

P-1889

COMMUNITY INITIATING SYSTEM FOR THE  
WORLD WIDE WEB – COMMUNE

By

Lakshmi Narayanan .S

Reg No 71204621012

of

Kumaraguru College of Technology

Coimbatore

A PROJECT REPORT

Submitted to the

FACULTY OF INFORMATION AND COMMUNICATION ENGINEERING

*In partial fulfillment of the requirements*

*for the award of the degree*

*of*

MASTER OF COMPUTER APPLICATIONS

July 2007



P 1889

Kumaraguru College of Technology

Coimbatore – 641 006

Department of Computer Applications

## BONAFIDE CERTIFICATE

Certified that this project report titled **Community Initiating System For The World Wide Web – Commune** is the bonafide work of **Mr. S.Lakshmi Narayanan (Reg No 71204621012)** who carried out the research under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form part of any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.




Project Guide



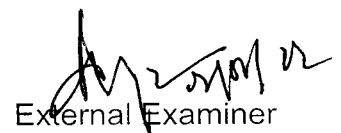
Head of Department

Submitted for the University Examination held on

02-07-2007



Internal Examiner



External Examiner



# Valiant Info Soft Pvt. Ltd.

Date: 29-05-2007

## TO WHOMSOEVER IT MAY CONCERN

Sub.: Regarding Student Project Training Completion

This is to confirm that **Mr. Lakshmi Narayanan S** (04MCA12) of M.C.A. department of **Kumaraguru College of Technology, Coimbatore** has finished his final year project training in our concern under the guidance of our concerned staffs. His code of conduct during the training period is good. The project topic is "**Community Initiating System for the World Wide Web**" and the training period is from 25-12-2006 to 25-05-2007.



## ABSTRACT

In the World Wide Web, people hangout on their favorite websites looking for information and more information. This Project "**COMMUNITY INITIATING SYSTEM FOR THE WORLD WIDE WEB**" provides a small utility named "**COMMUNE**", which enables the surfers of the same website to get-together.

In these days of internet age, many of the community initiating programs like orkut, ICQ, yahoo, msn messenger users are growing at a very high rate. This project is a completely based on an innovative idea by removing all the disadvantages in the existing chat systems and bringing in a new chat system to the user with at most flexibility in community development.

Sharing information is vital for early intervention to ensure that people get the services they require. It is also essential to protect the wastage of valuable time of surfers by sharing and group activity. This project provides a way to bring the surfers of the same website together, thereby creating as instant community, and by providing an instant messaging solution; the users could communicate across the community.

Thereby it is opening up endless possibilities for knowledge sharing and bringing together people with similar interests.

## ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of any task would be incomplete without mentioning the names of people who made it possible, whose constant guidance and encouragement crowns all efforts with success.

I convey my earnest thanks to **Mr.A.Muthukumar MSc., MCA., M.Phil., Assistant Professor, Kumaraguru College of technology** for her invaluable guidance, support and suggestions throughout the course of this project work.

I express my deep sense of gratitude to **Mr. Vivekanandhan, Associate Manager, Valiant Infosoft Pvt Limited, Chennai** for his invaluable guidance, support and suggestions throughout the course of this project work.

I would like to express my sincere thanks to **Dr.M.Gururajan, Head of the Department,** Department of Computer Science and Engineering, college of Engineering, Anna University for having permitted me to undertake this project.

Last but not the least; I am grateful to my parents and friends for their constant support and encouragement.

## TABLE OF CONTENTS

| CH. NO   | TOPIC                                  | PAGE NO.  |
|----------|--|-----------|
|          | Abstract                               | iii       |
|          | List of Tables                         | vii       |
|          | List of Figures                        | vii       |
| <b>1</b> | <b>INTRODUCTION</b>                    | <b>1</b>  |
|          | 1.1 System Overview                    | 1         |
|          | 1.2 Company Profile                    | 2         |
| <b>2</b> | <b>SYSTEM STUDY AND ANALYSIS</b>       | <b>3</b>  |
|          | 2.1 Problem statement                  | 3         |
|          | 2.2 Other Chat Environments            | 4         |
|          | 2.2.1 Orkut                            | 4         |
|          | 2.2.2 Myspace                          | 6         |
|          | 2.2.3 Yahoo Messenger                  | 7         |
|          | 2.2.4 Comparison Results               | 9         |
|          | 2.3 Proposed System                    | 10        |
| <b>3</b> | <b>SYSTEM DEVELOPMENT ENVIRONMENT</b>  | <b>12</b> |
|          | 3.1 Hardware Requirements              | 12        |
|          | 3.2 Software Technologies Used         | 13        |
|          | 3.2.1 Java Language                    | 14        |
|          | 3.2.2 Java Servlet                     | 15        |
|          | 3.2.3 Tomcat Webserver                 | 16        |
|          | 3.2.3 Web Browser                      | 17        |
| <b>4</b> | <b>SYSTEM DESIGN &amp; DEVELOPMENT</b> | <b>18</b> |
|          | 4.1 Use Case Diagram                   | 18        |
|          | 4.2 System Modules                     | 19        |
|          | 4.3 Modules Description                | 20        |
|          | 4.3.1 Registry Entry Module            | 20        |

|          |                                     |           |
|----------|-------------------------------------|-----------|
| 4.3.2    | Client Module                       | 22        |
| 4.3.3    | Server Module                       | 23        |
| 4.3.3.1  | Web Server Setup                    | 23        |
| 4.3.4    | Chat Module                         | 24        |
| 4.3.5    | Login Module                        | 24        |
| 4.3.6    | New User Registration               | 25        |
| 4.3.7    | Registered User Logging In          | 26        |
| 4.3.8    | Updating the Server                 | 27        |
| 4.4      | Database Design                     | 27        |
| 4.4.1    | Database Security                   | 29        |
| <b>5</b> | <b>TESTING &amp; IMPLEMENTATION</b> | <b>31</b> |
| 5.1      | Testing Overview                    | 31        |
| 5.1.4    | Testing Methodologies               | 33        |
| 5.2      | Test Cases                          | 36        |
| 5.3      | Implementation                      | 41        |
| <b>6</b> | <b>CONCLUSION AND ENHANCEMENTS</b>  | <b>42</b> |
| 6.1      | Conclusion                          | 42        |
| 6.2      | Suggestion for Further Development  | 43        |
|          | <b>REFERENCES</b>                   | <b>44</b> |

## LIST OF TABLES

|           |   |                                   |
|-----------|---|-----------------------------------|
| Table 2.1 | : | Chat Application Comparison Table |
| Table 4.1 | : | user_master                       |
| Table 4.2 | : | user_buddy_details                |
| Table 4.3 | : | user_website_details              |

## LIST OF FIGURES

|            |   |                                   |
|------------|---|-----------------------------------|
| Figure 1.1 | : | Valiant ship company logo         |
| Figure 2.1 | : | block diagram of existing system  |
| Figure 2.2 | : | Orkut webpage screen shot         |
| Figure 2.3 | : | my space webpage screen shot      |
| Figure 2.4 | : | yahoo messenger screen shot       |
| Figure 2.5 | : | proposed system block diagram     |
| Figure 4.1 | : | use case diagram                  |
| Figure 4.2 | : | welcome screen shot               |
| Figure 4.3 | : | login screen shot                 |
| Figure 4.4 | : | New User registration screen shot |
| Figure 4.5 | : | Chat welcome screen               |



# CHAPTER 1

## INTRODUCTION

### 1.1 SYSTEM OVERVIEW

In the World Wide Web, people hangout on their favorite websites looking for information and more information. This Project "Community initiating system for the world wide web" provides a small utility, which enables the surfers of the same website to get-together.

For example, let us look at a special interest website like "formulaonecars.com" at a given point of time, there could be 50 – 100 people (at least) browsing the site, looking for information.

These bunch of 50 -100 people have a common interest, they are looking for information on Formula One Sports cars. But unfortunately they can't talk to each other. It makes a "So near, yet so far" situation. If they could talk to each other and if they are able to share information about their interests, it could emerge as a new channel for knowledge sharing.

In these days of internet age, many of the community initiating programs like orkut, ICQ, yahoo, msn messenger users are growing at a very high rate. This project is a completely based on an innovative idea by removing all the disadvantages in the existing chat systems and bringing in a new chat system to the user with at most flexibility in community development.

This project provides a way to bring the surfers of the same website together, thereby creating as instant community, and by providing an instant messaging solution, the users could communicate across the community.

Thereby it is opening up endless possibilities for knowledge sharing and bringing together people with similar interests.

## 1.2 COMPANY PROFILE

Valiant Infosoft Pvt Limited , a wholly owned subsidiary of M/S Valiant Ship Management Limited ,Hong Kong is a major in development of shipping domain products.



**Figure 1.1 Valiant ship company logo**

Its major projects are into Shipping Domain with the efforts and expertise put in for over 3 decades in safe and profitable shipping along with an IT expertise of over a decade. Combined with the Technical and Commercial domain knowledge of shipping and ship management Valiant Infosoft was the first to offer LAN and Web compatible application to fit the needs of successful shipping companies which was one of their major projects.

Their main operations are in the domains like

- Ship management application
- Turnkey MRP [Manufacturing Resource Planning] Solutions
- Data Management and Migration Projects
- Health Management Solutions
- Online Logistics and cargo tracking application development
- HR outsourcing for IT, Marine and Industrial Domains
- Back office operation
- Secretarial and Front Office Management

<http://www.valiantship.com/software/>

## CHAPTER 2

### SYSTEM STUDY AND ANALYSIS

#### 2.1 PROBLEM STATEMENT

In a normal browsing environment, though these set of users surf the same website, they are not connected to each other. No interaction is possible between them.

Example : There may be several users at one possible time in Annauniv.edu webpage who cannot communicate with each other.

It makes a "So near, yet so far" situation. If they could talk to each other and if they are able to share information about their interests, it could emerge as a new channel for knowledge sharing.

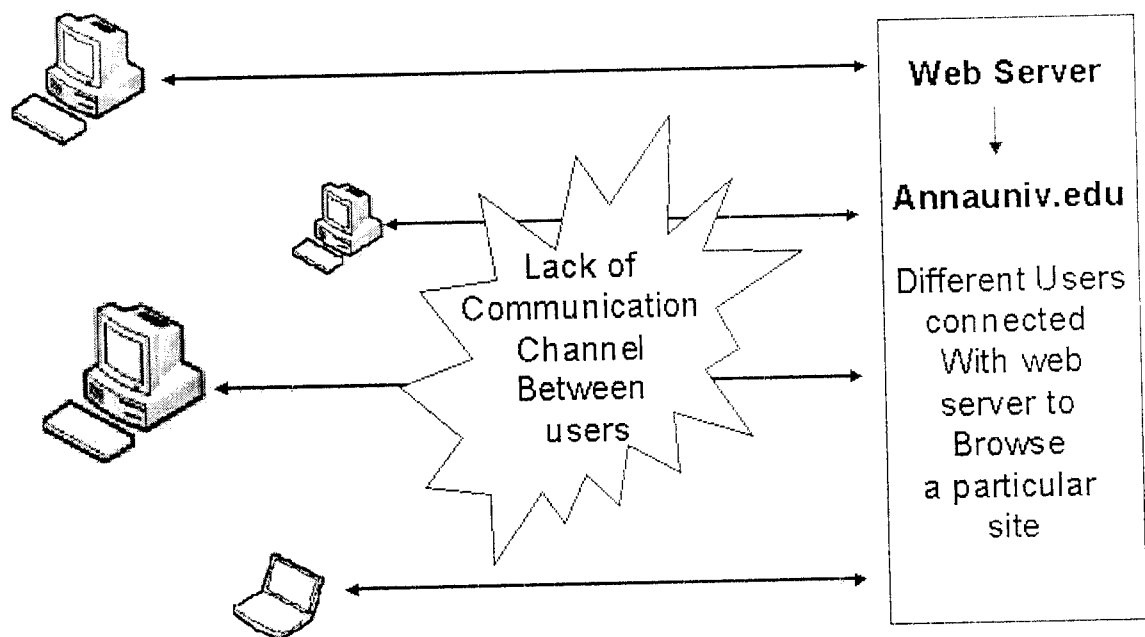


figure 2.1 block diagram of existing system

## 2.2 OTHER CHAT ENVIRONMENTS

### 2.2.1 ORKUT

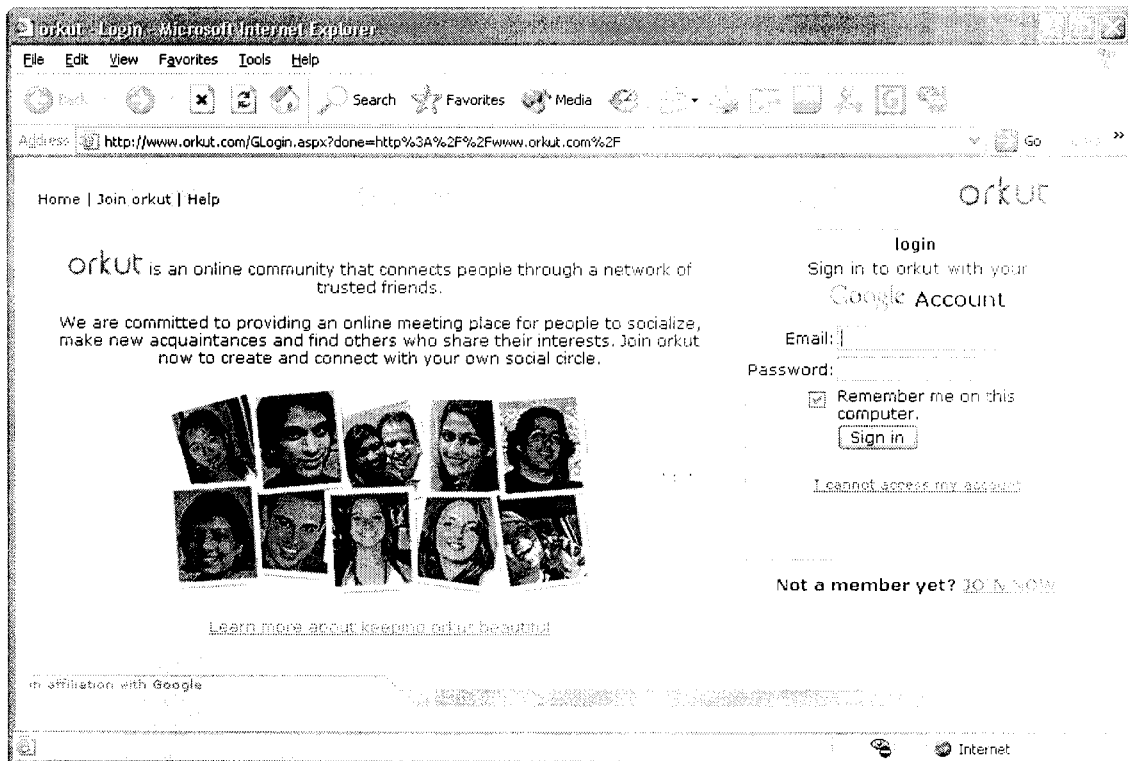
Orkut is an Internet social network service run by Google and named after its creator, Google employee Orkut Büyükkökten. It claims to be designed to help users meet new friends and maintain existing relationships. Similar to Friendster and MySpace, Orkut goes a step further by permitting the creation of easy-to-setup simple forums (called "communities") of users. For a short time between October 2006 and April 2007, Orkut permitted users to create accounts without an invitation. In April 2007, Orkut introduced polls in communities.

Orkut has a list of features unique to itself like "Scrapbook", "Communities" etc., apart from the normal features like messaging and photo album.

Scrapping, is popular among the Orkut community as a kind of offline chatting. You can scrap a person even if he is offline. The person to whom the scrap is sent will be able to see it when he visits his scrapbook. Contrary to personal messaging or email, scrap book entries are public, meaning that any one with an orkut account can read others scraps.

Another unique feature of orkut is "Communities". Anyone with an orkut account can create a community on anything. One can post topics, inform users about an event, ask them questions or just play games. There are more than a million communities on orkut on absolutely everything from pizza to pasta from Film star to superstar, from your pet to your teacher. The 1st 5 communities on orkut were started with in 24 hrs of launch of orkut.

In April 2007, orkut added a feature called "Polls" where in community users can be polled. Participation can be open for all or can be restricted to only the members of community.



**figure 2.2 Orkut webpage screen shot**

In addition to this there is a personal messaging feature. Users rarely use this feature. It is mainly used by community owners to ask others to join their community. Though the messages are meant to be personal they aren't. If you know the exact link of the message then you can visit read their messages.

The major success of orkut has come from its ability to create communities.

## 2.2.2 MY SPACE

MySpace is a popular social networking website offering an interactive, user-submitted network of friends, personal profiles, blogs, groups, photos, music and videos internationally. It is headquartered in Beverly Hills, California, USA,[1] where it shares an office building with its immediate owner, Fox Interactive Media; in turn, the owner of Fox Interactive (and therefore MySpace), News Corporation, is headquartered in New York City.

According to Alexa Internet, MySpace is currently the world's fifth most popular English-language website, the fifth most popular website in any language, and the third most popular website in the United States, though it has topped the chart on various weeks (it is possible that other websites have a greater number of unique visitors). The service has gradually gained more popularity than similar websites to achieve nearly 80% of visits to online social networking websites. It has become an increasingly influential part of contemporary popular culture, especially in English speaking countries.

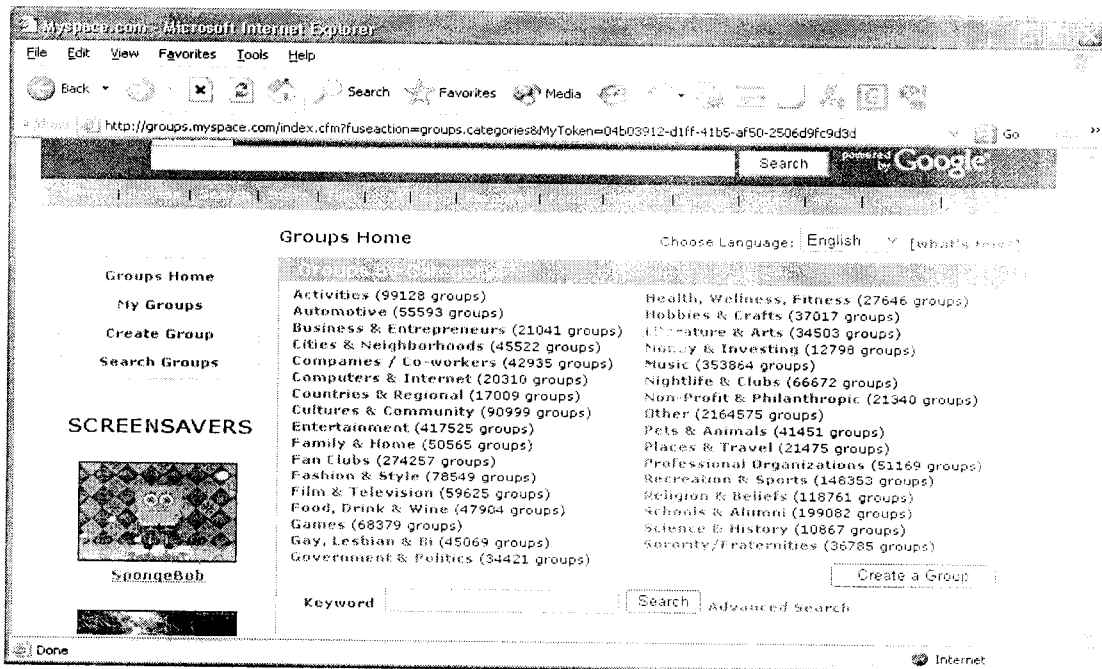


figure 2.3 myspace webpage screen shot

With the 100 millionth account being created on August 9, 2006, in The Netherlands and a news story claiming 106 million accounts on September 8,

2006, the site reportedly attracts new registrations at a rate of 230,000 per day.

\* Source wikipedia

### 2.2.3 YAHOO MESSENGER

Yahoo! Messenger is a popular advertisement-supported instant messaging client and protocol provided by Yahoo!. Yahoo! Messenger is provided free of charge and can be downloaded and used with a generic "Yahoo! ID" which also allows access to other Yahoo! services, such as Yahoo! Mail, where users can be automatically notified when they receive new email. Yahoo! offers PC to PC telephone, file transfers, web cam hosting, text messaging service, and chat rooms in various categories.

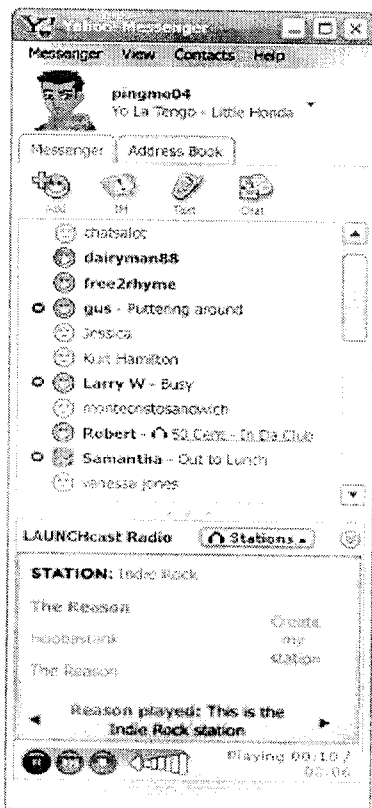


figure 2.4 yahoo messenger screen shot

In addition to instant messaging features similar to those offered by ICQ, it also offers (on Microsoft Windows) many unique features such as: IMVironments (customizing the look of Instant Message windows), address-book integration and Custom Status Messages. It was also the first major IM client to feature BUZZing and music-status. Another recently added feature is customized avatars.

Yahoo! Messenger 2.5.3 for Mac, released in 2003, is the current stable version for Mac OS X. It has fewer features than the Windows release and has been reported to be quite buggy. In June of 2006 Messenger for Mac 3.0b1 was released as a beta, with a final version due later. This beta has a more modern interface than 2.5.3 and includes avatars, display image viewing, BUZZ facility, and other features comparable to those on the Windows version. However, it is still lacking the more advanced features such as PC-to-PC calling and an address book.

Yahoo! Messenger was originally launched under the name Yahoo! Pager on March 9, 1998.



## 2.2.4 COMPARISON RESULTS

By comparing the key features of other chat application, we understand that,

| Chat feature application     | Orkut | Myspace | Yahoo  |
|------------------------------|-------|---------|--------|
| Chat rooms                   |       |         | A      |
| Communities                  | A     | A       |        |
| Scrap book                   | A     | A       |        |
| Ease of creating communities | GOOD  | GOOD    |        |
| Type of Chat                 |       |         | ONLINE |

**Table 2.1 chat application comparison table**

Two important features of any chat applications are

- community or group formation
- online chat

### Result:

The existing applications are not good at handling both the features of community creation and online chat together.

### Solution:

This project – community initiating system for the WWW will make better use of both these features and thus overcome the disadvantages of the existing systems.

## 2.3 PROPOSED SYSTEM

Conceptually, this idea of community initiating system origins from the Internet Forums and the Instant Messaging applications. A forum is a place in a website, where the visitors of the website can share their views and can interact with each other. But this provides an off line mode of communication. Here, user1 posts some message in the forum and the other users could see and respond to it. The topics generally are scrutinized by the owner of the site, to avoid unwanted content.

commune makes direct user-to-user interaction possible. A thin layer of web is formed upon users of the site, connecting the users of the same website, which makes the interaction with the fellow users possible.

The following diagram depicts the scenario of the proposed system,

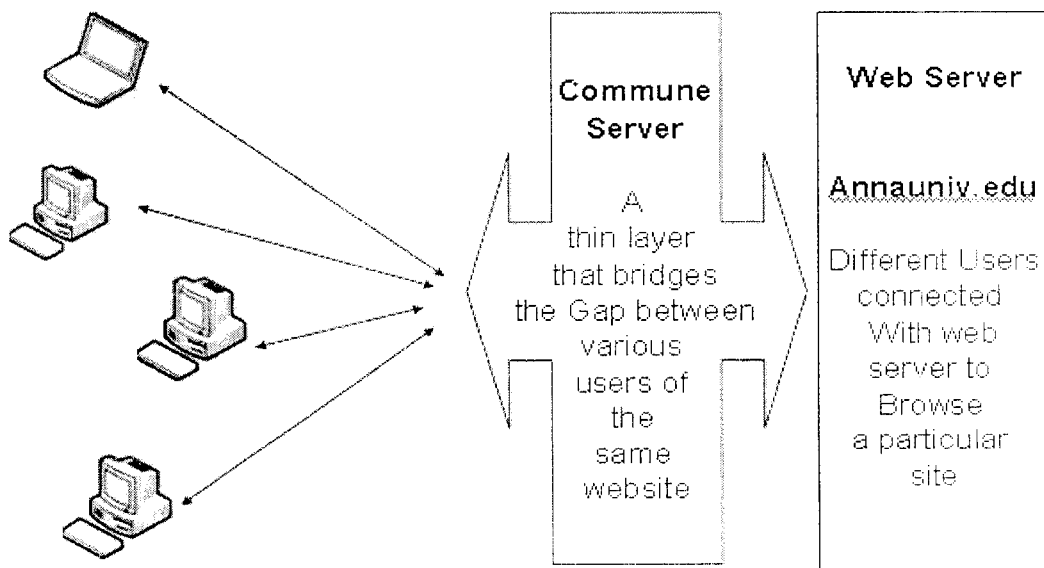


figure 2.5 proposed system block diagram

With this community initiating system, a thin client application is installed into the client's machine. The application comprises of entries in the registry of the operating system, which enables an icon in the browser's toolbar and an Explorer Bar.

A few handler files are installed in the client machine, which enables the client's browser to connect to the server for the first time. The browser window of the client contains a frame which is connected to the server. The client's activities are sent to the server and which in turn is broadcasted to the other users of the same website.

When a client is browsing a site and if he prefers to check out the other users of the site, he clicks on the commune icon in the toolbar. This makes an Explorer bar open up to the left of the Internet Explorer. The user logs in using his username and password. Once logged, the client information is sent tot the commune server, especially the URL he is browsing. Once this is sent to the server, the commune server software finds out the other users of the site from database. The list of users of the site is sent to the client and is listed out in the Explorer bar. The Applet in the client provides a simple chat option, through which the user can commute with the other users of the site.



P-1889

## CHAPTER 3

### SYSTEM DEVELOPMENT ENVIRONMENT

#### 3.1 HARDWARE REQUIREMENTS

The hardware support required for deploying the application:-

(Minimum system configuration required)

- PC with 300 megahertz (MHz) or higher processor clock speed recommended; 233-MHz minimum required.
- Intel Pentium/Celeron family, AMD K6/Athlon/Duron family, or compatible processor recommended.
- 128 megabytes (MB) of RAM or higher recommended (64 MB minimum supported; may limit performance and some features)
- 1.5 gigabyte (GB) of available hard disk space.\*
- Super VGA (800 × 600) or higher resolution video adapter and monitor
- CD-ROM or DVD drive
- Keyboard and Microsoft Mouse or compatible pointing device

## 3.2 SOFTWARE TECHNOLOGIES USED

These are the list of software that was used in the development of the project. Development depends both on Microsoft's and sun's technologies.

- MS Windows XP  
The client's operating system is MS Windows XP.
- Java JDK1.3  
Java JDK is used for build the server software.
- Tomcat web server  
The developed server is an extension of the web server.
- Internet Explorer 5.0 and above  
The handler files are installed as extension in the Browser.  
The Browser should be Java enabled.
- Java Servlets  
Java Servlets are used for build the server software.
- JavaScript, HTML and CSS  
JavaScript and HTML are used in the handler files as well as in the content that is pushed to the client from the commune server.
- Microsoft Visual Studio  
The utility tools like GUID Generator Component is used for generate Unique Id to make registry entries.
- Microsoft access  
The DMBS used for storing data that is necessary for the system

### 3.2.1 JAVA LANGUAGE

Java is an object-oriented applications programming language developed by Sun Microsystems in the early 1990s. Java applications are typically compiled to bytecode, although compilation to native machine code is also possible. At runtime, bytecode is usually either interpreted or compiled to native code for execution, although direct hardware execution of bytecode by a Java processor is also possible.

The language itself derives much of its syntax from C and C++ but has a simpler object model and fewer low-level facilities. JavaScript, a scripting language, shares a similar name and has similar syntax, but is not directly related to Java.

Sun Microsystems provides a GNU General Public License implementation of a Java compiler and Java virtual machine, in compliance with the specifications of the Java Community Process, although the class library that is required to run Java programs is not free software.

The Java project has seen many release versions. Since 1997 they are:

JDK 1.1.4 (Sparkler) September 12, 1997

JDK 1.1.5 (Pumpkin) December 3, 1997

JDK 1.1.6 (Abigail) April 24, 1998

JDK 1.1.7 (Brutus) September 28, 1998

JDK 1.1.8 (Chelsea) April 8, 1999

J2SE 1.2 (Playground) December 4, 1998

J2SE 1.2.1 (none) March 30, 1999

J2SE 1.2.2 (Cricket) July 8, 1999

J2SE 1.3 (Kestrel) May 8, 2000

J2SE 1.3.1 (Ladybird) May 17, 2001

J2SE 1.4.0 (Merlin) February 13, 2002

J2SE 1.4.1 (Hopper) September 16, 2002

J2SE 1.4.2 (Mantis) June 26, 2003

J2SE 5.0 (1.5.0) (Tiger) September 29, 2004

Java SE 6 (1.6.0) (Mustang) December 11, 2006 [4]

Java SE 7 (1.7.0) (Dolphin) anticipated for 2008

### **3.2.2 JAVA SERVLET**

The Java Servlet API allows a software developer to add dynamic content to a Web server using the Java platform. The generated content is commonly HTML, but may be other data such as XML. Servlets are the Java counterpart to non-Java dynamic Web content technologies such as PHP, CGI and ASP.NET. Servlets can maintain state across many server transactions by using HTTP cookies, session variables or URL rewriting.

The Servlet API, contained in the Java package hierarchy `javax.servlet`, defines the expected interactions of a Web container and a servlet. A Web container is essentially the component of a Web server that interacts with the servlets. The Web container is responsible for managing the lifecycle of servlets, mapping a URL to a particular servlet and ensuring that the URL requester has the correct access rights.

A Servlet is an object that receives a request (`ServletRequest`) and generates a response (`ServletResponse`) based on the request. The API package `javax.servlet.http` defines HTTP subclasses of the generic servlet (`HttpServlet`) request (`HttpServletRequest`) and response (`HttpServletResponse`) as well as a session (`HttpSession`) that tracks multiple requests and responses between the Web server and a client. Servlets may be packaged in a WAR file as a Web application.

However, servlets can be generated automatically by JavaServer Pages (JSP), or alternately by template engines such as WebMacro.

### 3.2.3 TOMCAT WEBSERVER

Apache Tomcat is a web container developed at the Apache Software Foundation (ASF). Tomcat implements the servlet and the Java Server Pages (JSP) specifications from Sun Microsystems, providing an environment for Java code to run in cooperation with a web server. It adds tools for configuration and management but can also be configured by editing configuration files that are normally XML-formatted. Tomcat includes its own internal HTTP server.

Tomcat is a web server that supports servlets and JSPs. The accompanying Tomcat Jasper compiler compiles JSPs into servlets.

The Tomcat servlet engine is often used in combination with an Apache HTTP Server or other web servers. Tomcat can also function as an independent web server. Earlier in its development, the perception existed that standalone Tomcat was only suitable for development environments and other environments with minimal requirements for speed and transaction handling. However, that perception no longer exists; Tomcat is increasingly used as a standalone web server in high-traffic, high-availability environments.

Tomcat is cross-platform, running on any operating system that has a Java Runtime Environment.

#### Tomcat 5.x Features

- implements the Servlet 2.4 and JSP 2.0 specifications
- reduced garbage collection, improved performance and scalability
- native Windows and Unix wrappers for platform integration
- faster JSP parsing



### 3.2.3 WEB BROWSER - MICROSOFT INTERNET EXPLORER

Windows Internet Explorer (also called Microsoft Internet Explorer) is a series of proprietary graphical web browsers developed by Microsoft and included as part of the Microsoft Windows line of operating systems starting in 1995.

After the first release for Windows 95, additional versions of Internet Explorer were developed for other operating systems: Internet Explorer for Mac and Internet Explorer for UNIX (the latter for use through the X Window System on Solaris and HP-UX). Only the Windows version remains in active development; the Mac OS X version is no longer supported.

It has been the most widely used web browser since 1999, peaking at nearly 90% market share with IE6 in the early 2000s—corresponding to over 900 million users worldwide by 2006.

Though released in 1995 as part of the initial OEM release of Windows 95, Internet Explorer was not included in the first retail, or shrink-wrap, release of Windows 95. The most recent release is version 7.0, which is available as a free update for Windows XP with Service Pack 2, and Windows Server 2003 with Service Pack 1, and is included with Windows Vista. Versions of Internet Explorer prior to 6.0 SP2 are also available as a separate download for versions of Windows prior to Windows XP. An embedded OEM version called Internet Explorer for Windows CE (IE CE) is also available for WinCE based platforms and is currently based on IE6. Another Windows CE/ Windows Mobile browser known as Pocket Internet Explorer is from a different code base and should not be confused with desktop versions of the browser.

## CHAPTER 4

### SYSTEM DESIGN & DEVELOPMENT

#### 4.1 USE CASE DIAGRAM

This use case diagram depicts the flow of user activities that are involved with the system.

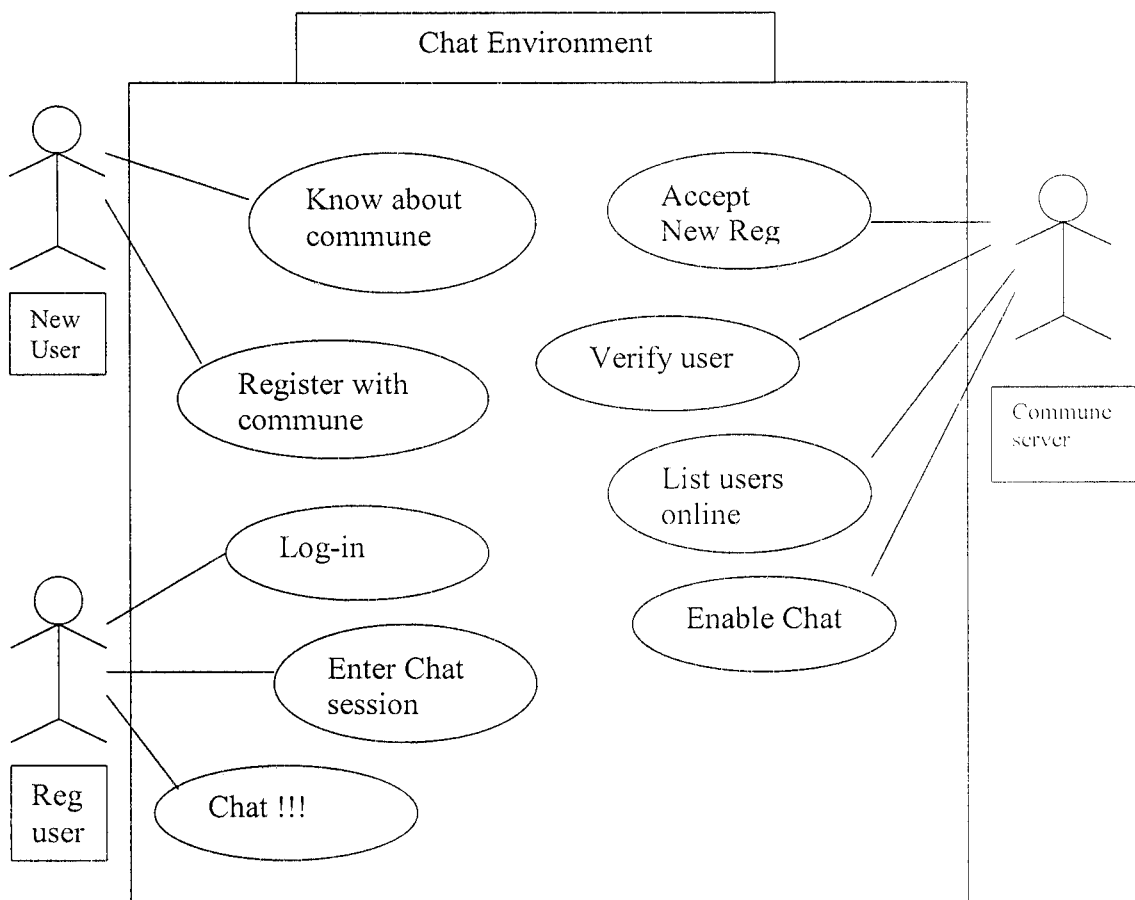


Figure 4.1 use case diagram

## **4.2 SYSTEM MODULES**

A software system is always divided into several subsystems which make it easier to develop and perform tests on the whole system. The subsystems are known as the modules and the process of dividing an entire system into subsystems is known as Decomposition.

The complete system is grouped into smaller modules to facilitate easy development. These smaller systems will be a part of the original system yet they will be independent in the sense that they will incorporate within them the major functionalities of the proposed system.

In this community initiating system development the modules are grouped under two major categories.

### **4.2.1 PROJECT MODULES**

- Segmenting the project into modules based on the developers views.

They are

- Registry Entry
- Server Module
- Client Module
- Chat Application

### **4.2.2 USER MODULES**

- segmenting the project into modules based on the users perspective.

They are

- User login
- New user registration
- Chat

## 4.3 MODULES DESCRIPTION

### 4.3.1 Registry Entry Module

This module is functionality is to add an entry in the registry of the OS to append the chat icon in the Internet explorer.

Internet Explorer provides a way add custom control components in the browser toolbar and the Explorer bars. This feature is available in Internet Explorer 5.0 or above. Toolbar buttons requires two sets of icon: one set with the active (color) icons and one set with the default (grayscale) icons. These icons can be stored in two .ico files or inside a resource (like a .dll or .exe file).

Before making the register entries, a Guaranteed Unique ID is to be generated using the GUID Generator Component for each the Explorer Bar and the Toolbar Icon.

#### 4.3.1.1 Explorer Bar

The Entry for the Explorer bar has to be made in the HKEY\_CLASSES\_ROOT\CLSID. The Key created has to be created with unique id. It should be like the following.

```
HKEY_CLASSES_ROOT\CLSID\{D0914066-5197-11d6-A1A4-
CBA81F292152}
```

The following is the .reg file of the entries made for the Explorer Bar, this can imported directly into the registry.

To Set the Width of the Explorer bar, the following entry is made.

```
HKEY_LOCAL_MACHINE\Software\Microsoft\Internet Explorer\Explorer
Bar\{D0914066-5197-11d6-A1A4-CBA81F292152}
```

The following is the .reg file of the entries made for the Explorer Bar UI properties, this can imported directly into the registry.

```
REGEDIT4
```

```
HKEY_LOCAL_MACHINE\Software\Microsoft\Internet Explorer\Explorer
Bar\{D0914066-5197-11d6-A1A4-CBA81F292152}
"BarSize"=hex: 97,00,00,00,00,00,00,00
```

The value can be changed, so that the width of the Explorer bar can be either increased or decreased.

#### **4.3.1.2 Toolbar Icon**

The entry for the toolbar should be made below the HKEY\_LOCAL\_MACHINE\Software\Microsoft\Internet Explorer\Extension and the key should be a unique value generated using GUID Component.

```
HKEY_LOCAL_MACHINE\Software\Microsoft\Internet
Explorer\Extension\{FFE7BC03-5132-11D6-A1A4-B5A2CE592465}
```

Once all these entries are made, open the Internet Explorer to and the Customize toolbar buttons, add the commune icon to the Toolbar. And the Commune icon is visible in the toolbar.

#### **4.3.1.3 Handler Files**

A .html file, cclient.htm is used to handle the user's request for the first time. This read-only file is stored in safe place and the location is mentioned in the registry. This file checks if the user is browsing file from the local machine or from a remote web server. Once the user is found to be a registered user, the information about the client is sent to the server to the server. This is made using a simple URL call, GET method in HTTP terms. This can done using POST method also, for increased security.

### 4.3.2 Client Module

The Commune client includes a few registry entries and handler files. The registry entries are made manually and can be automated if this project made a commercial product. The handler files are ".html" files, which contains HTML, JavaScript and cascading style sheet (CSS).

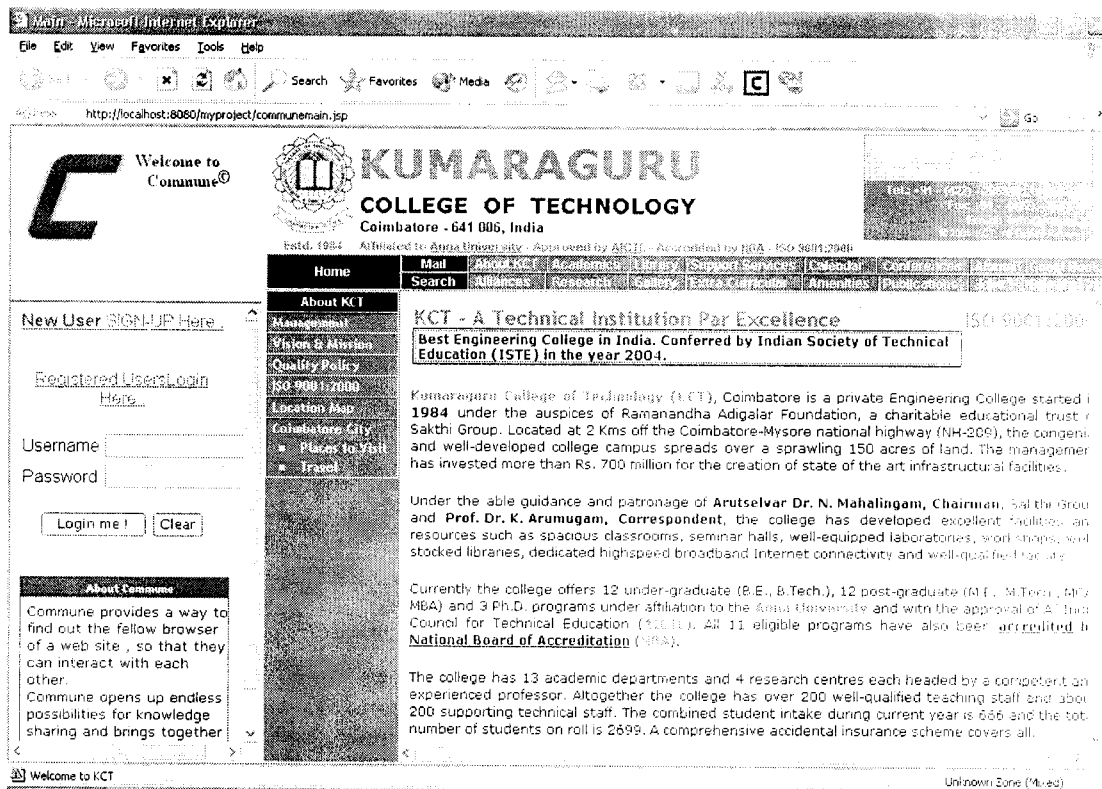


figure 4.2 welcome screen shot

Since Commune sits as an extension to the Internet Explorer, it does not involve complicated installation procedures and setup initiatives.

### **4.3.3 Server Module**

The Commune server, which is an extension of the web server contains of java class files. Servlets and applets.

Integrated Development Environment (IDE) or plain text editors can be used to create the above mentioned files and the utilities. Using an IDE will speed-up the task and save a lot of time, when developing an applets, i.e. when task involves more GUI work.

#### **4.3.3.1 Web Server Setup**

In a nutshell, a web server is waiting for client HTTP requests. When these requests arrive from browser clients the server does whatever is needed to serve the requests by providing the necessary content.

Tomcat is a servlet container and java server pages implementation. It may be used stand alone, or in conjunction with several popular web servers. Tomcat requires a java runtime environment conformant to JRE 1.1 or later, including any java2 platform system.

The Commune server is an extension of the web server. The requests sent from the Commune client is received & processed by a set of applications, which forms the Commune Server.

The server software, written in java, extending the servlet classes, resides in the classes folder of the tomcat web server.

Other resource files like Applet, HTML, GIF and JPEG files are stored in the public\_html directory.

### 4.3.4 Chat Module

The Client UI includes an Applet which lists out the peers of the website. The list is updated whenever there is change in the address bar or when a new user logs into the same url.

By clicking on the name or by selecting the user name and pressing the chat button a chat window pops up, using which the user can chat with the peer.

### 4.3.5 Login Module

As soon as the user clicks on the Commune button on the toolbar, the Commune bar opens with the cclient.htm. When the file loads, it validates the URL in the address bar and calls the Commune Server (CommuneServlet) with a request id of "showWelcomeScreen". The Servlet redirects the request to welcome.htm file. The User interface is show below.

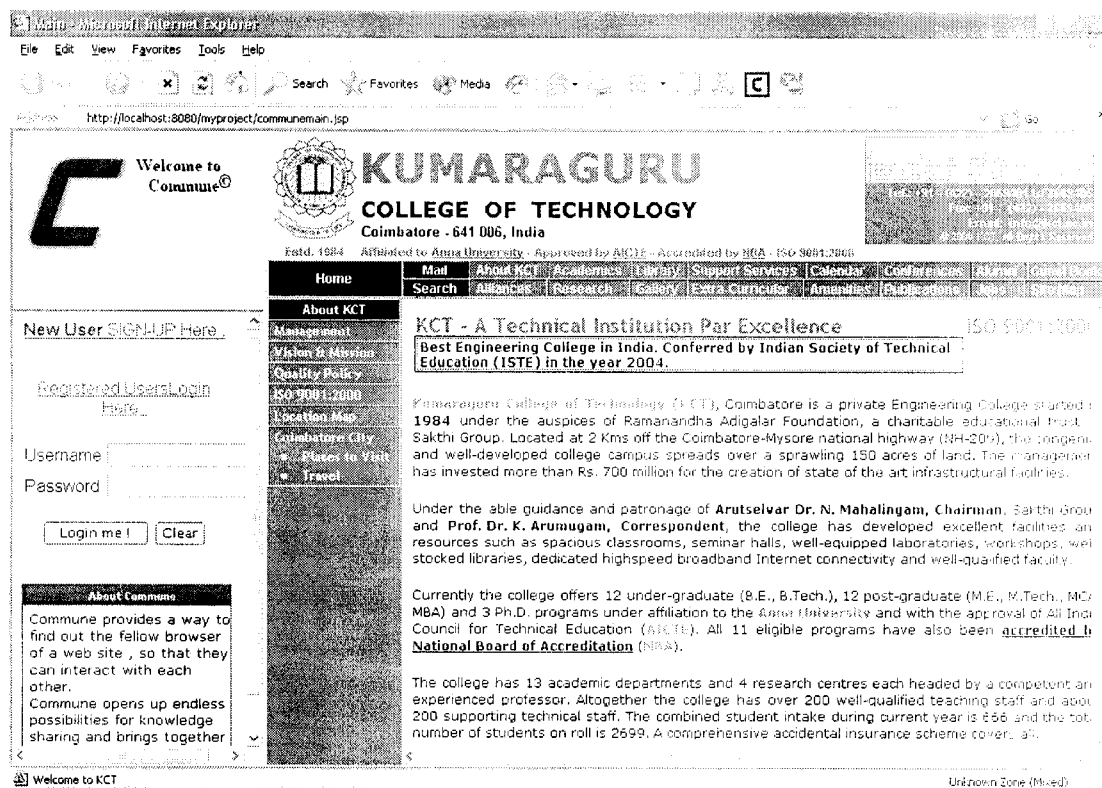
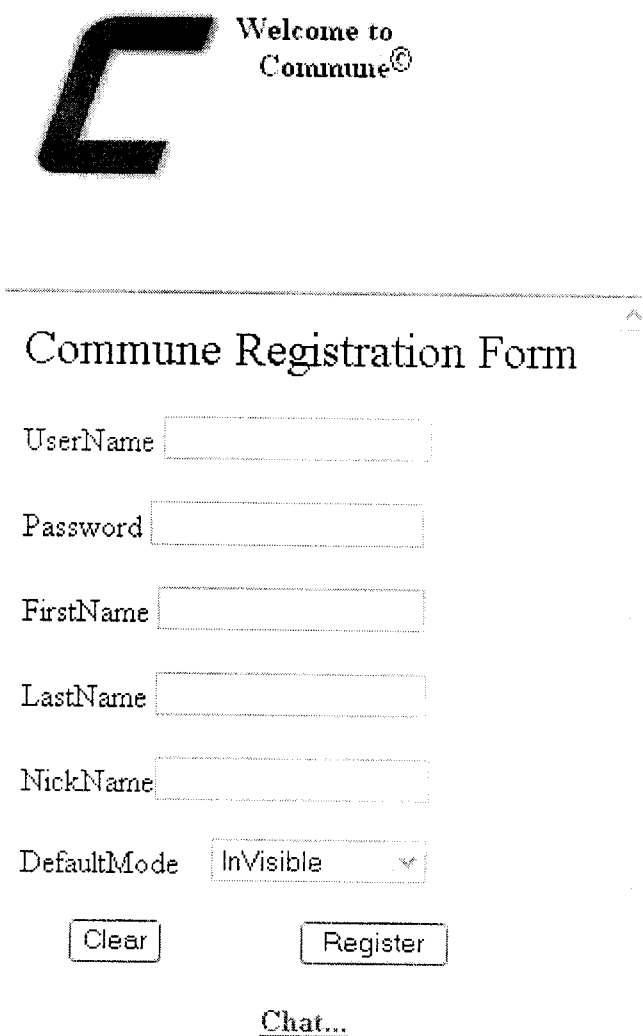


figure 4.3 login screen shot



### 4.3.6 New User Registration

The server handles the New User Registration via a HTML form provided in the Commune Bar. When the CommuneServlet receives a request for new user registration, it redirects the request to the register.htm file. The data entry form provided in this file is to be filled and submitted. The data submitted through this form is received by the CommuneServlet and stored in the user\_master table. The user interface for the New User registration is shown below.



The screenshot shows the Commune Registration Form. At the top left is a large stylized 'C' logo. To its right is the text 'Welcome to Commune®'. Below this is a horizontal line with a small upward-pointing arrow on the right. The main title of the form is 'Commune Registration Form'. The form contains several input fields: 'UserName', 'Password', 'FirstName', 'LastName', and 'NickName'. Below these is a 'DefaultMode' dropdown menu currently set to 'InVisible'. At the bottom of the form are two buttons: 'Clear' and 'Register'. Below the buttons is a link labeled 'Chat...'.

figure 4.4 New User registration screen shot

### 4.3.7 Registered User Logging In

After the registration is done, the user can login using the Login form provided in the welcome screen. The CommuneServlet validates the username and password and allows the user to enter the Commune zone. Once the user logs in, the user\_web\_details are updated about the user's arrival and the request is redirected to the CommuneServlet with request id as "showUserList". The client is now served with a list of users in List in an Applet. The user interface for the same is shown below.

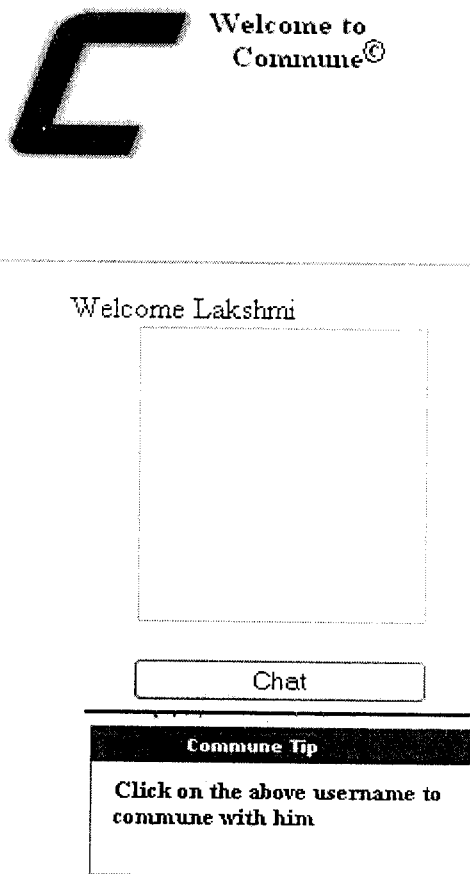


figure 4.5 Chat welcome screen

### 4.3.8 Updating the Server

Then onwards, whenever the users types in a URL (a new URL), it is notified to the Commune server (provided, the Commune bar is active). The CommuneServlet handles these requests. It receives the URL and the user\_id and updates the user\_website\_details table and notify to other users of the website

## 4.4 DATABASE DESIGN

Commune uses database as its persistence mechanism in the server for managing client's activities. The following is the schema of the tables used. The data structure contains three tables.

### Table 4.1 : USER\_MASTER

“user\_master” contains the details of the users of the Commune, when a new user registers him self through the registration form provided in the Commune Bar, the data is stored in the user\_master.

The user\_master maintains the complete list of users of Commune and their status, whether they are online or offline. The user can also have a nickname to hide his identity. The user\_mode allows the user to hide him self completely from other users, but he would be able to see other user.

| Table Field | Description                                    |
|-------------|--|
| User_Id     | Unique ID for the User                         |
| User_name   | Unique, User name of the User                  |
| Password    | Password for the account                       |
| First_name  | First name of the User                         |
| Last_name   | Last name of the user                          |
| Status      | Status of the user, online/offline             |
| Nickname    | Name to be displayed or as seen by other users |
| User_mode   | Visible/Invisible to the other commuters       |

**Table 4.2 : USER\_BUDDY\_DETAILS**

User\_buddy\_details contains the user's buddy ids. A Commune user can have a buddy list maintained for him. This table consists of the user id and their buddy ids. The addToBuddyList option provides a way to make an anonymous user his buddy. This is similar to that of an Instant Messenger. Additional columns can be added to add more features.

The user\_id and the buddy\_id both are maintained by their unique ids and not through their usernames, this makes the data structure efficient.

| Table Field | Description  |
|-------------|--|
| Line_no     | Key field for this table   |
| User_id     | Unique ID for the User, referred with user_master                              |
| Buddy_id    | Unique user id of the User's buddy, again referred with the user_master table. |

**Table 4.3 : USER\_WEBSITE\_DETAILS**

To hold the website details of the user.

| Table Field | Description                              |
|-------------|--|
| User_Id     | Unique ID for the User                   |
| User_name   | An ID to represent the URL unique        |
| Website_url | URL of the website, the user is watching |
| Start_time  | The time, user started browsing this URL |
| End_time    | The time, user stopped browsing this URL |
| Status      | Status of the record, active/Inactive    |

#### 4.4.1 Database Security

In order to enforce security to the user data that is stored in the database, the One-way Hash Algorithm has been used to encrypt the user password before storing into the database.

Most of the web sites today have some sort of a registration module where a user is asked to choose a username/password combination. This data gets stored in the database. They use the one-way Hash Encryption Algorithm to implement encryption.

#### One-way Hash Encryption

This scenario is a perfect candidate for "one-way hash encryption" also known as a message digest, digital signature, one-way encryption, digital fingerprint, or cryptographic hash. It is referred to as "one-way" because although you can calculate messages digest, given some data, you can't figure out what data produced a given message digest.

This is also a collision-free mechanism that guarantees that no two different values will produce the same digest. Another property of this digest is that it is a condensed representation of a message or a data file and as such it has a fixed length.

The data stored in the database ends up looking like this:

#### Example :

|                    |                                  |
|--------------------|----------------------------------|
| Actual password    | - "password"                     |
| Encrypted Password | - "5yfRRkrhJDbomacm2lsvEdg4GyY=" |

The java code that implements the algorithm is

```

import java.io.UnsupportedEncodingException;
import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;
import org.myorg.SystemUnavailableException;
import sun.misc.BASE64Encoder;
import sun.misc.CharacterEncoder;
public final class PasswordService {
    private static PasswordService instance;
    private PasswordService() {
    }
    public synchronized String encrypt(String plaintext) throws
SystemUnavailableException {
        MessageDigest md = null;
        try {
            md = MessageDigest.getInstance("SHA"); //step 2
        }
        catch(NoSuchAlgorithmException e) {
            throw new SystemUnavailableException(e.getMessage());
        }
        try {
            md.update(plaintext.getBytes("UTF-8")); //step 3
        }
        catch(UnsupportedEncodingException e) {
            throw new SystemUnavailableException(e.getMessage());
        }
        byte raw[] = md.digest(); //step 4
        String hash = (new BASE64Encoder()).encode(raw); //step 5
        return hash; //step 6
    }
    public static synchronized PasswordService getInstance() //step 1
    {
        if(instance == null) {
            return new PasswordService();
        }
        else {
            return instance;
        }
    }
}

```

## **CHAPTER 5**

### **TESTING & IMPLEMENTATION**

#### **5.1 TESTING OVERVIEW**

Testing is vital to the success of the system. System testing makes a logical assumption that if all the parts of the system are correct, the goal will be successfully achieved. Inadequate testing or non-testing leads to errors that may not appear until months later. This creates two problems:

- The time lag between the cause and the appearance of the problem
- The effect of system errors on files and records within the system

For any software that is newly developed, importance is given to the testing of a system. It is the last opportunity for the developer to detect and correct the errors before releasing the software to the organization. During the test we try to make sure that the software does exactly what is supposed to do. Testing is the process by which the programmers will generate a set of test data, which gives the maximum probability of finding all types of errors that can occur in the software.

##### **5.1.1 Purpose Of The Testing Stage**

1. Executing the programs to identify any errors that might have occurred while feeding the program to the system.
2. To affirm the quality of the software.
3. To find and eliminate any residual errors from the previous stage.
4. To validate the software as a solution to the actual problem.
5. To demonstrate the presence of all specified functionality in the software.
6. To ensure the operational reliability of the system.
7. To obtain the reports and submitting they examine by the operators, to test the reliability of outputs.

### 5.1.2 Testing Objectives:

- Testing is a process of executing a program with the insight of finding an error.
- A good test is one that has a high probability of finding an as yet undiscovered error.
- A successful test is one that uncovers an as yet undiscovered error.

### 5.1.3 Test Case Design:

The design of tests for software is as challenging as the initial design of the product itself. We must design tests that have the highest likelihood of finding the most errors with a minimum amount of time and effort.

Two important test approaches

- Black Box Testing
- White Box Testing

#### 5.1.3.1 Black Box Testing:

Black Box Testing methods focus on the functional requirements of the software. Black Box Testing enables the software engineer to derive sets of input conditions that will fully exercise all functional requirements of a programmer. It is a complementary approach that is likely to uncover a different class of errors than white box testing. Black Box Testing attempts to find errors in the following categories:

- Incorrect or missing functions
- Interface errors
- Errors in data structure or external database access
- Performance errors
- Initialization and termination errors



### **5.1.3.2 White Box Testing:**

White Box Testing is a test case design method that uses the control structure of the procedural design to derive test cases. Using white box testing methods, the software engineer can derive test cases that

- Guarantee that all independent paths within a module have been exercised at least once.
- Exercise all logical decision on their true or false sides.
- Execute all loops at their boundaries and within their operational bounds and
- Exercise internal data structures to ensure their validity.

## **5.1.4 Testing Methodologies**

### **5.1.4.1 Stress Testing**

In this test the application is tested in a manner that the application is demanding resources in an abnormal manner. The application is tested for maximum memory usage. The boundary conditions are also tested for arrays and other data structures used in the system.

### **5.1.4.2 Performance Testing**

Performance testing for the application is performed for finding the actual run-time performance. Time taken for search and locate the required data is tested. The data conversion accuracy and speed for the converter tool is also tested. Connectivity and data transfer performance for live update facility of the application is tested by using the test data sets provided by the system testing specialties of the company.

### **5.1.4.3 User Acceptance Testing**

User Acceptance Testing is the moment at which we prove that we have really done what we promised to do so. This testing is not the same as the testing conducted in the various stages of the life cycle. Since, here it involves the customer and intends to check that the tests dispose no errors at all. All the possible testing conditions and data that the software organization has never seen will fall within the specified operational conditions and behavior of the software. The proposed system is very efficient and perfect than the existing one. The modules available are very user friendly and also completely devoid of all the possible input errors. The user proves it to by using the sample data.

### **5.1.4.4 Unit Testing**

Unit testing focuses verification effort on the smallest unit of software design module. Using the detailed design description as a guide, important control paths are tested to uncover errors within the boundary of the module.

### **5.1.4.5 Integration Testing**

Integration Testing is a systematic technique for construction of the program structure while at the same time conducting test to uncover errors associated with interfacing (i.e., "putting the pieces together"). The objective is to take unit testing modules and build a program structure that has been dictated by design; there are two approaches to integration. The non-incremental integration uses the "Big Bang" approach –all the modules are combined in advance and tested as whole. In incremental integration the program is constructed and tested in small segments. There are some incremental integration strategies like Top-Down integration.

### **Top-Down Integration**

The integration module is performed in a series of four steps:

- The main control module is used as a test driver and is sustained for all modules directly subordinate to main control module.
- Depending on the integration approach selected (i.e. depth or breadth first), subordinate stubs are replaced one at a time with actual modules.
- Tests are conducted as each module is integrated.
- Regression testing (i.e. conduction all or some of the previous tests may be conducted to ensure that new errors have not been introduced.

### **Bottom-Up Integration**

The bottom-up integration strategy may be implemented with the following steps:

Low-Level modules are combined into clusters (or builds) that perform a specific software sub function.

- A driver (a control program for testing) is written to coordinate test case input and output.
- The cluster is tested
- Driver is removed and clusters are combined moving upward in the program structure

#### **5.1.4.6 Validation Testing**

Validation succeeds when the software functions in a manner that can be reasonably expected by the customers.

#### ***Validation Test Criteria***

Software validation is achieved through a series of black box tests that demonstrates conformity with requirements. A test plan outlines the class of test to be conducted and a test procedure defines specific test cases that will be used to demonstrate conformity with requirements.

## 5.2 TEST CASES

|   |   |
|---|---|
| <b>Test Case: 1</b>   |   |
| <b>Project No:</b>  | <b>Project Name:</b> Community initiating system for world wide web |
| <b>Name/Type:</b> Black box testing   | <b>Test Status:</b>   |
| <b>Prepared By:</b> Lakshminarayanan.   | <b>Date:</b>  |
| <b>Test Case Description:</b><br>This test case is designed to test the user login system   |   |
| <b>Purpose or Method:</b> <ul style="list-style-type: none"> <li>• This case validates and verifies username and password.</li> <li>• Using black box testing by specifying input values</li> </ul> |   |

| Step | Action   | Expected Result  | Pass/Fail |
|------|--|--|-----------|
| 1.   | Enter a username and password  | Accept in text boxes   |           |
| 2.   | Inclusion of symbols and invalid text (validation).                  | Process at client side itself  |           |
| 3.   | Click login after entering valid username and password(verification) | <ul style="list-style-type: none"> <li>➤ Encrypt the entered password.</li> <li>➤ Verify username and password with database</li> <li>➤ Correct username and password enter application else inform user and return to login screen</li> </ul> |           |
| 4    | User hitting cancel button   | Return to login screen after clearing text fields.   |           |

|   |   |
|---|---|
| <b>Test Case: 2</b>   |   |
| <b>Project No:</b>  | <b>Project Name:</b> Community initiating system for world wide web |
| <b>Name/Type:</b> Unit testing  | <b>Test Status:</b>   |
| <b>Prepared By:</b> Lakshminarayanan.   | <b>Date:</b>  |
| <b>Test Case Description:</b><br>To test whether the system is capable of listing all active users who are logged in and browsing same web pages. |   |
| <b>Purpose or Method:</b> <ul style="list-style-type: none"> <li>To check for correctness in listing the active users</li> </ul>                  |   |

| Step | Action                                 | Expected Result   | Pass/Fail |
|------|--|---|-----------|
| 1.   | User logs in (after successful case 2) | <ul style="list-style-type: none"> <li>➤ Enter details of user in database &lt;table name</li> <li>➤ Extract the domain name of current user</li> <li>➤ With current users, website domain name, search the database for other users</li> <li>➤ Store the name in a record set</li> <li>➤ Display the other user names in the chat window.</li> </ul> |           |

|   |   |
|---|---|
| <b>Test Case: 3</b>   |   |
| <b>Project No:</b>  | <b>Project Name:</b> Community initiating system for world wide web |
| <b>Name/Type:</b> Unit Testing (chat application)   | <b>Test Status:</b>   |
| <b>Prepared By:</b> Lakshminarayanan.   | <b>Date:</b>  |
| <b>Test Case Description:</b><br>This case is to check for correctness in chatting module.  |   |
| <b>Purpose or Method:</b> <ul style="list-style-type: none"> <li>• To facilitate chatting.</li> <li>• To check whether the chat messages are delivered across the users.</li> </ul> |   |

| Step | Action                           | Expected Result   | Pass/Fail |
|------|----------------------------------|---|-----------|
| 1.   | User clicks on an username       | <ul style="list-style-type: none"> <li>➤ Show a small chat applet</li> <li>➤ Create a chat session between these two users</li> </ul>           |           |
| 2.   | User enters text in chat applet. | <ul style="list-style-type: none"> <li>➤ Accept text</li> </ul>   |           |
| 3.   | User hits send button            | <ul style="list-style-type: none"> <li>➤ Transform text to other end user</li> <li>➤ Perform these steps for all user communication.</li> </ul> |           |

|  |   |
|--|---|
| <b>Test Case: 4</b>  |   |
| <b>Project No:</b>   | <b>Project Name:</b> Community initiating system for world wide web |
| <b>Name/Type:</b> System Testing   | <b>Test Status:</b>   |
| <b>Prepared By:</b> Lakshminarayanan.  | <b>Date:</b>  |
| <b>Test Case Description:</b><br>Testing the system as a whole and its interaction with browser.   |   |
| <b>Purpose or Method:</b> <ul style="list-style-type: none"> <li>• This case is to check whether the system loads into the browser.</li> <li>• Repeatedly calling the application from the browser and checking status during every call.</li> </ul> |   |

| Step | Action   | Expected Result   | Pass/Fail |
|------|--|---|-----------|
| 1.   | User Hits the 'commune' icon on the browser.                               | Commune login frame to be displayed at the left side of the browser |           |
| 2.   | User Hits the 'commune' icon when commune is already loaded in the browser | <.do nothing > (browser have this functionality)                    |           |
| 3.   | User hits close button   | Exit application  |           |

|  |   |
|--|---|
| <b>Test Case: 5</b>  |   |
| <b>Project No:</b>   | <b>Project Name:</b> Community initiating system for world wide web |
| <b>Name/Type:</b> Unit Testing (new user registration module)  | <b>Test Status:</b>   |
| <b>Prepared By:</b> Lakshminarayanan.  | <b>Date:</b>  |
| <b>Test Case Description:</b><br>This case is to check for correctness in new user registration module.  |   |
| <b>Purpose or Method:</b> <ul style="list-style-type: none"> <li>• To facilitate New user Registration.</li> <li>• New user registration , informing about existing ID , storing New user details in Table.</li> </ul> |   |

| Step | Action                         | Expected Result   | Pass/Fail |
|------|--------------------------------|---|-----------|
| 1.   | User clicks on Register button | Display new user registration form.   |           |
| 2.   | User clicks submit button.     | <ul style="list-style-type: none"> <li>➤ Accept text</li> <li>➤ Validate the username</li> <li>➤ Check whether all necessary details are entered.</li> <li>➤ Encrypt the password and store in database.</li> <li>➤ Display message to user.</li> </ul> |           |
| 3.   | Invalid Text entry             | ➤ Inform user.  |           |
| 4.   | User name already registered   | <ul style="list-style-type: none"> <li>➤ Inform user</li> <li>➤ Return to new user registration screen.</li> </ul>  |           |



### 5.3 IMPLEMENTATION

Implementation is the key to successful information system. It simply means converting a new system design into operation. Newly developed system attains its fulfillment only after the implementation of the newly system designed into operational one. This implementation involves various jobs like planning of implementation, transition and training.

In order to implement the system, planning is essential. Proper planning has been done to take care of the issues like the implication of the system environment, standby facilities, channel for communication, resource availability, methods of changeover.

The plan of implementation consists of steps like detection and correction of errors, making necessary changes in the system, checking the reports with that of existing system, training the involvement of user personnel, installation of hardware and software utilities.

The implementation plan was well executed and results were obtained. As per the observation some minor problem were identified and rectified. The system was found to be user friendly.

The system that was newly developed was found satisfactory in all aspects and considering the nature of business environment.

## CHAPTER 6

### CONCLUSION & FURTHER DEVELOPMENT

#### 6.1 CONCLUSION

This project “**Community Initiating System for the World Wide Web**” would be more popular if it's been adopted by the users of the internet. Implementation at a organizational level has provided positive results. As said earlier is has worn out the disadvantages of the other chat application in the internet and has brought out a new concept of chatting. This project can be further be developed with features like voice chat, video conferencing, file sharing etc

## 6.2 SUGGESTION FOR FURTHER DEVELOPMENT

This project can be further developed implementing with these key features,

- **VOICE CHATTING**

The means of communicating with voice that will create a significant increase in distant communications where two people from opposite ends of the world can talk almost free of cost.

- **VIDEO CONFERENCING**

Also known as a video teleconference is a set of interactive telecommunication technologies which allow two or more locations to interact via two-way video and audio transmissions simultaneously. It has also been called visual collaboration and is a type of groupware.

- **FILE SHARING**

Usually file sharing follows the peer-to-peer (P2P) model, where the files are stored by one user can be sent to another user.

Other user friendly features like,

- Saving chat history
- Offline message storing
- Virus scanning etc etc

## References

1. Budi Kurniawan (Author), Paul Deck (Author) 1999 How Tomcat Works.
2. Jerry Honeycutt 2005 ISBN 9780735622180 Microsoft® Windows® Registry Guide, Second Edition
3. Herbert Schildt (1999) Java 2: The Complete Reference, Fifth Edition
4. [www.sun.com/docs](http://www.sun.com/docs)