REAL-TIME COLLABORATIVE TEXT EDITOR

MINOR PROJECT REPORT

In partial fulfillment of the requirements
for the award of the degree of

BACHELOR OF TECHNOLOGY
IN
COMPUTER SCIENCE AND ENGINEERING
OF
COCHIN UNIVERSITY OF SCIENCE AND TECHNOLOGY
Under the guidance of
Mr. Titty Jacob

Submitted by :
CSU 051/11 Cijo George
CSU 051/ 16 George Mathew
CSU 051/30 R Srijith
CSU 051/50 Jibil Louis

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
GOVERNMENT MODEL ENGINEERING COLLEGE
THRIKAKKARA, KOCHI-21
2008

GOVERNMENT MODEL ENGINEERING COLLEGE

THRIKKAKARA, KOCHI-21

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Certificate

This is to certify that the project entitled

……

Submitted by

……

……

is a bonafide account of the work done by him/her under our supervision.

Guide Coordinator Head of the Department

Acknowledgement

We are extremely thankful to our principal Dr. Prof. Suresh Kumar P for providing us with a conductive environment and the requisite lab facilities.

We are indebted to our Project coordinator Ms. Preetha Theresa Joy for her constant help and support in light of this project.

We extend our sincere and heartfelt thanks to our guide, Mr. Titty Jacob for helping us throughout the course of this project by providing us with valuable advice.

We thank God almighty for all the blessings showered on us during this endeavor.

Last but not the least we would like to thank our friends for the support and encouragement they have given us during the course of our work.

Abstract

The project aims at designing a text editor which allows more than two people to edit the same text from different locations in a network. The project mainly focuses at providing a non-web based peer-to-peer editing application for the collaborative text editing needs of people connected through a Local Area Network. The application will be operational in the 'GNU/Linux' platform and is implemented using the various modules of 'PERL' programming language. The 'udp' protocol is used to send data over the network.

The basic application serves as a usual text editor with various text editing features. In addition to these, the editing application will also feature an option to 'collaborate' with other users in the network who are using the same application. This option allows the main user to list the IPs of users with whom he wants to collaborate. Upon filling inputting the required details for collaborating, each editor is connected with one another in a peer-to-peer fashion, after asking for confirmation from the users whose IPs have been entered. The text typed in by any user after this will be updated in 'real time in all other editors. Each user will be working on his own copy of the file which he can save in his system at any time.

Selecting the option to collaborate will also result in the addition of a chat facility in the editor interface, which allows the users to communicate with each other during collaborative text editing. The IPs of users who currently share the text will also be listed in the application interface. Even if the main user gets disconnected due to some reason, the others can continue editing, because of the peer-to-peer nature of the connection between the applications.

Table of Contents

1.0 System Study

1.1 Introduction

A collaborative editor is a software application that allows several people to edit a text file from different computers in a network. Web-based collaborative text editing applications does not allow real-time editing, i.e. text is not updated for all users in real-time. Also, such applications work in server-client mode. A non web-based real-time peer-to-peer collaborative editor thus gains its significance.

1.2 Purpose

The project focuses at providing a non-web based peer-to-peer editing application for the collaborative text editing needs of people connected through a Local Area Network.

The basic application serves as a usual text editor with various text editing features. Along with serving the purpose of a simple text editor, it can be converted into a collaborative editor by collaborating with a number of other users in the network with known IP address. Once the editors are connected in a peer-to-peer mode, all collaborating editors will be cleared and the text currently present in the editor of the user who initiated the collaboration will be sent to all editors and displayed. After this, the text typed in by any user in his editor will be updated in real time in all other editors. Each user works on his own copy of the file, which he can save at any time.

A chat facility is provided in the user interface when the editor works in collaborative editing mode, allows users to communicate with each other during collaborative text editing.

The application is coded in various modules of 'PERL' programming language like Tk, IO::Socket etc

1.3 Alternative Design Approaches

Web-based collaborative text editors can be designed, which works in server-client mode through a web-browser, where every user works on the same text file, which is stored in a server system. Non Web-based collaborative text editing applications can also be developed using languages like Java, C, C++ etc.

1.4 Platform

GNU/Linux

1.5 Software Requirements

Perl

1.6 Hardware Requirements

Personal Computers connected in a network, hardware required for network connections to setup a Local Area Network.

1.7 Advantages and Features

- Basic text editing facility
    - When the user doesn't want to collaborate with other users for shared text editing, the application can be used just like any other simple text editor.
    - Features of common text editors, like Open File, Save, Save As, Clear File, Cut, Copy, Paste, Undo, Redo, Select All, Go to Line Number etc.
    - Extended GUI features like Font Style, Font Size etc.
    - Line number and column number of the cursor is shown in the bottom bar of the editor.

- Real-time collaborative text editing facility
    - A user can invite other users to collaborate with him for shared text editing by entering the list of IPs with whom he wants to collaborate.
    - All such applications in the network will be connected in a peer-to-peer fashion for collaborative editing.
    - Changes in text will appear on all editors in the network in real-time.
    - Each user can select a unique colour while collaborating. Any text input from a user will appear in other editors in colour that the user selects.
    - After an user saves his file, all text in the file will appear black and new text arriving from other users will be coloured.
    - IPs of users who share the document is listed in the GUI.
    - Chatting Facility is provided to users when the editor is in collaborative editing mode.

- Chatting facility
    - Users who share the text document can make use of the chat facility to communicate with each other.
    - Text typed into the chat facility will appear in the chat window in the colour selected by that user.
    - The text entered will be labeled with the username entered by the user while collaborating.

1.8 Limitations

- Only users with a static IP address can be invited to collaborate.
- Features like Font Size and Font Style should be set before collaborating. After collaborating, users are not allowed to change these settings.
- Addition or Deletion of users in the IP list of collaborating users is not permitted.
- Username and colour, once set cannot be changed.
- Some special keys in the keyboard will not provide its functionality when the editor is in collaborative mode.

2.0 Software Requirement Specification

2.1 Introduction

This Software Requirements Specification (SRS) describes the scope & various functions of the proposed text editing software system & the various requirements of the system. It also specifies the advantages & limitations of the system.

2.2 Purpose

2.2.1 Scope

A collaborative editor is a software application that allows several people to edit a text file from different computers in a network. Web-based collaborative text editing applications does not allow real-time editing, i.e. text is not updated for all users in real-time. Also, such applications work in server-client mode. A non web-based real-time peer-to-peer collaborative editor thus gains its significance.

The proposed 'Real-Time Collaborative Text Editor' is a software having the following features.

- Basic text editing facility with extended GUI features like Font Style, Font Size etc along with the normal features of a text editor.
- Real-time collaborative editing facility, with text colouring to identify which user typed in a particular text.
- Chatting facility for users who collaborate for shared text editing.

The basic text editing application operates in systems using GNU/Linux Operating System and the collaborative editing mode works in multiple systems using GNU/Linux, over a network.

2.2.2 Definition, Acronyms and Abbreviations

GUI: Graphical User Interface

IP : Internet Protocol

2.2.3 References

1. http://en.wikipedia.org

2.2.4 Overview

The SRS gives an in-depth view of the functionality of the proposed software through different sections describing product perspective, product functions, user characteristics & general constraints. It also provides information about the various requirements of the software including functional requirements, external interface requirements & performance requirements.

.

2.3 General Description

2.3.1 Product Perspective

The 'Real-Time Collaborative Text Editor', will allow different users to edit the same text file at the same time from different locations over a network. The project focuses at providing a non-web based peer-to-peer editing application for the collaborative text editing needs of people connected through a Local Area Network.

The basic application serves as a usual text editor with various text editing features. In addition to these, the editing application will also feature an option to 'collaborate' with other users in the network who are using the same application. This option allows the main user to list the IPs of users with whom he wants to collaborate. Upon filling inputting the required details for collaborating, each editor is connected with one another in a peer-to-peer fashion, after asking for confirmation from the users whose IPs have been entered. All collaborating editors will be cleared and the text currently present in the editor of the user who initiated the collaboration will be sent to all editors and displayed. The text typed in by any user after this will be updated in 'real time' in all other editors. Each user will be working on his own copy of the file which he can save in his system at any time.

Selecting the option to collaborate will also result in the addition of a chat facility in the editor interface, which allows the users to communicate with each other during collaborative text editing. The IPs of users who currently share the text will also be listed in the application interface. Even if the main user gets disconnected due to some reason, the others can continue editing, because of the peer-to-peer nature of the connection between the applications.

Web-based collaborative text editing applications does not allow real-time editing, i.e. text is not updated for all users in real-time. Also, such applications work in server-client mode. A non web-based real-time peer-to-peer collaborative editor thus gains its significance.

2.3.2 Product Functions

- Basic text editing facility
    - When the user doesn't want to collaborate with other users for shared text editing, the application can be used just like any other simple text editor.
    - Features of common text editors, like Open File, Save, Save As, Clear File, Cut, Copy, Paste, Undo, Redo, Select All, Go to Line Number etc.
    - Extended GUI features like Font Style, Font Size etc.
    - Line number and column number of the cursor is shown in the bottom bar of the editor.

- Real-time collaborative text editing facility
    - A user can invite other users to collaborate with him for shared text editing by entering the list of IPs with whom he wants to collaborate.
    - All such applications in the network will be connected in a peer-to-peer fashion for collaborative editing.
    - Changes in text will appear on all editors in the network in real-time.
    - Each user can select a unique colour while collaborating. Any text input from a user will appear in other editors in colour that the user selects.
    - After an user saves his file, all text in the file will appear black and new text arriving from other users will be coloured.
    - IPs of users who share the document is listed in the GUI.
    - Chatting Facility is provided to users when the editor is in collaborative editing mode.

- Chatting facility
    - Users who share the text document can make use of the chat facility to communicate with each other.
    - Text typed into the chat facility will appear in the chat window in the colour selected by that user.
    - The text entered will be labeled with the username entered by the user while collaborating.

2.3.3 User Characteristics

The application mainly targets users with a basic knowledge in using computers & have prior experience in using any of the GUI or command line text editors. A basic knowledge of Linux based systems and computer networking is also essential.

2.3.4 General Constraints

- The port specified in the application for sending a receiving information packets between the different nodes in the network, should be open in all the computers which are to be collaborated.
- All computers in the network that needs to share the text should be running on GNU/Linux Operating System with the application installed.
- The IP addresses of the users with whom the file is to be shared should be known to the user who initiates the collaboration.

2.4 Specific Requirements

2.4.1 External Interface & Functional Requirements

2.4.1.1 User Interface

- An interface similar to that of a text editor, where the user can edit text files.
- An option for enabling collaborative text editing, which involves entering IP addresses of the computers with which the file should be shared manually.
- An interface for selecting the colour for a user.
- An interface to enter the username or display name of the user.
- A box in the GUI in which the IPs of users who are collaborating are shown.
- A chat window for users to communicate with each other through instant messaging.
- Menu options for other usual functions in a text editor such as Open File, Save, Save As, Clear File, Cut, Copy, Paste, Undo, Redo, Select All, Go to Line Number etc and also extended GUI features of the editor like Font Style and Font Size.
- An interface that shows the line number and column number of the current cursor position

2.4.2 Performance Requirements

The software is expected to work with any number of users in the network sharing a text file. Communication delay caused in the network will be negligible in most cases. No specific limits are set for the size of text file that can be shared using the software.

2.5 Design Considerations

2.5.1 Assumptions and Dependencies

- All users who need to view or edit a shared text file should be using a computer in the given network, which runs on GNU/Linux Operating System & have the application installed in the system.
- The network should be flawless without any connectivity problems between the different computers
- The specified port should be open in all computers which are to be collaborated.
- Information packets sent & received by a computer in the network should not be blocked by a 'Firewall' or any such application.

3.0 High Level Design

3.1Introduction

3.1.1 Application Overview

The project being developed is a text editor which allows more than two people to edit the same text from different locations in a network. The project mainly focuses at providing a non-web based peer-to-peer editing application for the collaborative text editing needs of people connected through a Local Area Network. The application will be operational in the 'GNU/Linux' platform and is implemented using the various modules of 'PERL' programming language. The 'UDP' protocol is used to send data over the network.

The basic application serves as a usual text editor with various text editing features. In addition to these, the editing application will also feature an option to 'collaborate' with other users in the network who are using the same application. This option allows the main user to list the IPs of users with whom he wants to collaborate. Upon filling inputting the required details for collaborating, each editor is connected with one another in a peer-to-peer fashion, after asking for confirmation from the users whose IPs have been entered. The text typed in by any user after this will be updated in 'real time' in all other editors. Each user will be working on his own copy of the file which he can save in his system at any time.

Selecting the option to collaborate will also result in the addition of a chat facility in the editor interface, which allows the users to communicate with each other during collaborative text editing. The IPs of users who currently share the text will also be listed in the application interface. Even if the main user gets disconnected due to some reason, the others can continue editing, because of the peer-to-peer nature of the connection between the applications.

3.1.2 Document Overview

The high level design will serve as a comprehensive guide for the development process by providing in-depth and detailed designs of the intended goals and architecture of the project.

3.1.3 Design Considerations and Constraints

- All users who need to view or edit a shared text file should be using a computer in the given network, which runs on GNU/Linux Operating System & have the application installed in the system.

- The network should be flawless without any connectivity problems between the different computers
- .The specified port should be open in all computers which are to be collaborated.
- Information packets sent & received by a computer in the network should not be blocked by a 'Firewall' or any such application.
- The port specified in the application for sending a receiving information packets between the different nodes in the network, should be open in all the computers which are to be collaborated.
- All computers in the network that needs to share the text should be running on GNU/Linux Operating System with the application installed.
- The IP addresses of the users with whom the file is to be shared should be known to the user who initiates the collaboration.

3.1.4Definitions, Acronyms and abbreviations

GUI : Graphical User Interface
IP : Internet Protocol

3.2Design Specifications

3.2.1High Level Overview

The basic application serves as a usual text editor with various text editing features. Along with serving the purpose of a simple text editor, it can be converted into a collaborative editor by collaborating with a number of other users in the network with known IP address. Once the editors are connected in a peer-to-peer mode, all collaborating editors will be cleared and the text currently present in the editor of the user who initiated the collaboration will be sent to all editors and displayed. After this, the text typed in by any user in his editor will be updated in real time in all other editors. Each user works on his own copy of the file, which he can save at any time.

A chat facility is provided in the user interface when the editor works in collaborative editing mode, allows users to communicate with each other during collaborative text editing.

3.2.2Low Level Overview

The application when first invoked, acts as a normal text editor. The user can edit text and make use of the basic features of the editor like Save, Save As, Cut, Copy, Paste, Select All etc. Extended features like Font Style and Font Size can also be used.

On invocation of the ‘collaborate’ option from the editor menu, the application asks for the users IP and the list of IPs to which request is to be sent for collaboration. A request is sent to all IPs entered by the user.

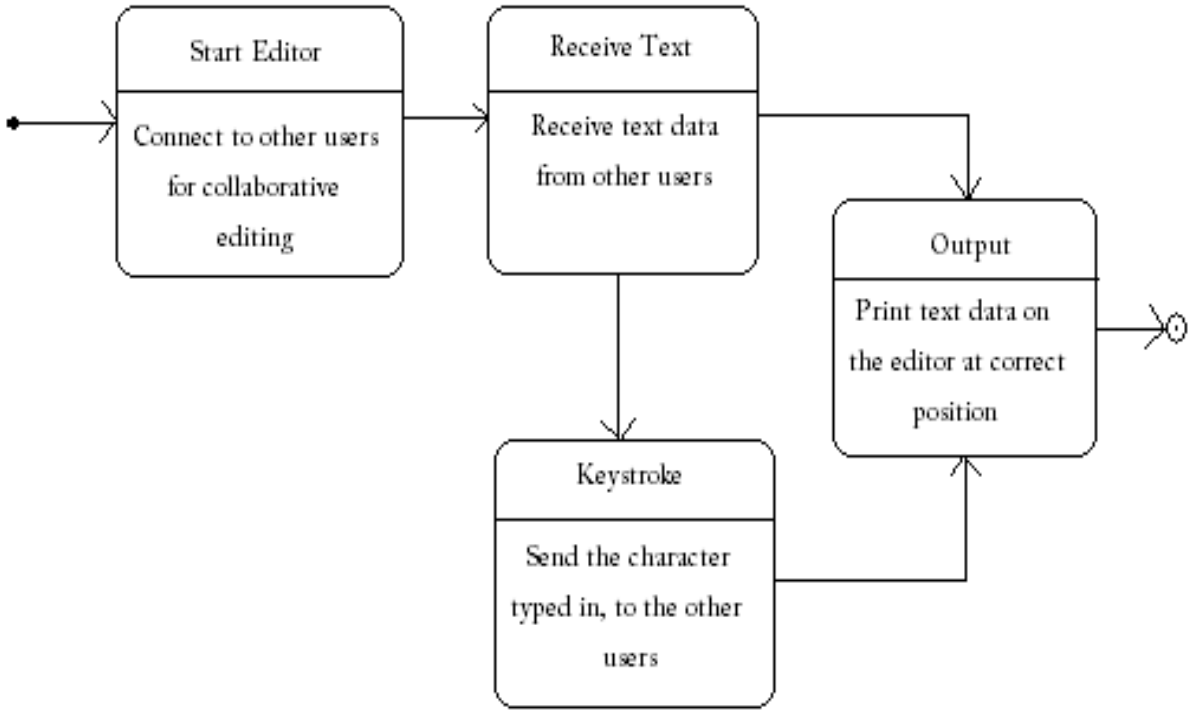
At each IP to which the request is sent, the application asks for confirmation from the user to allow collaboration. On giving the consent to collaborate, the application asks the user to pick a colour from a pool of colours. This will be the colour in which text typed by the user will be displayed in the editors of other users. It will also ask the user to enter a username to be displayed along wit chat messages.

After receiving response from each editor, the application which invoked the ‘collaborate’ option asks its user to select a color and to enter a username. It then connects to all the IPs that gives permission for collaborating. It also makes a list of IPs to connect to by each of the other editors and send the list to the respective editors. Upon receiving this list, these editors will also connect to each other based on the IP list. Thus s peer-to-peer connection is set up. All IPs to which connection is made is displayed in a box in the GUI. After this every keystroke in by each of the users is send to all other users and is updated in the editors. The text will be displayed other editors in the color of the user who typed in the text.

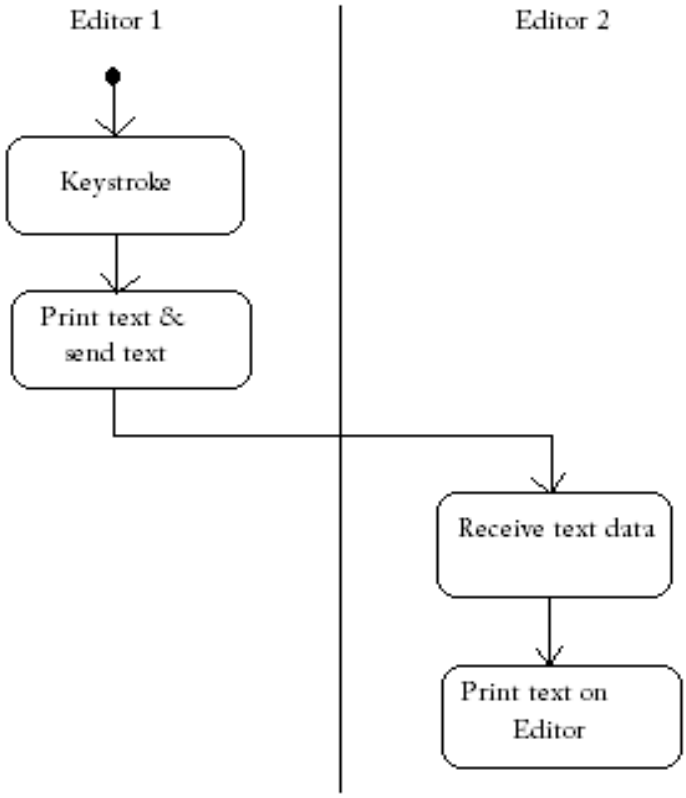
The message typed into the chat facility is also sent to all users in a similar way and displayed in the chat facilities of each editor, in the color of the user who typed in the message and labeled wit the username of the user.

Upon saving the file by any user, all text in his editor turns to black. New text arriving from other users will again be coloured.

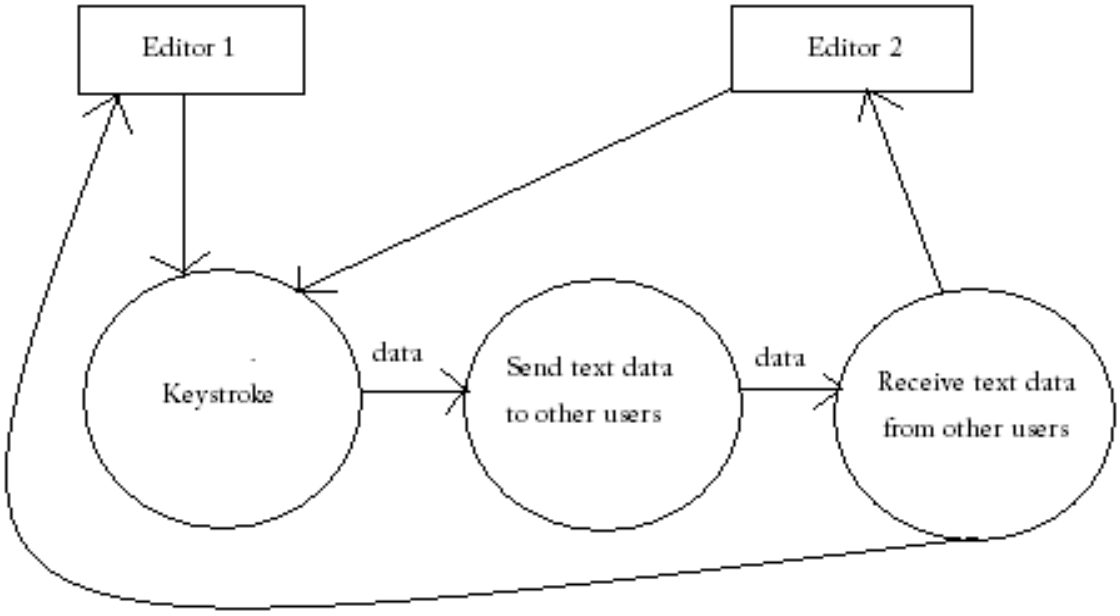
3.3 State Chart Diagram



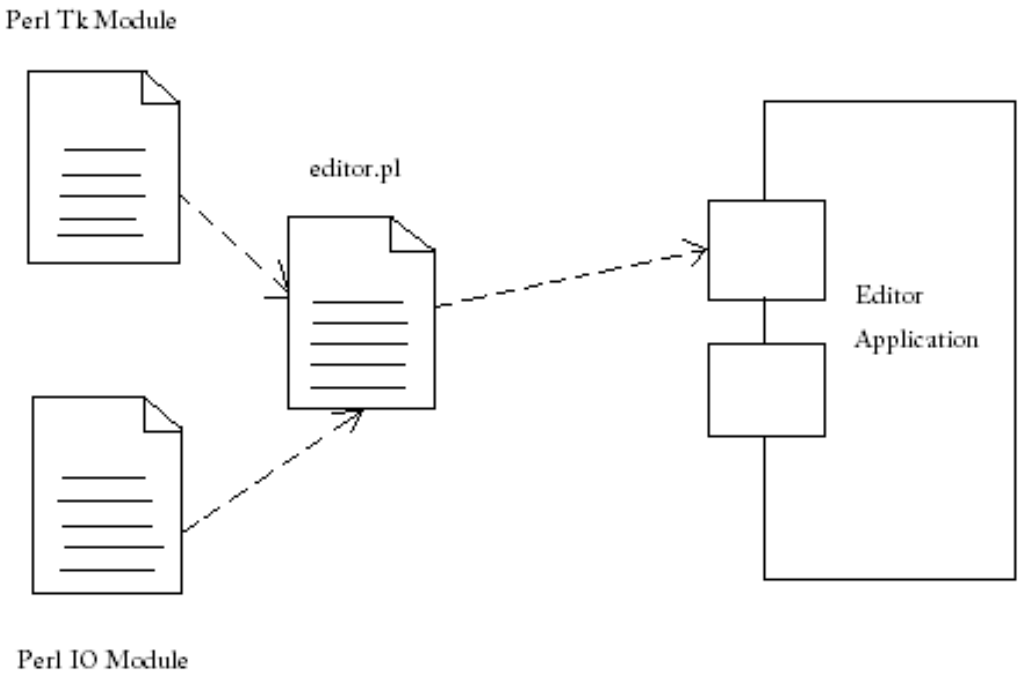
3.4 ActivityDiagram



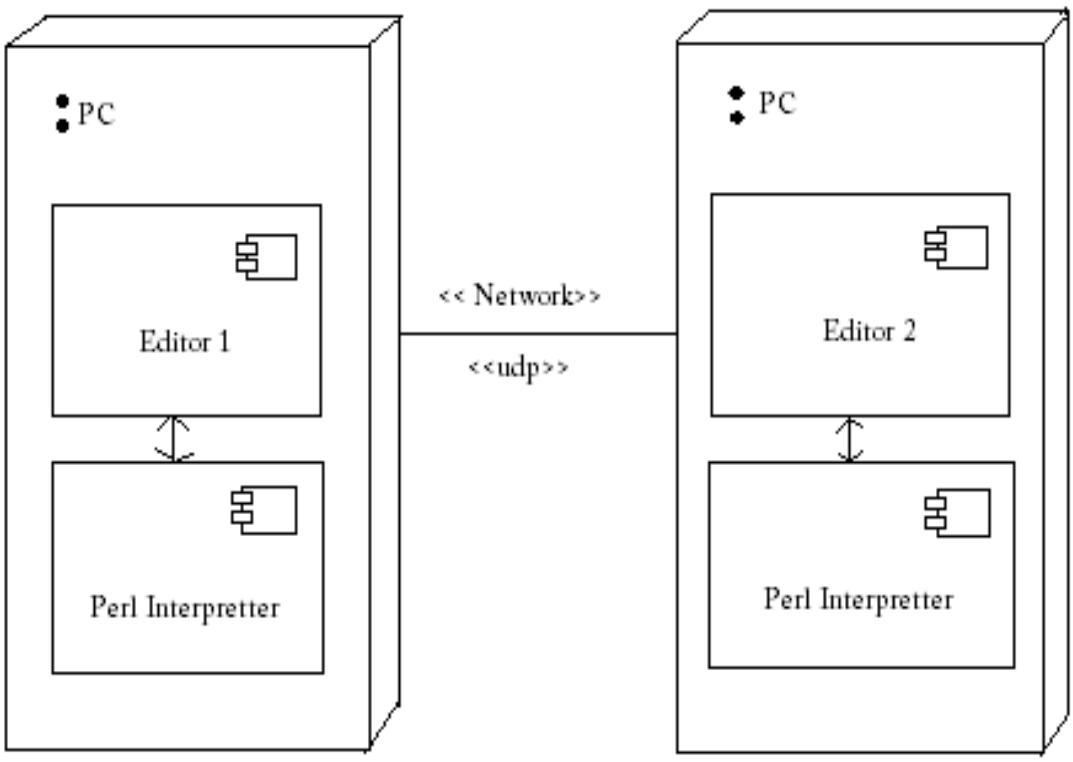
3.6 Data Flow Diagram



3.7 Component Diagram



3.8 Deployment Diagram



4.0 Detailed Design

4.1 Text Editor

Input: Key press
Output: Pressed key

Procedure

1. Accepts key input from the user

   $t = $text->get($line)

2. Displays the pressed key

   $text->insert("$line",$linenumber)

3. Displays the Menu Bar

   $menubarFrame = $mainFrame->Frame(-relief=>'raised', -borderwidth=>2)
   $fileDrop = $menubarFrame->Menubutton(-text=>'File', -underline=>0)

4. If the commands are pressed, then execute the commands

   $col=$fileMenu->command(-label=>'Collaborate',-
   command=>\&collaborate, -underline=>2)

5. Displays the Line Number

   $lineno->configure(-text=>"Line Number:$line Column Number:$col")

4.2 Networking

Input: Key press from Text Editor
Output: Transmits data over the Network

Procedure

1. Initialize the localport to receive any message

   $MySocket1=new IO::Socket::INET-
   >new(LocalPort=>2000,Proto=>'udp',Blocking => 0)

2. Initialize the client connections

```
foreach $ipl in allIP
    {
     $i+=1
     $out[$i]=new IO::Socket::INET-
>new(PeerPort=>2000,Proto=>'udp',PeerAddr=>$ipl)
    }
```

3. Sends a request to all computers that the user wants to collaborate with

```
for $a=0 to$i
{
 $out[$a]->send($msg)
}
```

4. Computers responds with the reply to the collaborate request

```
If(reply is yes)
{
 $msg='!&&&&&!'.$myIP.','.$color.',y'
 $temp->send($msg);
}
else
{
 $msg='!&&&&&!'.$myIP.','.$color.',n'
 $temp->send($msg)
}
```

5. Check for any new packets of data and if so is stored to $A

```
$MySocket1->recv($A,128)
```

4.3 Chat Engine

Input: The message is entered from keyboard into chat box.
Output: The message is received by the chat box at the receiver.

Procedure

1. Getting the characters from keyboard

```
my $chat = $txt->get
```

2. Calls the Function to sent the message to all users by binding that function to enter

$txt->bind('<Return>' sub { ……..});

3. Display the message in chat box

   $log->insert("end","$myName: $chat\n")

4. Send the message to all users

   ```
   $chat1.=$chat.",".$myName.",".$myIP;
   for($j=0;$j<=$i;$j++)
   {
     $out[$j]->send($chat1);
    }
   ```

5. Wait for next message

4.4 User Interface

Input: All the Individual Modules
Output: One single Graphical User Interface

Procedure

1. Integrating the Text editor to the main Window

   $textFrame = $mainFrame->NoteBook()

2. Integrating the Chat engine to the main window

   $log = $mainFrame->Scrolled('ROText',-scrollbars=>'ose',-height=>8,-background => 'white')-pack(-side=>'top',- fill=>'x',-pady=>4)

3. Integrating the Popup windows to the main window

   ```
   $main = MainWindow -> new()
   $main->Label(-text=>'Enter username : ')->pack(-pady=>4)
   ```

4. Integrating the Add-On features of Text editor such as Font, Search

   ```
   $family = $text->fontActual($f, -family)
   $font->configure(-size=>$fsize, -family=>$family)
   $text->FindPopUp()
   ```

5.0 Test Plan

5.1 Introduction

The project aims at designing a text editor which allows more than two people to edit the same text from different locations in a network.

5.2 Overview

5.2.1 Description of Document

This document is a Test Plan for the Network Analysis and Intrusion Control. It describes the testing strategy and approach to validate the quality of this product. It also contains various resources required for the successful completion of the project.

Preparation for this test consists of two major stages -

- The Test Approach sets the scope of system testing, the activities to be completed, the general resources required and the methods and processes to be used to test the release.

- Test Conditions/Cases documents the tests to be applied, the data to be processed and the expected results.

5.2.2 System Overview

The purpose of the system and the software to which this document applies is stated below. It shall describe the general nature of the system and the software; summarize the various modules and functions they perform. This project includes the following modules.

1. Text Editor.

    - Inserting into the editor
    - Using Basic Features
    - Using Advanced Features

2. Networking.

    - Connecting all computers
    - Testing collaboration

- Testing failure of a computer

3. Chat Engine

- Inserting into the chat box
- Network Connections
- Launching Chat Engine

4. User Interface.

- Opening the file
- Overloading the GUI
- Editing the text using the edit options
- Exit

5.3 Preparation for testing

various test cases were identified in the various modules through comparison with existing systems.

5.4 Scope of testing

1. To detail the activities required to prepare for and conduct the test.

2. To communicate to all the project group members the task they are to perform and the schedule to be followed in performing the tasks.

5.5 Testing

Module Name: **Text Editor**

Description of test case.	**Data used or field changed.**	**Expected Results.**	**Actual Results.**
Inserting in the editor.	Inserting text without Collaborate Option.	Text is entered without exceptions.	
Using Basic Features.	Using features such as Save, Include ,Close and Save	The features work without errors.	
Using Advanced Features	Add-On feature such as Undo, Redo, Include File etc	The features works without errors.	

Module Name: **Networking**

Description of test case.	**Data used or field changed.**	**Expected results.**	**Actual Results.**
Connecting all Computers.	Accepting IPs of all computers to be Collaborated with and connects to all other	The IP list is sent to all computers participating in the editing and all of them are connected with each other in Peer to Peer mode.	
Testing Collaboration.	All editors are used together to edit the same file and checked for updating without a problem.	Any numbers of Computers are able to be connected and are able to update their copy with the changes others make.	
Testing the Failure of a computer.	A computer's editor is closed and checked if all others can work without a problem.	All others are able to continue without a problem.	

Module Name: **Chat Engine**

Description of test case.	**Data used or field changed.**	**Expected results.**	**Actual Results.**
Inserting into the chat box	Typing the messages for chatting purpose.	Messages are displayed without exceptions.	
Network Failure	The host machine gives the IP addresses of the systems that are to be connected.	Systems connected without errors and messages relayed properly.	
Launching chat engine	Chat Engine is integrated along with the text editor and opens along with it	The chat engine is loaded along with the editor.	

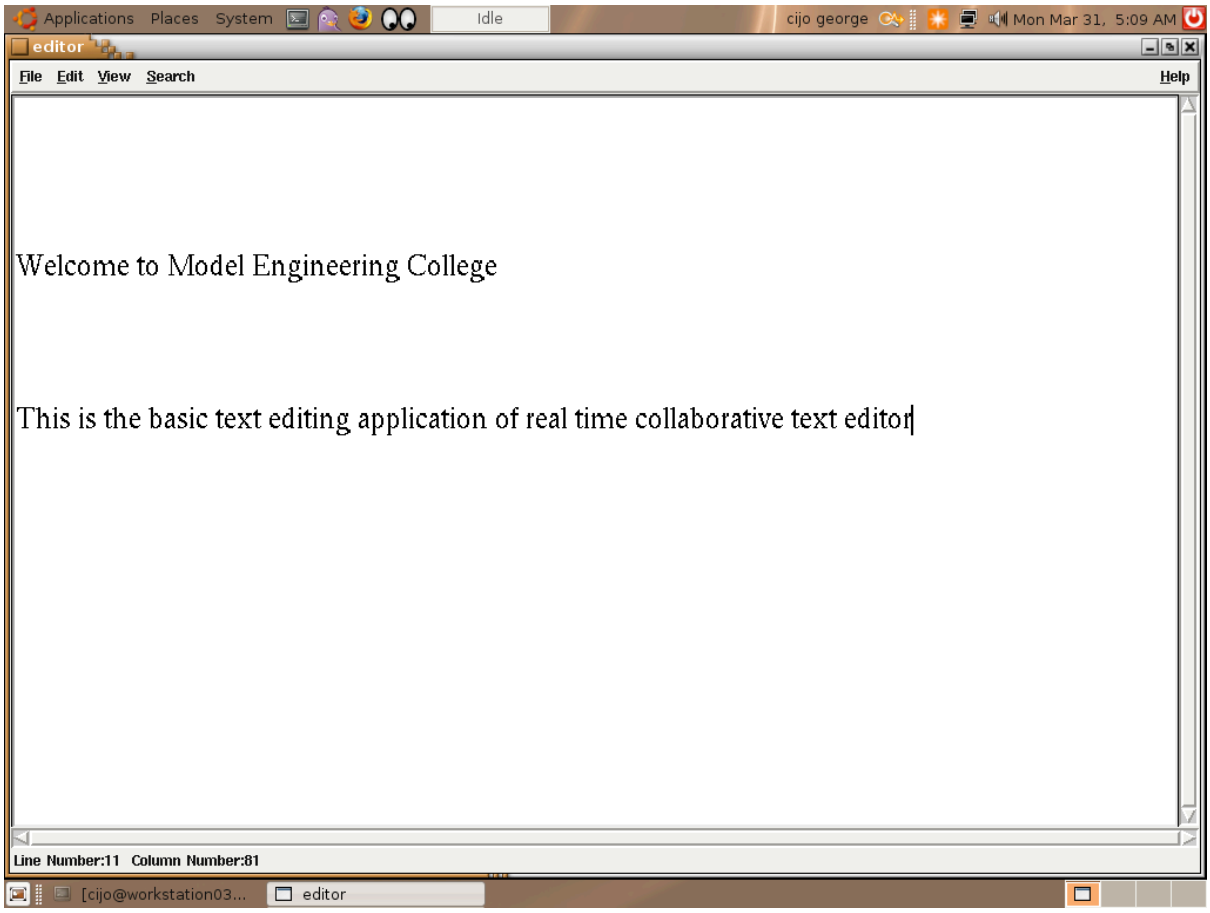
Module Name: **User Interface**

Description of test case.	**Data used or field changed.**	**Expected results.**	**Actual Results.**
Opening the file	The file exists in all computers and each user can open the file in their system.	The file opens without errors when the path provided is correct.	
Overloading the GUI	Typing in data continuously endlessly.	GUI does not crash even when it is overloaded and displays the typed messages	
Edit options provided in the text editor	The text is typed in any of the given fonts and can be of any of the given size.	The text that is typed in will be displayed in the exact format in all systems accessing the file.	
Exit	The exit button.	Exits from the program.	

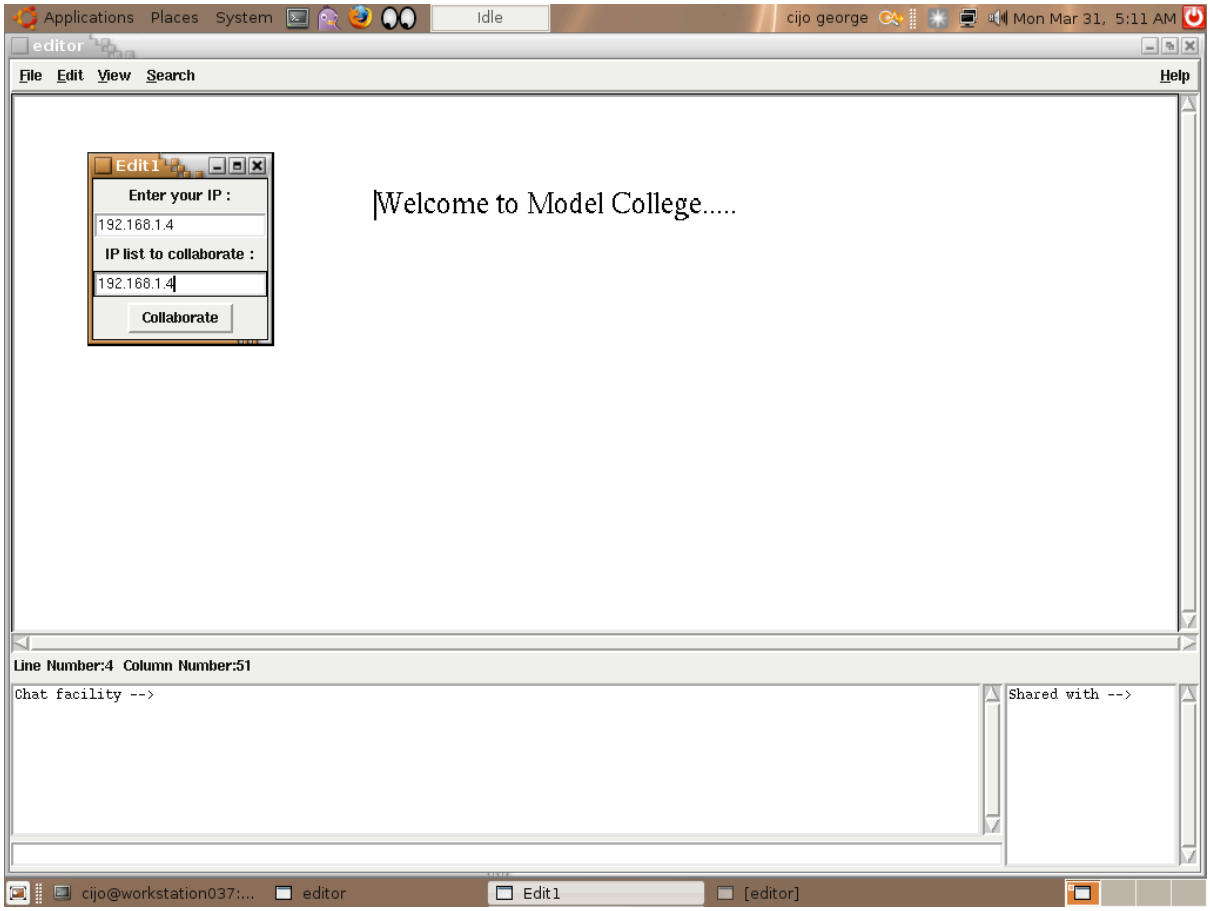
6.0 Implementation Details

6.1 Screenshots

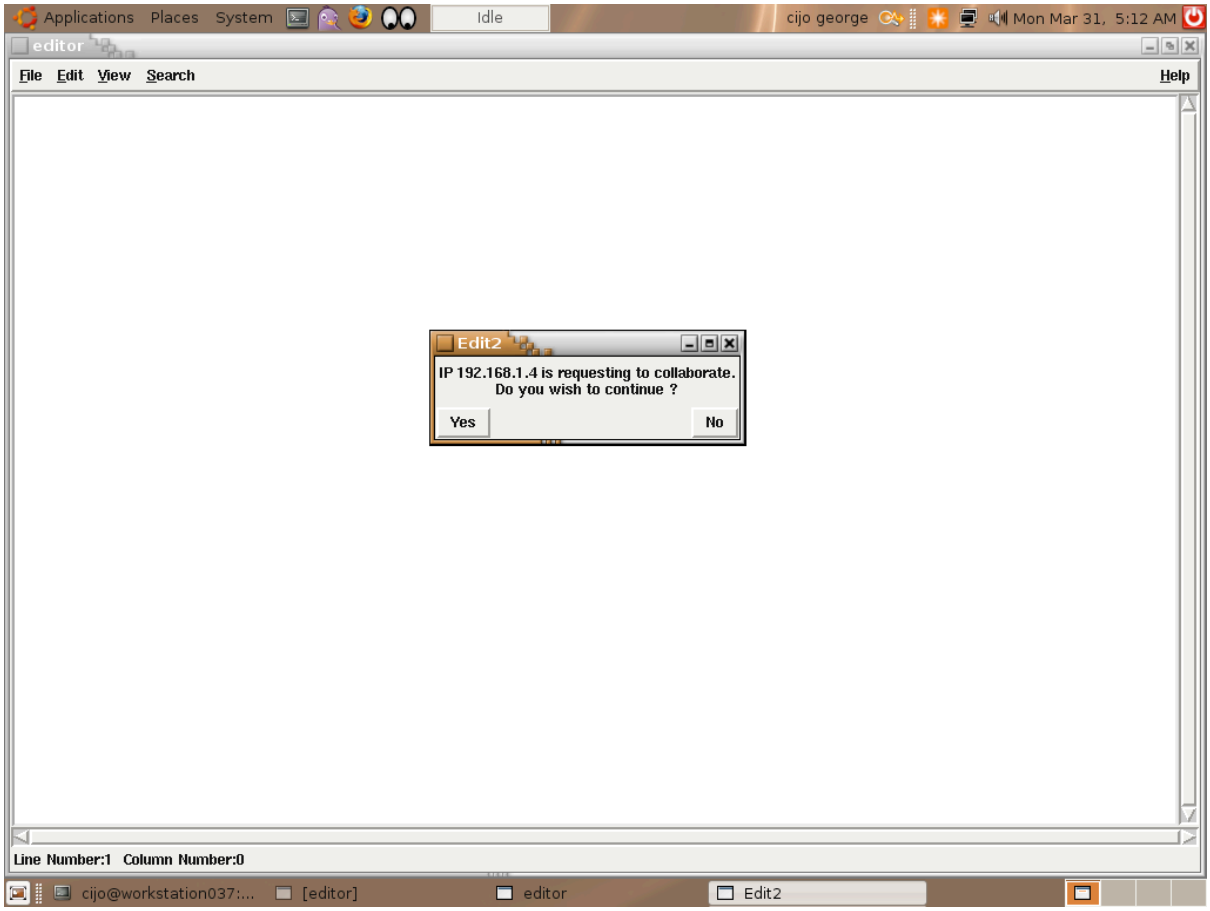
6.1.1 Basic Text Editor



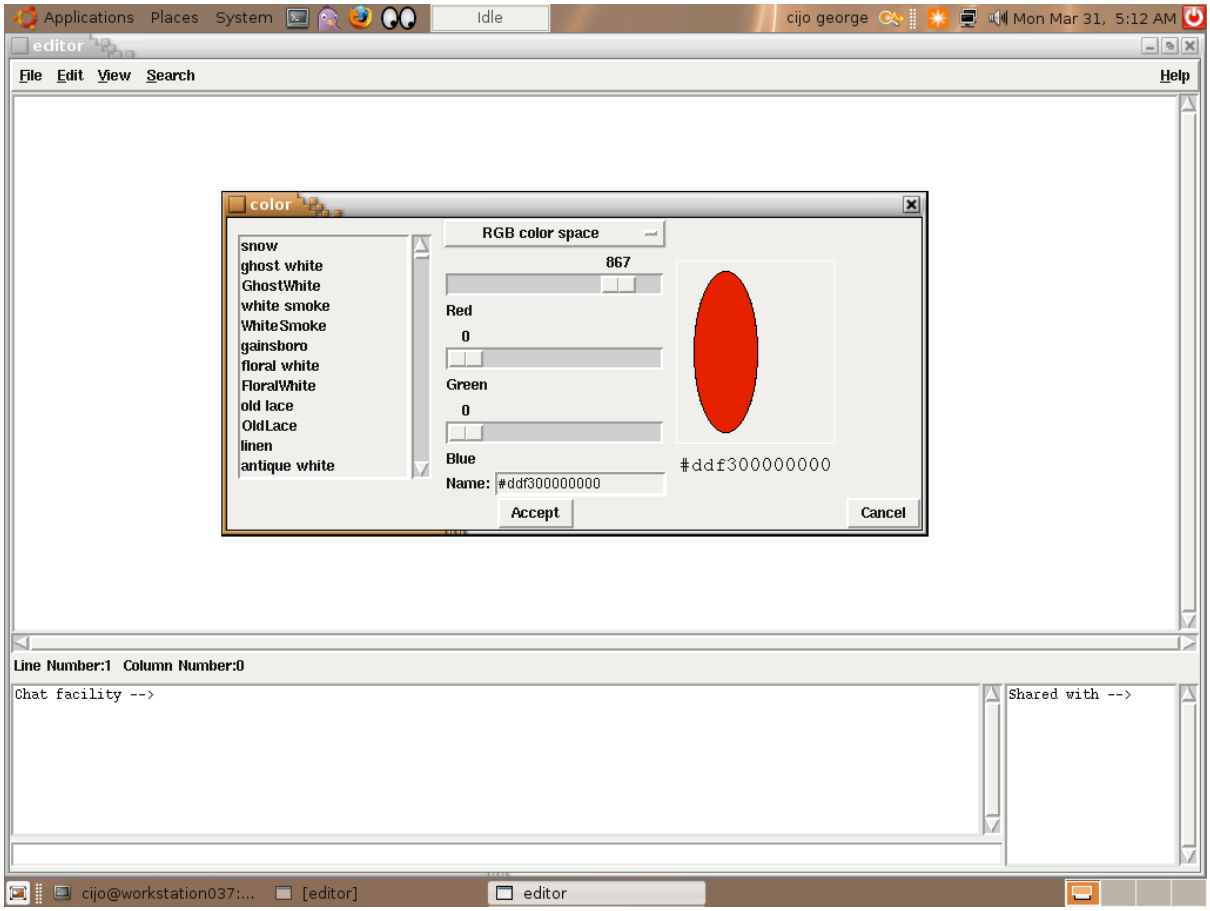
6.1.2Send Request to Collaborate



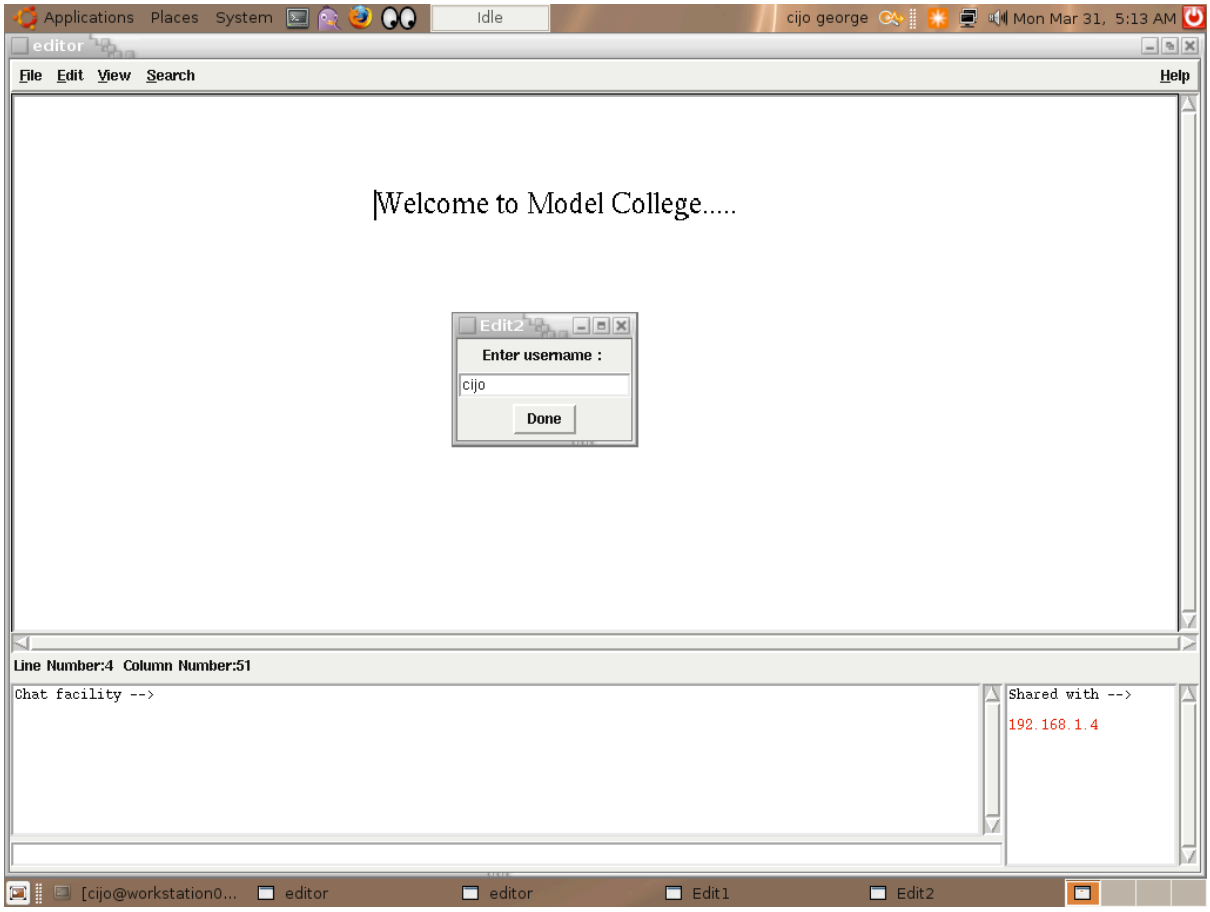
6.1.3Request Confirmation



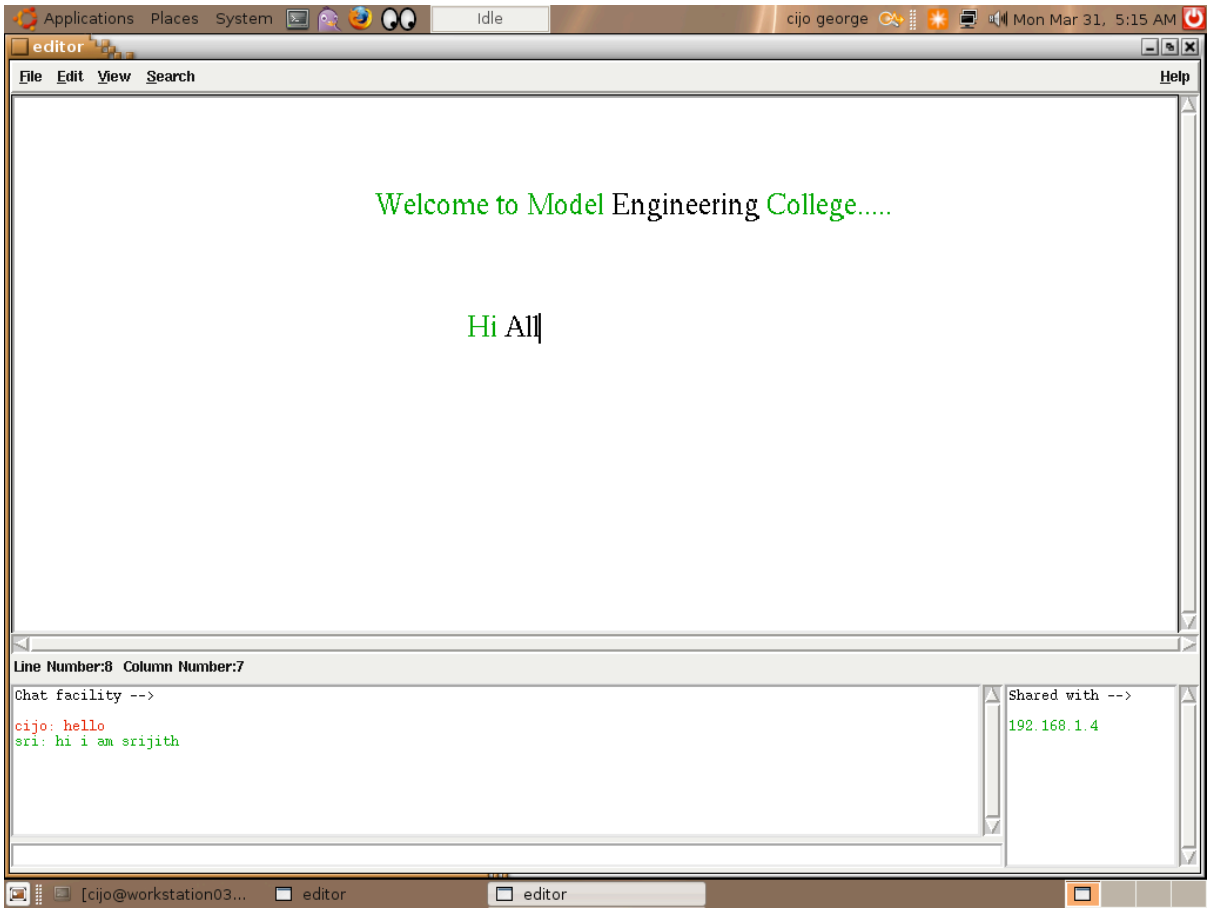
6.1.4Select Unique User Colour

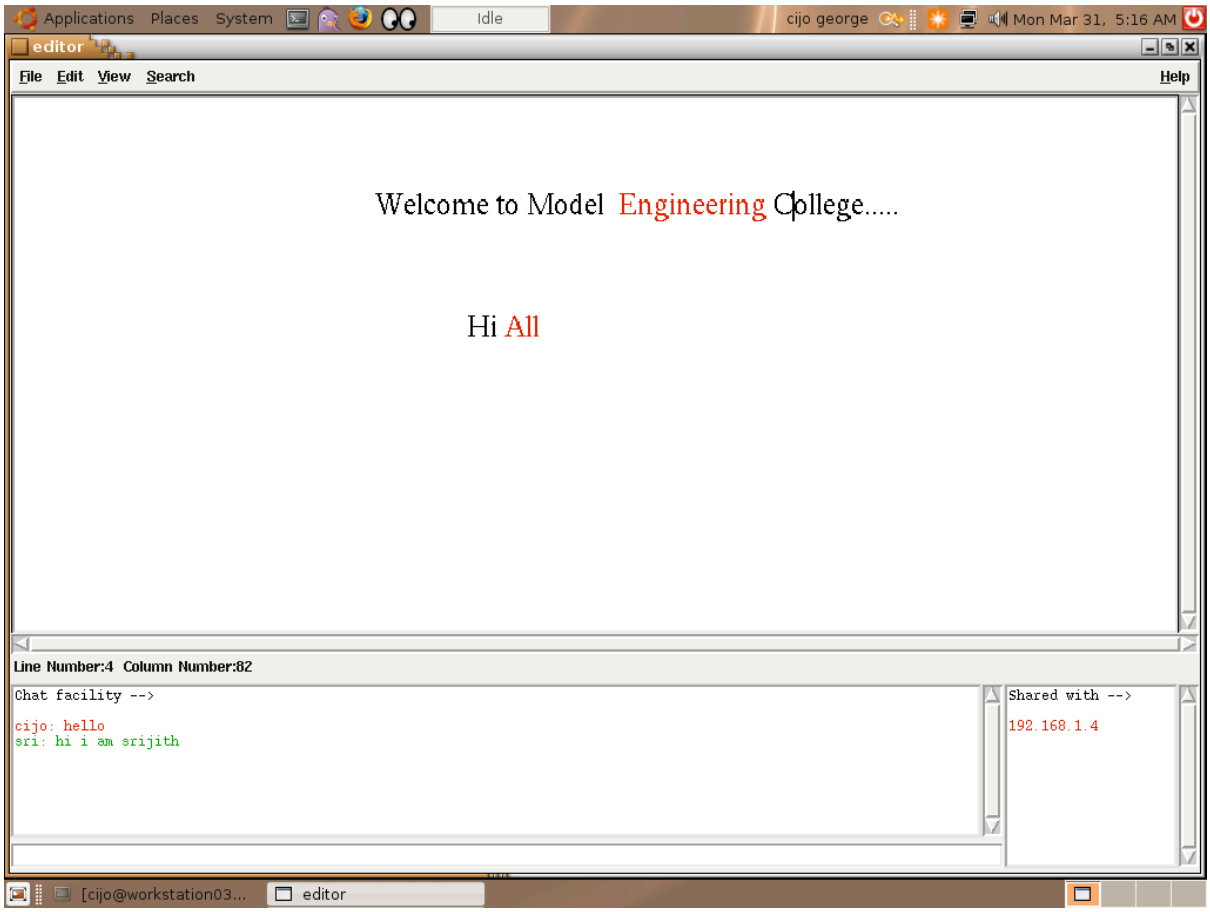


6.1.5Enter Username



6.1.6 Collaborate Text Editing





7.0 Conclusion

The concept of 'Real-time' is the core of this project. A collaborative editor is a software application that allows several people to edit a text file from different computers in a network. Web-based collaborative text editing applications does not allow real-time editing, i.e. text is not updated for all users in real-time. Also, such applications work in server-client mode. A non web-based real-time peer-to-peer collaborative editor thus gains its significance.

Developing the project in a new generation programming language like 'PERL' provides exposure to a language of the future and enables the addition of advanced features to the software.

8.0 Bibliography

- Mastering Perl/Tk : by Stephen Lidie, Nancy Walsh
- Programming the Network with Perl : by Paul Barry