WeCONNECT

A project submitted in partial fulfillment of the requirements for the degree
of

Bachelor of Technology
in
Computer Science & Engineering

Submitted
by

Bhabna Acharya(1011012221)

Lipsa Mishra(1011017060)

Rajashree Padhi(1011017011)

Sukanya Shivangi Mishra(1011019019)

Under the Guidance of

Mrs. Sipra Sahoo

Department of Computer Science & Engineering
Institute of Technical Education and Research, Bhubaneswar
(Siksha 'O' Anusandhan University, Odisha)

(April, 2014)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

INSTITUTE OF TECHNICAL EDUCATION & RESEARCH
(Under Sikhsha 'O' Anusandhan University, Odisha)
Jagamohan Nagar, Jagamara, Khandagiri, Bhubaneswar- 751030

Declaration

We hereby declare that the matter embodied in this project report is original and has not been submitted for the award of any other degree.

Submitted By:
Bhabna Acharya(1011012221)
Lipsa Mishra(1011017060)
Rajashree Padhi(1011017011)
Sukanya Shivangi Mishra(1011019019)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

INSTITUTE OF TECHNICAL EDUCATION & RESEARCH
(Under Sikhsha 'O' Anusandhan University, Odisha)
Jagamohan Nagar, Jagamara, Khandagiri, Bhubaneswar- 751030

CERTIFICATE

This is to certify that the project work entitled ”**WeCONNECT**” carried out by Bhabna Acharya (1011012221), Lipsa Mishra (1011017060), Rajashree Padhi (1011017011), Sukanya Shivangi Mishra (1011019019), students of 8th semester B.Tech in Computer Science and Engineering from Institute of Technical Education and Research, Bhubaneswar, under my guidance has been completed by them and worthy of acceptance for the degree of Bachelor of Technology in Computer Science and Engineering, under Siksha O Anusandhan University, Bhubaneswar, Odisha.

Prof.(Dr.)B.K.Pattanayak
Head of the Department
Assiciate Professor and HOD
Department of Computer Science and Engineering
Institute of Technical Education and Research,
Bhubaneswar, Odisha

Mrs. Sipra Sahoo
GUIDE
Assistant Professor
Department of Computer Science and Engineering
Institute of Technical Education and Research,
Bhubaneswar, Odisha

ACKNOWLEDGEMENT

We hereby express our deepest gratitude towards our **HOD.,B.K.Pattanayak**(Department of Computer Science and Engineering) for his guidance throughout this project, and our guide, Assistant Professor, **Mrs. Sipra Sahoo**, for her help and guidance throughout our project. Without their help and support we could not have completed our project on time.

We also thank **Mrs. Sashikala Mishra,** our project co-ordinator for her constant support and help.

Submitted By:
Bhabna Acharya(1011012221)
Lipsa Mishra(1011017060)
Rajashree Padhi(1011017011)
Sukanya Shivangi Mishra(1011019019)

Abstract

It is not economical and practical in this high-tech century that the students and professors of our university's various departments spread over a wide range of locations meet together for sessions, seminars, personal developments programs etc. after extensive travelling as well as wasting of valuable times. So we need to have solution for uniting students irrespective of locations, travelling, time-limits. Our solution must provide an enriched environment for collaboration and discussion among the students and faculty. This project will be based on a client-server type of application which helps in communication between students and faculties within our university. This application also helps to conduct seminars, meetings and other types of group discussions between the faculties and the students of the same college or between different colleges of the university. The person who is from one network who desires to chat or hold a video conference with another person belonging to the other network, then he has to send a request to the server. Then the server accepts the request and a successful chat conversation can be held. In the same way the video conferencing will be performed. Java is used for programming purposes in this project and is a standard application providing a host of benefits. It possesses a higher level of programming than other languages. It is also well known for its security features.The softwares we are going to use are NetBeans IDE 7.1.2 for writing the java codes, JM-Studio for transmitting and receiving the audio/video files, Many Cam for capturing the audio/video files, Dia software for drawing the ER-Diagrams and Usecase Diagrams, Oracle 10g for the database purpose and Winedit for writing the codes in latex and making the report.

Contents

List of Figures

Chapter 1

Introduction

1.1 Introduction

This project is based on having live video chat and voice chat among the faculties and students of a particular college. The at most priority must be given to the definition of the problem statement of the project to be undertaken. So the project is a set of file related with each other developed by the creators of the system to accomplish a specific purpose. Our purpose is to design project for **VOICE CHATTING and VIDEO CONFERENCING**.

As the name suggests, our project is based on audio/video transmission and reception. Through our application two or more persons in an intranet can Chat with one another and they can have Video Conferencing also.It is a Client-Server type application in which the Server handles all the traffic. The person (from one of the computer in the network) who wants to have chat or conferencing with another person requests to Server and after acceptance of request they can have successful chat or conferencing. The Server (which is a person indeed) can also have Voice Chatting or Conferencing with the clients.With the growth of Internet there has been a radical change in the method of software design and deployment. Software applications are becoming more and more distributed. Video conferencing is still relatively new for the Internet, but its popularity is growingquickly. It is a full-motion, two-way, video/audio system that permits two or more people in different locations to communicate with each other. Two-way video conferencing is often used for large groups and by colleges and universities that offer video courses.

Video conferencing or video communications is the ability to communicate with other people as if they were in the same room. For video conferencing to really succeed, participants need to be able to see, hear and use meeting tools regardless of whether participants are in the same room or across the other side of the world. Video conferencing is defined as (by www.whatis.com): a live connection between people in separate locations for the purpose of communication, usually involving audio and often text as well as video. At its simplest, video conferencing provides transmission of static images and text between two locations. At its most sophisticated, it provides transmission of full-motion video images and high-quality audio between multiple locations.

This project will be based on a client-server type of application which helps in communication between students and faculties within our university. This application also helps to conduct seminars and meetings.Each time an user is registered the application and user details remains pending till the ADMIN approves the information. And the user details are stored in the database only after the ADMIN approves it. We work in modules. The person who is from one network who desires to chat with another person from other network, sends request to the server. After accepting the request from the other person , one can hold video and audio conferencing and can even do instant messaging. For every audio and video conference the request will be sent to the admin. There will be only one presenter at one moment. Others can just be the receivers.

The receiver address will be connected through the ip address of each pc and the transmission will be done through a port address. The port address will be provided by the server. And after the presentation is over the presenter can stop transmitting video.

Our application is programmed in Java programming language. The other tools that weve used to build our application are JDK1.6 (Java Development Kit) ,JMF 2.0 (Java Media Framework) and RTP (Real-time transport Protocol).

JMF is a package that is used to develop softwares related to audio and video. It enables to capture media data (audio/video) and to transmit to target device. RTP is the protocol designed to handle Real-Time traffic on the intranet/internet that lies between UDP and application program used with UDP.

1.2 History

Video conferencing, as a technology, has been around for approximately 40-50 years. However, the growth of video conferencing has depended heavily on the availability to run on a reliable digital communications network. It wasnt until the early 1990s that ISDN standards were introduced and, finally, video conferencing could begin to grow. Systems in the mid to late 1990s were expensive and only really a technology that larger multinational companies could afford and then only at the higher levels in the organisation. The decrease in endpoint price, increase in quality and functionality as well as global events, such as the Gulf wars, increased terrorist activity and, more recently, climate change concerns have fuelled the growth of video conferencing. In addition, just as the standardisation of ISDN networks fuelled the initial adoption in the early 1990s, the ability to run video conferencing over computer data networks has also fuelled growth in the past three years or so.

1.3 Why Use Video Conferencing?

Video conferencing can speed up business processes and procedures in the same way that e-mail has revolutionised the way we share information. The most common reason for implementing video conferencing is to save travel costs. However other benefits are often more important. First, it isnt just the cost of travel but the cost of the time taken to travel. Travel also causes wear and tear on an individual which reduces their effectiveness. Business travel isnt going to disappear completely but video conferencing can significantly reduce it and really make a difference to productivity.Conferences can exist in several forms. At its simplest, an audio conference is a connection between two individuals using a standard telephone. More complicated, it can involve numerous sites all connected with video, audio and data, all on different networks and speeds. On top of the time saving benefits there are also significant benefits in changing the way we do business. Video conferencing greatly improves communication between remote sites both within a company and between suppliers and customers. Product or project development times can be decreased and easily involve experts where ever they are. This can increase profit and the quality of the end result.

Meetings are often more effective over video and can be held more often. This, in turn, enables companies to make decisions within smaller time frames solving urgent problems more quickly and also enabling companies to react to market changes faster. Video conferencing has personal benefits too. Less time away from the family is one benefit but also when you do have to travel you can use video conferencing to communicate back home as well.

1.4 Video conference components

→ Display
→ Cameras
→ Microphone
→ Amplifier
→ Speakers
→ Echo cancellation
→ Networks
→ User interface Cables
→ Peripheral equipment

In order to understand how video conferencing works it is important to recognize the component parts of the system. The course will look closely at the following areas:

The display for example a plasma screen
→ Cameras
→ Microphones
→ Sound mixers
→ Speakers
→ Echo cancellation
→ The networks used to carry video conference traffic

The user interfaces for example the remote control and the web interface Cables Peripheral equipment for example a laptop or second camera and the Codec which is the main brain of the system.
A monitor of some sort is needed to display the far end picture. There are four main types:

- Plasma screens
- LCDs (Liquid Crystal Display)
- Projectors
- CRT (Cathode Ray Tube) what most of us would refer to as a TV.

1.5 Which components to Use?

Plasma:

- **PROS:** screen's phosphor coating creates lifelike color .
- **CONS:** vulnerable to burn in

LCD:

- **PROS:** panels weigh less than plasma and use less energy
- **CONS:** picture slightly less natural than top plasmas; limited size

TV:

- **PROS:** cheap and excellent image quality
- **CONS:** heavy, bulky, no HD support

Projector:

- **PROS:** large size image, light weight
- **CONS:** camera and projector positioning can be difficult

There are advantages and disadvantages of all 4 types. Plasma screens look professional, have larger screen sizes and take up less space. However they can be expensive, are very heavy which you need to consider when wall mounting and they can suffer burn in. Burn in occurs when the same image is displayed for long periods of time to the point when the image is visible even when the plasma is off. LCD screens are lighter and cheaper than Plasma screens but the image isn't as good. However when ever we talk about picture quality we should remember that the image will only be as good as the weakest point and this isn't often the display. Like a plasma, an LCD takes up little space and looks professional.TVs or CRT monitors, particularly 100hz versions, have an excellent quality image. However they are bulky and seen as yesterdays technology which they are. They also do not support High Definition formats. Projectors can be a good choice in certain room layouts, particularly where a very large image projection is required. You will have to be careful though about where you position the camera.

Chapter 2

Objective

It is not economical and practical in this high-tech century that the students and professors of our universitys various departments spread over a wide range of locations, meet together for sessions, seminars, personal developments programs etc., after extensive travelling as well as wasting of valuable times. So we need to have solution for uniting students irrespective of locations, travelling, time-limits. Our solution must provide an enriched environment for collaboration and discussion among the students and faculty.

This project will be based on a client-server type of application which helps in communication between students and faculties within our university. This application also helps to conduct seminars,meetings and other types of group discussions between the faculties and the students of the same college or between different colleges of the university. The person who is from one network who desires to chat or hold a video conference with another person belonging to the other network, then he has to send a request to the server. Then the server accepts the request and a successful chat conversation can be held. In the same way the video conferencing will be performed. Java is used for programming purposes in this project and is a standard application providing a host of benefits. It possesses a higher level of programming than other languages. It is also well known for its security features.

This project will be based on a client-server type of application which helps in communication between students and faculties within our university.

The person who is from one network who desires to chat or hold a video conference with another person belonging to the other network, then he has to send a request to the server. Only when the server accepts the request, the audio/video file transmission and receiving can take place successfully.

Application:

- To make application satisfying customer request.
- To provide good user interface.
- To provide better way of communication.

Chapter 3

Requirements and Specification

3.1 Feasibility Analysis

Technical Feasibility:

This project doesnt required advance and higher technology. It requires only knowledge of Core Java and Java Media Framework (JMF).

Economical Feasibility:

The development of project will cost too much as it requires microphone and Web Camera as extra hardware besides computer. It involves very few persons and it does not require any outside professionals. As project is based on java so we donot have any extra cost of setting up a network. The tools that we have used are freeware and other tools and these can be downloaded from internet.

Operational Feasibilty:

This project is a live project. Person using these applications does not require extra technical or computer skill. It can be very easily operated and used.

3.2 Software Required

The software used in the project are:

- NetBeans IDE 7.0.1
- JM Studio
- Many Cam
- Oracle 10g

- Dia software 0.97.2
- basic-Miktex 2.9.5105
- winedit 80

3.2.1 Description

NetBeans IDE 7.0.1:

NetBeans is an integrated development environment (IDE) for developing primarily with Java, but also with other languages, in particular PHP, C/C++, and HTML5. It is also an application platform framework for Java desktop applications and others.The NetBeans IDE is written in Java and can run on Windows,OS X,Linux,Solaris and other platforms supporting a compatible JVM. The NetBeans Platform allows applications to be developed from a set of modular software components called modules. Applications based on the NetBeans Platform (including the NetBeans IDE itself) can be extended by third party developers. The NetBeans Team actively support the product and seek future suggestions from the wider community. Every release is preceded by a time for Community testing and feedback. NetBeans IDE 7.0 was released in April 2011. On August 1, 2011, the NetBeans Team released NetBeans IDE 7.0.1, which has full support for the official release of the Java SE 7 platform.

JM Studio:

JMStudio is a stand alone Java application that uses the JMF 2.0 API to play, capture, transcode, and write media data. JM Studio also uses the JMF RTP APIs to receive and transmit media streams across the network. Before you can run JM Studio, you must have JMF 2.1.1 installed. JM Studio lets you play media streams from a variety of sources–files, URLs, or RTP transmissions. JMStudio enables you to transcode a media stream and write it to a file. Transcoding is the process of converting a media stream or some of its tracks from one media format to another. To transcode and save a media file, URL, or captured media stream with JMStudio.

Many Cam:

ManyCam is a freeware program that allows users to have the ability to use their webcam with multiple chat applications simultaneously, such as Skype, MSN, or Youtube. Users can also add live CGI graphics inside any webcam application, such as changing the appearance of their face, eyes, hair, background and more. ManyCam Pro is a paid upgrade, it provides users with a professional quality live video production studio. There is also a simple module for creating effects included within the software, which mainly consists of adding image files, videos, and converts them to a ManyCam friendly format. Face effects and backgrounds can also be created. The ManyCam website has a database of more than 7,000 webcam video effects, most of which were contributed by users. ManyCam uses a webcam or video camera as input for the software itself and then replicates itself as an alternative source of input. Because of this, ManyCam works well with nearly all chat software such as Windows Live Messenger that can use alternative video sources. On December 10, 2013 version 4.0.44 for Windows was released. This version of the software was a big improvement from previous versions and includes several new features and a new user interface. ManyCam is also available for Mac. The latest version 2.0.38 was released on October 30, 2013. This version includes a new virtual video driver with 64-bit support as well as other bug fixes. On October 2, 2013 ManyCam LLC was acquired by Visicom Media Inc. Visicom Media Inc. was established in 1996 and is a leading developer and pioneer of Internet application technologies.

Oracle 10g:

The Oracle Database, commonly referred to as Oracle RDBMS or simply as Oracle, is an object-relational database management system produced and marketed by Oracle Corporation. Larry Ellison and his friends, former co-workers Bob Miner and Ed Oates, started the consultancy Software Development Laboratories (SDL) in 1977. SDL developed the original version of the Oracle software. The name Oracle comes from the code-name of a CIA-funded project Ellison had worked on while previously employed by Ampex. An Oracle database systemidentified by an alphanumeric system identifier or SID comprises at least one instance of the application, along with data storage. An instanceidentified persistently by an instantiation number comprises a set of operating-system processes and memory-structures that interact with the storage. (Typical processes include PMON (the process monitor) and SMON (the system monitor).) Oracle documentation can refer to an active database instance as a "shared memory realm".Users of Oracle databases refer to the server-side memory-structure as the SGA (System Global Area). The SGA typically holds cache information such as data-buffers, SQL commands, and user information. In addition to storage, the database consists of online redo logs (or logs), which hold transactional history. Processes can in turn archive the online redo logs into archive logs (offline redo logs), which provide the basis (if necessary) for data recovery and for the physical-standby forms of data replication using Oracle Data Guard. If the Oracle database administrator has implemented Oracle RAC (Real Application Clusters), then multiple instances, usually on different servers, attach to a central storage array. This scenario offers advantages such as better performance, scalability and redundancy. However, support becomes more complex, and many sites do not use RAC. In version 10g, grid computing introduced shared resources where an instance can use (for example) CPU resources from another node (computer) in the grid. The Oracle DBMS can store and execute stored procedures and functions within itself. PL/SQL (Oracle Corporation's proprietary procedural extension to SQL), or the object-oriented language Java can invoke such code objects and/or provide the programming structures for writing them.

Dia software:

Dia is free and open source general-purpose diagramming software, developed originally by Alexander Larsson. Dia uses a controlled single document interface (SDI) similar to GIMP and Inkscape. Dia has a modular design with several shape packages available for different needs: flowchart, network diagrams, circuit diagrams, and more. It does not restrict symbols and connectors from various categories from being placed together. Dia has special objects to help draw entity-relationship models, Unified Modeling Language (UML) diagrams, flowcharts, network diagrams, and simple electrical circuits. It is also possible to add support for new shapes by writing simple XML files, using a subset of Scalable Vector Graphics (SVG) to draw the shape.Dia loads and saves diagrams in a custom XML format, which is, by default, gzipped to save space, and can print large diagrams spanning multiple pages. It can also be scripted using the Python programming language.

Basic-Miktex 2.9.5105:

MiKTeX is a typesetting system for Microsoft Windows that is developed by Christian Schenk. It consists of an implementation of TeX and a set of related programs. MiKTeX provides the tools necessary to prepare documents using the TeX/LaTeX markup language, as well as a simple tex editor (TeXworks). The name comes from Christian Schenk's login: MiK for Micro-Kid. MiKTeX can update itself by downloading new versions of previously installed components and packages, and has an easy installation process. Additionally, it can ask users whether they wish to download any packages that have not yet been installed but are requested by the current document. The current version of MiKTeX is 2.9 and is available at the MiKTeX homepage. Since version 2.7, MiKTeX has support for XeTeX, MetaPost and pdfTeX and compatibility with Window

Winedit 80:

WinEdt is a shareware Unicode (UTF-8) editor and shell for Microsoft Windows. It is primarily used for the creation of TeX (or LaTeX) documents, but can also be used to edit HTML or any other type of text file. It can be configured to run as a front-end for a variety of TeX systems, including MiKTeX, fpTeX and TeX Live. WinEdt's highlighting schemes can be customized for different modes and its spell checking functionality supports multi-lingual setups, with dictionaries (word-lists) for many languages available for downloading from WinEdt's Community Site. It supports DVI and PDF workflow.

Chapter 4

Modules

4.1 User Registration

Before you host your first video conference, you must register the sites you commonly connect with. The benefits of registering your site include the following:

- Full site and contact details can be retained for easy access.
- The video meeting components can be checked (certified) in advance to ensure reliable operation.
- Reservations and scheduling can be made for all elements of your video conference, including the room and video meeting components.
- Site facilities can be publicized to assist conference planning.
- Registering a site can be done online or via the phone with a video coordinator.

  You will need to provide:

- user name, password and address details
- Site contact information
- System information (video numbers, video conference equipment details)
- Maximum operating speed (bandwidth)
- port number
- Any special details or needs

→ Certification calls After registration and with prior notification a certification call will be made to the site. These calls are designed to ensure that all site details are correct and that any equipment or network issues are re- solved prior to a business conference.
→ Prior notification of certification calls will normally be by email to the site contact.

→ Joining your conference
→ Set-up time A standard set-up time of 15 minutes is recommended for all conferences. This allows time for all video rooms to be added to the conference prior to the start of the meeting.
→ For more than six sites or for very important conferences, it is recommended that the set-up time be extended to 30 minutes. The set-up time can be varied by agreement when scheduling the conference.

4.1.1 Controlling your conference

Picture display modes:

A variety of options is available for how the pictures from the different sites are displayed on the screens. The mode may be preselected at the time of scheduling the conference. It may also be changed during the conference by contacting the video coordinator.

End-of-conference warning:

If requested at the time of scheduling, warning tones can be added to the call to notify participants 10 minutes prior to the end of the conference.

Extending the conference:

If requested at the time of scheduling, the conference can automatically run over by up to 30 minutes (subject to site availability). A conference can be further extended by calling the video coordinator at the number provided at the start of the conference at least 10 min- utes before the end of the scheduled time. Extension is subject to equipment and site availability. Charges are only made for additional time actually used.

It is possible to start video conferencing in real time without a prior reservation.

The service includes capabilities such as:

- Add video and audio participants (number of participants varies by package)
- Choose the initial video display for your conference and change the mode during the course of the conference .
- Connect and disconnect individual participants without interrupting the conference.
- Select different transmission rates for different sites

The maximum operating bandwidth to each site for a self-serve conference is 384Kbps. Initiating and controlling a self-serve video conference is done online.

4.2 User Authentication

The first thing users do when they log on to an IM network is authenticate themselves to the system. Again, several approaches are possible here, with clear implications for security. Some

IM systems do not go through the full authentication process that is done in other contexts (e.g., SSL/TLS), since both the user and the system share a secret key known only to the two of them: the users name and pass- word. While the initial system sign-up is typically done using HTTP secured by SSL/TLS, once the name and password are decided, login authentication is typically done by exchanging hashes of the shared secret. In this way, the password is never transmitted in the clear over the network, although the user name is. Both AIM and YMSG work this way. The advantage to this approach is that expensive crypto operations are avoided, such as RSA public key or AES shared key encryption. Instead, relatively cheaper authentication algorithms based on MD5 and/or SHA are used. The disadvantage is that confidentiality is not provided; observers can monitor the packet exchanges and determine who has logged in, even if they cannot determine the password. Since the hash algorithms are well known, and the challenge and hash result are sent in the clear, the systems are vulnerable to dictionary attacks. Users must therefore use passwords that are difficult to crack. In addition, performing the exchange in the clear could lead to connection hijacking; for example, AIM uses the cookie returned by the log on server as a credential sent in the clear to the BOS server. This credential must be used within 30 seconds or the connection will be terminated by the BOS server. This suggests that there is a window of opportunity where an adversary could monitor the conversation, capture the cookie, and use it to impersonate the victim to the BOS server.

4.3 Streaming Audio and video

Streaming, or web casting, enables you to deliver high-impact, rich-media messages (including video and audio) and presentations to a wider audience. The technology takes audio and video files and transmits them efficiently over the internet or your corporate intranet. They can be accessed by anyone with a personal computer and a web browser. Streaming conferences can be viewed live or archived for future viewing.

Requirements Personal computer with:

- Internet access
- Sound card
- Speakers
- Real Player or Windows Media Player

4.3.1 Data Transfer

One of the key issues in any IM or chat protocol is how protocol headers and payloads are encoded. The representation of this data can take two forms. Historically, many network protocols have used a binary representation of data in network byte order; examples include TCP and IP. Application-layer protocols such as HTTP and SMTP have tended to use a text- based approach. The main advantage to the binary representation is that it makes most efficient use of space on the network. The advantage of the text-based approaches is that the representation is closer to the way humans view information, and thus debugging is easier.

4.4 Security

Instant messaging may soon become an indispensable business tool; however, the risks of using anunsecured IM platform in corporations are high. This section explores the security issues introduced with the use of corporate instant messaging and offers best practices that can help in deploying asecure IM platform. Video conferences are no less immune to the need for security than any other collection of data we use today. Because secrets and other sensitive information are often part and parcel of video conferencing, security is vitally important. Security is also important because it's now the law. Government regulations for banks, medical providers and many other organizations stipulate that all data pertaining to customers, patients and clients must be kept secure. Military agencies are, of course, are also required to maintain stringent security protocols for electronic transmission, as in video conferencing. Securing a video conference has much to do with making the material invulnerable to theft or electronic eavesdropping by third parties, as the video and audio shuttle along the many networks they often travel. (How the conference is securely stored after the program concludes is a different question.) As a result, data transmission encryption – effectively scrambling the conference in transit – is crucial to the security of the material. Using the encryption that comes standard with your video conferencing application usually provides enough security, but for highly sensitive data, such as military information, special encryption protocols are probably needed. Before we start a video the admin needs to approve the presenter,the contents to be transferred and the time of video conference. When there is an user registration the details remain pending till the admin approves them. Then stored in database for further queries and receiving or presenting the video.

Chapter 5

Design and Implementation

5.1 Why Java Platform

Java is ideally suited to become the standard application development language for wireless devices, providing us with lots of benefits.
Here are some of the most important ones given below:

Cross platform compatibility:

The Java application can easily transfer between different devices and different platforms as long as the JVM has been developed for those devices. The java programmes can be run in any environment.

Object Oriented Programming:

Java has a better abstraction mechanisms and higher level programming constructs than C++. Object-oriented programming (OOP) is a programming paradigm that represents concepts as ”objects” that have data fields (attributes that describe the object) and associated procedures known as methods. Objects, which are usually instances of classes, are used to interact with one another to design applications and computer programs.

Huge java developer community:

Java has become the most popular programming language taught in schools and universities.

Security:

Java is known for its security features (class file verification, cryptography possibilities etc.). The

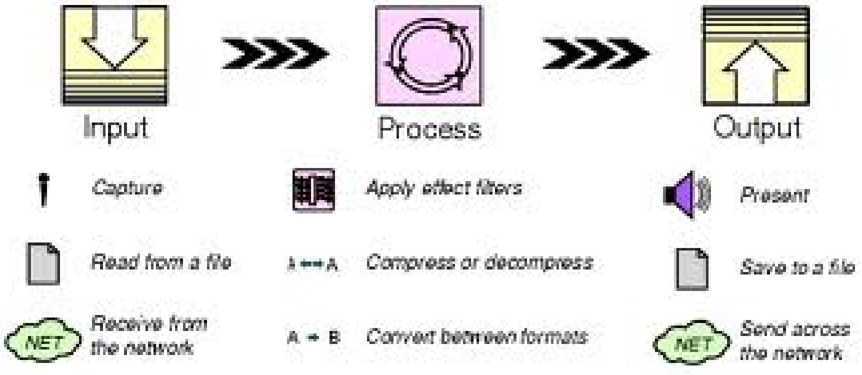


Figure 5.1: Media processing model

Java platform provides a number of features designed to improve the security of Java applications. This includes enforcing runtime constraints through the use of the Java Virtual Machine (JVM), a security manager that sandboxes untrusted code from the rest of the operating system, and a suite of security APIs that Java developers can utilise. Despite this, criticism has been directed at the programming language, and Oracle, due to an increase in malicious programs that revealed security vulnerabilities in the JVM, which were subsequently not properly addressed by Oracle in a timely manner.

Dynamic:

Java classes can be easily downloaded dynamically over the network, and easily integrated with the running application. Java programmes can be written in a computer and can be executed in another computer.Java doesn't have the capability to dynamically add properties. Nor does it have the ability to dynamically create classes at runtime or change them at runtime. The Java Compiler API allows to compile Java classes at runtime. Technically a Java source file could be written to look exactly how one wants, it could be compiled and loaded. Java bytecode libraries can rewrite classes at runtime. This is used by such libraries as JPA (and others). The classes in java can also be modified.

5.2 Java Media Framework

5.2.1 Streaming Media

Any data that changes meaningfully with respect to time can be characterized as time based media. Audio clips, movie clips and animation are common form of time-based media. They can be obtained from various sources like network files, camera, microphones, and live broadcasts.

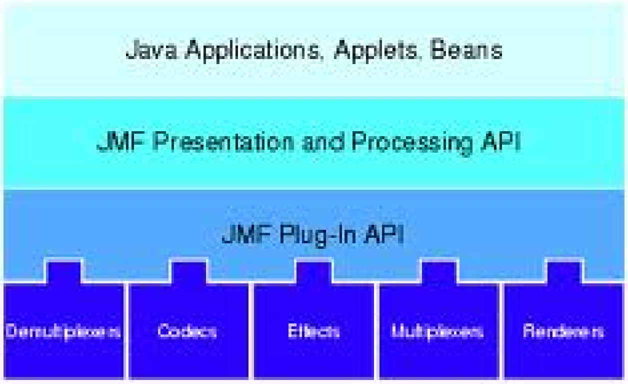

Java Applications, Applets, Beans
JMF Presentation and Processing API
JMF Plug-In API
Demultiplexers
Codecs
Effects
Multiplexers
Renderers

Figure 5.2: High-level JMF Architecture

Time-based media is also referred as streaming media. it is delivered in a steady stream that must be received and processed within a particular time frame to produce acceptable results. Media data is in media streams that are obtained from a local file, acquired over network or captured from a camera or microphone.

These time-based media is represented through o/p devices like speakers and monitors. An output destination for media data is referred as a Data-sink. In many cases, the presentation of the media stream cant begin immediately. This is latency experienced before begin of presentation. The data in a media stream is manipulated before it is presented to the user.

Time-based media can be captured from a live source for processing and playback. For this we need capture devices. Capture devices can be characterized as either push or pull source. A still camera is a pull source, where the user controls when to capture an image. A microphone is a push source, where the live source continuously provides a stream of audio.

Java Media framework (JMF) provides architecture and messaging protocol for managing the acquisition, processing and delivery of time-based media data. JMF is designed to support most standard media content types such as AIFF, AVI, GSM, MIDI, MPEG, WAV etc. JMF implementations can leverage the capabilities of the underlying operating system, while developers can create programs that feature time-based media by writing to the JMF API. With JMF, developers can easily create applets and applications that present, capture, manipulate and store time-based media.

5.2.2 Architecture

When you play a movie using VCR, you provide the media stream to the VCR by inserting a video tape. The VCR reads and interprets the data and sends appropriate signals to TV and speakers. JMF uses this same basic model. A data source encapsulates the media stream much like a video tape and a player provides processing and control mechanism similar to VCR. Playing and capturing

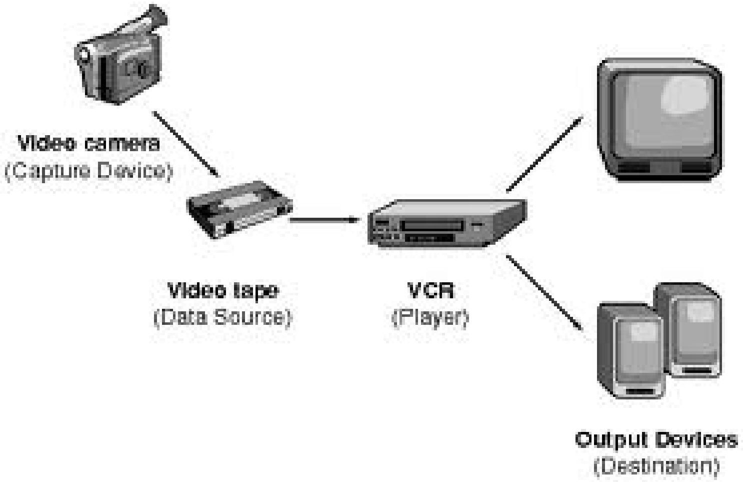


Figure 5.3: Recording, Processing and presenting time-based media.

audio and video with JMF requires input and output devices.

Time Model:

JMF keeps time to nanosecond precision. A particular point in time is typically represented by Time object. Classes that support JMF time model implement Clock to keep track of time for particular media stream. This interface defines the basic timing and synchronization operations that are needed to control the presentation of media data. A Clock uses a Time-Base to keep track of the passage of time while a media stream is being presented.
To keep track of current media time, a Clock uses:

- **The time-base start time** → the time that its Time-Base reports when the presentation begins.
- **The media start time** → the position in the media steam where presentation begins.
- **The playback rate** → how fast the Clock is running in relation to its Time-Base. The rate is a scale factor that is applied to the Time-Base.

The current media time is calculated as follows:
MediaTime =MediaStartTime + Rate (TimeBaseTime - TimeBaseStartTime)

Managers:

The JMF API consists of interfaces that define behavior and interaction of objects. By using intermediary objects called managers, JMF makes it easy to integrate new implementations of key interfaces that can be used with existing classes.
There are four managers:

- **Manager**→ handles the construction of Players, Processors, DataSources and DataSinks.
- **PackageManager**→ maintains a registry of packages that contain JMF classes.
- **CaptureDeviceManager**→ maintains a registry of available capture devices.
- **PlugInManager**→ maintains a registry of available JMF Plug-In processing components.

If you extend JMF functionality by implementing a new plug-in, you can register it with the PlugInManager to make it available to Processors that support the plug-in API.

<u>Event Model:</u>

JMF uses a structured event reporting mechanism to keep JMF-based programs informed of the current state of the media system and enable JMF-based programs to respond to media-driven error conditions such as out-of data and resource unavailable conditions. Whenever a JMF object needs to report on the current conditions, it posts a MediaEvent. MediaEvent is subclassed to identify many types of events.

<u>Data Model:</u>

JMF media players use Data Sources to manage the transfer of media-content. A Data Source encapsulates both the location of media and the protocol and s/w used to deliver the media. A Data Source is identified by either a JMF Media Locator or a URL. JMF defines several types of Data Source objects categorized according to how data transfer is initiated:

- **Pull Data-Source** → the client initiates the data transfer and controls the flow of data from pull data-sources. There are 2 types of pull Data-Sources: **PullDataSource and PullBufferDataSource**, which uses a Buffer object as its unit of transfer.
- **Push Data-Source** → the server initiates the data transfer and controls the flow of data from push data-sources. Push data-source include broadcast media, multicast media and Video on Demand. There are 2 types push data sources: **PushDataSource and PushBufferDataSource**, which uses a Buffer object as its unit of transfer.

JMF defines two types of specialty data sources, cloneable data sources and merging data sources. Those data sources are defined below:

A **Cloneable** data source can be used to create clones of either a pull or push Data Source. The clones dont necessarily have the same properties as the cloneable data source used to create them or the original Data Source.

A **Merging Data Source** can be used to combine the Source Streams from several Data Sources into a single Data Source. This enables a set DataSources to be managed from a single point of control.

<u>Control:</u>

Objects that have a Format control can implement the Format Control interface to provide ac-

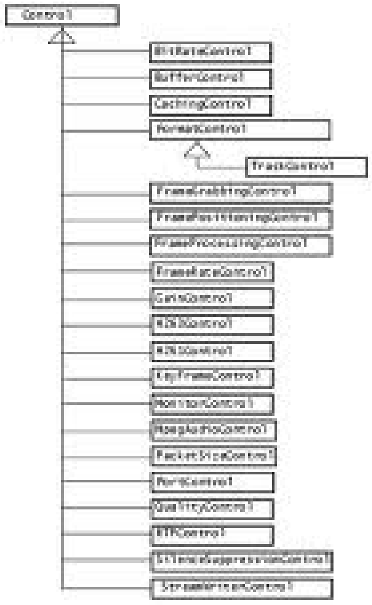


Figure 5.4: JMF Controls

cess to the Format and also provides methods to querying and setting the format. One of the Format Control is Track Control provides mechanism for controlling what processing a Processor object performs on a particular track of media data. Through it, we can specify what format conversions are performed in individual tracks and select Effect, Codec and Renderer plug-ins used by the Processor. Other controls such as Port Control (which defines methods for controlling o/p of a capture device) and Monitor Control (which enables media data (captured or encoded) to be previewed) enable user control over the capture process. Buffer Control enables user-level control over the buffering done by a particular object.
JMF Control provides mechanism for setting and querying attributes of an object. A Control often provides access to a corresponding user interfaces component that enable user control over an objects attributes. They define methods for retrieving associated Control objects. DataSource and PlugIn use the Controls interface to provide access to their Control objects.
JMF also has several codec controls to enable control over hardware or software encoders and decoders. A control provides access to a Component that exposes its control behavior to the end user. If you dont want to use the default control components provided by a particular implementation, you can implement your own and use the event listener mechanism to determine when they need to be updated.

5.2.3 Presentation

The presentation process is modeled by the Controller interface. It defines basic state and control mechanism for an object that controls presents or captures time-based media. It defines phases that a media controller goes through and provides mechanism for controlling the tran-

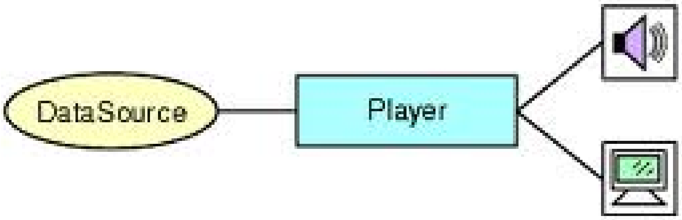


Figure 5.5: JMF Player Model

sitions between those phases. The JMF API has two types of Controllers: Players and Processors. They are constructed for a data source and not reused to present other media data.

Players:

A Player processes an input stream and renders it at precise time. A DataSource is used to deliver input media stream to the Player. The rendering destination depends on the type of media being presented. A Player does not provide any control over the processing that it performs how it renders the media data.

5.2.4 Processing

The processing of the media data is split into several stages:

- Demultiplexing is the process of parsing the input stream. If the stream contains multiple tracks, they are extracted and output separately.
- Pre-processing is the process of applying effect algorithm to the extracted from the input stream.
- Transcoding is the process of converting each track from one input format to another.
- Post-Processing is the process of applying effect algorithms to decoded tracks.
- Multiplexing is the process of interleaving the transcoded media tracks into a single output stream. E.g. separate audio and video tracks might be multiplexed into single MPEG-1 data stream.
- Rendering is the process of presenting the media to the user.

The Processing at each stage is done by separate processing component which are plug-ins. There are five types of JMF plug-ins:

- **Demultiplexer** → parses media streams such as WAV, MPEG, etc.
- **Effect** → performs special effect processing on a track.
- **Codec** → performs data encoding and decoding.

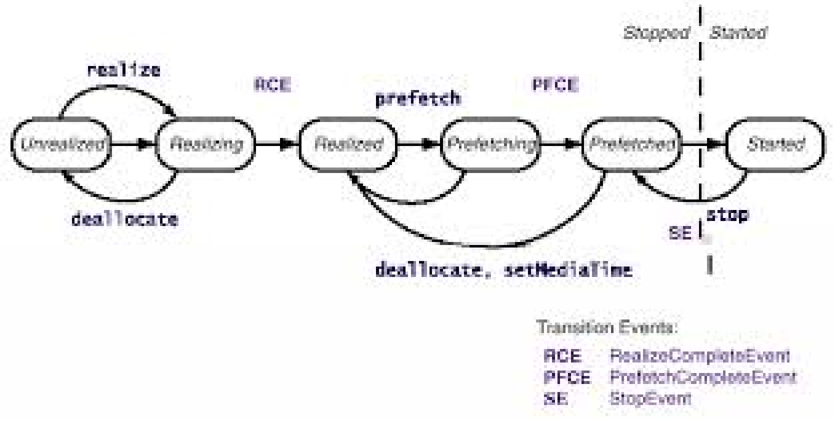


Figure 5.6: JMF Player states

RCE - RealizedCompleteEvent PFCE - PrefetchCompleteEvent SE - StopEvent A Player can be in one of the six states. The Clock interface defines the two primary states: Stopped and Started. To facilitate resource management, Controller breaks down the stopped state into five standby states: Unrealized, Realizing, Realized, Prefetching and Prefetched. Let us look at each states:

- A Player in Unrealized state has been instantiated but does not yet know anything about its media.
- When realize is called, a Player moves to the Realizing state. It is in process of determining its resource requirements. This might include rendering resources other than exclusive-use resource.
- After that, Player moves to Realized state. In this state, it knows what resources it needs and information about type media it is to present.
- When prefetch is called, Player enters into Prefetching state. Here, it preloads its media data, obtains exclusive-use resources and does whatever else it needs to do to prepare itself to play.
- When a Player finishes Prefetching, it enters into the Prefetched state. A Prefetched Player is ready to be started.
- Calling start puts Player into Started state. A Started Player objects time-base time and media time are mapped and its clock is running.

A Player posts TransitionEvent as it moves from one state to another. The ControllerListener interface provides a way for your program to determine what state a Player is in and to respond appropriately. Using this event reporting mechanism, you can manage a Player objects start latency by controlling when it begin Realizing and Prefetching.

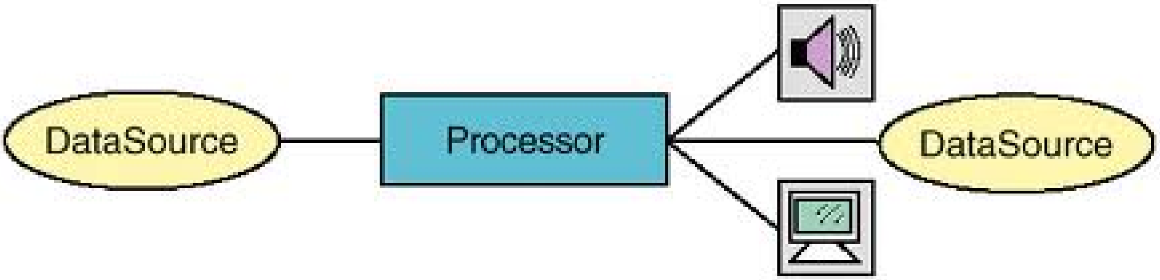


Figure 5.7: JMF Processor model

Processor can also be used to present media data. A Processor is just a specialized type of Player that provides control over what processing is performed on the input media stream. It supports all of the same presentation controls as a Player. A Processor can output media data through a DataSource so that it can be presented by another Player or Processor, further manipulated by another Processor or delivered to some other destination, such as file. Additional custom Control types might be supported by a particular Player or Processor implementation to provide other control behaviors and expose custom user interface components. A Player or Processor generally provides two standard user interface components, a visual component and a control-panel component. A Processor allows application developer to define the type of the processing that is applied to the media data. This enables the application of effects, mixing and composing in real-time.

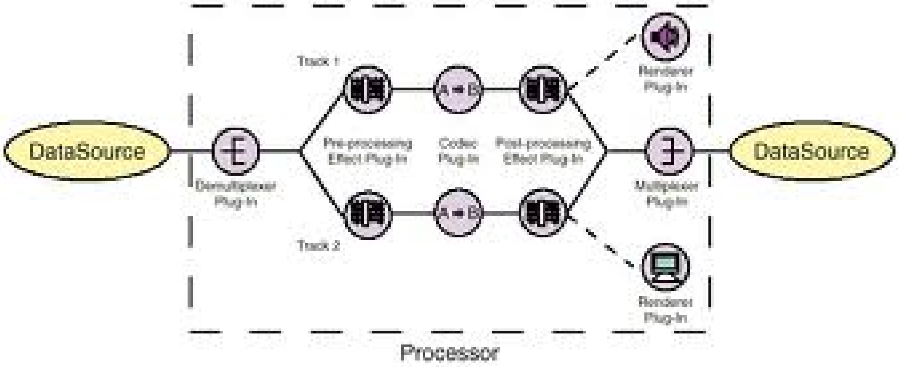


Figure 5.8: JMF Processor model

- **Multiplexer** → combines multiple tracks of input data into a single interleaved output stream.

- **Renderer** → processes the media data and delivers it to a destination.

5.2.5 Capture

A multimedia capturing device can act as a source for multimedia data delivery. E.g. a microphone can capture raw audio input / a digital video capture board might deliver digital video from camera. Such Captured devices are abstracted as Data Source. E.g. a device that provides timely delivery of data can be represented as Push Data Source. Some devices deliver multiple data streams. The corresponding Data Source can contain multiple Source Streams that map to the data streams provided by the device.

5.2.6 Media Data Storage and Transmission

A Data Sink is used to read media data from a Data Source and render the media to some destination. A particular Data sink might write data to a file, write data across the network, or function as an RTP broadcaster. Like Players, Data Sink objects are constructed through Manager using a Data Source. A Data Sink can use a Stream Writer Control to provide additional control over how data is written to the file.

5.3 Real-time Transport Protocol

5.3.1 Streaming Media

When media content is streamed to a client in real-time, the client can begin to play the stream without having to wait for the complete stream to download. The term *streaming media* is often used to both technique of delivering content over the network in real-time and the real-time media content that is delivered. Through this, streaming media is changing the way people communicate and access information.

Transmitting media data across the network in real-time requires high network throughput. It is easier to compensate for lost data than to compensate for large delays in receiving the data. Consequently, the protocols used for static data such as TCP dont work well for streaming media.

So, underlying protocols other than TCP are typically used. Such a protocol is User Datagram Protocol (UDP). UDP is a general transport layer unreliable protocol. Its a low level networking protocol on top of which more application specific protocols are built. The Internet standard for transporting real-time data (audio-video) is the Real-time Transport Protocol (RTP).

5.3.2 RTP Architecture

The media data for a session is transmitted as a series of packets known as RTP stream. Each data packet in a stream contains two parts, a structure header and the actual data.

Data Packets:

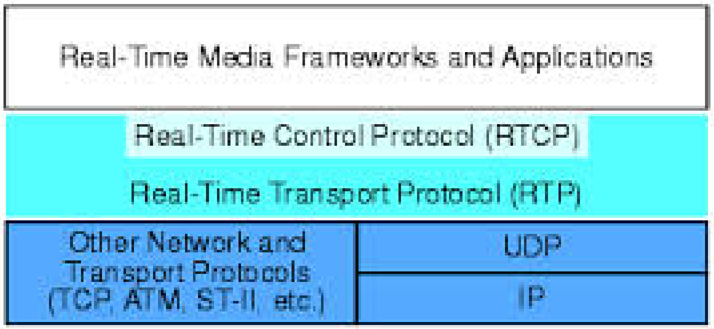

Real-Time Media Frameworks and Applications
Real-Time Control Protocol (RTCP)
Real-Time Transport Protocol (RTP)
Other Network and Transport Protocols (TCP, ATM, ST-II, etc.)
UDP
IP

Figure 5.9: RTP architecture

RTP provides end-to-end network delivery services for the transmission if real-time data. RTP can be used over both unicast and multicast network services. Over unicast, separate copies of the data are sent from the source to each destination. Over multicast, the data is sent from the source only once and the network is responsible fro transmitting the data to multiple locations. This is more efficient for multimedia application such as video conferences. RTP enable you to identify the type of data being transmitted, determine what order the packets of data should be presented in and synchronize media streams from different sources. RTP data packets are not guaranteed arrive in the order they were sent and not guaranteed to arrive at all. This is augmented by a control protocol (RTCP) that enables you to monitor the quality of data distribution and provides control and identification mechanism for RTP transmissions. An RTP session is an association among set of applications communicating with RTP. A session is identified by a network address and a pair of ports. One port is used for media data and other port is used for control data. A participant is a single machine, host or user participating in the session. Each media type is transmitted in different session. E.g. if both audio and video are used in conference, one session is for the audio and second session is used to transmit video data. This enables participants to choose which media types they want to receive.

The header of an RTP data packet contains:

- **The RTP version number (V):** 2 bits. The version defined by the current specification is 2.
- **Padding (P):** 1 bit. If the padding bit is set, there are one or more bytes at the end of the packet that are not part of the payload. The very last byte in the packet indicates the number of bytes of padding. It is used by some encryption algorithms.
- **Extension (X):** 1 bit. If the extension bit is set, the fixed header is followed by one header extension. This extension mechanism enables implementations to add information to the RTP Header.
- **CSRC Count (CC):** 4 bits. The number of CSRC identifiers that follow the fixed header. If the CSRC count is zero, the synchronization source is the source of the payload.
- **Marker (M):** 1 bit. A marker bit defined by the particular media profile.
- **Payload Type (PT):** 7 bits. An index into a media profile table that describes the payload format. The payload mappings for audio and video are specified in RFC 1890.
- **Sequence Number:** 16 bits. A unique packet number that identifies this packets position in the sequence of packets. The packet number is incremented by one for each packet sent.
- **Timestamp:** 32 bits. Reflects the sampling instant of the first byte in the payload. Several consecutive packets can have the same timestamp if they are logically generated at the same time N for example, if they are all part of the same video frame.
- **SSRC:** 32 bits. Identifies the synchronization source. If the CSRC count is zero, the payload source is the synchronization source. If the CSRC count is nonzero, the SSRC identifies the mixer.
- **CSRC:** 32 bits each. Identifies the contributing sources for the payload. The number of contributing sources is indicated by the CSRC count field; there can be up to 16 contributing sources. If there are multiple contributing sources, the payload is the mixed data from those sources.

Control Packets:

Control data packets (RTCP packets) are periodically sent to all of participants in the session. They can contain information about the quality of service for the session participants, the source of the media and statistics pertaining to the data that has been transmitted so far. There are several types of RTCP packets:

- **Sender Report**→ A participant that has recently sent data packets issues a Sender Report that contains the total number of packets and bytes sent and other information used to synchronize media streams from different sessions.
- **Receiver Report**→ A participant periodically issues Receiver Reports for all sources from which they are receiving data packets. It contains information about number of packets lost, the highest sequence number received and a timestamp used to estimate round trip delay.

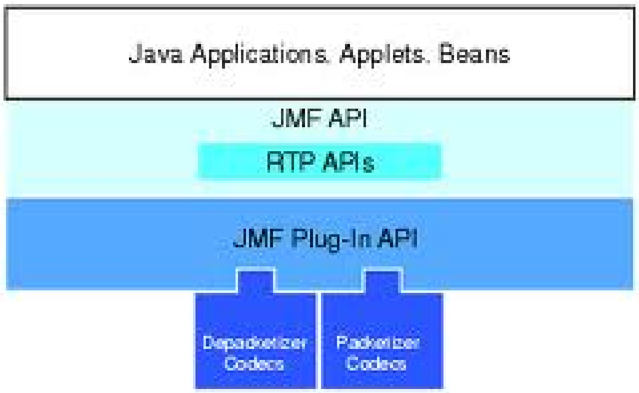


Figure 5.10: High-level JMF RTP architecture

- **Source Description**→ all compound RTCP packets must include a source description (SDES) element that contains Canonical name (CNAME) that identifies the source. Additional information might be included.
- **BYE**→ When a source is no longer active, it sends an RTCP BYE packet. It may include the reason that the source is leaving the session.
- **Application Specific**→ RTCP APP packets provide mechanism for applications to define and send custom information via RTP control port.

5.3.3 RTP Application

RTP applications are often divided into those that need to able to receive data from the network (RTP Clients) and those that need to be able to transmit data across the network (RTP Servers). Some applications do both.

5.4 JMF RTP Application Programming Interface

5.4.1 RTP API Architecture

JMF enables the playback and transmission of RTP streams through the APIs defined in the javax.media.rtp, javax.media.rtp.event and javax.media.trp.rtcp packages. JMF RTP APIs are designed to work with the capture, presentation and processing capabilities of JMF. Players and Processors are used to present and manipulate RTP media streams. JMF can be extended to support additional RTP-specific formats and dynamic payloads through the standard plug-in mechanism.

Session Manager:

In JMF, a SessionManger is used to coordinate an RTP session. The session manager keeps track of the session participants and the streams that are being transmitted. It maintains the state of the session as viewed from the local participant. It also handles the RTCP control channel and supports

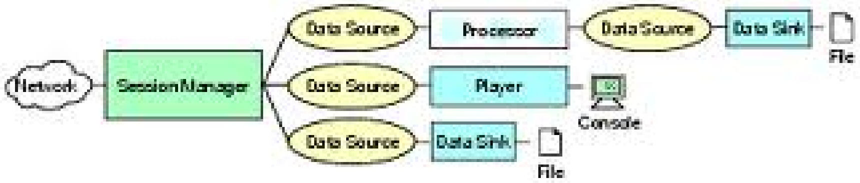


Figure 5.11: RTP reception
User can play incoming RTP streams locally, save them to a file or both. Similarly they can use APIs to transmit captured or stored media streams across network. The outgoing streams can also be played locally, saves to a file or both.

RTCP for both sender and receiver. The SessionManager interface defines methods that enable an application to initialize and start participating in session, remove individual streams created by the application and close the entire session.

Session Statistics:

The session manager maintains statistics for all of the RTP and RTCP packets and received in the session. It provides access to global reception and transmission statistics: GlobalReceptionStats- maintains global reception statistics for the session. and GlobalTransmissionStats- maintains cumulative transmission statistics for all local senders. Statistics for a particular recipient are: ReceptionStats- maintains source reception statistics for an individual participant. TransmissionStats- maintains transmission statistics for an individual send stream.

Session Participants:

Each participant is represented by an instance of a class that implements the Participant interface. Participants can be passive or active. There is exactly one local participant that represents the local client/server participant. A participant can own more than one stream, each of which is identified by the synchronization source identifier SSRC) used by the source of stream.

Session Streams:

For each stream of RTP data packets there is an RTPStream object. There are 2 types of RTPStream: ReceiveStream represents a stream thats being received from a remote participant. SendStream represents a stream of data coming from Processor or input DataSource that is being sent over the network.

RTP Events:

RTP-specific events are used to report on the state of the RTP session and streams. To receive notification of RTP events, you need to implement the appropriate listener and register it with the session manager:
SessionListener: Receive notification of changes in the state of the session. SessionStreamListener:

Receives notification of changes on the state of an RTP stream that is transmitted. ReceiveStreamListener: Receives notification of changes on the state of an RTP stream that is received. RemoteListener: Receives notification of events or RTP control messages received from a remote participant.

RTP Data:

Data Handlers: The JMF RTP APIs are designed to be transport-protocol independent. A custom RTP Data Handler can be created to work over a specific protocol. The RTPPushDataSource class defines the basic elements of JMF RTP data handler. It has both input data stream and output data stream and is used for the data channel or control channel of an RTP session. A custom RTPSocket can be used to construct Player through the Manager. JMF defines name and location for custom RTPSocket implementation: <protocol package-prefix>.media.protocol.rtpraw.Datasource

Data Formats: All RTP specific data uses an RTP specific format encoding as defined in the AudioFormat and VideoFromat classes.

AudioFormat define 4 standard RTP encoding strings:

public static final String ULAWRTP = JAUDIOG711ULAW/rtp; public static final String DVIRTP = dvi/rtp; public static final String G723RTP = g723/rtp; public static final String GSMRTP = gsm/rtp; VideoFormat defines 3 standard RTP encoding strings: public static final String JPEGRTP = jpeg/rtp; public static final String H261RTP = h261/rtp; public static final String H263RTP = h263/rtp;

RTP Controls: RTP API defines one RTP-specific control, RTPControl. RTPControl provides a mechanism to add a mapping between a dynamic payload and a Format and methods for accessing session statistics and getting the current payload Format.

5.4.2 Reception

The presentation of a RTP stream is handled by a Player. A MediaLocator is used to construct Player which has form: rtp://address:port[:ssrc]/content-type/[ttl]

The Player is constructed and connected to the first stream in the session. If there are multiple streams in the session, session manager is required. User can receive notification from the session manager whenever a stream is added to the session and construct a Player for each new stream.

5.4.3 Transmission

A session manager can also be used to initialize and control a session so that you can stream data across the network. The data to be streamed is acquired from a Processor.

1. Create, initialize, and start a SessionManager for the session.

2. Construct a Processor using the appropriate capture DataSource

3. Set the output format of the Processor to an RTP-specific format. An appropriate RTP packetizer codec must be available for the data format you want to transmit.

4. Retrieve the output DataSource from the Processor.

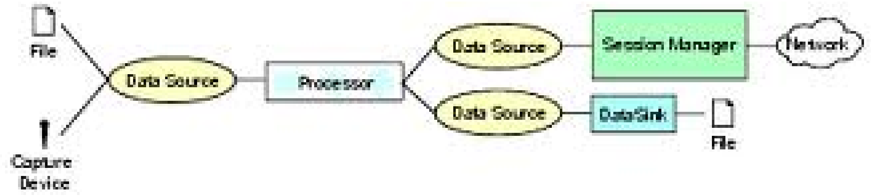


Figure 5.12: RTP transmission

5. Call createSendStream on the session manager and pass in the DataSource.

For example, to create a send stream to transmit data from a live capture source, you would: Transmission is controlled through the SendSteram start and stop methods. When it is first started, the SessionManager behaves as a receiver. As soon as SendStream is created, it begins to send out RTCP sender reports and behaves as a sender host as long as one or more send streams exist. If all SendStream are closed, it becomes a passive receiver.

5.5 Network Security

As internet is becoming more and more commercial and larger, the security of the computer system is becoming much important. So the growth of the internet has spawned additional attacks to the computer system. The objective of security is to protect the computer systems from unauthorized persons and at the same time allow access to the authorized users. In this chapter we will briefly look at the basic principles of computer system security. We will focus mainly on the internet security and will discus the encryption algorithms that we are using in our secure video conferencing system.

Some Definitions:

- **Security Policy:** A security policy is a formal statement of the rules which must be followed by people who are given access to an organization's information. The policy should specify the mechanisms through which these requirements can be met.
- **Security Attacks:** Security Attack is an action that compromises the security of the information owned by an organization for example, an unauthorized access to the information.
- **Security Mechanisms:** Mechanisms are designed to detect, prevent, or recover from the security attack
- **Security Services:** A service is that enhances the security of the information transfers of an organization. The services make use of security mechanisms.

Security services:

- **Authentication**: The authentication service is concern with assuring that the communicating entity is the one that it claims to be.
- **Access Control:** This service controls the access to the resources.

- **Data Confidentiality:** protects from unauthorized disclosure of transmitted data.
- **Data Integrity:** Data integrity assures that data received are exactly as sent by an authorized entity. That means the data received contains no modifications, insertion, deletion, or replay.
- **Non repudiation:** prevents either sender or receiver from denying a transmitted message.
- **Availability Services:** Protects a system from denial-of-service attacks.

5.6 Client-Server Architecture

We developed our Secure Video Conferencing System (SVCS) over client-server architecture because of the security needs. The system includes a centralized server and distributed client. Distribution of audio and video data between the client nodes is done using point-to-point connections between clients. The server manages the connection between the clients. It server is also responsible of distribution of session key for encryption of audio video streams and controlling the QoS.
Functions of the server:

Server side application performs following functions: -

- Registration of the users: Each user must be registered at the server in order to take part in the conference. At the time of registration user will provide his information to the server along with his/her public key. User will get a unique user-id and servers public key. These public keys will be used at the time of authentication. Authentication of users: At the time of joining a conference by a user the server and the user, both will authenticate each other using their private-public key pairs.
- Authentication of users: At the time of joining a conference by a user the server and the user, both will authenticate each other using their private-public key pairs.
- Distribution of the session key: After authenticating a user, the server will send the session key to the client application running at the user side in a secure manner. This session key will be common for all users taking part in a particular conference.
- Maintaining the state of the conference: Whenever a user will join or leave a conference the server will inform the other users of that conference and provide them necessary information in order to maintain the state of the conference.
- Controlling QoS: Server will also responsible of controlling quality of service (QoS) of the audio/video data exchanged between the clients. In order to do that,the server gets the feedbacks from the clients about every media stream they are receiving. For each client, the server maintains QoS statistics from the feedbacks of all receivers of that client and whenever required it sends the QoS control signals to the client to maintain the quality of media stream being sent by that client.

Functions of the client:

Following are the basic functions performed by the client side application:-

- Session setup: The client application will interact with the server to get the session key and information of other users to setup a conferencing session.
- Capturing the audio/video stream: The client application will be responsible of capturing users real time audio and video streams from the capturing devices.
- Compression and Encryption of the audio/video streams: The client will compress the audio and video streams to reduce the bandwidth requirements. it will also encrypt each packet of the audio video stream before transmission using the session key got from the server.
- Creation of RTP session: The client application will create RTP sessions for transmitting real time audio/video streams of the users to other users taking part in the conference. Two separate sessions will be created for audio and video.
- Opening RTP sessions: The client application will open RTP sessions for each user whose audio/video streams the user wants to receive.
- Decryption and Decompression: Decryption and Decompression will be done both audio and video streams of each user to get the streams in their original form.
- Rendering the audio/video stream: Finally the incoming audio and video streams will be rendered and will be sent to the speakers and display unit respectively.

- Rendering the audio/video stream: Finally the incoming audio and video streams will be rendered and will be sent to the speakers and display unit respectively.

5.7 Audio/Video Transmitter

The transmitter module is a JMF/RTP based application. JMF manager class is used to create a merged data source object for audio and video. This data source object is passed to the manager class to create a processor object. While the Processor is in the configured state, its track control objects are obtained for the individual audio and video tracks. Codecs for compression and encryption are set to both of the audio and video tracks. After setting the codecs both of the tracks, processors realize method is called. While the processor is in realized state the output data source is created from the processor. The output data source object is passed to the RTP manager object to create RTP sessions for audio and video transmission.

Chapter 6

Security and Advantages of Video-conferencing

6.1 Security during video Transmission

Video conferences are often archived for later use. Since the information discussed in these video conferences could be sensitive, data storage needs to be secure and separate from all other networks. It's not advisable to use standard computers and hard drives to store video-conference data, since these machines are the most susceptible to intrusion, either from internal or external sources. Data transmission is the most vulnerable area of video-conferencing security since the data must travel over so many public and private networks to reach its destination. Encryption and network security are the keys to protecting data transmission during a video conference.Another video conferencing security concern is something called data radiation. All electronic devices give off a certain amount of radiation that can be intercepted by hackers. Copper phone line cable, for example, can act as an antenna, broadcasting data to those who know how to decipher it . And the data radiation from a video screen can be read up to a kilometer away.

6.2 Advantages

- Simultaneous lectures in and off the classroom Instructors can web cast and deliver live lectures to the students sitting inside and outside the classroom enabling complete interaction with Q/A features, live chat, and real-time quizzes.

- Expert lectures. Live conferencing lets K-12 and universities to invite the interest of regional and international experts on a subject and let them communicate and instruct audience from their preferred or existing locations.

- Lecture retention. Live conferencing can be used to retain classroom lectures through instant recording. The recorded lectures can be distributed to the audience with or without editing in an instant to eliminate manual note taking and encourage students to focus on the lectures during the class.

Live conferencing equipment with cutting edge technologies and comprehensive features provide several benefits:

- **Cost saving -** Most live conferencing equipment and solutions integrate with existing hardware or corporate infrastructure to save cost. Further, live conferencing saves traveling and lodging cost spent in attending in-person seminars or meeting.
- **Reduced communication time -** Live conferencing enables instant communication with just few clicks. It requires no traveling and can be connected any time beyond geographical boundaries. Live conferencing saves communication time through organization wide network of networking points.
- **Increased efficiencies -** Live conferencing improves student performance, comprehension, and employee satisfaction with immediate communication and leverages managers and instructors to address problems right away.
- **Ease of Use -** Live conferencing equipment is usually easy-to-use built with familiar interfaces and standard technologies.

Chapter 7

Diagrams

A use case diagram at its simplest is a representation of a user's interaction with the system and depicting the specifications of a use case. A use case diagram can portray the different types of users of a system and the various ways that they interact with the system. This type of diagram is typically used in conjunction with the textual use case and will often be accompanied by other types of diagrams as well.

An entityrelationship model (ER model) is a data model for describing the data or information aspects of a business domain or its process requirements, in an abstract way that lends itself to ultimately being implemented in a database such as a relational database. The main components of ER models are entities (things) and the relationships that can exist among them.

A flowchart is a type of diagram that represents an algorithm, work flow or process, showing the steps as boxes of various kinds, and their order by connecting them with arrows. This diagrammatic representation illustrates a solution to a given problem. Flowcharts are used in analyzing, designing, documenting or managing a process or program in various fields.

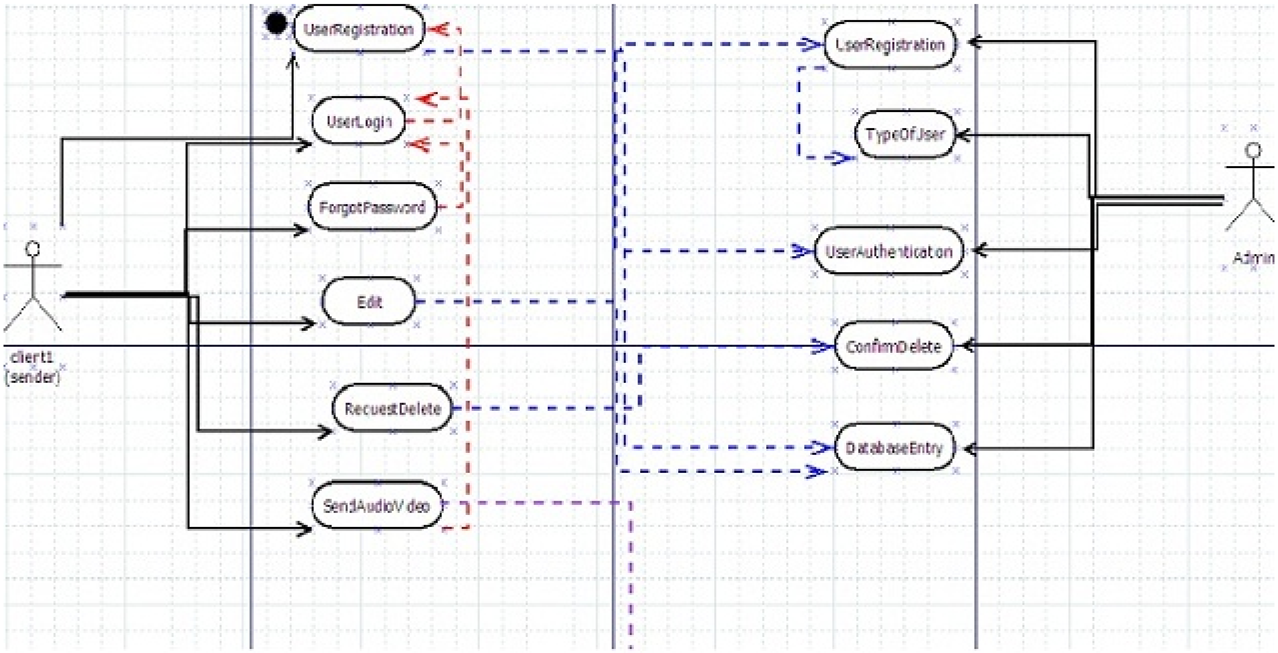


Figure 7.1: Use-case Diagram 1

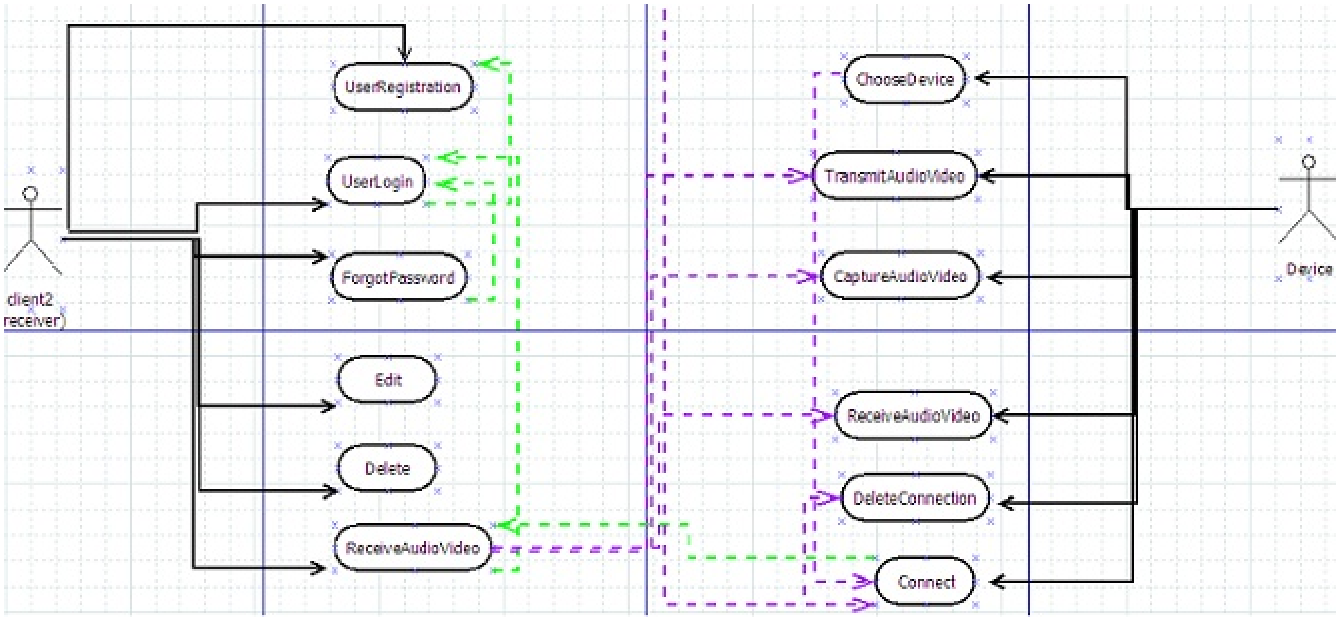


Figure 7.2: Use-case Diagram 2

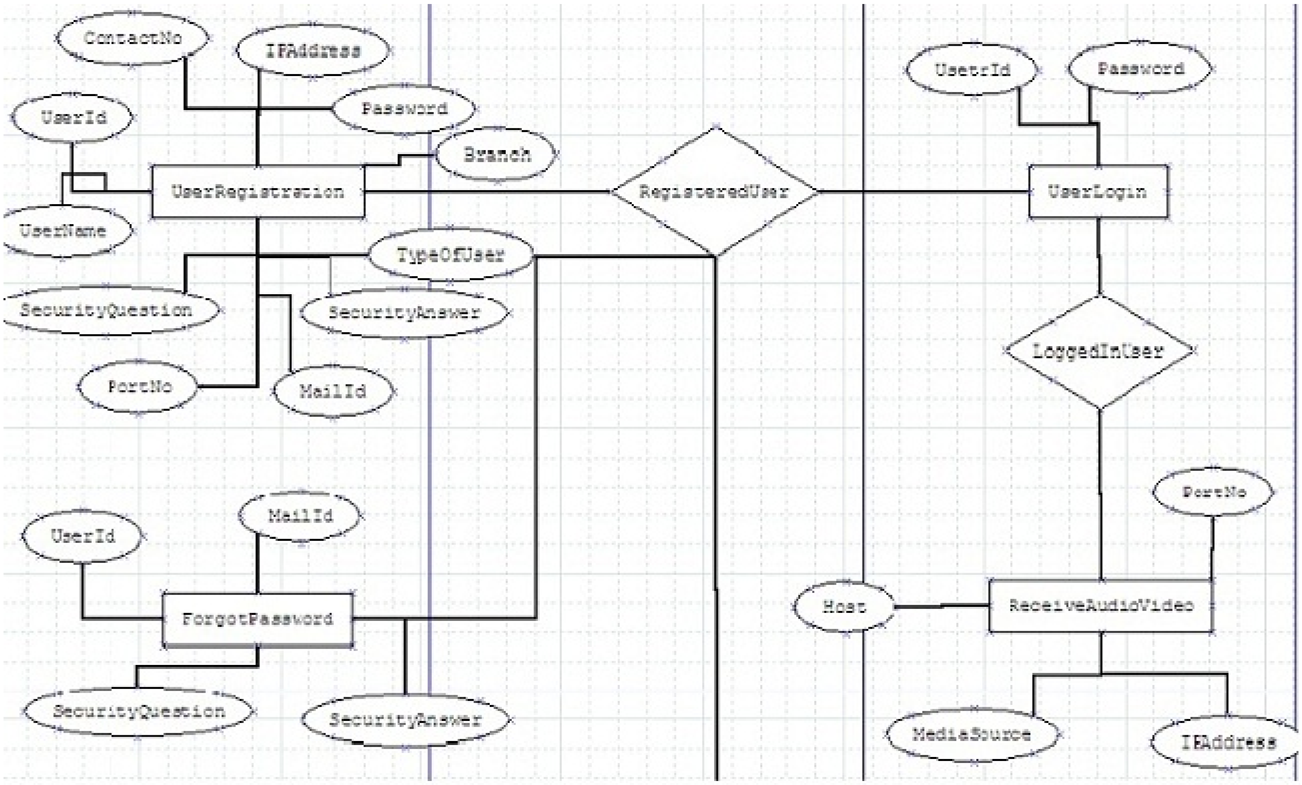


Figure 7.3: ER Diagram 1

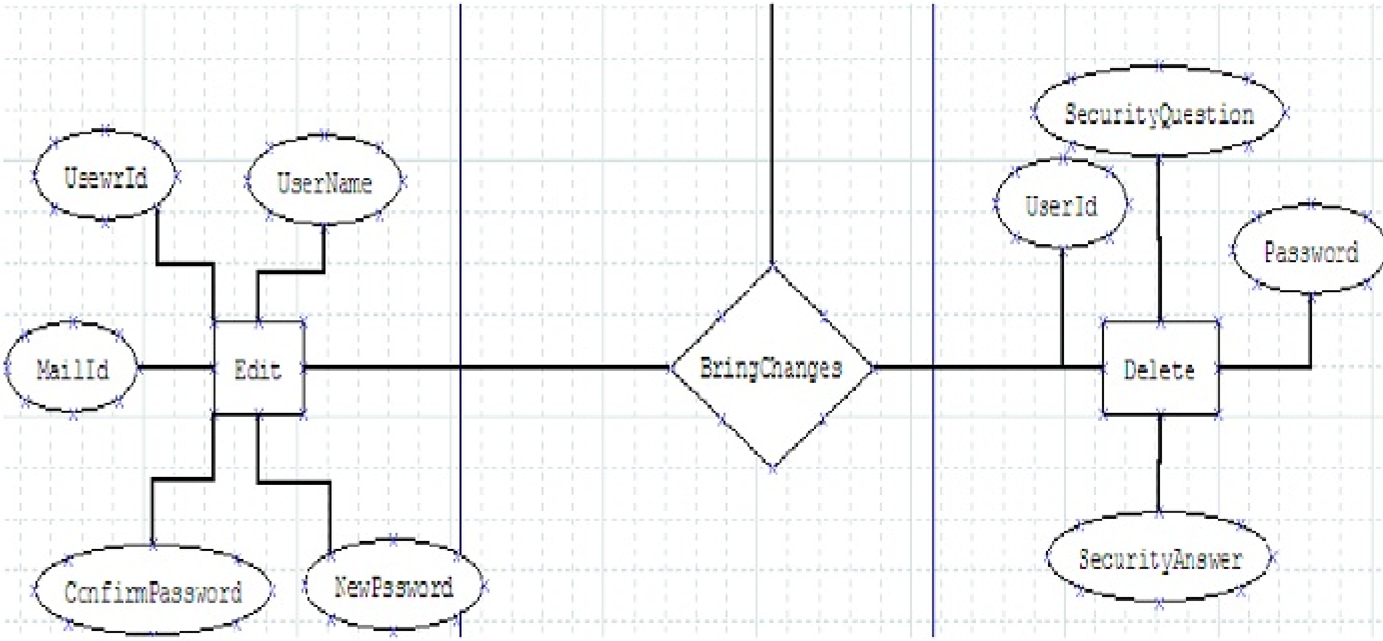


Figure 7.4: ER Diagram 2

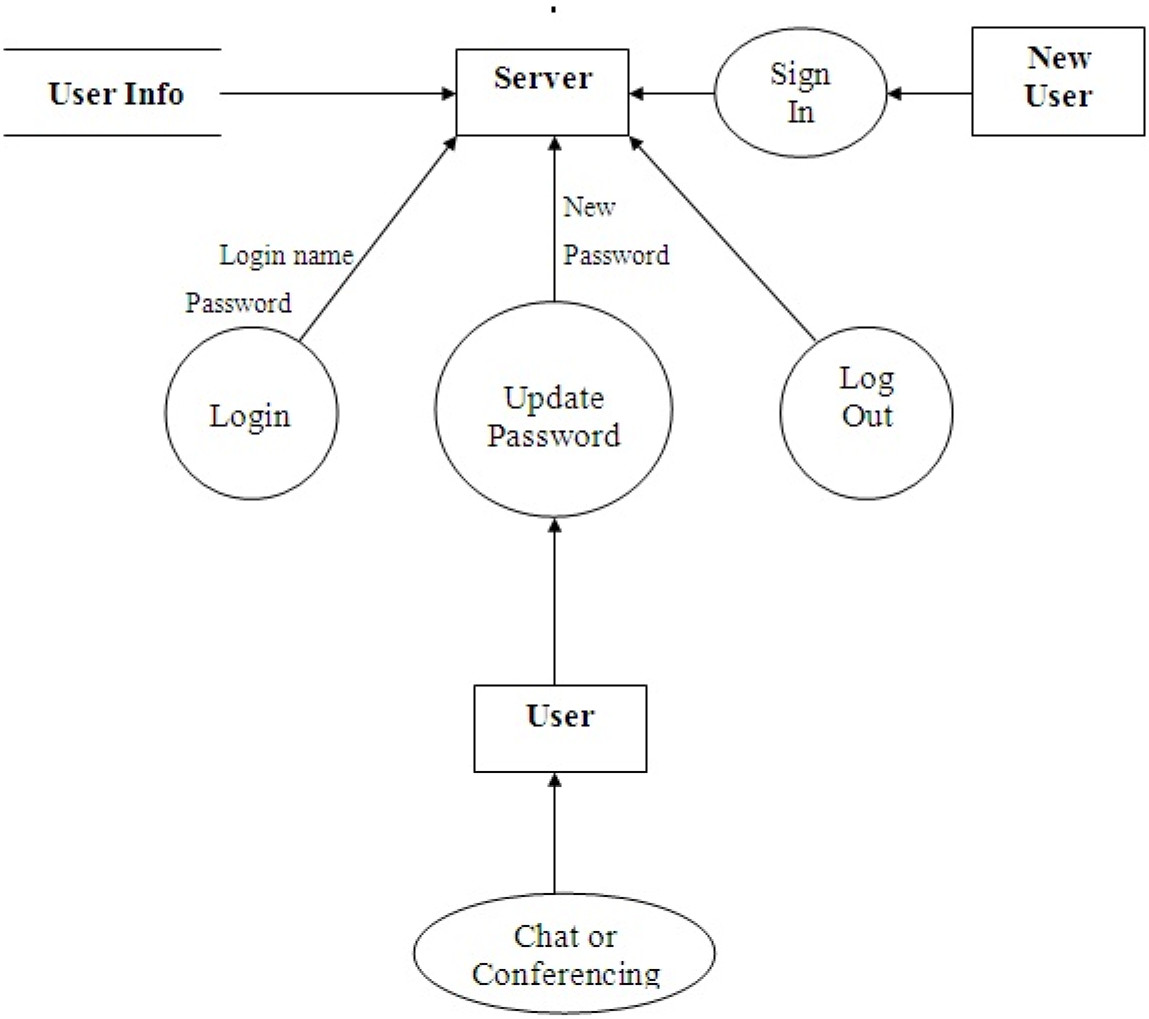


Figure 7.5: Data Flow Diagram

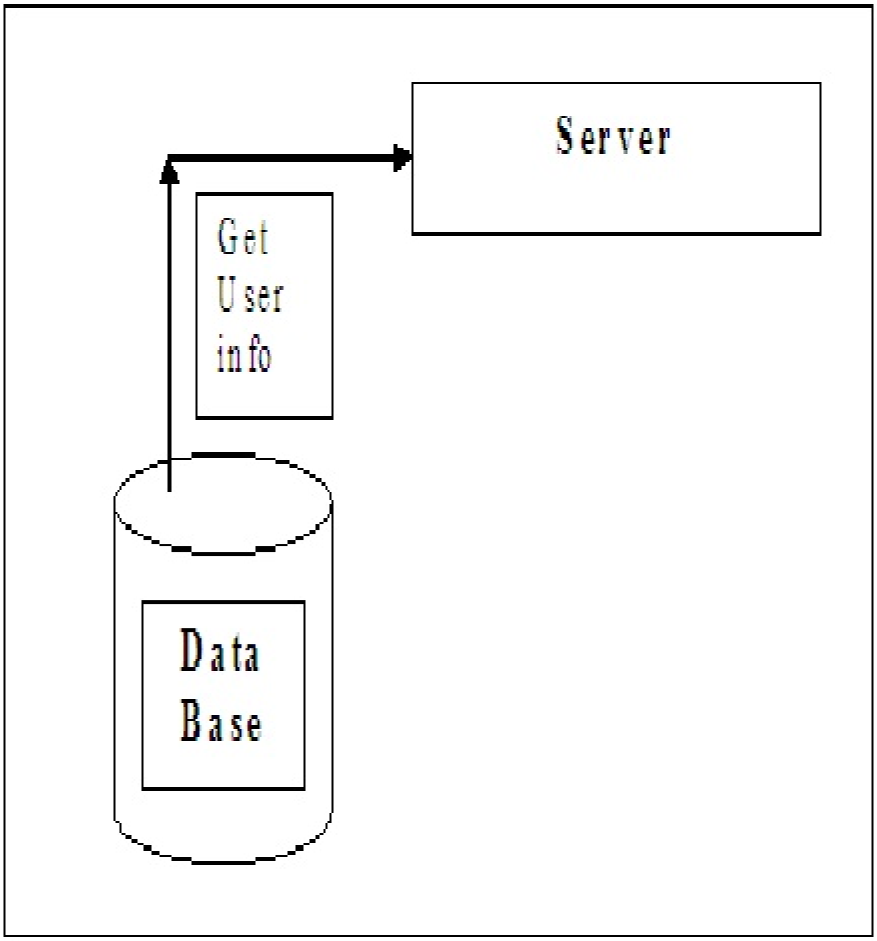


Figure 7.6: During initialization

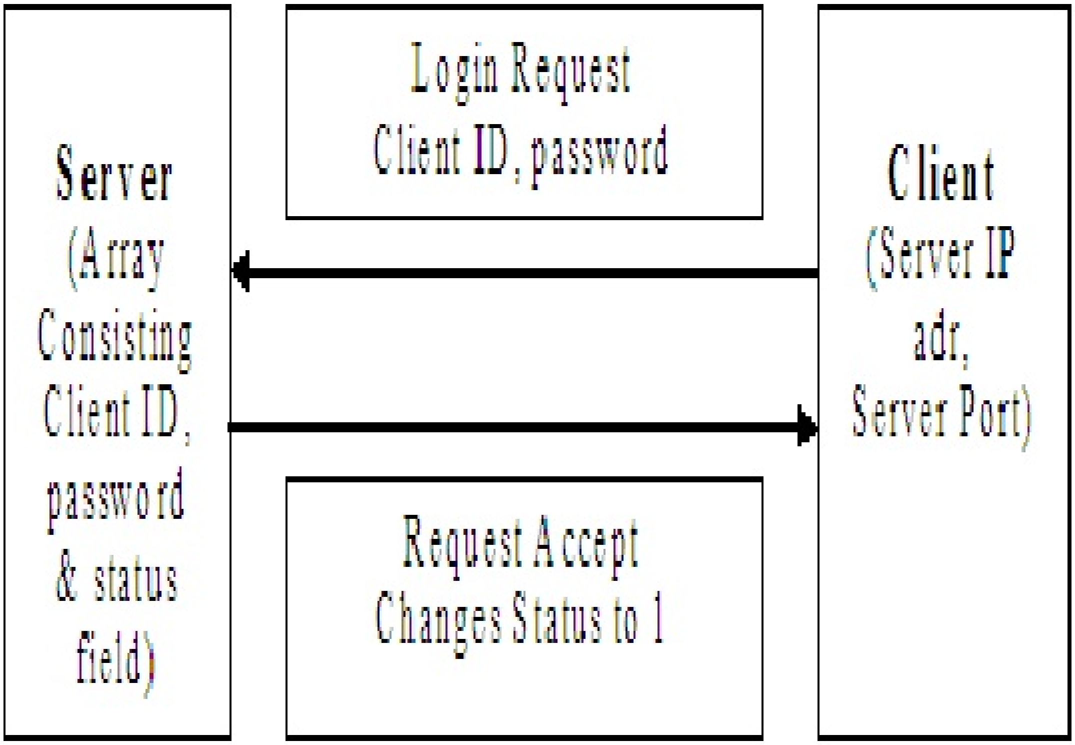


Figure 7.7: Login Request sent by first Client

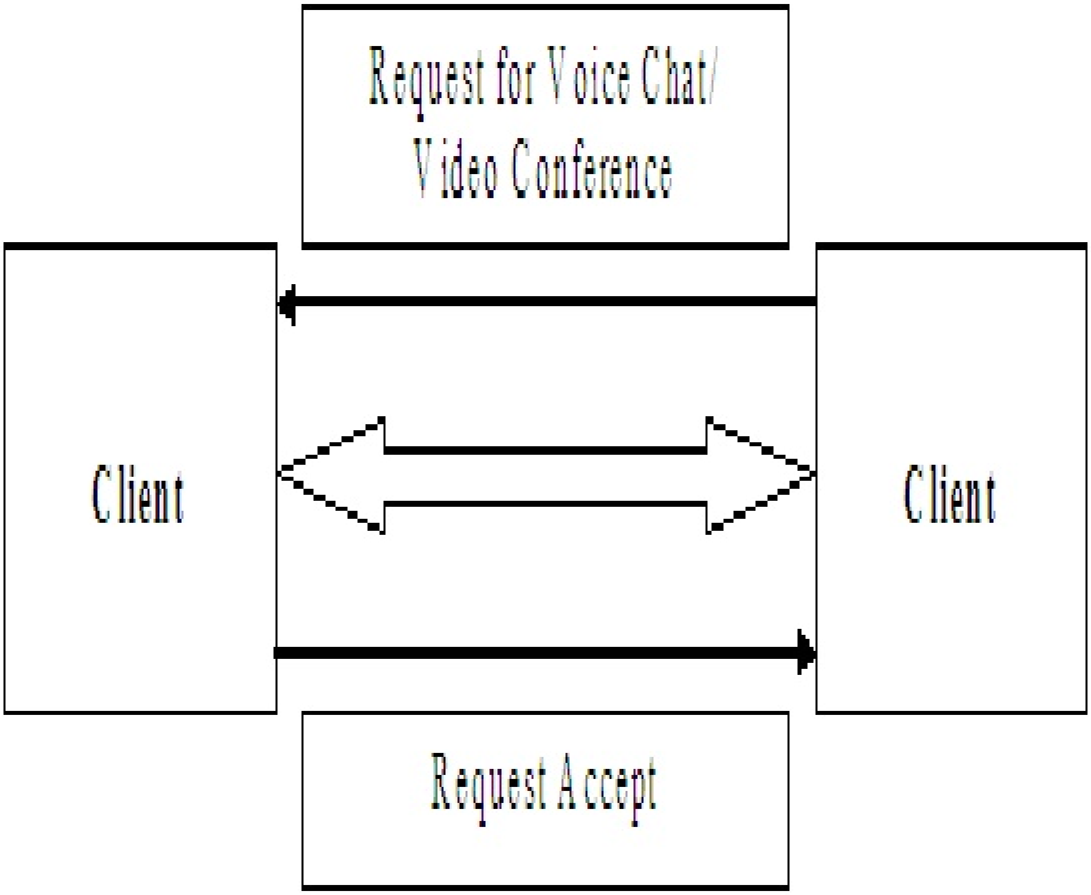


Figure 7.8: Voice Chat / Video Conference Request and Acceptance

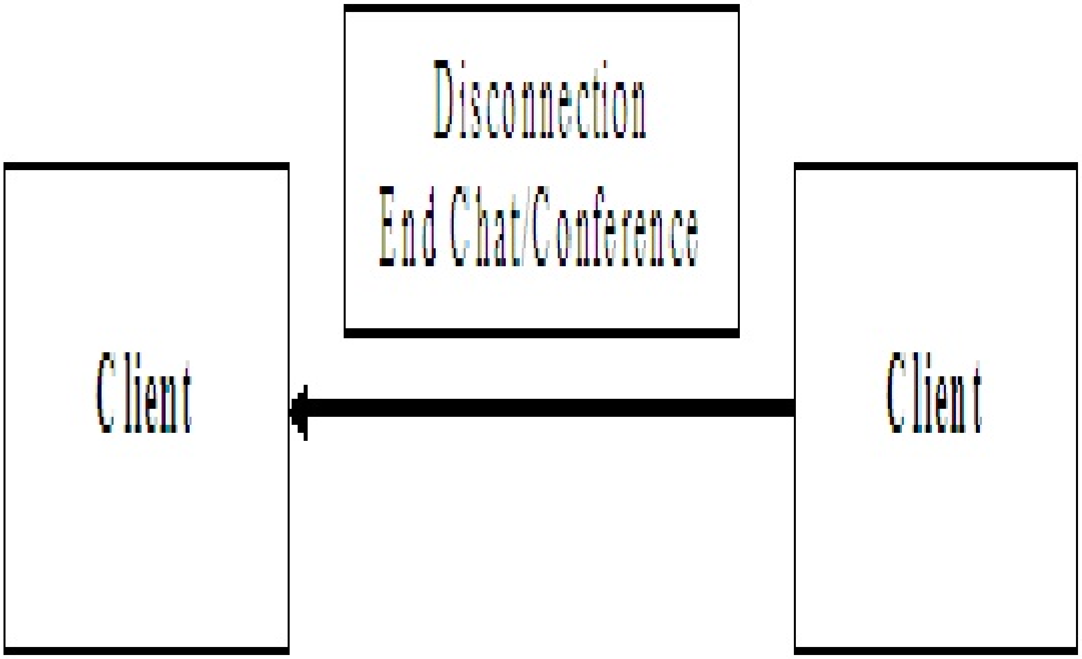


Figure 7.9: A Client ends Chat/Conference

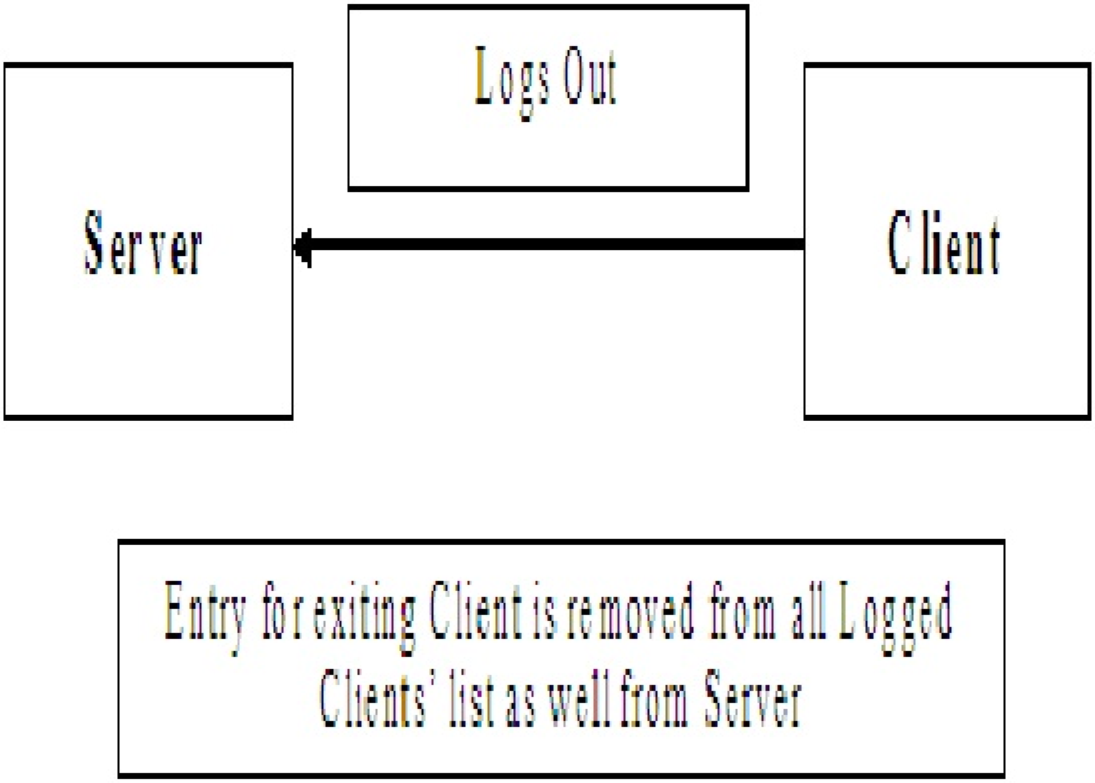

Logs Out
Server
Client
Entry for exiting Client is removed from all Logged Clients' list as well from Server

Figure 7.10: A Client logs out

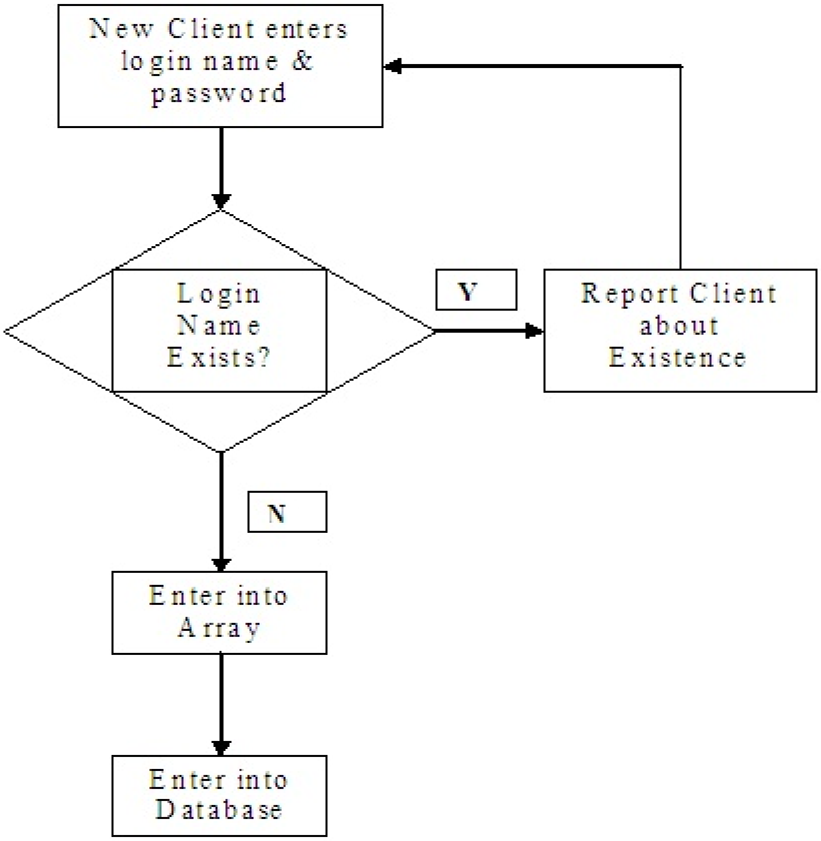

New Client enters login name & password
Login Name Exists?
Y
Report Client about Existence
N
Enter into Array
Enter into Database

Figure 7.11: New User Creation

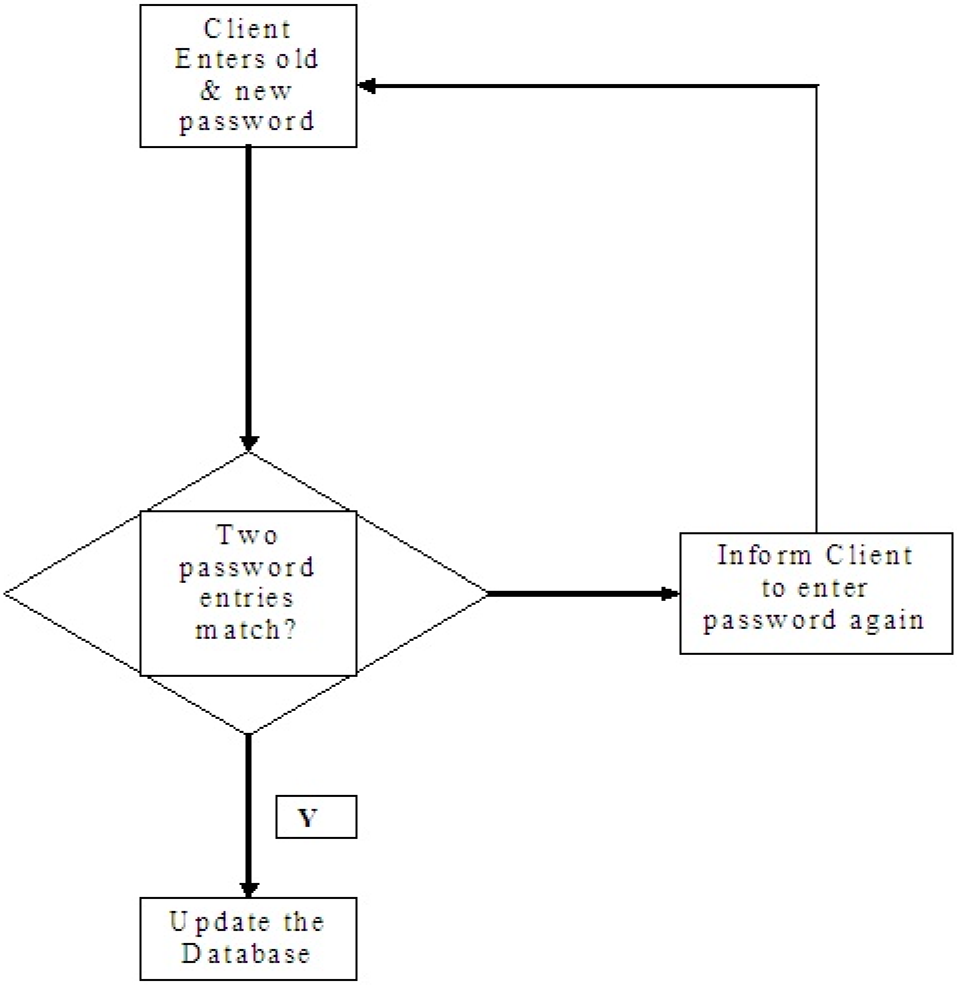

Client Enters old & new password
Two password entries match?
Inform Client to enter password again
Y
Update the Database

Figure 7.12: User changes Password

Chapter 8

Diagram Descriptions

USE CASE 1: CONNECT TO HOST

This use case describes the tasks that the client/s actor needs to undertake in order to connect to the host. This includes providing a username and IP Address. In addition the actor will need to be connected to a LAN to carry out these tasks.

Pre-conditions: Connected to LAN.

Basic flow:

- **Client:** Inputs user name and hosts IP.
- **System:** Checks if user name is not already in use and IP is valid and connects to server.

Alternative flow:

- **Client:** Inputs invalid user name and/or IP.
- **System:** Generates message stating invalid user name and/or IP.
- **Client:** Re-enters valid username and IP.
- **System:** Returns to valid state.

Post-condition: Connected to host.

USE CASE 2: UPDATES CONTACTS LIST

This use case describes the tasks the system carries out to update the contacts list when the-client/s log in or log out of the video conferencing system. These tasks include adding or removing a contact from the list and re-sending the list.

Pre-conditions: None

Basic flow:

- **Client:** Connects.
- **System:** Adds user to contact list and re-sends list.
- **Else if Client:** Disconnects.
- **System:** Removes user from contact list and re-sends list.

Alternative flow: None

Post-condition: List is updated.

USE CASE 3: STOP CONFERENCE

This use case describes the tasks that the host actor needs to undertake in order to stop a conference. This includes selecting client/s from the contacts list and clicking the stop button.

Pre-conditions: Conference in progress

Basic flow:

- **Host:** Chooses client from list and clicks stop conference.
- **System:** Stops streaming audio/video.

Alternative flow: None

Post-condition: System returns to ready state.

USE CASE 4: START UNICAST CONFERENCE

This use case describes the tasks that the host actor needs to undertake in order to start a unicast conference. This includes selecting a client from the contacts list and clicking the start chat button.

Pre-conditions: Users Connected

Basic flow:

- **Host:** Select one client from list and click Start chat button.
- **System:** Transmit audio and video.

Alternative flow:

- **System:** Generates message No audio/video device detected.

- **Post-condition:** Unicast session started.

Post Condition: None

USE CASE 5: START MULTICAST CONFERENCE

This use case describes the tasks that the host actor needs to undertake in order to start a multicast conference. This includes selecting multiple clients from the contacts list and clicking the start chat button.

Pre-conditions: Users Connected

Basic flow:

- **Host:** Select multiple clients from list and click Start chat button.
- **Host:** Select multiple clients from list and click Start chat button.

Alternative flow:

- **System:** Generates message No audio/video device detected.
- **Post-condition:** Multicast session started.

USE CASE 6: START BROADCAST CONFERENCE

This use case describes the tasks that the host actor needs to undertake in order to start a broadcast conference. This includes clicking the broadcast button.

Pre-conditions: Users Connected

Basic flow:

- **Host:** Select multiple clients from list and click Start chat button.
- **System:** Transmit audio and video.

Alternative flow:

- **System:** Generates message No audio/video device detected.

Post-condition: List is updated.

USE CASE 7: LAUNCH SERVER

This use case describes the task the host actor needs to undertake in order to launch the server. This includes providing an IP address and username.

Pre-conditions: Connected to LAN.

Basic Flow:

- **Host:** Inputs user name and IP Address
- **System:** checks that IP address is valid and starts server.

Alternative flow:

- **Host:** Inputs invalid IP address.
- **System:** Generates message stating invalid IP.

Post-condition: Server started.

USE CASE 8: CLOSE SERVER

This use case describes the task the host actor needs to undertake in order to close the server.

Pre-conditions: Server on.

Basic Flow:

- **Host:** Clicks Disconnect button.
- **System:** closes server.

Alternative Flow: None.

Post Condition: Application closed.

USE CASE 9: LOG OUT

This use case describes the tasks that the client/s actor needs to undertake in order to log out of the application.

Pre-conditions: Connected to host.

Basic flow:

- **Client:** Clicks Disconnect button.
- **System:** Disconnects client from server.

Alternative Flow: None

Post Condition: List updated.

Chapter 9

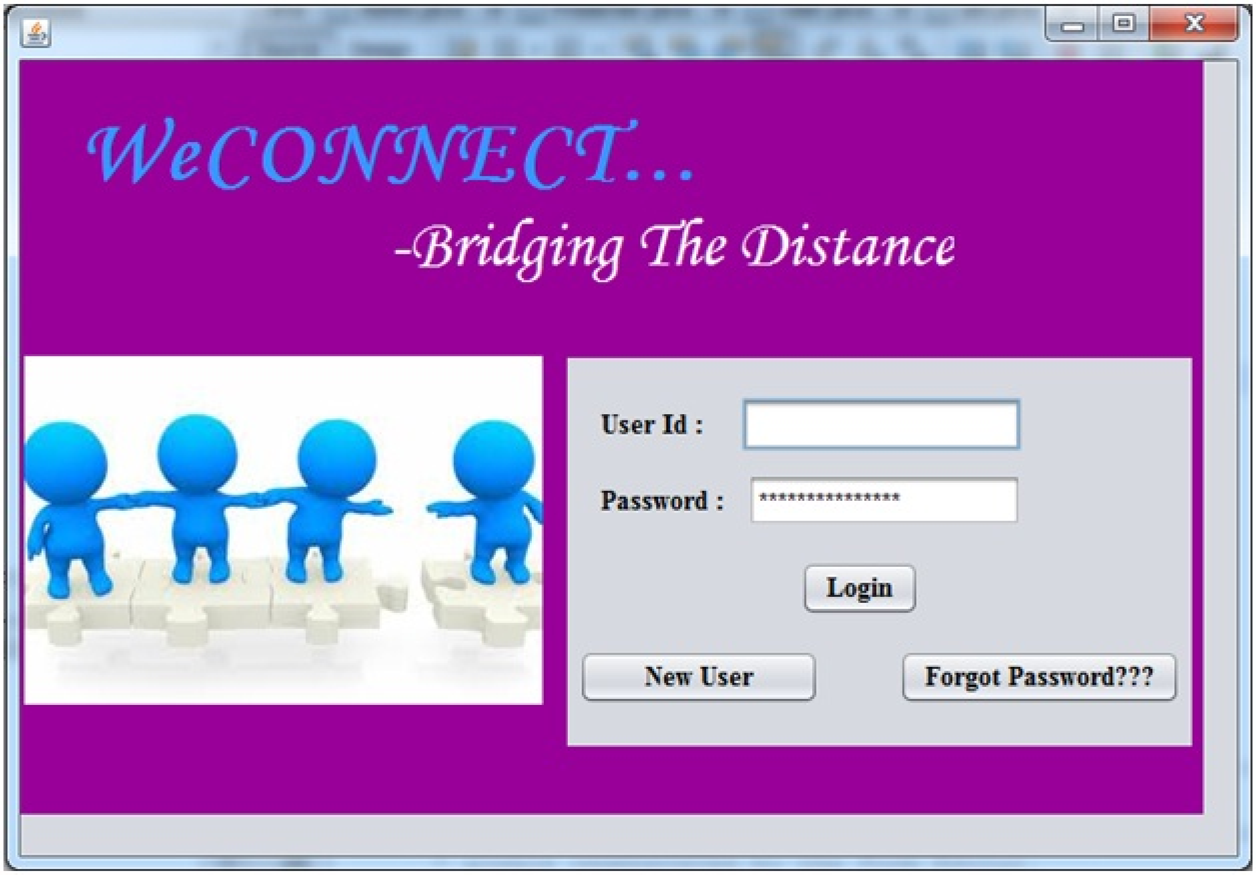
Results and Screenshots

A screenshot, screen capture (or screen-cap), screen dump, screen grab is an image taken by the computer user to record the visible items displayed on the monitor, television, or another visual output device. Usually, this is a digital image using the operating system or software running on the computer, but it can also be a capture made by a camera or a device intercepting the video output of the display.

Screenshots can be used to demonstrate a program, a particular problem a user might be having, or generally when display output needs to be shown to others or archived. For example, after being emailed a screenshot, a Web page author might be surprised to see how their page looks on a different Web browser and can take corrective action. Likewise with differing email software programs, (particularly such as in a cell phone, tablet, etc.,) a sender might have no idea how their email looks to others until they see a screenshot from another computer and can (hopefully) tweak their settings appropriately.

This chapter contains the output of the codes which are written using NetBeans IDE 7.0.1. The screenshots in this chapter are the images which are obtained on the output screen of the computer. This chapter also contains description of each screenshot along with the images or screenshots.

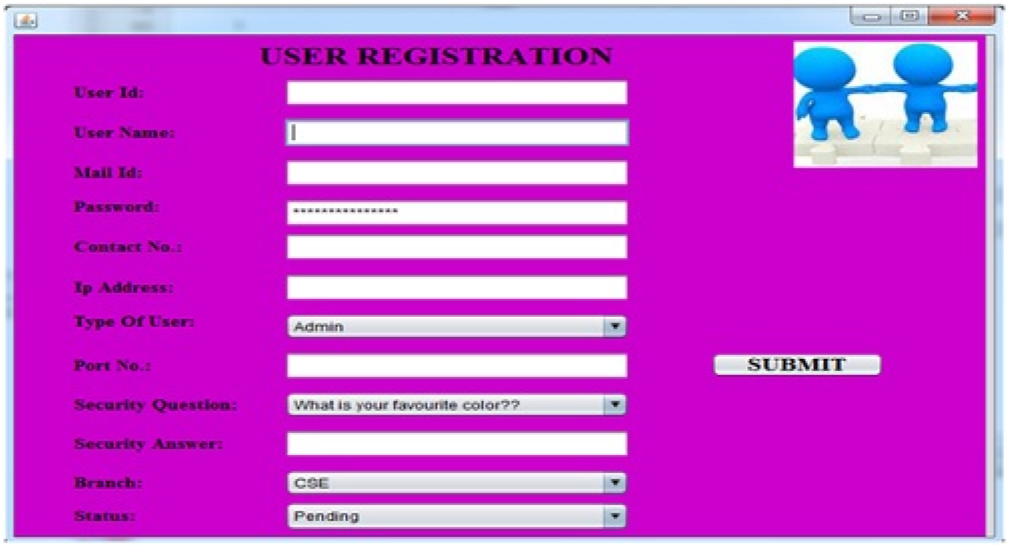
The screenshots or outputs or results of our project are in the form of colourful images, which gives a better look and feel to our project presentation.



User Id :

Password : ****************

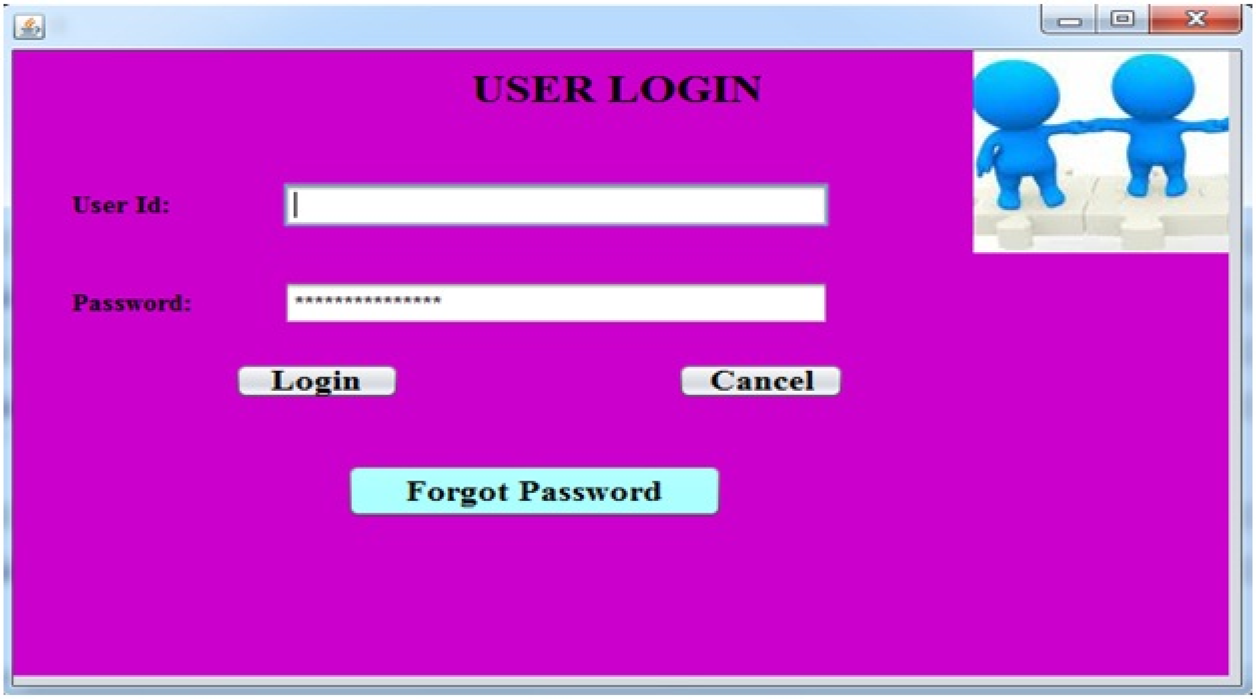
Figure 9.1: Home Page OF WeCONNECT
This is the home page of our project WeConnect. By the help of this, a new user can sign in with the help of new User or a registered user can login it with its user id and password.



USER REGISTRATION

User Id:	
User Name:	
Mail Id:	
Password:	****************
Contact No.:	
Ip Address:	
Type Of User:	Admin
Port No.:	
Security Question:	What is your favourite color??
Security Answer:	
Branch:	CSE
Status:	Pending

Figure 9.2: User Registration
A new user can sign in with the help of user registration page. An user has to enter all its information asked in the user registration page and then he has to click on the submit button to save his information in the database .

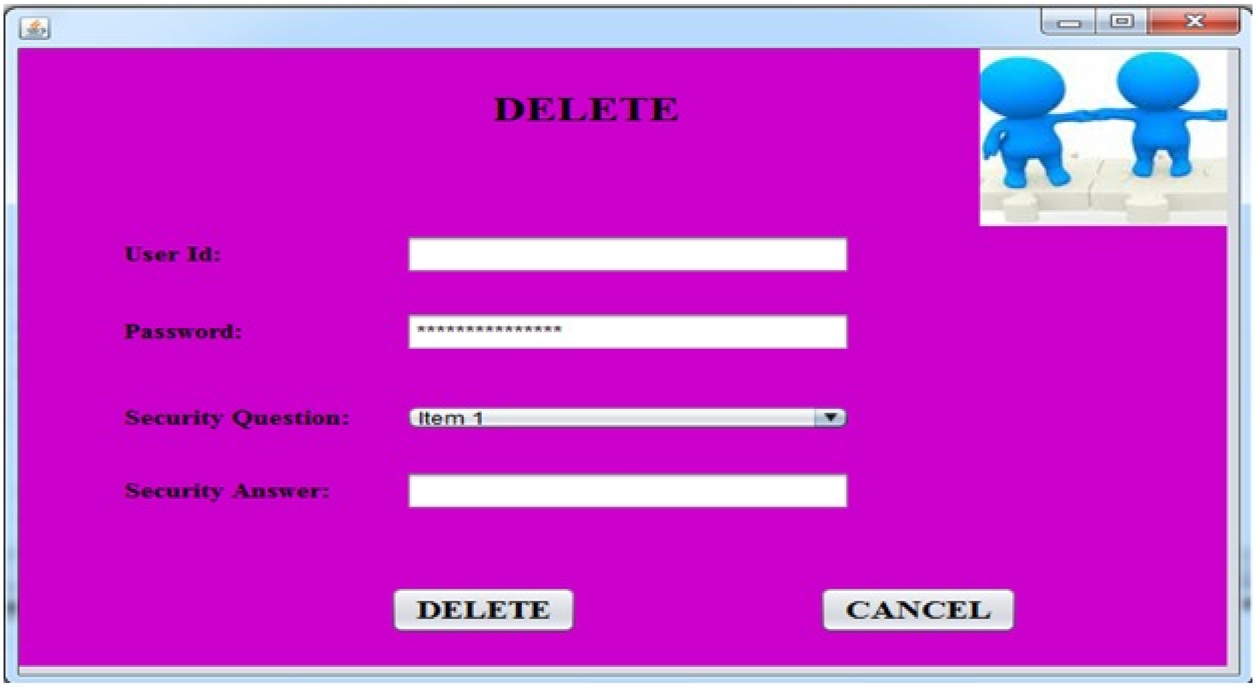


USER LOGIN

User Id:

Password: ****************

Figure 9.3: User Login
This is the login page where an existing user has to enter its user id and password and can get logged in. If an user has forgotten its password then he can go for the forgot password option to set a new password.



DELETE

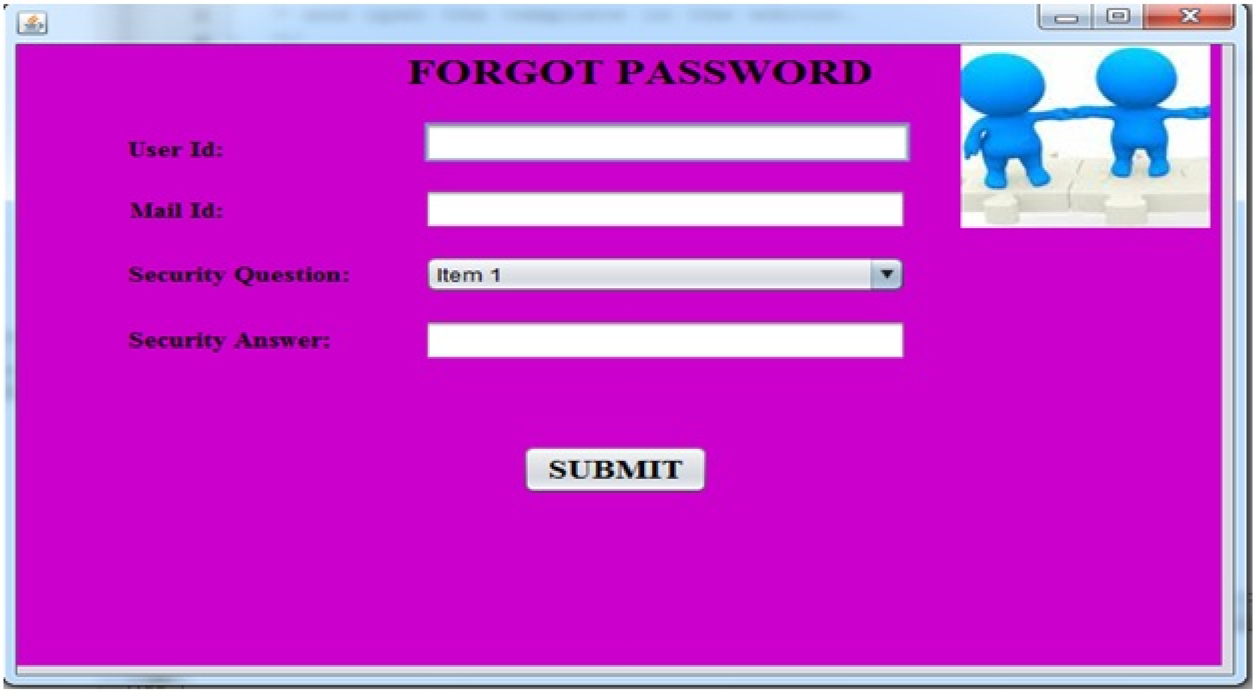
User Id:

Password: ****************

Security Question: Item 1

Security Answer:

Figure 9.4: Delete Page
This is the delete page where if an user wishes to delete his account then he has to provide its user id and password and answer the security question to deactivate his account completely.



FORGOT PASSWORD

User Id:

Mail Id:

Security Question: Item 1

Security Answer:

Figure 9.5: Forgot Password Page
This is the forgot password page if an user has forgotten his password then he can reset his password with the help of this page.

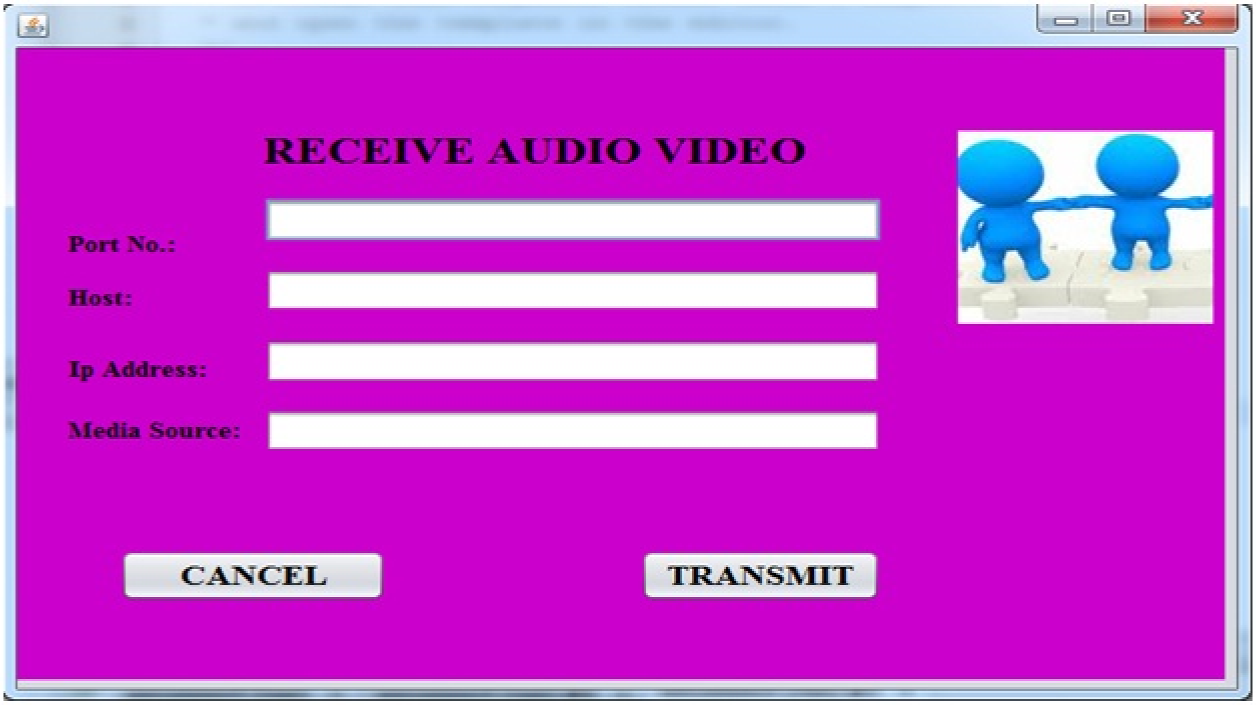
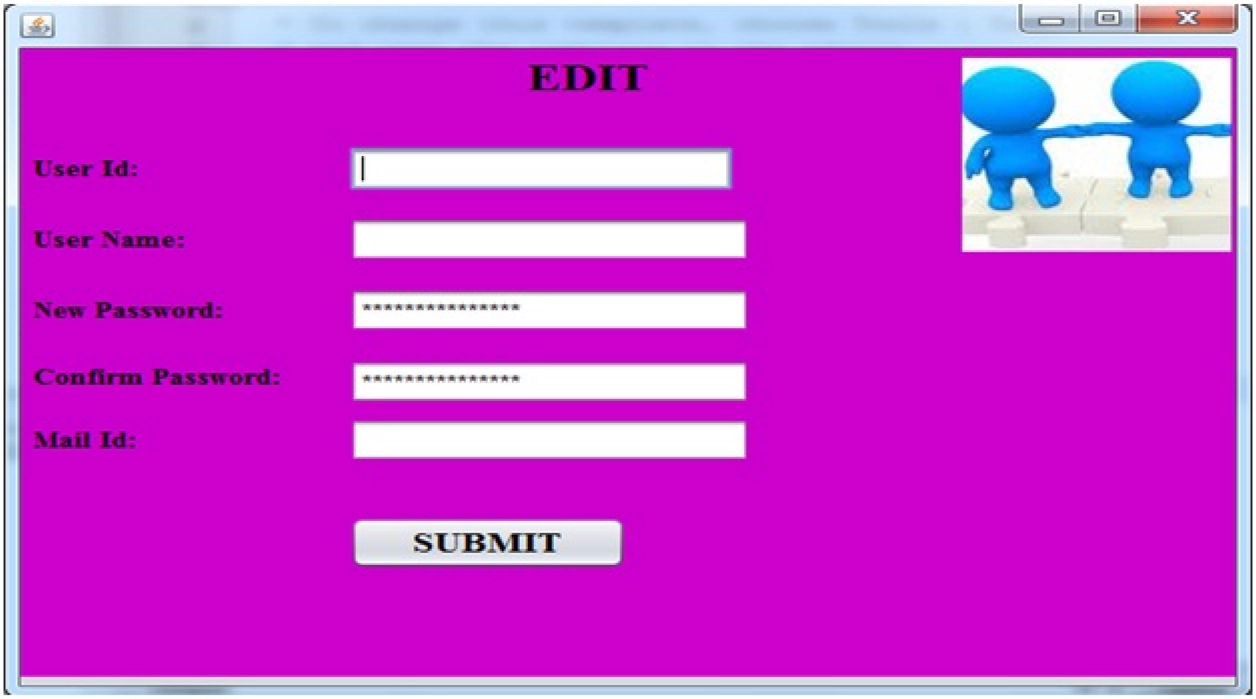


Figure 9.6: Receive Audio/Video Page
This is the receive audio video page with the help of which an user can receive real time audio and video after logging into the software. The user has to enter the port no,host name,his IP address,and media source to start with the video conferencing.



User Id:

User Name:

New Password: ***************

Confirm Password: ***************

Mail Id:

Figure 9.7: Edit Page
This is the Edit page with the help of which the user can change its information like password and submit it to save the changed information.

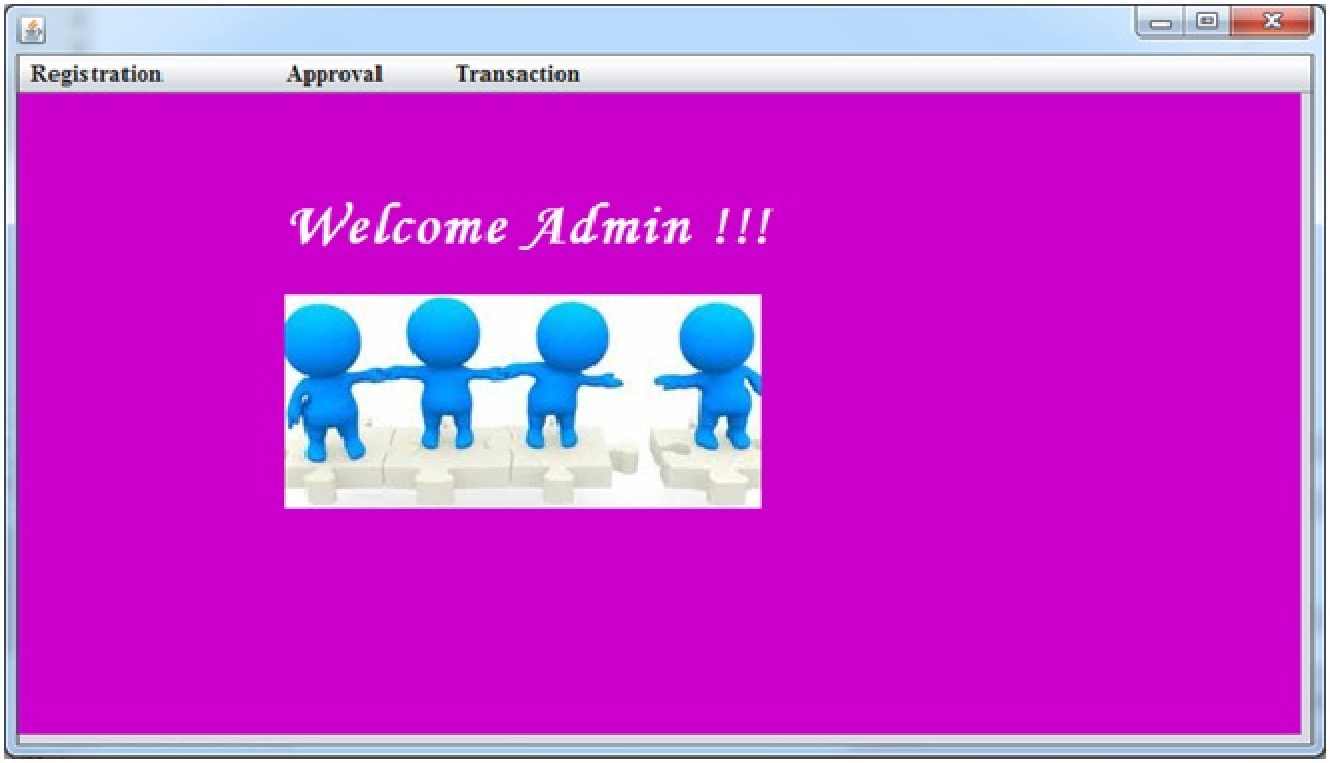


Figure 9.8: Admin Page
This is the Admin page where the Admin can create a new user,give approval to the user and can receive,capture and transmit videos.

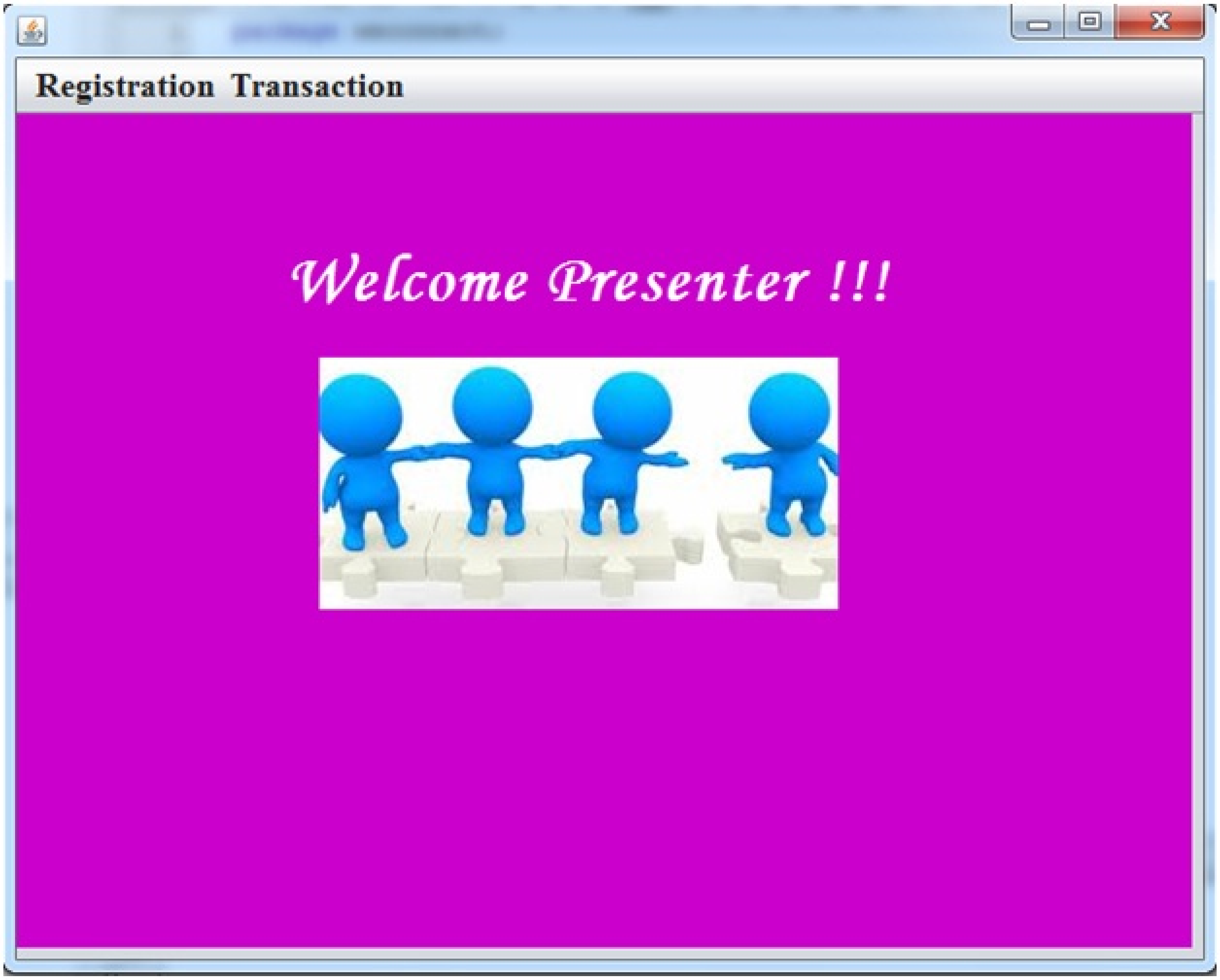


Figure 9.9: Presenter Page
This is the presenter page where the selected presenter can present in the forum . The approval to be the presenter is given by the Admin.

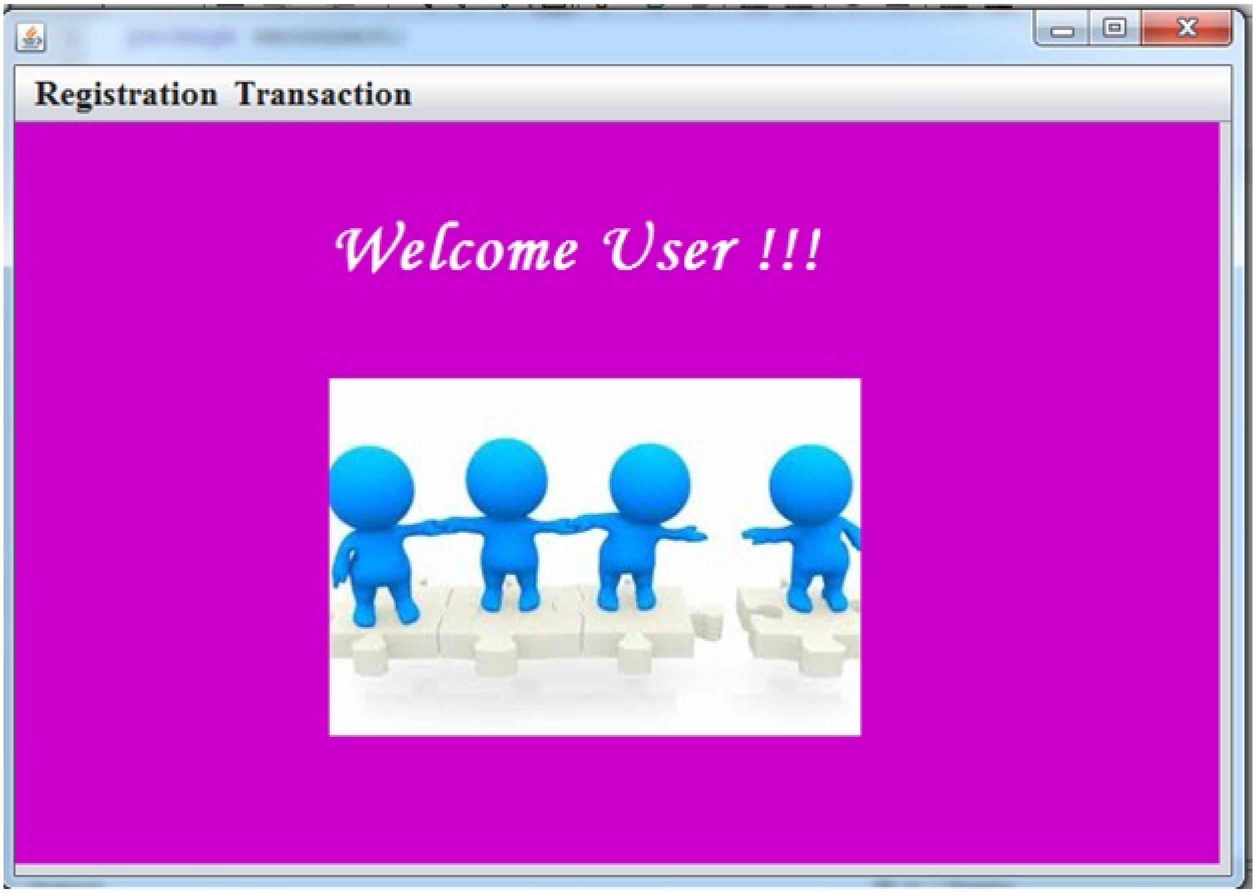
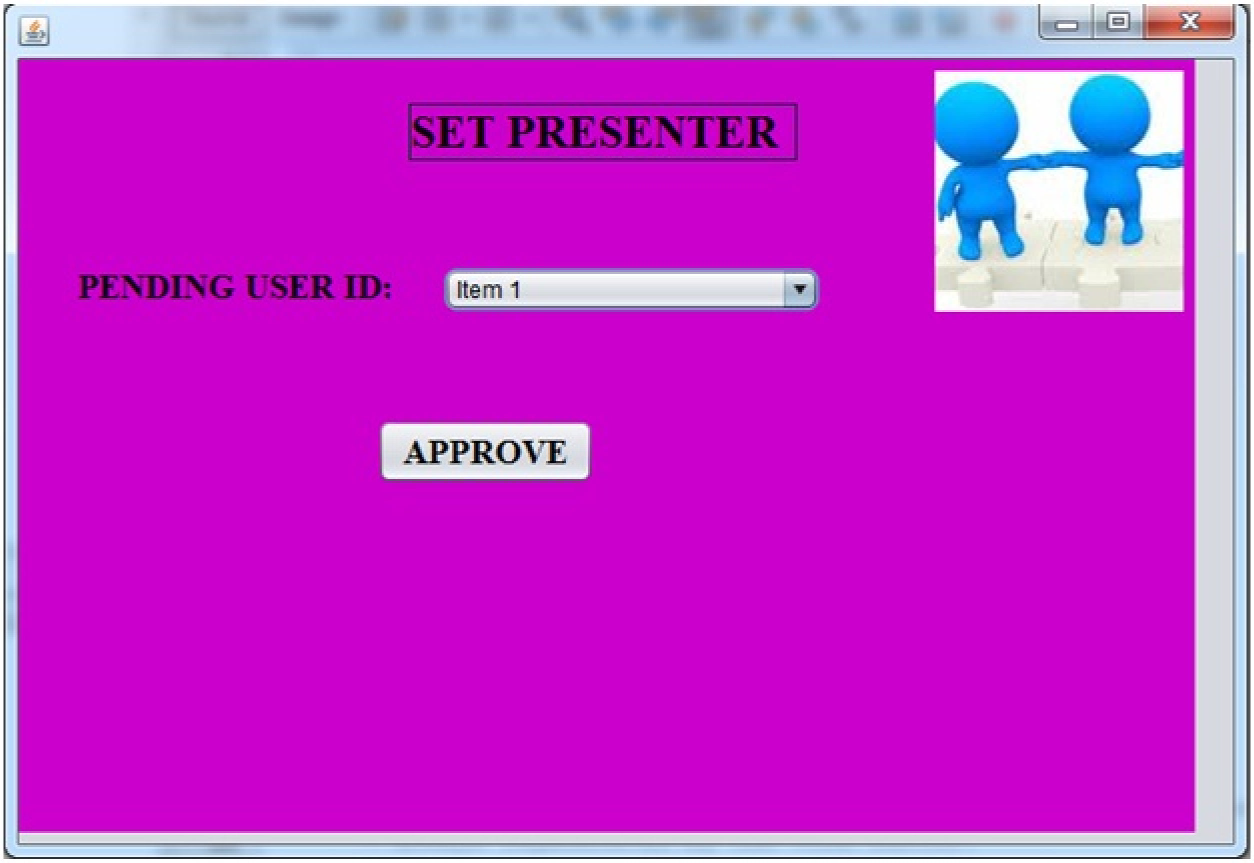


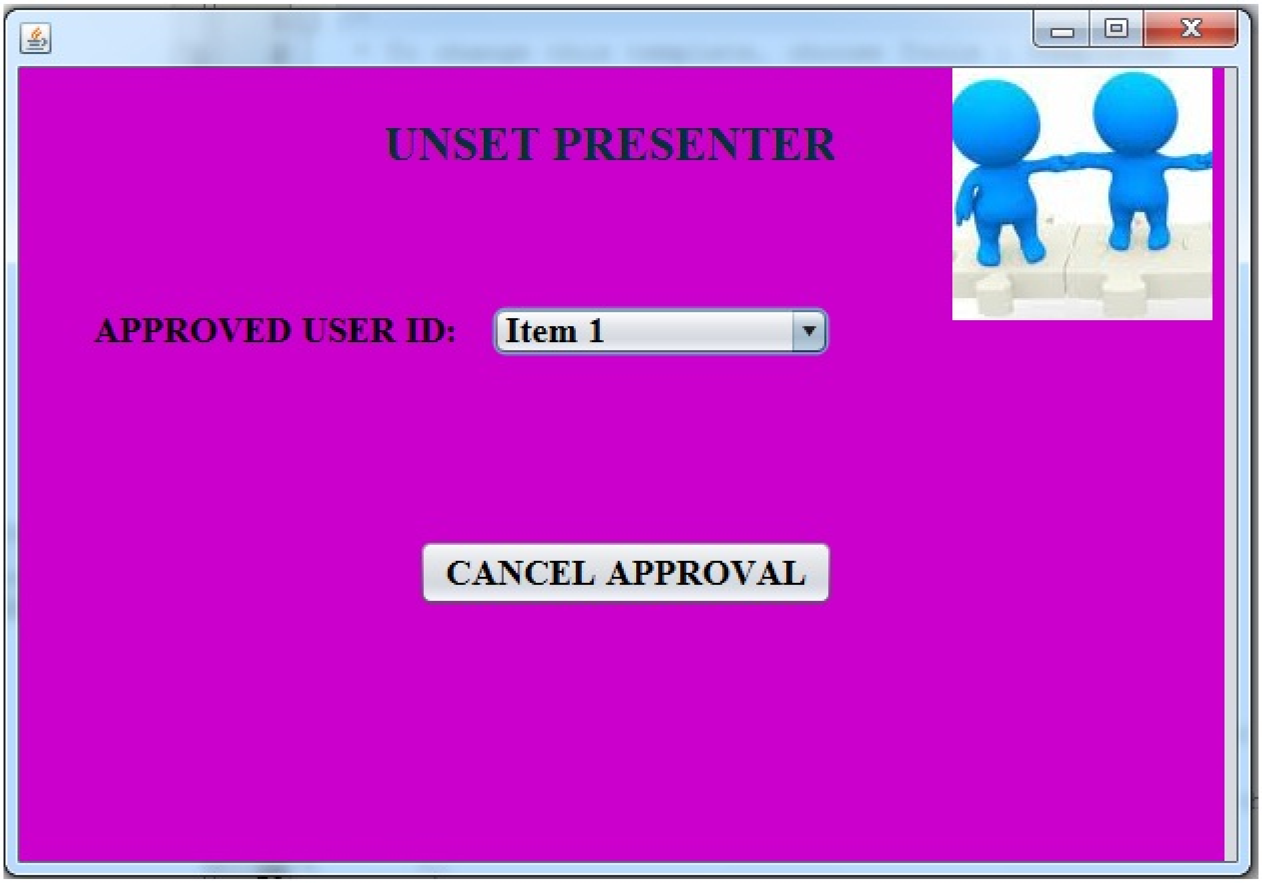
Figure 9.10: User Page
This is the user page where an user after getting logged in can create a new account,edit his information and can receive and capture videos.



SET PRESENTER

PENDING USER ID: Item 1

Figure 9.11: Set Presenter Page
This is the set presenter page visible only to the Admin. It is viewed when the Admin decides to make an user as presenter. So here all the pending requests by the users who want to be the presenter will be viewed.



UNSET PRESENTER

APPROVED USER ID: Item 1

Figure 9.12: Unset Presenter Page
This is the unset presenter page where the Admin unsets an existing presenter when the presenter finishes his work. It shows all the set presenter list so that the Admin can unset it.

Chapter 10

Conclusion

We have described our experiments or the project as an application for audio/video chat based on Java platform using tools Java Media Framework and Real-time Transport protocol.

Due to unavailability of time we have only implemented Voice Chat and Video Conferencing which helps the user to converse with the faculties, friends and others within the same university. However this is not the end. In future, more application can be developed in the field of audio and video reception and transmissions on the Java platforms or other platforms.

This project been designed as an application, which satisfies the communication gap between the students and faculties. It will also provide a good user interface and a good way of developing better communication between the students and faculties of an university. This project will help the students to do their assignments successfully, to get their doubts cleared by their teachers and for other learning purposes too. Thus by using this project, conversation can be implemented between students, friends and faculties too.

Overall, the Intelligent Video conferencing project is progressing well. Video conferencing is a system which is designed with huge potentials to education and the communication systems in general. However being innovative and dynamic it requires a degree of commitment from whoever makes use of it. It needs systematic design and preparation to be fully successful; otherwise it will be yet again another useful technological resource which represents nothing more than a piece of equipment. Accurate planning is needed not only at the design level but also at a technical and pedagogic level. Keeping the primary stake holders targeted for each individual video conferencing session helps establish clear aims and objectives which are the root of the success of such a system, in whichever way and set up it might be used. Making the system as transparent as possible to the participants helps not only increase their confidence in the system, but also helps them to attain the level of skills and satisfaction which learners can only achieve through actively participating in their own learning process.

Chapter 11

Future Work

Chat Rooms:

Due to time limitation we have just use a single chat room. The application can be expanded to include more chat rooms. So the clients can chat or conference in particular chat-room they intend and each client have list of logged clients for the room where he is currently doing activities.

Voice-mail:

We can introduce a new feature called Voice-mail in the application. Voice-mail is a facility which can be used by a client to send voice-message to a target client which was in off-line at the time when the message was sent. The target client can listen voice-messages sent by different clients whenever it logs in.

Video-mail:

We can have another interesting feature known as Video-mail. In Video-mail a client will record a video-clip and send to the recipient client which is in off-line state. The recipient client will see the video-messages when it logs in.

File Transfer:

We want to give facility of transferring files other than between the clients. The files may be a text document, worksheets and other types of files. Various clients that are having voice or video conversation can send and receive files to one another.

Internet:

We want to apply our project in such a way, so that we can access the audio/video files through inernet.

Developing as an application:

We want to convert our work or project into an application which can be downloaded though the internet service and can be used by the users in their respective PCs or mobile devices or any other devices.

Instant messaging and peer-to-peer communications:

While most instant messaging systems use centralized servers to transmit all messages, some systems do offer peer-to-peer messaging.In such a model, clients contact the IM server to locate other clients.Once the client chat program has located its peer, it contacts the peerdirectly.The peer-to-peer scheme offers better security than the client-server-client scheme when both users re on the same local area network because messages do not travel over the Internet.However, if one users located outside the corporate network, messages sent between machines are exposed to potential droppers, just assign the client-server-client scheme.

Instant Messaging and Chatting:

Instant messaging (IM) and Internet chat communication have seen enormous growth over the last several years. IM is the private network communication between two users, whereas a chat session is the network communication between two or more users. Chat sessions can either be private, where each user is invited to join the session, or public, where anyone can join the session. There are on the order of 100 million Internet IM users, where a user is defined as a unique name on one of the major public IM networks AOL Instant Messenger (AIM), Microsoft Messenger (MSN), or Yahoo! Messenger (YMSG). To date, little has been documented about the network protocols used by these systems. The protocols are not standardized, many of them are proprietary, and they are even seen as a control point in this business by the companies involved. This is demonstrated by the repeated attempts of the IM services to lock out users of other systems, in an attempt to keep their customers private. However, enough information is available to determine the broad characteristics of these systems. We have also used packet tracing of IM traffic in order to glean further details into these protocols and systems.

Bibliography

[1] http://en.wikipedia.org/wiki/HTML

[2] http://en.wikipedia.org/wiki/CSS

[3] http://en.wikipedia.org/wiki/JavaScript

[4] www.navicat.com

[5] http://dataflowdiagram.blogspot.in/2009/10/symbol-used-in-data-flow-diagram-dfd.html

[6] http://en.wikipedia.org/wiki/NetBeans

[7] www.servlet.com