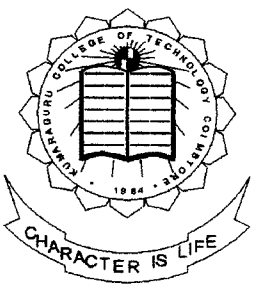
ELECTRONIC MAIL SERVER

A PROVISION FOR SELECTIVE E-MAIL DOWNLOADING

P-504



Project report presented by

Gunasekaran.R

Karthik.T.R

Saravanakumar P.A.J

Sriram Prasad.N

Guided by

Mrs.S.Devaki B.E., M.S.,

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE AWARD OF BACHELOR OF ENGINEERING DEGREE IN COMPUTER SCIENCE AND ENGINEERING BY **THE BHARATHIAR UNIVERSITY**

Department of Computer Science and Engineering

KUMARAGURU COLLEGE OF TECHNOLOGY

COIMBATORE – 641 006

KUMARAGURU COLLEGE OF TECHNOLOGY

COIMBATORE 641 006

Department of Computer Science and Engineering

Certificate

This is to certify that the report entitled

ELECTRONIC MAIL SERVER

has been submitted by

KARTHIK T.R , SRIRAMPRASAD.N , GUNASEKARAN.S
PAS. SARAVANAKUMAR

Mr._____ fulfillment for the award of the degree of Bachelor of Engineering in Computer Science and Engineering branch of Bharathiar University,Coimbatore-641 046 during the academic year 1997-2001.

10/3/2001

Guide

12/3

Head of Department

Date:

Certified that the candidates with University Registration No_____ was examined in the project viva-voce examination held on 12-03-2001

Internal Examiner

External Examiner

ACKNOWLEDGEMENT

We wish to express our sincere and heartfelt gratitude to **Dr.K.K. PADMANABHAN, B.Sc. (Engg). M.Tech.Ph.D.** Our esteemed principal, Kumaraguru College of Technology for giving us the needed encouragement in starting this project and carrying this out successfully.

Our heartfelt thanks and a deep sense of gratitude to **Prof.Thangaswamy, B.E (Hons), Ph.D.,** the head of department of Computer Science and Engineering for his benevolent attitude and spurring encouragement.

We would like to thank our guide **Asst.Prof. Mrs.S.Devaki B.E., M.S.,** Department of Computer Science and Engineering without her motivation and guidance we would not have been able to embark on a project of this magnitude.

We would also like to thank our class advisor **Ms.A.Lavanya B.E.,** for her constant support and guidance, without which we would not have been able to complete our project successfully.

It is our duty to express our thanks to **Mr.V.Krishnan, Project Manager , Atna Technologies (p) Ltd. ,** Coimbatore for guiding us to do this project work and his kind encouragement during the course of the project.

We reciprocate the kindness shown to us by the staff members of the department of computer science, people at home and our beloved friends who have supported and helped us in getting this project done.

DECLARATION

We, R.Gunasekaran, P.A.J.Saravana Kumar, N.Sriram Prasad, T.R.Karthik hereby declare that the project work entitled '**Electronic Mail Server**' submitted to Kumaraguru College of Technology, Coimbatore (Affiliated to Bharathiar University) is a record of original work done by us under the supervision and guidance of Mrs.S.Devaki B.E., M.S., Department of Computer Science and Engineering.

Name	**Register No**	**Signature**
Gunasekaran.R	9727K0138	
Saravana Kumar.P.A.J	9727K0173	
Sriram Prasad.N	9727K0183	
Karthik.T.R	9727K0514	

Counter Signed:

Staff In charge: Mrs.S.Devaki B.E., M.S.,

Assistant Professor

Department of Computer Science and Engineering

Kumaraguru College of Technology

Coimbatore – 641 006

Place: Coimbatore

Date: 10 – 03 - 2001

SYNOPSIS

The aim of our project is to promote selective e-mail downloading which would greatly reduce the traffic in the Internet. This project is a pilot project, which is implemented in Intranet. In addition to traffic reduction it curtails the unwanted permanent connection with the host during mail transaction. Even though we emphasize on e-mail downloading we have empowered our project with sending mails.

PROCESS

- User enters the login name and pass code to enter into the mail program after it has been started.
- User is presented with options for Sending, Receiving, Receiving old mails and Write new ones.
- The user can select his choice from the options. According to his selection the user is presented with appropriate screen.
- If the user selects either send or receive mails, only then the connection is established with the server.
- Unlike the ordinary web based mail the user enjoys the privilege of sending mails collectively rather than one by one.
- Viewing and selecting the needed mails from the header list displayed to the user accomplish selective downloading.
- All the transactions and user details are maintained by MS-Access database.

CONTENTS

INTRODUCTION

In order to fulfill the caliber of engineering degree in computer science we, have landed upon a project, which is full-fledged with the new cutting edge technology. So the idea of selective e-mail downloading rang in our mind. We also enhanced our project with characteristics such as

- Curtailing the connection time between the client and the server.
- Reducing the network traffic to a greater extent.
- Ability to view the header of the mails without downloading them completely.
- A powerful centralized database serving as the backbone of the project.

In the forth coming pages we will see the vivacity of the project in a user friendly way.

ABOUT MS-ACCESS

In other database-management programs, he term database is used to refer to tables that hold data. Access uses the term more broadly. An access database consists of the tables that hold the data and all the related objects, such as queries, forms and reports that are used to manage data.

When you open a database, access displays the database window, sometimes called the database container, because it contains all the objects that contain the database.

We can create tables in the design view or the typical wizard. We also have the capability to create forms. Forms let us control how data is displayed on the screen. We also have an option for printing data using the reports option.

The salient features of access are :

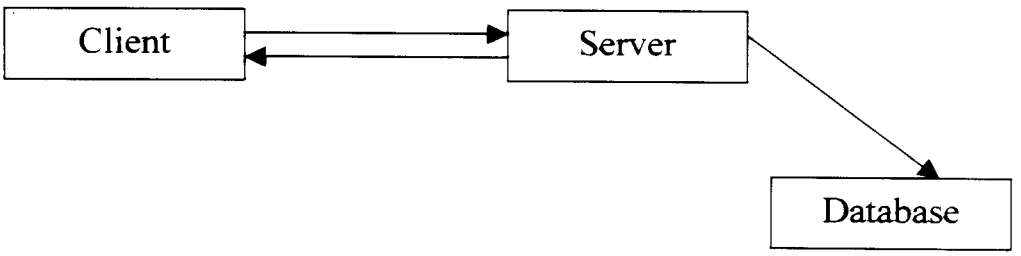
- Macros that let you automate and speed up your work, they are also used when you develop applications. A macro is a list of actions. Access performs all the actions in the list when you run the macro.

- We have a lot of utilities and special techniques present in access, viz.
- Creating windows shortcuts.
- Using access utilities to manage database and their objects.

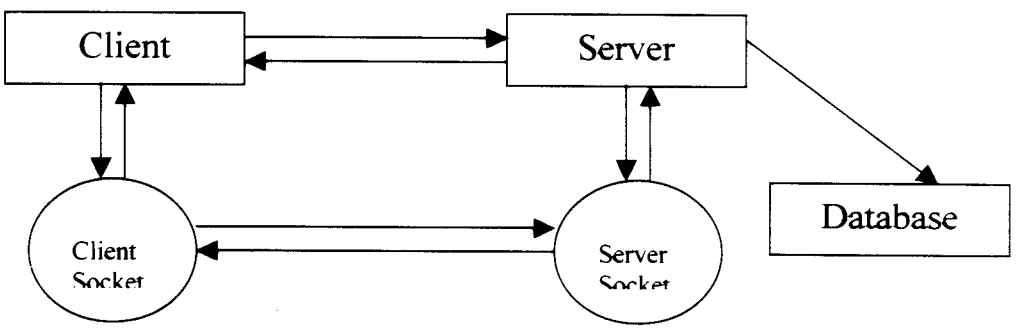
- Using hyperlink data type.
- Creating web pages.
- Creating indexes based on single or multiple fields.
- Working on both embedded and link OLE objects in access, tables and queries.
- Working with bound and unbound OLE objects in forms and reports.
- Attaching a table from another database application so that access and the other applications can use it simultaneously.
- Customizing the access-working environment using the options dialogue box.

To sum up, access begins with database utilities that let you compact, convert, encrypt and repair databases and object utilities that let you rename, delete, cut, copy and paste.

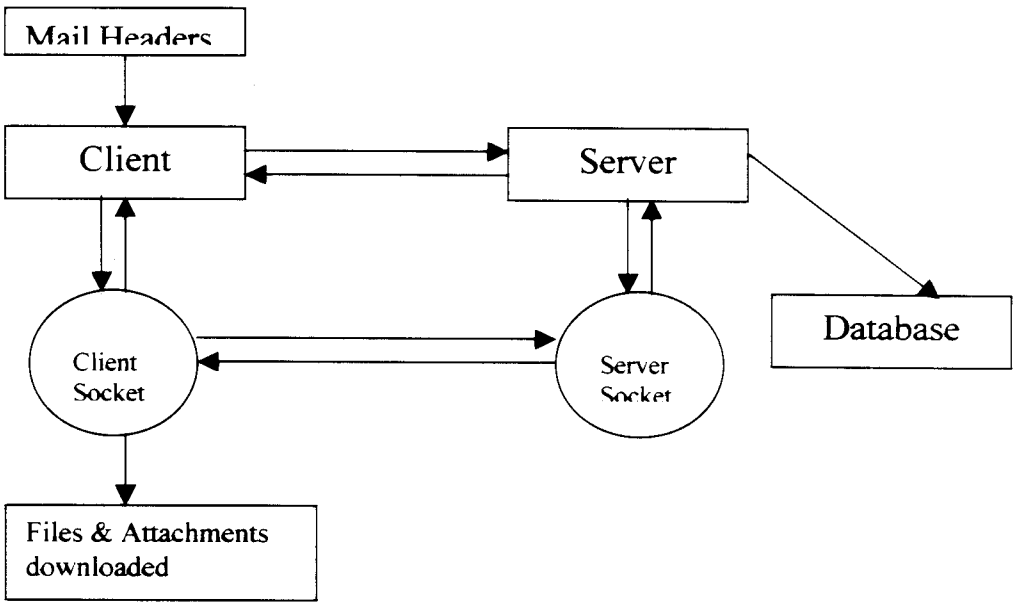
LOGIN SCREEN



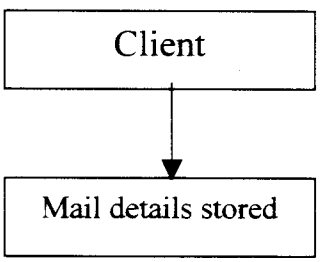
RECEIVE NEW MAILS / OLD MAILS



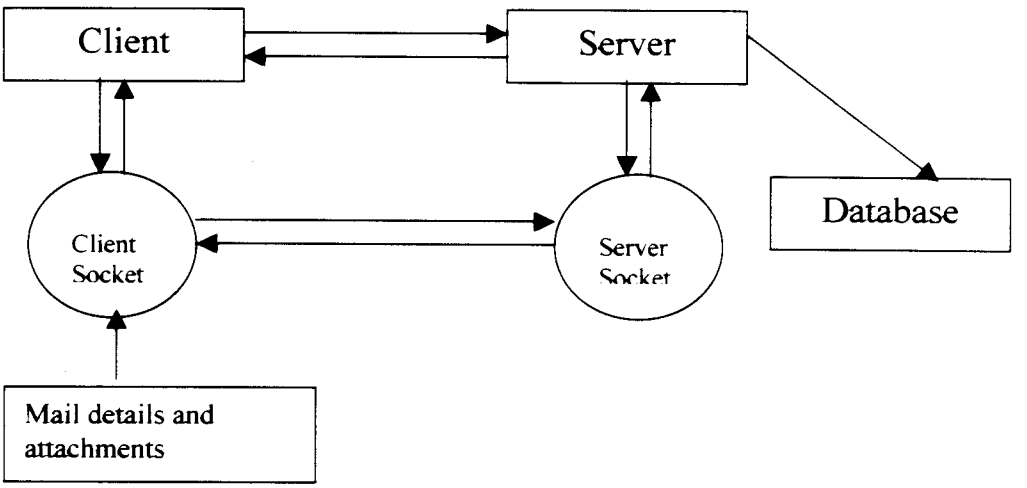
SELECTIVE E-MAIL DOWNLOADING



WRITING NEW MAILS



SENDING MAILS



DATABASE DESIGN

LOGIN

FIELD NAME	TYPE	SIZE
IDNO	Text	50
Lname	Text	50
Passwd	Text	50

MASTER

FIELD NAME	TYPE	SIZE
Msgno	Auto number	Long int
From1	Text	50
To1	Text	50
Sub	Text	50
Mesg	Text	255
Attachname	Text	50
Apath	Text	50
Visited	Boolean	Yes/no
Date	Date/time	date

USER DETAILS

FIELD NAME	TYPE	SIZE
Name	Text	50
Dofb	Date/time	Date/time
Qualification	Text	50
Address	Text	50
Phno	Text	50
Idno	Text	50

SOFTWARE REQUIREMENT SPECIFICATION

1 Introduction

1.1 Purpose

The purpose of this document is to describe all unambiguous and specific software requirements, interface requirements; performance requirements, design constraints, software attributes and other issues relevant to the selective E-mail download system.

Scope

This document describes the requirements of the system and serves dual purpose – (a) It is description of the framework of the task for use by developers and (b) it will also be the basis for validating the final delivered system.

1.3 Acronyms, Abbreviations and Terms

- Client: Person(s) or organization that is paying for the product being developed. They also decide on the main functions of the product.

- Contract: Legally binding document agreed upon by client and supplier/developer. It includes the technical, organizational, cost and scheduled requirements of a product.

- Developer: Is the same as supplier within this report.

- E-mail server: The application, which deals in receiving mails and sending mails

  Product: E-mail server.

- SRS: Software Requirement Specification.

- Supplier: Person(s) or organization that is developing the product being discussed. Also is referred to as developer in this case.

  User: The person who have registered with the Server

1.4 References

1. IEEE Std 830, Guide to Software Requirements Specification.

1.5 Developers' Responsibilities' Overview

The developers are responsible for (a) designing, developing and installing the software system and (b) seeking information (for clarification or otherwise) from the clients, where necessary.

Any change made to the requirement in the future will have to go through a formal change approval process. The developer may not make any alteration without the permission of the client.

2. General Description

2.1 Product Function Overview

The product is E-mail Server and client application, where the user logs on to the server and checks for new mails and sends new mails and also has access over old mails

The user runs the client application and enters the login and pass word. Only the valid users enter the next screen where he has the option to receive old, new mails and also write new mails. The mails and attachments the user sends to the server are put to the particular users and the database, which holds the information about the messages and other details, is updated

2.1 User Characteristics

This application is for people who are familiar with executing programs in windows environment.

3. SPECIFIC REQUIREMENTS

3.1 External Interface Requirements

3.1.1 User Interfaces

The users are provided with screens with options for entering details about the mail and for selective downloading of mails.

3.1.2 Hardware Interface

A server, which will host Server application, computers and modems for development and telephone lines.

3.1.3 Software Requirements

Windows 98 OS with java

4 Performance Constraints

Depends on the speed of the intranet connection, network traffic and client system

5 Design Constraints

5.1 Software constraints:

Components and tools needed for development:

(i) JDK 1.2.
(ii) JSDK 2.1.
(iii) MS Access.

For users:

(i) Intranet connection.
(ii) Java Runtime Environment

5.2 Hardware constraints:

(i) Adequate disk space required for storage.

6 Acceptance Criteria

The developers are required to take the client on a comprehensive tour about the application and ensure that all agreed specifications are met.

ELECTRONIC MAIL

Electronic mail is nothing but passing the message-using computer networks. The network can be either Internet or intranet. Internet is network of more than a million computers and any one can access it. So e-mails have become popular, economical and fastest way to communicate. We made alive the project with the help of intranet where in it proved to be the testing ground of our pilot project.

E-mails are passed on from one location to the other by the electronic connection the computer has with the host or server. The sending process checks for the computer name and sends the mail to that computer and the computer on the other end just checks for the username if the username matches with the database then the mail is put in his mailbox.

At present most e-mail servers use SMTP (Simple Mail Transfer Protocol) for sending e-mails and for receiving they use either POP3 (Post Office Protocol) or IMAP4 (Internet Mail Access Protocol). Selective e-mail download facility can be availed from e-mail servers that support IMAP.

Internet and intranet e-mails provide paper free communication thereby mobilizing the idea of green world. Wherever you go what ever you do as long you are a valid user, the information is at your fingertips.

PROJECT DESCRIPTION

SCREEN 1

The initial screen is a validation screen where the user enters his login name and password. When the user clicks OK, the client is checked whether he is a valid user or not.

The client contacts the server and the server checks with the Database and returns valid, if the user is a registered user else the server returns invalid.

SCREEN 2

Only valid users get the next screen, where they have the options for sending, receiving new, receiving old, or writing new mails. On selecting an option a particular servlet is invoked depending on the option chosen.

SCREEN 3

On selecting to receive new or old mails, the client enters this interactive screen where he views the details about the mail like, from where the mail has come and what is its subject etc. The user selects the message no's that are to be downloaded. On clicking the download option, a dialog box appears asking the client where the mails are to be downloaded. On specifying the path the files are downloaded to the particular folder or drive

The user is also provided with option to delete the selected mails.

For downloading of mails the server creates a socket and makes it listen in a particular port, and asks the client contact in the particular port. When the

client makes connection to that port no, the server initiates the download of the files.

On pressing either of the buttons the program terminates after the process is over.

SCREEN 4

On selecting to write a new mail, the client enters the screen which contains fields that are to be filled by the client like to whom the mail is, subject of the mail, attachment if any, message etc.

The client after entering a mail clicks the OK button, the mail is saved and the fields are cleared for next mail. On pressing the CANCEL button the user is presented with screen2

SCREEN 5

On pressing the SEND button the mails are send to the appropriate users and the program terminates. The Servlet receives the list of files to be uploaded and the server creates Socket, the client contacts this socket and the files are transferred to the appropriate places in the server.

PROGRAMMING AMBIENCE

HARDWARE DESCRIPTION (MINIMUM REQUIREMENTS)

Client Side

Processor	: Pentium MMX @ 166MHz
Main Memory	: 64 MB
Hard Disk Drive	: 200 MB
Floppy Disk Drive	: 1.44 MB
Display Type	: VGA Color
Mouse	: Standard 2 – button
Keyboard	: Standard 104 Keys US Layout

Server Side

Processor	: Pentium 4 @1.5 GHz
Main Memory	: 2 GB
Hard Disk Drive	: 80 GB
Floppy Disk Drive	: 1.44 MB
Display Type	: SVGA Color
Mouse	: Standard 2 – button
Keyboard	: Standard 104 Keys US Layout

DESCRIPTION SOFTWARE & PACKAGES USED

Platform	: Windows NT 4
Back end	: MS Access 2000
Front end	: Java Application
Programming Language	: Java, JDBC, Servlets

PLATFORM

This application is developed on Microsoft Windows NT 4 operating system. Because of the popularity Windows NT 4 enjoys, it is apt choice for the operating system. Windows NT 4 is highly user friendly and the main advantage of this famous GUI based OS is the huge application base available over it.

A lot of front end and back end tools are available for Windows NT 4 platform. The users are very much familiar with its interfaces that feel convenient to work with. This also curtails the amount training for the novice. Thereby our choice for Windows NT 4 is the right one to promote our application.

JAVA

Java was developed at Sun Microsystems under interesting circumstances. Working on Java originally began with the goal of creating a platform independent language and OS for consumer electronics. The original intention was to use C++ but as the work progressed the Java developers came to an idea of creating a unique language that would serve them in a better way.

Java is both a programming language and an environment for executing programs written in Java. Unlike traditional compilers that convert source code into machine level instructions, the Java compiler Java source code into byte code which runs in Java Virtual machine, which, has been developed for every operating systems in the World. Java is suitable for many general programming tasks and in fact many of the Java tools are written in Java. It is a compiler development custom that a language has come to an age where its compiler can be written in the same language. According to this custom

The multithreading concept is Java has enhanced parallel processing environment to great heights. For instance we can take the example of banking tasks. So due to its simplicity, robustness, architecturally neutral in nature, dynamic and secure nature we prefer Java to other languages.

JAVA AS AN INTERNET LANGUAGE

The wonder of Java as an Internet development language is related to its capability to solve two key problems in the Internet arena.

- The WWW content is passive and static
- Delivery of WWW content is dependent on configuration of each users web browser.

The Internet helped catapult Java to the forefront of programming and Java in turn has had a profound effect on the Internet. The reason for this quite simple i.e., Java expands the universe of objects that can move about freely in cyber space.

As desirable as dynamic, networked programs are, they also present serious problems security and portability. As you will see Java addresses those concerns and by doing so has opened the door to an exciting new form of program called the applet.

Java goes far beyond being a mere derivative of C++ adding scope to a greater extent. Moreover Java strength as an Internet language helps in client server development. Java strength in terms of network awareness, security, portability and performance make it ideally suitable corporate client server development as well as Internet development.

JDBC

JDBC is a Java API that documents a standard framework for dealing with tabular and relational data. While JDBC 2.0 begins a move to make SQL semitransparent to the programmer SQL is still the *lingua franca* of the standard database engines and represents a major industry victory in the effort to separate data from code.

SQL is a standardized language used to create, manipulate examine and manager relational database

- A database is essentially a smart container for tables.
- A table is a container comprised of rows.
- A row is conceptually a container comprised of columns.
- A column is a single data item having a name, type and value.

A database approximates a file system to appreciable level, a table approximates a file, a row approximates a record or structure and a column approximates a field or variable. Because SQL is an application specific language a single statement can be very expressive and can initiate high-level actions such as sorting and merging on an entire set of data. However we must connect to a database before sending SQL commands and each database vendor has a different interface to do so, as well as different extensions of SQL.

JDBC 1.0

The JDBC 1.0 API provided the basic framework for data access consisting primarily of the following interfaces and classes.

- Driver
- Driver Manager
- Connection
- Statement
- Prepared statement
- Callable statement
- Result set
- Database meta data
- Result set meta data
- Types

SERVLETS

Java language was originally intended for use in small-embedded devices. Businesses have recognized Java's potential on the server – Java is inherently suitable for large client – server applications. The cross platform nature of Java is extremely useful for heterogeneous collection of servers running various flavors of the UNIX, Windows and Macintosh operating systems.

Java's modern, object oriented, memory protected design allows developers to cut development cycles and increase reliability. In addition, Java's built-in support for networking and enterprise API's providing access to legacy data, easing the transition from older client – server systems.

Java servlets are a key component of server side Java development. A servlet is a small plug able extension to a server that enhances the server's functionality. Servlets allow developers extend and customize any java enabled server or any custom server with a hitherto unknown degree of portability, flexibility and ease.

Java servlet is a generic Server extension. Servlets are commonly used with Web Servers, where they can take the place of CGI-Scripts. A servlet is similar to a proprietary server extension. Except that it runs inside a JVM(Java Virtual Machine) on the server, so it is safe and portable. Servlets operate solely within the Domain of the Server unlike the applets that require the support of the web browser.

CONCLUSION

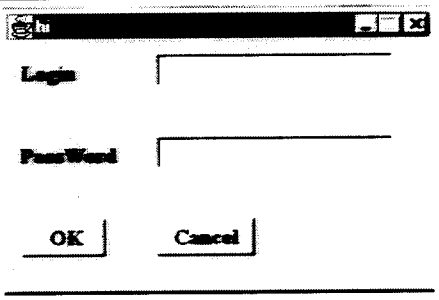
The selective e-mail downloading has significantly reduced the time between the client and the server. It also uses the hard disk space effectively. The project has emphasized greatly on our objectives and the result obtained is fruitful on convincing.

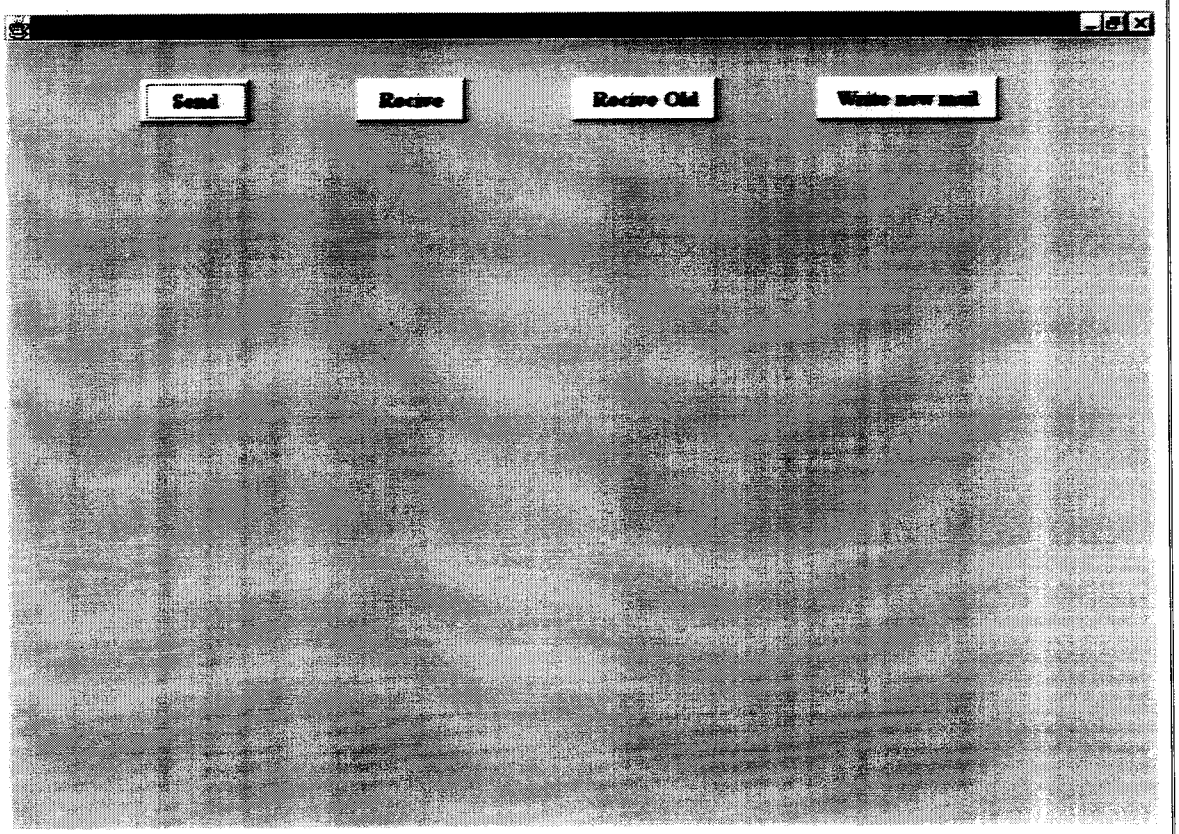
Although it is a pilot project the underlying principles can be implemented in a full-fledged way in the Internet.

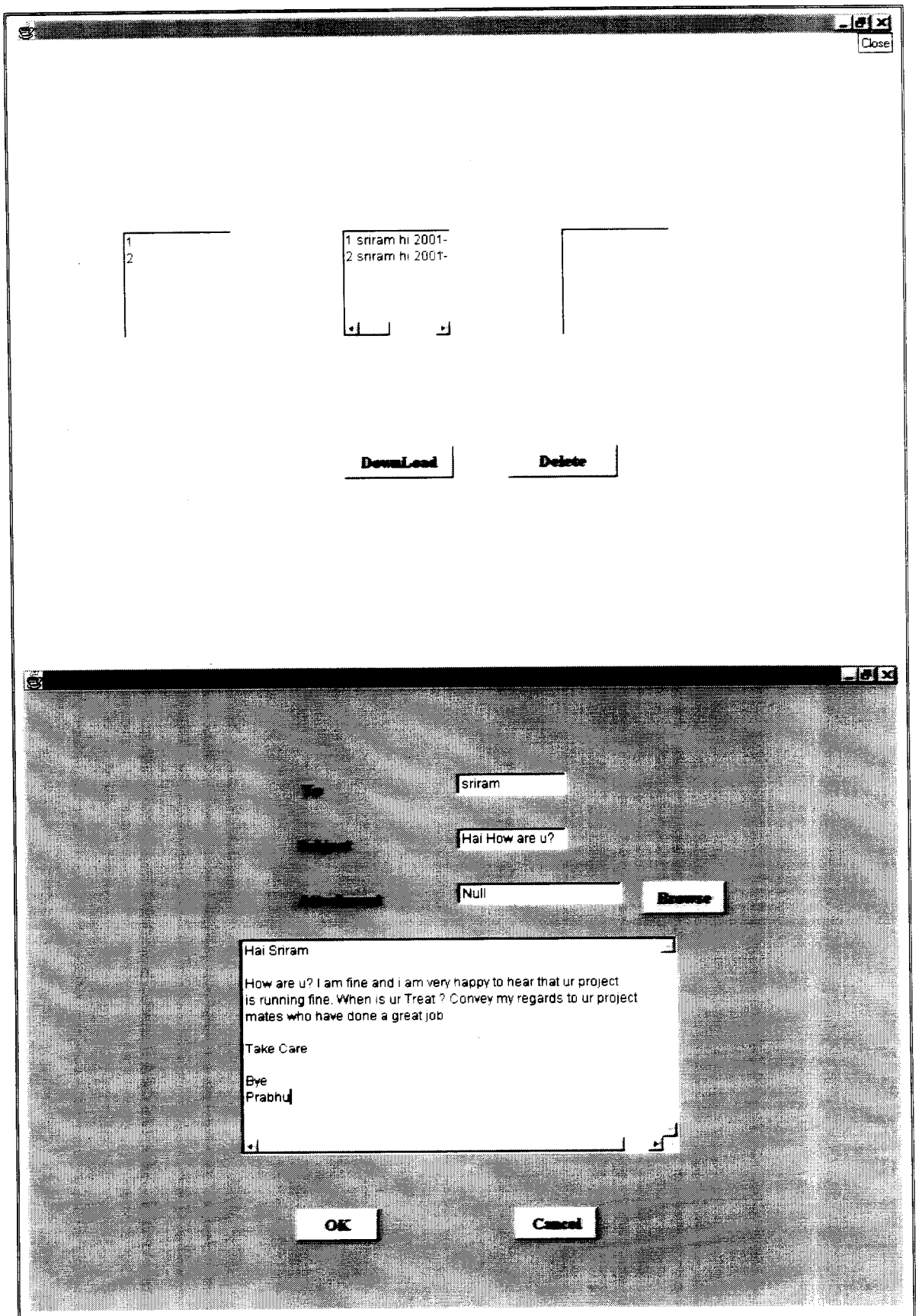
FUTURE ENHANCEMENT

The application has been designed and developed flexibly according the current objectives. Since the new additional requirements may sprout in future, the project is capable of being modified to the future requirements.

Further development may be made the direction of the project towards the internet with a robust database.


Login
PassWord
OK
Cancel


Send
Recive
Recive Old
Write new mail


Close
1
2
1 sriram hi 2001-
2 sriram hi 2001-
DownLoad
Delete
sriram
Hai How are u?
Null
Browse
Hai Sriram
How are u? I am fine and i am very happy to hear that ur project is running fine. When is ur Treat ? Convey my regards to ur project mates who have done a great job
Take Care
Bye
Prabhu
OK
Cancel

Sample Coding

CLIENT SIDE CODING

CODES FOR THE LOGIN PAGE

```
import java.awt.*;
import java.awt.event.*;
import java.util.*;
import java.net.*;
import java.io.*;

public class aw1 implements ActionListener,KeyListener
{
   Frame f;
   Button b1,b2,b3,b4;
   TextField tf1,tf2;
   Label l1,l2;
   GridBagLayout g;
   GridBagConstraints gs;
   String lg,pd,tmp,dt1;
   char c;
   aw41 next;
   public aw1()
   {
      c = '*';
      lg = new String("");
      pd = new String("");
      tmp = new String("");
      dt1 = new String("");
      f = new Frame("hi");
      b1 = new Button("   OK  ");
      b1.setFont(new Font("TimesRoman",Font.BOLD,15));
      b1.addActionListener(this);
      b2 = new Button(" Cancel ");
      b2.setFont(new Font("TimesRoman",Font.BOLD,15));
      b2.addActionListener(this);
      tf1 = new TextField("",20);
      tf2 = new TextField("",20);
      tf2.addKeyListener(this);
```

```
//tf2.echoCharIsSet(false);
tf2.setEchoChar(c);
l1 = new Label("Login");
l1.setFont(new Font("TimesRoman",Font.BOLD,15));
l2 = new Label("PassWord ");
l2.setFont(new Font("TimesRoman",Font.BOLD,15));
g = new GridBagLayout();
gs = new GridBagConstraints();
f.setLayout(g);
gs.insets = new Insets(10,10,10,10);
gs.weightx = 1.0;
gs.weighty = 1.0;

gs.gridx = 0;
gs.gridy = 0;
gs.anchor = GridBagConstraints.NORTHWEST;
gs.gridwidth = 5;
gs.gridheight = 1;
g.setConstraints(l1,gs);
f.add(l1);

gs.gridx = 2;
gs.gridy = 0;
gs.gridwidth = GridBagConstraints.REMAINDER;
gs.gridheight = 1;
g.setConstraints(tf1,gs);
f.add(tf1);

gs.gridx = 0;
gs.gridy = 1;
gs.gridwidth = 5;
gs.gridheight = 1;
g.setConstraints(l2,gs);
f.add(l2);

gs.gridx = 2;
gs.gridy = 1;
gs.gridwidth = GridBagConstraints.REMAINDER;
gs.gridheight = 1;
g.setConstraints(tf2,gs);
```

```
        f.add(tf2);

        gs.gridx = 0;
        gs.gridy = 3;
        gs.gridwidth = 2;
        gs.gridheight = 1;
        g.setConstraints(b1,gs);
        f.add(b1);

        gs.gridx = 2;
        gs.gridy = 3;
        gs.gridwidth = GridBagConstraints.RELATIVE;
        gs.gridheight = 1;
        g.setConstraints(b2,gs);
        f.add(b2);

        f.setSize(300,200);
        f.setLocation(320,200);
        f.setResizable(false);
        f.setBackground(new Color(100,200,150));
        f.addWindowListener(new w());
        f.show();
    }
    public void actionPerformed(ActionEvent e)
    {
            lg = tf1.getText();

         if (e.getSource() == b1)
          {
            httpobj obj;
            try
            {

            Properties prop = new Properties();
            prop.put(lg,pd);
            URL serv = new URL("http://localhost:8080/servlet/log ");
            obj = new httpobj(serv);
            InputStream in1 = obj.sendGetmessage(prop);
            DataInputStream out1 = new DataInputStream(new
BufferedInputStream(in1));
```

```
        dt1 = out1.readLine();
        System.out.println(" from post " + dt1);
        in1.close();
        out1.close();
        }
       catch(Exception e2)
       {
       System.out.println(" ERR : " + e2);
       }

      if(dt1.equals("valid"))
       {
       f.setVisible(false);
       System.out.println("user = " + lg);
       next = new aw41(lg);
       }
      else
      System.exit(0);
     }

   else
       System.exit(0);
}

public void keyPressed(KeyEvent ke)
{
   StringBuffer bf = new StringBuffer(pd);
   StringBuffer bf1 = new StringBuffer(tf2.getText());

   if(ke.getKeyCode() > 47 && ke.getKeyCode()<96)
   pd = pd + ke.getKeyChar();
   if(ke.getKeyCode() == 8)
   {
   bf.deleteCharAt(pd.length()-1);
   pd = bf.toString();
   }
   if(ke.getKeyCode()==127)
   {
      if(bf1.length() > 0)
      tf2.setText(bf1.toString() + "*");
```

```
        }
    }

    public void keyReleased(KeyEvent ke)
    {}
    public void keyTyped(KeyEvent ke)
    {}

    public class w extends WindowAdapter
    {
        public void windowClosing(WindowEvent e)
        {
            System.exit(0);
            }
    }
    public static void main(String args[])
    {
        aw1 a = new aw1();
        }
}
```

CODES FOR OPTION SCREEN

```
import java.awt.*;
import java.awt.event.*;
import java.net.*;
import java.io.*;
import java.util.Properties;
import java.util.LinkedList;
import java.util.Iterator;
public class aw41 extends Frame implements ActionListener
{
        Socket ss;
        httpobj obj;
        BufferedInputStream in;
        BufferedOutputStream out;
        java.net.URLConnection con1;

        Button send,recv,old,ok,wnew,browse,save,cancel,rr;
```

```
Label to,sub,bcc,attach;
TextField to1,sub1,bcc1,attach1;
TextArea ta;
String lg;
String att,attpt;
details d;
objfile o;
String dt1;
int j,chk;
boolean de;
public aw41(String tmp)
{
        de = true;
        chk = 0;
        att = new String("none");
        lg = new String(tmp);
        o = new objfile(lg+"1");
        setLayout(null);
        con1 = null;
        send = new Button("Send");
        send.setFont(new Font("TimesRoman",Font.BOLD,15));
        send.addActionListener(this);
        send.setBounds(100,50,75,30);
        add(send);

        recv = new Button("Recive");
        recv.setFont(new Font("TimesRoman",Font.BOLD,15));
        recv.addActionListener(this);
        recv.setBounds(250,50,75,30);
        add(recv);

        old = new Button("Recive Old");
        old.setFont(new Font("TimesRoman",Font.BOLD,15));
        old.addActionListener(this);
        old.setBounds(400,50,100,30);
        add(old);

        wnew = new Button("Write new mail");
        wnew.setFont(new Font("TimesRoman",Font.BOLD,15));
        wnew.addActionListener(this);
```

```
        wnew.setBounds(570,50,125,30);
        add(wnew);

        setBackground(new Color(100,150,250));
        setVisible(true);
        addWindowListener(new w());

        show();



    }
public void actionPerformed(ActionEvent e)
{
    if(e.getSource()== wnew)
    {
        wnew.setVisible(false);
        old.setVisible(false);
        recv.setVisible(false);
        send.setVisible(false);

        to = new Label("  To ");
        to.setFont(new Font("TimesRoman",Font.BOLD,15));
        to.setBounds(250,100,75,30);
        add(to);

        to1 = new TextField(25);
        to1.setBounds(400,100,100,20);
        add(to1);

        sub = new Label(" Subject ");
        sub.setFont(new Font("TimesRoman",Font.BOLD,15));
        sub.setBounds(250,150,75,30);
        add(sub);

        sub1 = new TextField(20);
        sub1.setBounds(400,150,100,20);
        add(sub1);

        attach = new Label(" Attachment ");
```

```
attach.setFont(new Font("TimesRoman",Font.BOLD,15));
attach.setBounds(250,200,90,30);
add(attach);

attach1 = new TextField(40);
attach1.setBounds(400,200,150,20);
attach1.setEditable(false);
attach1.setText("Null");
add(attach1);

browse = new Button("Browse");
browse.setFont(new Font("TimesRoman",Font.BOLD,15));
browse.addActionListener(this);
browse.setBounds(570,200,75,30);
add(browse);

ta = new TextArea(25,25);
ta.setBounds(200,250,400,200);
add(ta);

ok = new Button(" OK ");
ok.setFont(new Font("TimesRoman",Font.BOLD,15));
ok.addActionListener(this);
ok.setBounds(250,500,75,30);
add(ok);

cancel = new Button(" Cancel ");
cancel.setFont(new Font("TimesRoman",Font.BOLD,15));
cancel.addActionListener(this);
cancel.setBounds(450,500,75,30);
add(cancel);

rr = new Button(" Enter all details ");
rr.setFont(new Font("TimesRoman",Font.BOLD,15));
rr.addActionListener(this);
rr.setBounds(650,400,100,30);
rr.setVisible(false);
add(rr);

show();
```

```
}
if(e.getSource()==cancel)
{
 remove(browse);
 remove(cancel);
 remove(to);
 remove(sub);
 remove(ok);
 remove(attach);
 remove(to1);
 remove(sub1);
 remove(attach1);
 remove(ta);
 wnew.setVisible(true);
 old.setVisible(true);
 recv.setVisible(true);
 send.setVisible(true);
 if(chk >0)
 {
    o.save(lg+"1");
    rr.setVisible(false);
  }

}
if(e.getSource()==browse)
{

  FileDialog fd = new FileDialog(this,"FILE SELECT");
  fd.setVisible(true);
  String add = new String(fd.getDirectory());
  att  = new String(fd.getFile());
  add = add + att;
  System.out.println(" addd" + add);
  attach1.setText(add);
  }

 if(e.getSource()==ok)
 {

     int ch = 0;
```

```
            String tmp33 = new String("");
            tmp33 = new String(to1.getText());
            if(tmp33.length()>0) ch++;
            tmp33 = new String(sub1.getText());
            if(tmp33.length()>0) ch++;
            tmp33 = new String(ta.getText());
            if(tmp33.length()>0) ch++;
            if(ch == 3)
            {
            d = new
details(lg,to1.getText(),sub1.getText(),att,attach1.getText(),ta.getText());
            o.add(d);
            to1.setText("");
            sub1.setText("");
            attach1.setText("");
            ta.setText("");
            att = new String("none");
            System.out.println(" obj created");
            chk++;
            }
            else
            {     rr.setVisible(true);
                 ok.setVisible(false);
             }

        }
        if(e.getSource()==rr)
        {
            rr.setVisible(false);
            ok.setVisible(true);
         }
        if(e.getSource()==old)
        {
         Properties p = new Properties();
         p.put("lname",lg);
         p.put("visited","true");

        try
          {
            URL serv = new URL("http://localhost:8080/servlet/log2");
```

```
            obj = new httpobj(serv);
            InputStream in1 = obj.sendGetmessage(p);
            DataInputStream out1 = new DataInputStream(new
BufferedInputStream(in1));
            dt1 = out1.readLine();
            System.out.println(" port no = " + dt1);
            in1.close();
          }
        catch(Exception ee)
          {
          System.out.println(" ERR : " + ee);
          }


          try
          {

            System.out.println("port no " + dt1);
            ss = new Socket("127.0.0.1",Integer.parseInt(dt1));
            in = new BufferedInputStream(ss.getInputStream());
            System.out.println("contacting port no " + dt1);
          }
          catch(Exception ed)
          {
            System.out.println(" error client " + ed);
           }

          try
          {
           FileOutputStream out = new FileOutputStream(lg);
            while(de)
             {

                  j = in.read();
                  out.write(j);

                  if(j == -1)
                  {
                   de = false;
                   ss.close();
```

```
            in.close();
            out.close();
           }
           else
           {

           System.out.print((char)j);
           }

      }
   }
   catch(Exception mm)
   {
     System.out.println(" err while dload " + mm);
   }

 setVisible(false);
dloadmesgs jkl = new dloadmesgs(lg);

}

if(e.getSource()==send)
{
   Properties p = new Properties();
   LinkedList fls = new LinkedList();
   Iterator itr;
   int zzz = 0;
   String nnn = new String("");
  try
   {
     FileInputStream fin = new FileInputStream("att"+lg+"1");
     ObjectInputStream in = new ObjectInputStream(fin);
     fls = (LinkedList)in.readObject();
     in.close();
     fin.close();
   }
   catch(Exception zxcv)
   {
     System.out.println(" err in 1 " + zxcv);
    }
```

```
            itr = fls.iterator();
           while(itr.hasNext())
           {
            nnn = (String)itr.next();
            zzz++;
            }
           p.put("no",Integer.toString(zzz));

            itr = fls.iterator();
           zzz = 1;
           while(itr.hasNext())
           {
            nnn = (String)itr.next();
            p.put(Integer.toString(zzz),nnn);
            zzz++;
            }

         fls = new LinkedList();
        try
        {
           FileInputStream fin = new FileInputStream("clatt"+lg+"1");
           ObjectInputStream in = new ObjectInputStream(fin);
           fls = (LinkedList)in.readObject();
           in.close();
           fin.close();
        }
        catch(Exception zxcv)
        {
           System.out.println(" err in 1 " + zxcv);
         }
      try
        {
           URL serv = new URL("http://localhost:8080/servlet/log3");
           obj = new httpobj(serv);
           InputStream in1 = obj.sendGetmessage(p);
           DataInputStream out1 = new DataInputStream(new
BufferedInputStream(in1));
           dt1 = out1.readLine();
           System.out.println(" port no = " + dt1);
```

```
        in1.close();
        out1.close();
      }
    catch(Exception ee)
      {
     System.out.println(" ERR 2: " + ee);
      }

     Iterator lll = fls.iterator();
    while(lll.hasNext())
    {
      try
      {

        System.out.println("port no " + dt1);
        ss = new Socket("127.0.0.1",Integer.parseInt(dt1));
        in = new BufferedInputStream(ss.getInputStream());
        out = new BufferedOutputStream(ss.getOutputStream());
        System.out.println("contacting port no " + dt1);
      }
      catch(Exception ed)
      {
        System.out.println(" error client " + ed);
       }

      try
      {
        boolean tttt = true;
        FileInputStream fin = new FileInputStream((String)lll.next());
        while(tttt)
        {
              j = fin.read();
              out.write(j);
              if(j==-1)
              {
              out.close();
              in.close();
              fin.close();
              }
         }
```

```
        }
        catch(Exception vb)
        {
          System.out.println(" err in file upload " + vb);
        }
      }
    }

      if(e.getSource()==recv)
      {
       Properties p = new Properties();
       p.put("lname",lg);
       p.put("visited","false");

      try
        {
          URL serv = new URL("http://localhost:8080/servlet/log2");
          obj = new httpobj(serv);
          InputStream in1 = obj.sendGetmessage(p);
          DataInputStream out1 = new DataInputStream(new
BufferedInputStream(in1));
          dt1 = out1.readLine();
          System.out.println(" port no = " + dt1);
          in1.close();
        }
       catch(Exception ee)
        {
        System.out.println(" ERR : " + ee);
        }


        try
        {

          System.out.println("port no " + dt1);
          ss = new Socket("127.0.0.1",Integer.parseInt(dt1));
          in = new BufferedInputStream(ss.getInputStream());
          System.out.println("contacting port no " + dt1);
        }
        catch(Exception ed)
```

```
        {
          System.out.println(" error client " + ed);
         }

        try
        {
         FileOutputStream out = new FileOutputStream(lg);
          while(de)
          {

                j = in.read();
                out.write(j);

                if(j == -1)
                {
                 de = false;
                 ss.close();
                 in.close();
                 out.close();
                }
                 else
                 {

                 System.out.print((char)j);
                 }

            }
         }
         catch(Exception mm)
         {
           System.out.println(" err while dload " + mm);
         }

        setVisible(false);
       dloadmesgs jkl = new dloadmesgs(lg);

       }
}
```

```
public class w extends WindowAdapter
{
    public void windowClosing(WindowEvent e)
    {
       System.exit(0);
       }
}



public static void main(String args[])
{

    aw41 a = new aw41("sriram");
    }
}
```

SERVER SIDE CODING

OPTION SCREEN SERVLET

```
import java.io.*;
import java.util.*;
import java.net.*;
import java.sql.*;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class log2 extends HttpServlet
{
    long mno[];
    int no;
    int ct;
    LinkedList dat;
    Iterator itr;
    static int i = 10024;
    String lname,visited;

   public static void incr()
   {
      i++;
      if(i>10124) i = 10024;
      }


     public log2()
     {
      dat = new LinkedList();
      lname = new String("");
      visited = new String("");
     }

   public void doGet(HttpServletRequest req, HttpServletResponse res)
throws ServletException , IOException
```

```
  {
    String s[];
    String s1;
    lname = req.getParameter("lname");
    visited = req.getParameter("visited");

    System.out.println("lname" + lname);
    try
     {
      DriverManager.registerDriver(new sun.jdbc.odbc.JdbcOdbcDriver());
      Connection con = DriverManager.getConnection("jdbc:odbc:sri");
      Statement st = con.createStatement();
      ResultSet rs;
      if(visited.equals("false"))
      rs = st.executeQuery("select * from master where to1 = '"+lname+"'
and visited = false");
      else
      rs = st.executeQuery("select * from master where to1 = '"+lname+"'
and visited = true");
      while(rs.next())
      {
       System.out.println(" insde sql query ");
      dform d = new
dform(rs.getString(1),rs.getString(2),rs.getString(4),rs.getString(9),rs.getStri
ng(6));
      dat.add(d);
      System.out.println(" data= " + d.toString());
      }
      con.close();

     }

    catch(Exception e1)
    {
     System.out.println(" inside sql err :" + e1);
    }

    try
    {
       FileOutputStream fout = new FileOutputStream(lname);
```

```
        ObjectOutputStream oos = new ObjectOutputStream(fout);
        oos.writeObject(dat);
        System.out.println(" file created ");
        System.out.println(" dat = " + dat);
        oos.close();
        fout.close();
        dat = new LinkedList();

    }

    catch(Exception e2)
    {
     System.out.println(" err in crea file " + e2);
     }

    incr();

    System.out.println(" port no = " + i);
    res.setContentType("text/plain");
    PrintWriter out = res.getWriter();
    out.println(Integer.toString(i));
    out.close();
    upserv1 u = new upserv1(i,lname);

   }


}
```

SQL QUERY CODING

```
import java.io.*;
import java.sql.*;
import java.util.*;

    public class sqltry1
  {
     String from1,sub1,date1,mesg1;
     long msgno[];
     LinkedList all;
```

```
    String tmp;
    public sqltry1(long msgnos[])
    {
     msgno = new long[msgnos.length];
     all = new LinkedList();
     tmp = new String("");
     from1 = new String("");
     sub1 = new String("");
     date1 = new String("");
     mesg1 = new String("");
     for(int ct=0;ct<msgnos.length;ct++)
     {
     msgno[ct] = msgnos[ct];
//       System.out.println(msgno[ct]);
      }
      System.out.println(" inside sqltry cons ");
     }
    public LinkedList verify()
    {
     int i = 0;
     try
     {
      DriverManager.registerDriver(new sun.jdbc.odbc.JdbcOdbcDriver());
      Connection con = DriverManager.getConnection("jdbc:odbc:sri");
      Statement st = con.createStatement();
      ResultSet rs;
      for(i=msgno.length-1;i>=0;i--)
      {
       System.out.println(" inside for loop " + i);
       System.out.println(" msgno = " + msgno[i]);
       rs = st.executeQuery("select * from master where msgno ="
+msgno[i]);
       while(rs.next())
       {
        from1 = rs.getString(2);
        sub1 = rs.getString(4);
        date1 = rs.getString(9);
        mesg1 = rs.getString(5);
        tmp = "from " + from1 + " sub "+sub1+" date " +date1+" message "
+ mesg1 + "\n";
```

```
    /*    System.out.println(tmp);
          System.out.println(" Valid user "); */
          all.add(tmp);
          tmp = rs.getString(7);
          if(tmp.equals("Null"))
          {
             System.out.println(" no attachment ");
          }
          else
          {
             all.add("attach");
             all.add(tmp);
          }
        }
       }
      }
      catch(Exception e)
      {
            System.out.println(" err :"+e);
            e.printStackTrace();
            }

     return all;
     }

  }
```

CODE FOR SERVER SIDE SOCKET CREATION

```
import java.net.*;
import java.io.*;
import java.util.*;

public class upserv1
{

      ServerSocket s;
      Socket s1;
      int no,j;
```

```
BufferedInputStream in;
BufferedOutputStream out;
boolean t;
public  upserv1(int n,String lname)
{
        this.no = n;

        try
          {
             s = new ServerSocket(no);
          }
        catch(Exception e)
          {
             System.out.println(" error " + e);
          }

         try
         {
            t = true;
            System.out.println(" Server Ready ");
            s1 = s.accept();
            System.out.println(" Client ready ");
            in = new BufferedInputStream(s1.getInputStream());
            out = new BufferedOutputStream(s1.getOutputStream());

            System.out.println(" inside itr loop :");
            File f = new File(lname);
            FileInputStream fin = new FileInputStream(f);
            BufferedInputStream bfin = new BufferedInputStream(fin);

             while(t)
             {

               j = bfin.read();
               out.write(j);
               if(j!= -1)
               {
               System.out.print((char)j);
               }
               else
```

BIBLIOGRAPHY

1. Patrick Naughton & Herbert Schildt., "**The complete reference java 2",** The McGraw-Hill companies inc,New York,Third Edition.

2. John Zukowski,"**Mastering java2** ", BPB Publications, New Delhi, First Edition.

3. Jason Hunter, **"Java Servlet Programming",** Oreilly Publications, New Delhi, First Edition.

4. Bruce Eckel, **"Thinking in java",** BPB Publications, New Delhi, Second Edition.