CORPORATE COMMUNICATION SYSTEM OVER IP

FOR

CBK INFOTECH INDIA (P) LTD., BANGALORE

PROJECT REPORT

P-1227

Submitted in partial fulfillment of the requirements for the award of the

degree of

M.Sc Applied Science Software Engineering

Of Bharathiar University

Coimbatore

Submitted By

EZHILARASAN.N

Reg. No. 0137S0030

Guided By

EXTERNAL GUIDE
Mrs.V.Leena, M.C.A,
Software Developer.

INTERNAL GUIDE
Mr.S.Ganesh Babu, M.C.A,
Lecturer.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

KUMARAGURU COLLEGE OF TECHNOLOGY

COIMBATORE – 641 006

SEPTEMBER 2004

Certificates

KUMARAGURU COLLEGE OF TECHNOLOGY

(Affiliated to Bharathiar University)

Department of Computer science and Engineering

Coimbatore – 641 006

CERTIFICATE

This is to certify that the project work entitled

CORPORATE COMMUNICATION SYSTEM OVER IP

Done By

EZHILARASAN.N

Reg. No. 0137S0030

Submitted in partial fulfillment of the requirements for the award of the degree M.Sc Applied Science Software Engineering of Bharathiar University.

Professor and Head **Internal Guide**

Submitted for the University examination held on 29/09/04

Internal Examiner **External Examiner**

CBK Infotech India (P) Ltd.

4/3, Sree Complex,
M.S.R.E.C. Road
Opp. NIAS, Mathikere,
Bangalore - 560 054
E-mail : cbk@satyam.net.in
Ph. : 3602402 / 3603995

Ref No. CBK/CT/2009/2004/02 Date:20.09.2004

TO WHOMSOEVER IT MAY CONCERN

This is to certify that **Mr. N. Ezhilarasan, (Reg . No. 0137S0030)** of Kumaraguru College Of Technology, Coimbatore (affiliated to **Bharathiar University**), has successfully completed the project titled **'Corporate Communication System Over IP'** in compliance with the requirement of partial fulfillment of the **Master Of Science (Software Engineering).** He was associated with us during the period from June -2004 to September-2004.

As per our company policy the source code is a property of CBK Infotech India (P) Ltd. And it cannot be disclosed.

For and on behalf of CBK Infotech India Pvt Ltd.

HR Manager

Declaration

DECLARATION

I hereby declare that the project work entitled

CORPORATE REMOTE TASK MANAGER

Done at

CBK INFOTECH INDIA (P) LTD, BANGALORE.

And submitted to

KUMARAGURU COLLEGE OF TECHNOLOGY

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

M.Sc. APPLIED SCIENCE (Software Engineering)

Is a report of work done by me during my period of study in

Kumaraguru College of Technology, CBE – 641 006

Under the supervision of

Mr.S.Ganesh Babu, M.C.A,

Lecturer, Dept of Computer science & Engineering,

Kumaraguru College of Technology, Coimbatore.

Place: Coimbatore

Date: 24/09/04

Signature of the Candidate

EZHILARASAN.N

(Reg. No. 0137S0030)

Acknowledgement

ACKNOWLEDGEMENT

I am extremely thankful to our beloved principal Dr K.K PADMANABAN, Kumaraguru College of Technology, for being a source of inspiration throughout my study in our college.

I owe my sincere gratitude to Dr. S. THANGASAMY, Head of Department, Computer Science and Engineering, Mr. K.R. BASKARAN, Course Coordinator, Master of Science (Software Engineering), Kumaraguru College of Technology, who took keen interest in me on every effort throughout this course.

I am indebted and extremely thankful to my internal guide Mr. S. GANESHBABU, M.C.A, for his valuable guidance and help throughout the course of the project.

I am grateful to Mr. CHANDRA BHUSHAN, CBK Infotech India (P) Ltd, Bangalore, for giving me the great opportunity to take up this project and for providing the excellent facilities to execute the project. I also wish to express my sincere thanks to Mrs. V. LEENA, Software Developer, CBK Infotech India (P) Ltd, Bangalore for his valuable guidance and constant encouragement during the course of the project.

I would fail in my duties if I don't thank Lord Almighty, my Family and my Friends for their invaluable support to see through this project.

Synopsis

SYNOPSIS

The goal of the project is to develop an application, which fulfill all possible communication system such as Messaging, Voice Communication, Fax system, along with the utilities like Activity reminder system and address book for the corporate network. The application will offer a real time point-to-point voice service providing virtual duplex voice communication over the corporate network.

The objective of the project is to establish a voice communication system using "Dataline Application programming interface using PCM format". Fax Over IP using "Datagram Protocol" and other messaging system using "TCP/IP" and reminder system & Address book using "FirstSQLJ", so that any corporate can using this application as their complete communication system as per their need.

Existing system is purely platform dependent. But the CCSOIP (Corporate Communication System Over IP) is Platform independent. A platform is the hardware or software environment in which a program runs. Some of the most popular platforms like Windows 2000, Linux, Solaris, and MacOS. Most platforms can be described as a combination of the operating system and hardware. The Java platform differs from most other platforms in that it's a software-only platform that runs on top of other hardware-based platforms.

Table of Contents

TABLE OF CONTENTS

Chapter 1

INTRODUCTION

1. INTRODUCTION

1.1 PROJECT OVERVIEW

The major aim of the project is to develop an application, which fulfils all possible communication system such as Messaging, Voice Communication, Fax system, along with the utilities like Activity reminder system and Address book for the corporate network. The application will offer a real time point-to-point voice service providing virtual duplex voice communication over the corporate network..

Corporate Communication System over IP (CCSOIP) uses the Internet Protocol (IP) to transmit voice as packets over the IP network. So Voice communication can be achieved on any data network that uses IP, like Internet, Intranets and Local Area Networks (LAN). Here the voice signals are digitized and converted to IP packets and then transmitted over the IP network. At the other end the IP packets are quantized and converted to voice signal.

Since it became popular, this has attracted many developers to develop software applications for better communications than offered by plain text. The applications are those that support voice communication. In this project we developed a voice communications application, which offers a real time point-to-point voice service providing virtual duplex voice communication over the Internet.

This project is implemented using JAVA as the platform, which provides the developers, the advantages of platform independence, availability of API's for implementation of the features, which are included in the Corporate Communication System Over IP.

1.2 ORGANIZATION PROFILE

CBK Infotech: Partners of TCS (Tata consultancy Services) is a world class software led IT services. CBK Infotech is an IT service company providing a range of value added software services to:

- Hardware product companies
- Software product companies
- End-user in large and medium business organization.

The information is the hallmark of today's world. A driver for productivity and the ability to offer quality solutions on information super high way are the key to development CBK Infotech India (P) Ltd. (hereby mentioned as CBK as well as company) has mirrored.

The essence of true development since 1988 by enhancing growth with the presence of social justice. In promoting and cherishing the growth of those associated with clients who are the true partners in progress.

There is no shortcut to success so as in the case of IT industry too. It is never possible without innovation, an eye for vision, a strong will to succeed and unlimited quality service. Quality objectives, precise and time bound are the root criteria for success and development is not an exception with CBK.

CBK will leave no stone unturned to reach its customer to the topmost rung of ladder success. A result that is translated at CBK, i.e. - in tune with technology with time and trust, truth and tradition, and requirement is the principle assets. CBK has two divisions working at the moment - Training division as Compu Home and a software development division. It is the development division that is offering this project training as detailed in this document.

CBK provides the state of the art technology like COM, COM+, Active-X, ASP, 3-tier solutions etc. and limited support of its clients in India and abroad. CBK also provides Consultancy services for all IT related matters to its clients. With the revolving strategy and re-

structuring, CBK has now started offering Web based solutions and gearing towards providing the E-Commerce / M-Commerce solutions to its existing and new clients.

Training Tie-up

CBK INFOTECH having status of ATC of TATA CONSULTANCY SERVICES for committed to excellence in corporate training. CBK have the unique advantage of combining a management perspective with in-depth technical knowledge in all our training solutions CBK offer a wide range of training programs to meet the requirements of corporate clients.

Other Tie-ups

The following are the measure Tie-ups of CBK INFOTECH in the areas of Software Development, Consultancy and Training.

1. Project Development Partner of TCS
2. Technology Team Development of RGSL
3. Prototype Development for VFM Software Solutions (p) LTD.

Work related areas @ CBK:

1. Business Application Development
2. Developing Device And Device Drivers
3. Web enabled applications development
4. Client / Server Applications Development
5. Embedded System
6. Research & Development in WAP and WEB related conversing technologies
7. Corporate training
8. High-end User Training (Vocational)
9. Industrial Automation
10. Data Processing

11. One-Wire and Tini Technology
12. Palm top/Hand Held PC Application Development/ E-CRM.....

It is the policy of CBK to design, develop, deliver, maintain and support high quality software solutions. This is done not only to meet the client's requirements but also to exceed their expectations by being their true partners to the ladder of success. CBK extend its services to its clients by providing skilled manpower resources on contractual basis. This leads to a dedicated human resources development program.

Chapter

2

SYSTEM STUDY AND ANALYSIS

2. SYTEM STUDY AND ANALYSIS

2.1 SOFTWARE REQUIREMENT SPECIFICATION

The different desirable characteristics of an SRS are: Understandable, Unambiguous, Completeness, Verifiable, Consistency, Modifiable, and Traceable.

Different modules in the proposed system are:

- Client process
- Server process
- Recording and play back speech
- FileTransfer.

✓ Client and Server Processes

CCSOIP provides a point-to-point voice communication over the Internet. CCSOIP will work on a single server, which will notify the user when a client someone@somewhere.net will be trying to contact the user via CCSOIP.

The CCSOIP consists of a single server. The server can initiate a connection, as well as accept a connection request from another server. This eliminates the needs for running of two separate programs, which only serves to tie up system resources, and thus slow down other programs.

One server sends a message to the remote server requesting a connection with remote server. If the remote server is running, the remote server will inform the user of a connection request, and the user will either accept or reject the connection.

The server will then send appropriate message to the server, either accepting or rejecting the connection. If the remote server user accepted the connection, the connection will then be established, and the recording processes will be initiated.

✓ **Recording and play back speech**

The recording device should be detected engaged before going for the recording process and also the information of the device will be captured. CCSOIP will provide a "virtual" duplex voice communication with a half duplex sound card. The software will automatically open a channel for either recording or playback.

✓ **File Transfer**

CCSOIP uses UDP (User Datagram Protocol) for transfer of files between the peers of the network. The person working on the peer who wish to transfer a particular file to another peer, first sends a request to the desired destination about the transfer of file and its complete details. Once the transmitting side, receives the acceptance of file transfer from the destination, the transfer of file begins and the destination is notified once the file transfer terminates.

2.2 EXISTING SYSTEM

We already have the facility of voice chatting using computer on Internet. The existing systems of voice chatting having some demerits: In this system the client has to wait for a long time to be connected to another system. A dedicated server is required to take care of the clients.

The reason behind is that the client cannot directly connect to another client. The process is that at first, the client has to connect to the server & server will take care how to connect to the other client.

As of now in voice chatting system the noise is the main problem. The client will listen original voice along with the noise. Some time it may happen that the client not able to recognize the original voice. There is no facility to avoid noise.

The whole process of chatting is very slow. Since the process is slow it is also not cost effective and it sends the voice packets using TCP protocol so every time it has to wait for acknowledgement or response. The problem associated with the existing system like delay for connection, noise, slow processes degrade its performance. In a nut shell the existing voice chatting is not cost effective and some time very difficult to recognize the original voice.

The ITU-T recommendation G711 is a 64 kbps coder for multimedia communications. This is named as Pulse Code Modulation (PCM). The data transmission by this algorithm is 64 kbps. The quality of this is good. And its complexity is lowest in terms of cost. Complexity can be measured in terms of computing speed in MIPS, of random access memory (RAM), and of read-only memory (ROM). So the complexity determines cost.

The transmission medium used is TCP/IP. Two ends are connected with connect () function and the data will be sent using send () function from the sender, and the data will be received using the receive () function at receiver end.

- **Telephone system**

The telephone has two copper wires coming out of it that go directly to the telephone company's nearest end office (also called local central office).

The two wire connections between each subscriber's telephone and the end office are known as Local loop. If a subscriber attached to the same end office, the switching mechanism within the office sets up, a direct electrical connection between the two local loops. This connection remains intact for the duration of the call.

If the called telephone is attached to another end office a different procedure has to be followed. Each end office has a number of outgoing lines to one or more nearby switching centers, called Toll office. These

lines are called connecting Trunks. If both the callers and callees end office have a Toll connecting Trunk to the same toll office, the connection may be established within the toll office.

If the caller and callee do not have a toll office in common, the path will have to be established somewhere higher up in hierarchy. There are primary, sectional and regional offices that are connected. The toll primary, sectional and regional exchanges communicate with each other via high bandwidth inter-toll Trunks.

In telephone system we are sending analog signal. There are advantages of digital signal over analog signaling. Some advantages are given as below:

- Digital transmission is that voice, data, music, and image can be interspersed to make more efficient of the data rates are possible using existing lines.
- Digital transmission of much cheaper than analog transmission.
- Maintenance of digital system is easier than maintenance of an analog one.
- PSTN (Public Switched Telephone Network) were design to transmit the human voice in a more or less recognizable form.
- A cable running between two computers can transfer data at speeds of 10 to power 7 to 10 to power 8 bits/sec. The error rate is usually so low it is hard to measure. A dial up line has a maximum data rate on the order of 10 to power 4 bits/sec and error rate of roughly 1 per 10 to power 5 bits/sec.

The bit rate time's rate performance of a local cable is thus 11 orders of magnitude better than a voice grade telephone line.

Using CCSOIP (Corporate Communication System Over Internet Protocol) we can chat in voice with different users working on the different peers of the corporate network.

2.3 PROPOSED SYSTEM

Corporate Communication System over IP will be providing real time voice communication over the Internet, along with the fax, messaging and reminder system with the facility of address book. This software will be tested on two machines that will be placed side by side. The targeted time difference between the moment the user on machine A will start talking and the moment the user on machine B will hear on the user on machine A.

CCSOIP will include a fax over IP with the facilities of reminder and messaging system. A user friendly graphical user interface will be developed using Java Swings.

CCSOIP is aimed to develop using JAVA and its API correspondingly for implementation which gives benefits of platform independence and all the benefits of JAVA language.

CCSOIP will have over all in it the modules as listed below:

- Implementation of messaging system in the form of messages between the PC's that are peers of the corporate network.
- Implementation of Voice communication between the PC's that is part of the network of the corporate.
- Implementation of Transfer of different information that are in the form of various files which forms the functionality of FAX between the peers.
- Building the Address book that helps out the users of CCSOIP for their utilization of the same in the process of communication.
- Building of the Reminder system for the end users of CCSOIP.
- Building the Graphical User Interface for integration of all the features as listed above and to provide the end user the user friendliness for the utilization.

For implementation of communication between the peers in the form of Messages, TCP/IP protocol will be utilized which will be reliable for the functionality.

For communication of Voice and data that exists in the form of files, "USER DATAGRAM PROTOCOL" will be used which gives the faster transfer of voice and data between the peers of the network.

For implementation of Reminder System and Address book for the benefit of end users, Database known as "FirstSQLJ" will be utilized. The advantage of using "FirstSQLJ database" is as below:

- ✓ It is completely developed in JAVA.
- ✓ It can be embedded into the application, which makes database portable along with the application.
- ✓ The functionality is equivalent to any RDBMS

The reminder system which is a utility in CCSOIP will have features of alert system as per the user requirements.

For voice communication, the features of alert of incoming calls, user status at both the ends of communication will be implemented.

In CCSOIP, it is possible to communicate to more than one user at the same time for messaging and fax system. For voice communication, it is made one to one communication.

Chapter

3

PROGRAMMING ENVIRONMENT

3. PROGRAMMING ENVIRONMENT

3.1 HARDWARE CONFIGURATION

Domain	:	Networking
Processor	:	Pentium III
RAM	:	128 MB
Hard Disk	:	10GB
Monitor	:	VGA Color Monitor
Network Card (Speed)	:	100mbps
Network	:	any supporting TCP/IP

3.2 DESCRIPTION OF SOFTWARE AND TOOLS USED

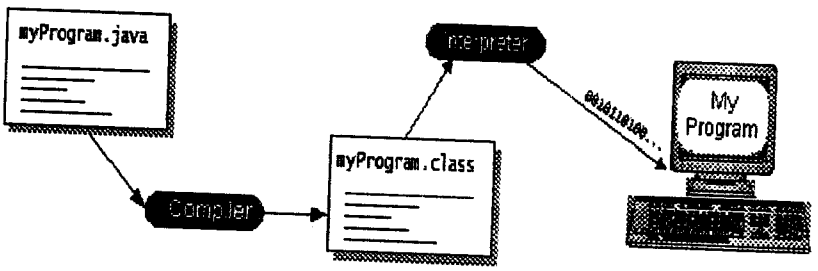
➢ The Java Programming Language

The Java programming language is a high-level language that can be characterized by all of the following buzzwords:

- Simple
- Object oriented
- Distributed
- Interpreted
- Robust
- Secure
- Architecture neutral
- Portable
- High performance
- Multithreaded
- Dynamic

With most programming languages, either compile or interpret a program so that we can run it on our computer. The Java programming language is unusual in that a program is both compiled and interpreted. With the compiler, first you translate a program into an intermediate language called *Java bytecodes* —the platform-independent codes interpreted by the interpreter on the Java platform. The interpreter parses and runs each Java bytecode

instruction on the computer. Compilation happens just once; interpretation occurs each time the program is executed. The following figure illustrates how this works.



Java bytecodes help make "write once, run anywhere" possible. We can compile our program into bytecodes on any platform that has a Java compiler. The bytecodes can then be run on any implementation of the Java VM. That means that as long as a computer has a Java VM, the same program written in the Java programming language can run on Windows 2000, a Solaris workstation, or on an iMac.

➢ The Java Platform

A platform is the hardware or software environment in which a program runs. We've already mentioned some of the most popular platforms like Windows 2000, Linux, Solaris, and MacOS. Most platforms can be described as a combination of the operating system and hardware. The Java platform differs from most other platforms in that it's a software-only platform that runs on top of other hardware-based platforms.

The Java platform has two components:

- Java Virtual Machine
- Java Application Programming Interface

The Java API is a large collection of ready-made software components that provide many useful capabilities, such as graphical user interface (GUI) widgets. The Java API is grouped into libraries of related classes and interfaces; these libraries are known as packages.

Java packages used in CCSOIP:

- javax.swing.* - Used for developing GUI interface
- java.sql.* - Used for Database connectivity
- java.util.* - Provides classes for Date and Time
- java.awt.* - Provides classes for Event handling
- java.io.* - Provides classes for I/O functions
- java.net.* - Provides classes to create sockets

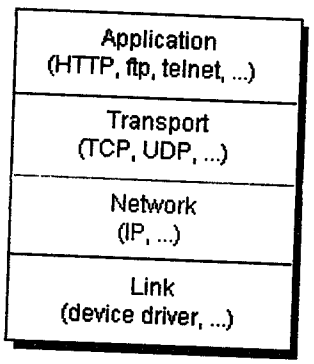
➢ Overview of the Swing

The Swing package is part of the Java™ Foundation Classes (JFC) in the Java platform. The JFC encompasses a group of features to help people build GUIs; Swing provides all the components from buttons to split panes and tables. Here we are using the containers JFrame and JPanel for the designing the GUI. And also we use the swing components for the design.

The Swing package was first available as an add-on to JDK 1.1. Prior to the introduction of the Swing package, the Abstract Window Toolkit (AWT) components provided all the UI components in the JDK 1.0 and 1.1 platforms. Although the Java 2 Platform still supports the AWT components. You can identify Swing components because their names start with J. The AWT button class, for example, is named Button, whereas the Swing button class is named JButton. In addition, the AWT components are in the java.awt package, whereas the Swing components are in the javax.swing package.

➢ Overview of the Networking

Computers running on the Internet communicate to each other using either the Transmission Control Protocol (TCP) or the User Datagram Protocol (UDP), as this diagram illustrates:



When we write Java programs that communicate over the network, we are programming at the application layer. Typically, we don't need to concern our self with the TCP and UDP layers. Instead, we can use the classes in the java.net package. These classes provide system-independent network communication. However, to decide which Java classes our programs should use, we need to understand how TCP and UDP differ.

➢ TCP

When two applications want to communicate to each other reliably, they establish a connection and send data back and forth over that connection. This is analogous to making a telephone call. If you want to speak to Aunt Beatrice in Kentucky, a connection is established when you dial her phone number and she answers. You send data back and forth over the connection by speaking to one another over the phone lines. Like the phone company, TCP guarantees that data sent from one end of the connection actually gets to the other end and in the same order it was sent. Otherwise, an error is reported.

TCP provides a point-to-point channel for applications that require reliable communications. The Hypertext Transfer Protocol (HTTP), File Transfer Protocol (FTP), and Telnet are all examples of applications that require a reliable communication channel. The order in which the data is sent and received over the network is critical to the success of these applications. When HTTP is used to read from a URL, the data must be received in the order in which it was sent. Otherwise, you end up with a jumbled HTML file, a corrupt zip file, or some other invalid information.

The TCP protocol is for the Voice Communication. We send the encoded data as packets through the TCP protocol. To establish a TCP connection we are using InetAddress class, Socket class and Server Socket class.

- **UDP**

The UDP protocol provides for communication that is not guaranteed between two applications on the network. UDP is not connection-based like TCP. Rather, it sends independent packets of data, called datagrams, from one application to another. Sending datagrams is much like sending a letter through the postal service: The order of delivery is not important and is not guaranteed, and each message is independent of any other.

For many applications, the guarantee of reliability is critical to the success of the transfer of information from one end of the connection to the other. However, other forms of communication don't require such strict standards. In fact, they may be slowed down by the extra overhead or the reliable connection may invalidate the service altogether.

Consider, for example, a clock server that sends the current time to its client when requested to do so. If the client misses a

packet, it doesn't really make sense to resend it because the time will be incorrect when the client receives it on the second try. If the client makes two requests and receives packets from the server out of order, it doesn't really matter because the client can figure out that the packets are out of order and make another request.

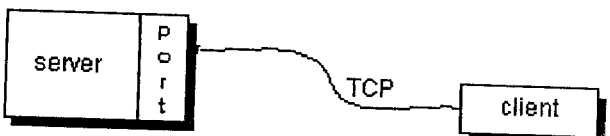
The UDP protocol provides for communication that is not guaranteed between two applications on the network. UDP is not connection-based like TCP. Rather, it sends independent packets of data from one application to another.

UDP protocol is used for file transfer and messaging between the systems connected in network. Datagram Socket and Datagram Packet are used for this process. Datagram Socket is a collection of Datagram packages. Datagram packet is a collection of data, that send through the network.

➢ Understanding Ports

Data transmitted over the Internet is accompanied by addressing information that identifies the computer and the port for which it is destined. The computer is identified by its 32-bit IP address, which IP uses to deliver data to the right computer on the network. Ports are identified by a 16-bit number, which TCP and UDP use to deliver the data to the right application.

In connection-based communication such as TCP, a server application binds a socket to a specific port number. This has the effect of registering the server with the system to receive all data destined for that port. A client can then rendezvous with the server at the server's port, as illustrated here:



In datagram-based communication such as UDP, the datagram packet contains the port number of its destination and UDP routes the packet to the appropriate application.

Port numbers range from 0 to 65,535 because ports are represented by 16-bit numbers. The port numbers ranging from 0 - 1023 are restricted; they are reserved for use by well-known services such as HTTP and FTP and other system services. These ports are called well-known ports. Our applications should not attempt to bind to them.

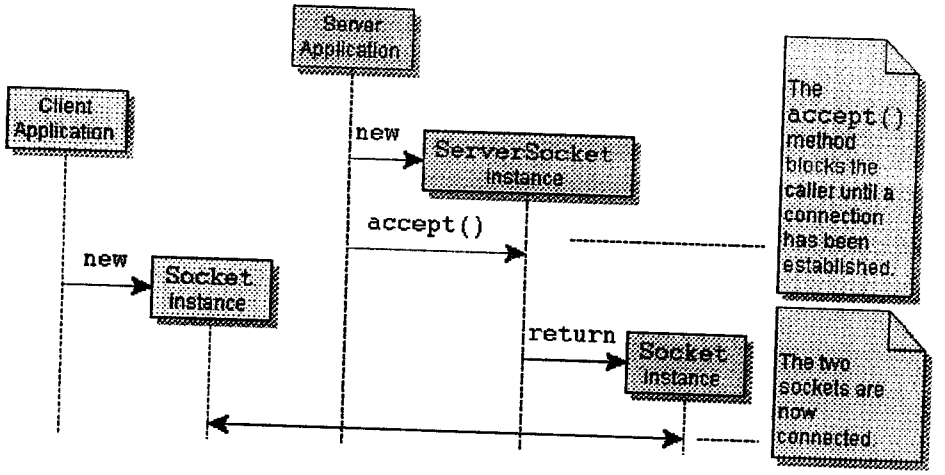
➢ Making TCP Connections

These classes are related to making normal TCP connections:

- ServerSocket class
- Socket class

For simple connections between a client and a server, ServerSocket and Socket are all that we probably need.

ServerSocket represents the socket on a server that waits and listens for requests for service from a client. Socket represents the endpoints for communication between a server and a client. When a server gets a request for service, it creates a Socket for communication with the client and continues to listen for other requests on the ServerSocket. The client also creates a Socket for communication with the server. The sequence is shown below:

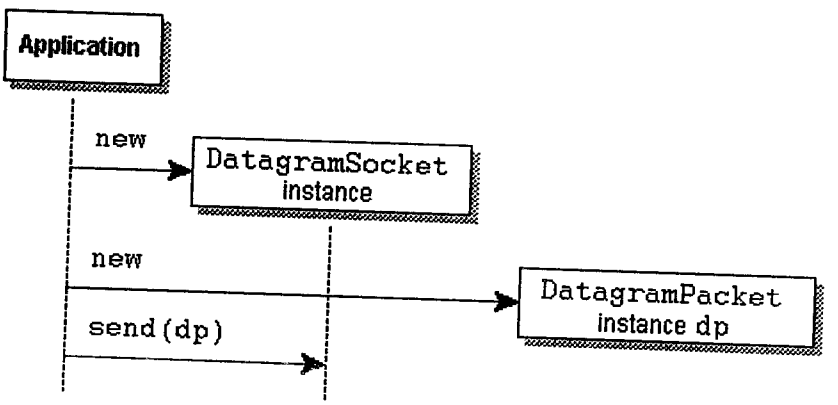


Once the connection is established, getInputStream() and getOutputSteam() may be used in communication between the sockets.

➢ Sending/Receiving Datagram Packets via UDP

The following are related to sending and receiving datagram packets via UDP.

- DatagramPacket
- DatagramSocket



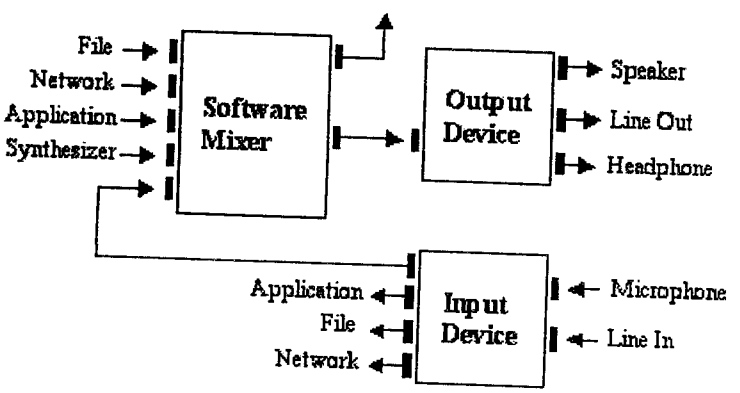
➢ Sampled Audio

The javax.sound.sampled package handles digital audio data, which the Java Sound API refers to as sampled audio. *Samples* are successive snapshots of a signal. In the case of audio, the signal is a sound wave. A microphone converts the acoustic signal into a corresponding analog electrical signal, and an analog-to-digital converter transforms that analog signal into a sampled digital form.

The term "sampled audio" is used here slightly loosely. A sound wave could be sampled at discrete intervals while being left in an analog form. For purposes of the Java Sound API, however, "sampled audio" is equivalent to "digital audio."

➢ Audio Configurations

The Java Sound API does not assume a specific audio hardware configuration; it is designed to allow different sorts of audio components to be installed on a system and accessed by the API. The Java Sound API supports common functionality such as input and output from a sound card (for example, for recording and playback of sound files) as well as mixing of multiple streams of audio.



A Typical Audio Architecture

In this example, a device such as a sound card has various input and output ports, and mixing is provided in the software. The mixer might receive data that has been read from a file, streamed from a network, generated on the fly by an application program, or produced by a MIDI synthesizer. (The javax.sound.midi package, discussed next, supplies a Java language interface for synthesizers.) The mixer combines all its audio inputs into a single stream, which can be sent to an output device for rendering.

➢ Formatted Audio Data

Formatted audio data refers to sound in any of a number of standard formats. The Java Sound API distinguishes between data formats and file formats.

◆ Data Formats

A data format tells you how to interpret a series of bytes of "raw" sampled audio data, such as samples that have already been read from a sound file, or samples that have been captured from the microphone input. You might need to know, for example, how many bits constitute one sample (the representation of the shortest instant of sound), and similarly you might need to know the sound's sample rate (how fast the samples are supposed to follow one another). When setting up for playback or capture, you specify the data format of the sound you are capturing or playing.

In the Java Sound API, a data format is represented by an AudioFormat object, which includes the following attributes:

- Encoding technique, usually pulse code modulation (PCM)
- Number of channels (1 for mono, 2 for stereo, etc.)
- Sample rate (number of samples per second, per channel)
- Number of bits per sample (per channel)
- Frame rate

- Frame size in bytes
- Byte order (big-endian or little-endian)

PCM is one kind of encoding of the sound waveform. The Java Sound API includes two PCM encodings that use linear quantization of amplitude, and signed or unsigned integer values. Linear quantization means that the number stored in each sample is directly proportional (except for any distortion) to the original sound pressure at that instant-and similarly proportional to the displacement of a loudspeaker or eardrum that is vibrating with the sound at that instant. Compact discs, for example, use linear PCM-encoded sound. Mu-law encoding and a-law encoding are common nonlinear encodings that provide a more compressed version of the audio data; these encodings are typically used for telephony or recordings of speech.

A nonlinear encoding maps the original sound's amplitude to the stored value using a nonlinear function, which can be designed to give more amplitude resolution to quiet sounds than to loud sounds.

A frame contains the data for all channels at a particular time. For PCM-encoded data, the frame is simply the set of simultaneous samples in all channels, for a given instant in time, without any additional information. In this case, the frame rate is equal to the sample rate, and the frame size in bytes is the number of channels multiplied by the sample size in bits, divided by the number of bits in a byte.

➢ Line

A line is an element of the digital audio "pipeline"-that is, a path for moving audio into or out of the system. Usually the line is a path into or out of a mixer (although technically the mixer itself is also a kind of line).

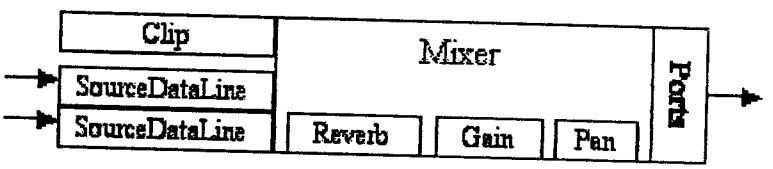
Audio input and output ports are lines. These are analogous to the microphones and speakers connected to a physical mixing console.

Another kind of line is a data path through which an application program can get input audio from, or send output audio to, a mixer. These data paths are analogous to the tracks of the multitrack recorder connected to the physical mixing console.

One difference between lines in the Java Sound API and those of a physical mixer is that the audio data flowing through a line in the Java Sound API can be mono or multichannel (for example, stereo). By contrast, each of a physical mixer's inputs and outputs is typically a single channel of sound. To get two or more channels of output from the physical mixer, two or more physical outputs are normally used (at least in the case of analog sound; a digital output jack is often multichannel). In the Java Sound API, the number of channels in a line is specified by the AudioFormat of the data that is currently flowing through the line.

✓ Lines in an Audio-output Configuration

Let's now examine some specific kinds of lines and mixers. The following diagram shows different types of lines in a simple audio-output system that could be part of an implementation of the Java Sound API:



Configuration of Lines for Audio Output

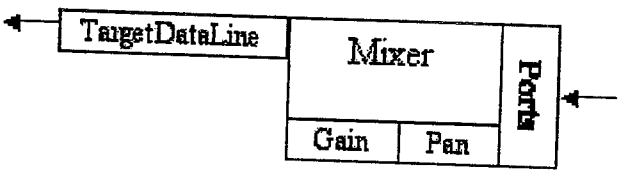
In this example, an application program has gotten access to some available inputs of an audio-input mixer: one or more clips and source data lines. A clip is a mixer input (a kind of line) into which you can load audio data prior to playback; a source data line is a mixer input that accepts a real-time stream of audio data. The application program preloads audio data from a sound file into the clips. It then pushes other audio data into

the source data lines, a buffer at a time. The mixer reads data from all these lines, each of which may have its own reverberation, gain, and pan controls, and mixes the dry audio signals with the wet (reverberated) mix. The mixer delivers its final output to one or more output ports, such as a speaker, a headphone jack, and a line-out jack.

Note that this is just one example of a possible mixer that is supported by the API. Not all audio configurations will have all the features illustrated. An individual source data line might not support panning, a mixer might not implement reverb, and so on.

✓ Lines in an Audio-input Configuration

A simple audio-input system might be similar:



Configuration of Lines for Audio Input

Here, data flows into the mixer from one or more input ports, commonly the microphone or the line-in jack. Gain and pan are applied, and the mixer delivers the captured data to an application program via the mixer's target data line. A target data line is a mixer output, containing the mixture of the streamed input sounds. The simplest mixer has just one target data line, but some mixers can deliver captured data to multiple target data lines simultaneously.

➢ FirstSQL/J

FirstSQL/J is a full Relational DataBase Management System (RDBMS), compliant with ANSI SQL/92, and it is also an Object DataBase Management System (ODBMS), providing extensive object capabilities through Java. FirstSQL/J seamlessly integrates the Object and Relational facilities into a complete Object/Relational DataBase Management System (ORDBMS).

Standard Java DataBase Connectivity (JDBC) is the primary client interface for FirstSQL/J. JDBC Classes provide access to the SQL and relational facilities and to the object capabilities of FirstSQL/J.

The primary physical entity in FirstSQL/J is the database. A FirstSQL/J Database contains all system and user information. The internal structure of a Database consists of a set of logical entities.

Three top level components comprise the logical structure of the database:

- a set of users
- a set of catalogs
- definition_schema -- system schema for database definitions

The user set defines external agents with authorized access to the database. A user has a name, a set of administrative privileges and an optional password.

The catalogue set contains user data in the database. Each catalog has a name and contains a set of user schemas. Each schema contains user data in the form of tables, views, classes, indexes and SQL privileges.

The schema -- definition_schema, contains descriptions of the logical elements of the database -- users, catalogs and their contents (schemas, tables, classes ...) and the system tables and indexes in definition_schema.

The ORDBMS maintains the data in the definition_schema, but it is available for read-only access by user sessions. The set of catalogs for a database is defined when the database is built. Other elements -- users, schemas, tables and classes are created with DDL commands.

For implementation of Reminder System and Address book for the benefit of end users,. Database known as "FirstSQLJ" will be utilized. The advantage of using "FirstSQLJ database" is as below:

- It is completely developed in JAVA.
- It can be embedded into the application, which makes database portable along with the application.
- The functionality is equivalent to any RDBMS

Chapter 4

SYSTEM DESIGN

4. SYSTEM DESIGN

➢ SYSTEM DESIGN

Reviewing the study phase activities and making decisions about which functions are to be performed by the hardware, software and Human ware started design phase. The output, input and file design for each of the programs was done. Finally, the generalized system was explained to the management to approval.

The steps involved in designing phase were:

1. The function to be performed is identified
2. The input, output and file design is performed
3. The system and component cost requirements is specified

➢ Logical system design

In the Logical System Design the following are done.

1. The specific objective of the design process is clarified
2. The constraint for the designing process is identified
3. The resources are available for designing the system is identified
4. The inputs and outputs of the system are analyzed
5. The data which is to be stored is structured in order to produce the outputs
6. The nature of the processing and procedure related to computers are classified
7. The objectives of the management in analysis and a design which is a compromise between the factors such as cost, Reliability, Accuracy, Security control, Expandability, Availability etc is framed

➢ Physical system design

In the physical system design the following activities are done.

1. A generalized system is framed
2. The input design is framed
3. The output design is framed
4. The file design is prepared
5. The design-phased documentation is prepared

4.1 INPUT DESIGN

Input design or form design consists of designing the screens for accepting the input. The user inputs are collected as screen entries. The screen has been designed in such a way to provide GUI features to the user. The input screens are designed in a way as to control the amount of input required, avoid delay and keep processing simple.

The form layout is designed to be user friendly. Layout labels are made self-explanatory. Common set of entries are grouped into a frame for easy identification. Drop down lists are provided in the case of item selection .The user can choose from the valid data from the list provided for all activities that takes place through the form such as additions ,deletions etc. Input data is validated in the screen itself. Appropriate error message and warnings are displayed for the user's convenience.

The input design is the process of entering data to the system. The input design goal is to enter data to the computer as accurate as possible. Here inputs are designed effectively so that errors made by the operations are minimized. The inputs to the system have been designed in such a way that manual forms and the inputs are coordinated where the data elements are common to the source document and to the input. The input is acceptable and understandable by the users who are using.

The major objective that is done during the input design is:

1. Data is collected from the source.
2. Transfer of data to an input form is done.
3. Data is converted to a computer acceptable form.
4. The converted data is verified
5. Data is checked for its accuracy
6. Data is transmitted to the computer
7. Validation of Input data is done
8. Data corrections are done to eliminate the error

The lists of inputs to be processed are

- For voice communication we are giving the input as voice
- For the Messaging and Fax we are giving the input as text.

4.2 DATABASE DESIGN

The database approach to system design places great emphasis on integration, integrity and independence of data. The master table contains the data that are fixed and do not change frequently. The transaction tables are maintained to record daily transactions. Tables have been normalized to avoid data redundancy. Primary key and foreign key are provided for integrity.

A database is a collection of interrelated data stored with minimum redundancy to serve many users quickly and efficiently. The general objective is to make information access easy and flexible for the user. In Java, we can create our databases and create tables for storing data. Database design is based on several objectives such as controlled redundancy, ease of learning and use, data independence, more information at low cost, accuracy and integrity, recovery from failure, privacy, security and enhanced performance.

Clients

Name	Null	Type
ID	Not Null	Number(10)
IPADDR	Not Null	Varchar(20)
NAME	Not Null	Varchar(25)

Reminder

Name	Null	Type
ID	Not Null	Number(10)
MSG	-	Varchar(500)
MSGDATE	-	Times stamp
STATUS	-	Char(1)
PRIORITY	-	Number(10)

Remindertime

Name	Null	Type
ID	Not Null	Number(10)
MSGID	-	Number(10)
RTIME	-	Varchar(15)

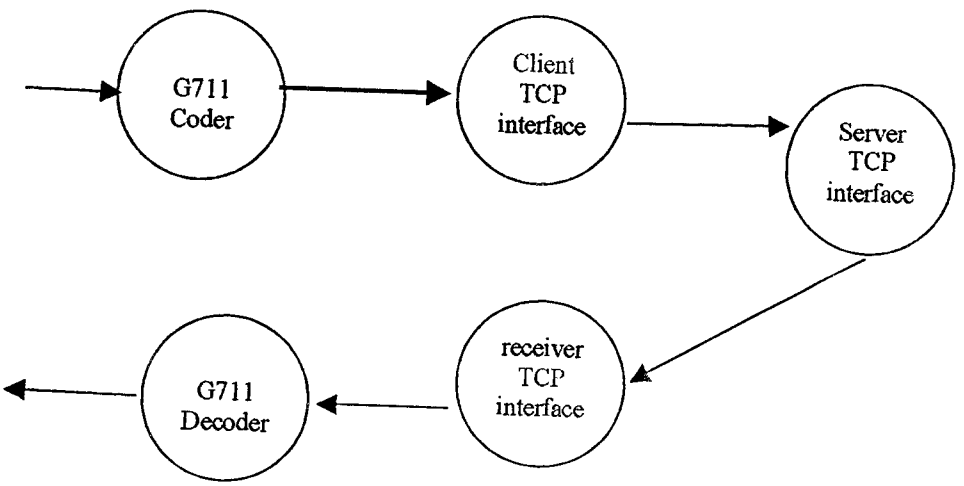
4.3 PROCESS DESIGN

The Process Design for CCSOIP is enlisted below:

- Implementation of messaging system in the form of messages between the PC's that are PEERS of the corporate network.
- The implementation of voice communication between the PC's that are part of the network of the corporate.
- Implementation of transfer of different information that are in the form of various files which forms the functionality of fax between the PEERS.
- Building the address book that helps out the users of CCSOIP for their utilization of the same in the process of communication.
- Building the remainder system for the end users of CCSOIP.
- Building the graphical user interface for integration of all the features as listed above and to provide the end user the user friendliness for the utilization.

4.4 DATA FLOW DIAGRAM

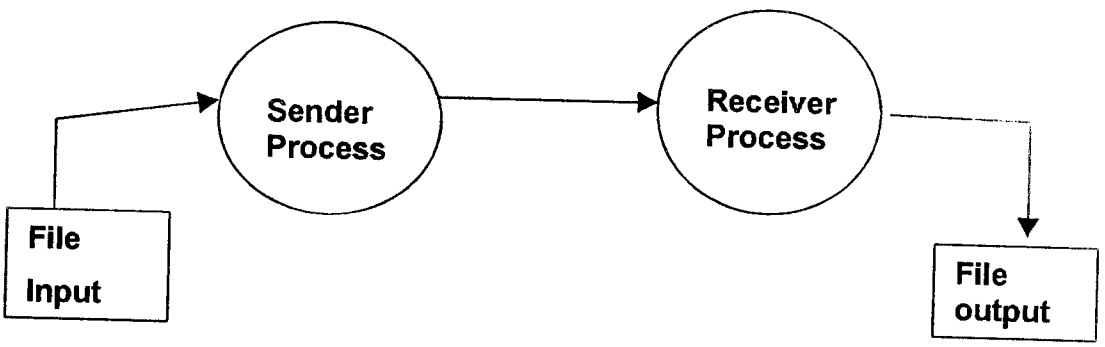
DFD for Voice Communication



Data Dictionary:

1. Speech from the external devices
2. Converted PCM data frames
3. PCM data packets in UDP network with destination address
4. Speech packets with source and destination address in network
5. PCM data frames at receiver end

DFD for File Transfer



Data dictionary

File Data: Data from Input File.

File Data Packets: Data from file in UDP packets format.

File Data2: Data obtained at the receiving end to written to a file in the format used at the transmitting end.

Chapter

5

System Implementation and Testing

5. SYSTEM IMPLEMENTATION AND TESTING

5.1 SYSTEM IMPLEMENTATION

Implementation is the state in the project where the theoretical design is turned a working system. The most crucial stage in achieving a new successful system and in confidence on the new system for the user that it will work efficiently and effectively.

The system can be implemented only after thorough testing is done and if it is found to work to the specifications. It involves careful planning, investigation of the current system and its constraints on implementation. Two major test of preparing the implementation are Education and Training of users and testing the system.

More complex the system being implemented, the more Involved will be the system analysis and design effect required for implementation.

A crucial phase in the system life cycle is the successful implementation of the new system design. Implementation includes all those activities that take place to convert from the old system to the new one. The new system may be completely new, replacing an existing manual or automated system or it may be major implementation becomes necessary so that a reliable system based on the requirements of the, organization can be provided. Successful implementation may not guarantee improvement in the organization using the new system, but improper installation will prevent it. It has been observed that even the best system cannot show good result if the analysts managing the implementation do not attend to every important detail. This is an area where the systems analysts need to work with utmost care.

This unit discusses the three aspects of implementation:

- Training personnel
- Conversion procedures

- Post-implementation review

In each area, the particular elements of that aspect are discussed, along with the methods of handling each aspect efficiently and effectively.

After the system is implemented and conversion is complete, a review should be conducted to determine whether the system is meeting expectations and where improvements are needed. A post implementation review measures the system's performance against predefined requirements. It determines how well the system continues to meet performance specification. It also provides information to determine whether major re-design or modification is required.

A post implementation review is an evaluation of a system in terms of the extent to which the system accomplishes stated objectives and actual project costs exceed initial estimates. It is usually a review of major problems that need converting and those that surface / during the implementation phase.

The post implementation study begins with the review team, which gathers and reviews request for evaluation. Unexpected change in the system that affects the user or system performance is a primary factor that prompts system review. Once request is filed, the user is asked how well the system is functioning to specifications or how well the measured benefits have been realized. Suggestions regarding changes and improvements are also asked for.

5.2 SYSTEM TESTING

Software testing is a critical element of the software quality assurance and represents the ultimate review of specification, design and coding.

The test plan is summarized as follows....

Phase 1: Unit Testing

Phase 2: Integration Testing

Phase 3: GUI Testing

Phase 1: Unit Testing:

In CCSOIP there are three modules available.

- Voice communication
- Messaging
- File transfer

✓ Voice Communication

Testing the voice communication module includes the following process.

- Testing whether the client send voice as packets to server
- Testing whether the server receives the voice.
- Testing whether the proper acknowledgement received by client.

Input: Sending voice through Headphone.

Expected Output: Response from the End user.

Actual Output: Response from the End user

Test Result: Pass

✓ Messaging

Testing the Messaging module includes the following process.

- Testing whether the client send data to server
- Testing whether the server receives data.
- Testing whether the proper acknowledgement received by client.

Input: Message and IP address

Expected Output: Response from end user.

Actual Output: Response from end user.

Test Result: Pass

✓ File Transfer:

Testing the File transfer module includes the following process.

- Testing whether the client send files to server
- Testing whether the server receives the files.
- Testing whether the proper acknowledgement received by client.

Input: Files and IP address.

Expected Output: Accepting or rejecting the file by End user.

Actual Output: Accepting or rejecting message from End user.

Test Result: Pass

Phase 2: Integration Testing:

It is testing the subsystems as a whole, Units are assumed to had been tested and passed. Individual programs may work correctly in isolation but may not work when integrated, so integration testing is inevitable.

In this system the three modules and GUI module are integrated and tested the integrity.

Input: Options for sending the voice and messaging details.

Expected Output: Proper warning message should be displayed.

Actual Output: Proper warning messages are displayed.

Phase 3: GUI Testing:

As one of the main objectives of this system is efficient GUI the interfacing units need be tested for bugs.

The following are the tests performed and verified for their correctness.

- Will the windows open properly based on related menu based commands?
- Is the Active dialog box is properly highlighted?
- Does the dialog box properly close?
- Are all the menu functions and pull-down and sub-functions listed?
- Is alphanumeric data entry properly encoded and input to the system.
- Is invalid data properly recognized?
- Are data input message intelligible?

Chapter 6

CONCLUSION

6. CONCLUSION

Corporate Communication System Over IP is effective and efficient tool. It is a user-friendly tool to communicate with the peers in network.
The goals achieved by the software are,

- ✓ Simplifies the operations
- ✓ Less processing time and increased speed is available
- ✓ Avoiding errors by minimizing human interaction through user-friendly screens when entering data and inquiring the information from the table
- ✓ Further expansion is also possible

The system developed is expected to eradicate all kinds of errors and work efficiently and effectively. This system reduces clerical work, which is very vital for the progress of an organization. It is highly interactive and is user friendly as the menu enables it. The system provides accurate updating data validation and integrity is observed in the system.

The system is designed such that it can submit more reports to the management in order to get a clear picture of the operational activities of the section. The management can henceforth decide on various measures to improve the functioning of the section as well as organization.

The system is designed in such an extendable fashion to incorporate the future changes into the system easily. The various user friendly features are introduced in this project.

The operation required by the user to operate the system is only basic computer knowledge.

Chapter 7

SCOPE FOR FUTURE DEVELOPMENT

7. SCOPE FOR FUTURE DEVELOPMENT

- ✓ In order to reduce bandwidth; CCSOIP can use ADPCM compression to compress speech. ADPCM can reduce the CCSOIP bandwidth from 128-Kbits/sec to 32-Kbits/sec.

- ✓ A complete rewrite to include a better Voice Activation Detector, a better voice codec with high compression ratio.

- ✓ Support for H.323. This will allow CCSOIP to talk to Linux based voice applications.

- ✓ Use of H.323 based codec's such as G711, G723, and/or G729a, etc. Using these vocoders will ensure interpretability with H.323 compliant software and also provide us with better voice compression than what can be achieved with the IMA ADPCM codec.

And anything else, which one can think off....

Chapter 8

BIBLIOGRAPHY

8. BIBLIOGRAPHY

Books:

- E.Balagurusamy, 'Object-Oriented Programming', Tata McGraw Hill Publication, 1999, Page No 1-25, 97-114.

- Reference for Comments on INTERNET, RFC No 0793 (TCP Protocol).

- Grady Booch, 'Object Oriented Analysis and Design', the Benjamin/Gum Publishing Company, Second Edition, 1994, topic-class diagram.

- Robert Schildt, 'Java 2 Complete Refrence', Eastern Economy Edition, Fifth Edition, 2003, Page No 1-385.

Web Sites:

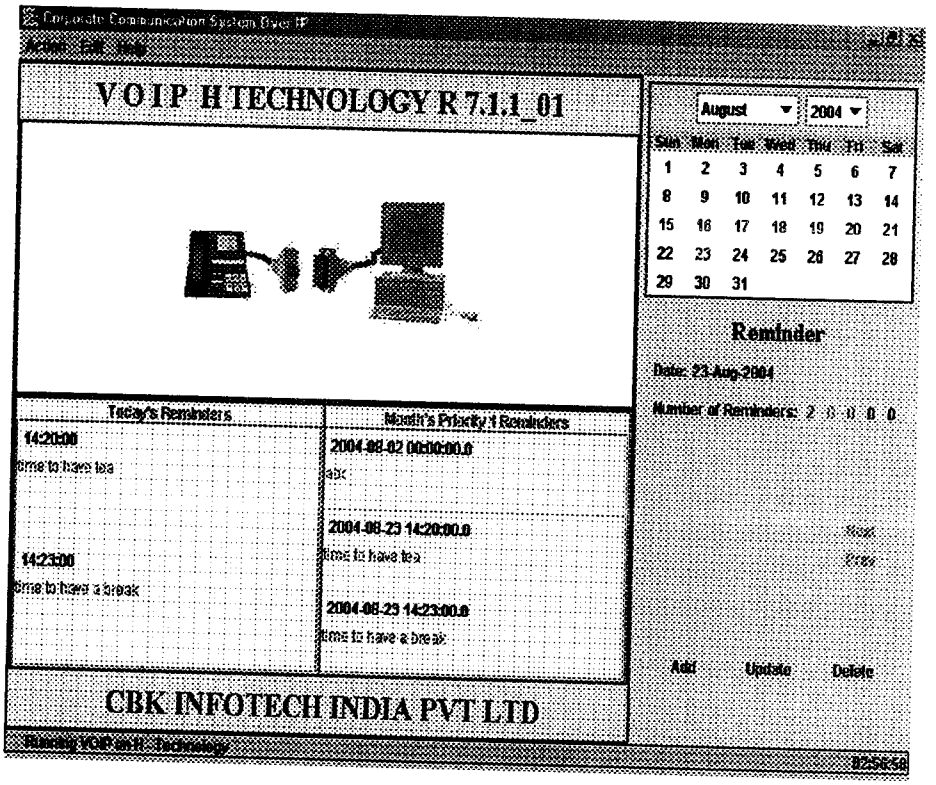
- www.firstsql.com
- www.google.com
- www.Java.sun.com

Chapter 9

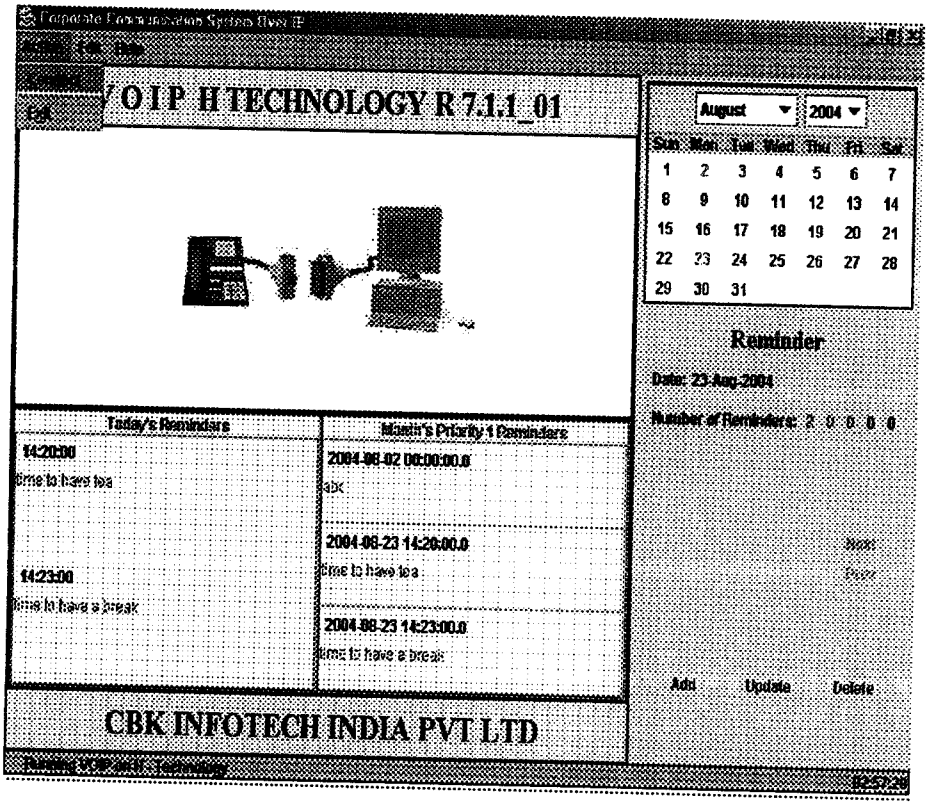
9. APPENDIX

9.1 SAMPLE SCREENS

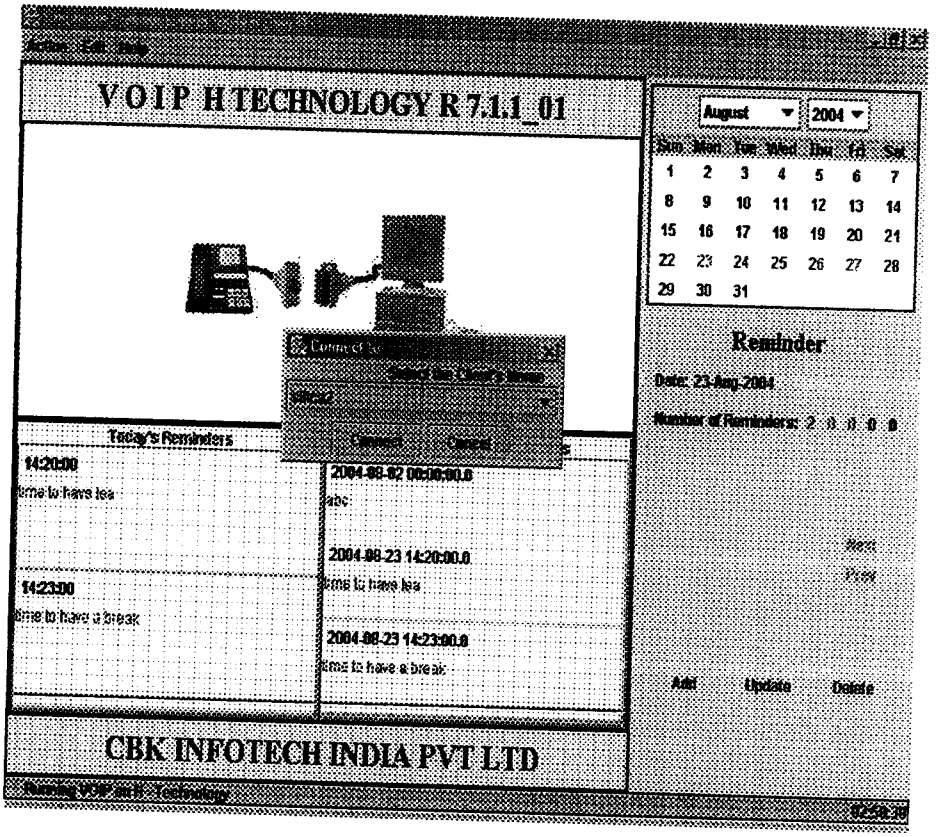
Main window



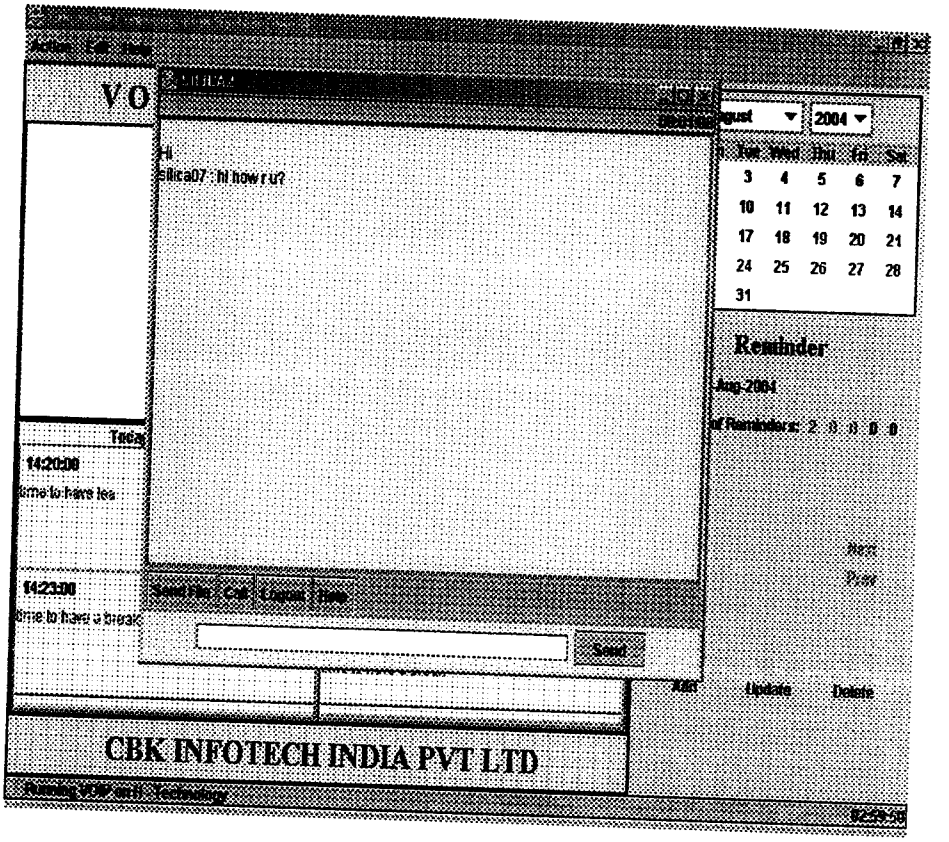
Connect to client



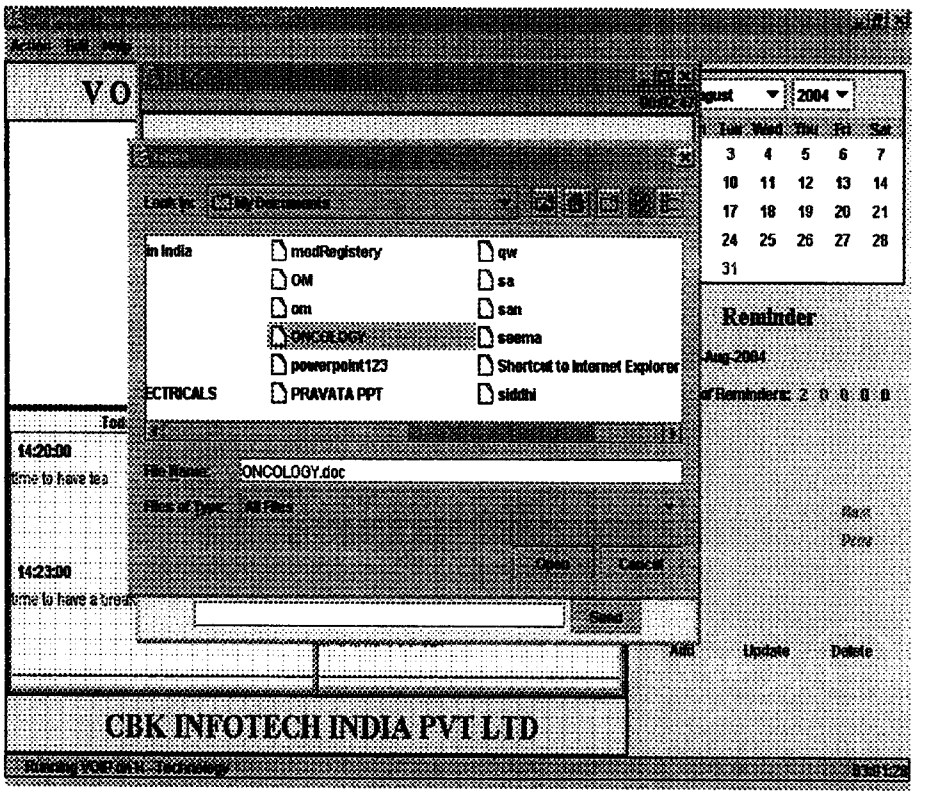
Selecting Client Name



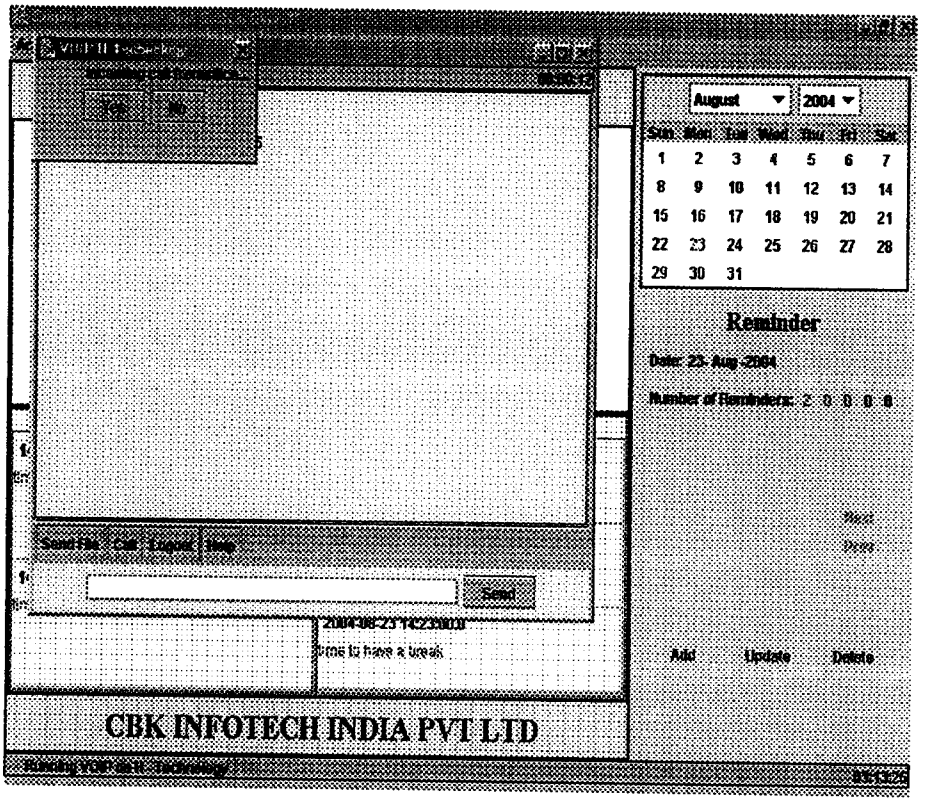
Message Window



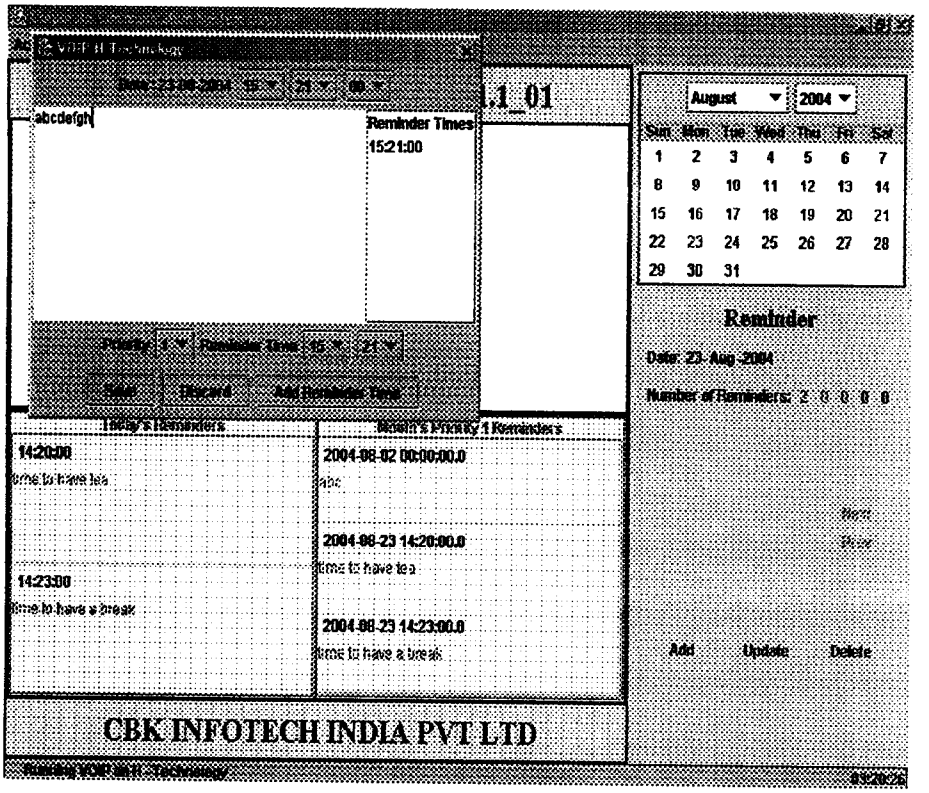
Sending a File



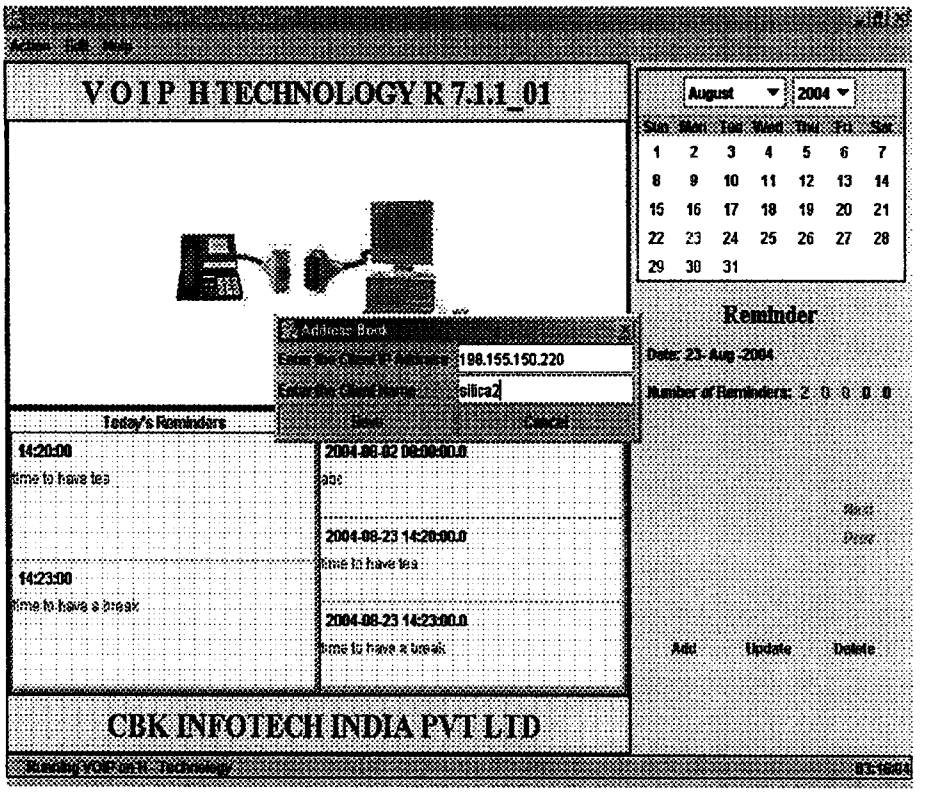
Incoming Call from Client



Reminder System



Adding a New Client



9.2 SAMPLE CODE

```
//Desigining the main panel
import javax.swing.*;
import java.awt.event.*;
import java.awt.*;
import javax.swing.border.*;
public class  MainPanel extends JPanel {
   ClientDisplay clients;
        ReminderDisplay reminder;
        public MainPanel() {
                super(new BorderLayout());
                Image logo =
Toolkit.getDefaultToolkit().getImage("Cbk.jpg");
                ImageIcon logoicon = new ImageIcon(logo);
                JLabel logolabel = new JLabel(logoicon);
               JLabel label = new JLabel("CCSOIP Beta Version-1.0");
              label.setFont(new Font("TimesRoman",Font.BOLD,25));
                label.setForeground(Color.blue);
      JPanel dummy = new JPanel();
                 dummy.setBackground(new Color(215,255,255));
                 dummy.add(logolabel);
                 dummy.add(label);
                add(dummy,"South");
         label = new JLabel("VOIP  H TECHNOLOGY R 7.1.1_01");
        label.setFont(new Font("TimesRoman",Font.BOLD,25));
        label.setForeground(Color.blue);
```

```
        dummy = new JPanel();
                dummy.setBackground(new Color(226,250,236));
                dummy.add(label);
                add(dummy,"North");
                dummy = new JPanel(new GridLayout(2,1,0,0));
                clients = new ClientDisplay();
                dummy.add(clients);
                reminder = new ReminderDisplay();
                dummy.add(reminder);
                add(dummy,"Center");
                setBackground(Color.white);
                setBorder(new LineBorder(Color.blue,2));
                reminder.start();
        }
}
```