLabel1.setBounds(0, 0, 600, 450);

pack();

}// </editor-fold>

private void chatActionPerformed(java.awt.event.ActionEvent evt) {

dispose();

Login login= new Login();

login.setVisible(true);

// TODO add your handling code here:

}

private void helpActionPerformed(java.awt.event.ActionEvent evt) {

dispose();

Help help=new Help();

help.setVisible(true);

// TODO add your handling code here:

}

private void aboutActionPerformed(java.awt.event.ActionEvent evt) {

dispose();

About about=new About();

about.setVisible(true);

// TODO add your handling code here:

}

private void maintenanceActionPerformed(java.awt.event.ActionEvent evt) {

dispose();

Admin mnt=new Admin();

mnt.setVisible(true);

// TODO add your handling code here:

}

private void password\_changeActionPerformed(java.awt.event.ActionEvent evt) {

dispose();

Password\_change pc=new Password\_change();

pc.setVisible(true); // TODO add your handling code here:

}

private void StartServerActionPerformed(java.awt.event.ActionEvent evt) {

dispose();

Administrator admin=new Administrator();

admin.setVisible(true);

// TODO add your handling code here:

}

/\*\*

\* @param args the command line arguments

\*/

public static void main(String args[]) {

/\* Set the Nimbus look and feel \*/

//<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">

/\* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.

\* For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html

\*/

try {

for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {

if ("Nimbus".equals(info.getName())) {

javax.swing.UIManager.setLookAndFeel(info.getClassName());

break;

}

}

} catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(MultiUserChatSystem.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(MultiUserChatSystem.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(MultiUserChatSystem.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(MultiUserChatSystem.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

}

//</editor-fold>

/\* Create and display the form \*/

java.awt.EventQueue.invokeLater(new Runnable() {

public void run() {

new MultiUserChatSystem().setVisible(true);

}

});

}

// Variables declaration - do not modify

private javax.swing.JButton StartServer;

private javax.swing.JButton about;

private javax.swing.JButton chat;

private javax.swing.JButton help;

private javax.swing.JLabel jLabel1;

private javax.swing.JTextField jTextField1;

private javax.swing.JButton maintenance;

private javax.swing.JButton password\_change;

// End of variables declaration

}

Other codes are for the following modules:

Admin,About,Help,Login,Password Change and maintainance

8. Testing

When developing a strategy for unit testing, there are three basic organizational approaches that can be taken. These are top down, bottom up and isolation. In our case of Multi user chat system,we simply use top down approach. There are two sub options in our project first one is Form1 mode, which is the listing part and second one is Form2 mode which is chat box.

In the first case we just test for the correct resolution of names of systems connected to a network. And in the second mode our motive is to obtain a two way communication between the host user and remote user. And we are very much successful here in our test case.

The test strategies included different types of testing as described below:-

1. Logical Testing:-We used this to test every aspect of both modes, report and query as soon as it is implemented, using valid, invalid and extreme data test data will be added to test each code module and results compared with the expected results. Sufficient data will be added to ensure that there is at least one entry in each category.Subsequent tests will often involved adding new data, which will be deleted when the test works satisfactorily.As per our requirement we have also included some field such as character size and then queries were performed after that results were tabulated and then the module were free from extra field.

2. Functional Testing:- In this menu,items were tested to ensure no functions has been missed out. This is done for the smooth working of the project.

3. System Testing: - This is done after the completion of system; we carried out all queries again to ensure that no errors have been introduced.

10.Limitation

In Limitation,there are mainly two limitations of the project and that are:-

 The firewall is to be disabled for intra network.

• The project is dependent on the specific algorithm used.

11. Future enhancement and Conclusion

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:

• File transfer: this will enable the user to send files of different formats to others via the chat application.

• 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 further enhance the feature of calling into video communication.

12. Bibliography

W3 schools.

System analysis and design 8th edition.

System Analysis and Design with UML Version 2.0(1)

Web Technologies - A Computer Science Perspective - J. Jackson (Pearson, 2007) BBS