VIDEO CONFERENCING SYSTEM

By

S. SRINIDHI (1518106087)

of

SONA COLLEGE OF TECHNOLOGY

Salem – 636 005

A PROJECT REPORT

Submitted to the

U15IT604R SOFTWARE DESIGN AND TESTING LABORATORY

In partial fulfillment of the requirements

for the award of the degree

of

BACHELOR OF TECHNOLOGY

IN

INFORMATION TECHNOLOGY

ANNA UNIVERSITY

CHENNAI

APRIL 2021

BONAFIDE CERTIFICATE

Certified that this project report title **VIDEO CONFERENCING SYSTEM** is the bonafide work of **S. SRINIDHI** who carried out the project work under my supervision.

Mr.P.IYANNAR., M.E., Ph.D,
Assistant Professor (Senior Grade),
Department of Information Technology,
Sona College of Technology,
Salem-636005.

Dr.J.AKILANDESWARI,
Professor and Head
Department Of Information Technology
Sona College of Technology,
Salem-636005.

Submitted for End Semester Practical Examination held on ……..

INTERNAL EXAMINER **EXTERNAL EXAMINER**

TABLE OF CONTENTS

CHAPTER NO	TITLE	PAGE NO
	LIST OF FIGURES	5
1	**PROBLEM DEFINITION**	7
2	**PROBLEM SCOPE**	8
3	**SOFTWARE REQUIREMENTS SPECIFICATION**	15
4	**SYSTEM ANALYSIS**	20
	4.1 ATTRIBUTE MATRIX	20
	4.2 TRACEABILITY MATRIX	21
	4.3 TRACEABILITY TREE	23
5	**SYSTEM DESIGN**	26
	5.1 USE CASE DIAGRAM	26
	5.2 ACTIVITY DIAGRAM	27
	5.3 CLASS DIAGRAM	32
	5.4 SEQUENCE DIAGRAM	37
	5.5 COLLABORATION DIAGRAM	41
	5.6 COMPONENT DIAGRAM	44
	5.7 DEPLOYMENT DIAGRAM	50
6	**SYSTEM DEVELOPMENT**	51
	6.1 VIDEO CALLS	51
	6.2 CREATE ROOM	51
	6.3 CHAT	51
	6.4 SCREEN SHARING	51
	6.5 LEAVE MEETING	51
7	**SYSTEM IMPLEMENTATION**	52
	7.1 SYSTEM PRE-REQUISITES	52
8	**SYSTEM TESTING**	53

	8.1 TEST PLAN	53
	8.2 TEST CASES	53
	8.2.1 Unit Testing	53
	8.2.2 Functional Testing	53
	8.3 TEST REPORTS	54
	8.3.1 Unit Testing	54
	8.3.2 Functional Testing	54
9	**CONCLUSION AND FUTURE ENHANCEMENT**	55
APPENDIX 1	**SAMPLE CODE**	56
APPENDIX 2	**SAMPLE SCREENSHOTS**	66

LIST OF FIGURES

FIGURE NO	TITLE	PAGE NO
2.1	Block Diagram	11
3.1	Video Conference	14
4.1	Stack Holder Requirement Attribute Matrix	20
4.2	Use Case Requirement Attribute Matrix	21
4.3	Stack Holder Vs Stack Holder Requirement	22
4.4	Use Case Vs Use Case Requirement	22
4.5	Use Case Vs Stack Holder Requirement	23
4.6	Stack Holder Requirement Traceability Tree	24
4.7	Use Case Requirement Traceability Tree	25
5.1	Use Case Diagram	26
5.2	Login Activity Diagram	27
5.3	Create Room Activity Diagram	28
5.4	Screen Sharing Activity Diagram	29
5.5	Mute Audio/Video Activity Diagram	30
5.6	Logout Activity Diagram	31
5.7	Login Class Diagram	32
5.8	Create Room Class Diagram	33
5.9	Screen Sharing Class Diagram	34
5.10	Mute Audio/Video Class Diagram	35
5.11	Logout Class Diagram	36
5.12	Login Sequence Diagram	37
5.13	Create Room Sequence Diagram	38
5.14	Screen Sharing Sequence Diagram	39
5.15	Mute Audio/Video Sequence Diagram	39
5.16	Logout Sequence Diagram	40
5.17	Login Collaboration Diagram	41
5.18	Create Room Collaboration Diagram	42
5.19	Screen Sharing Collaboration Diagram	42

5.20	Mute Audio/Video Collaboration Diagram	43
5.21	Logout Collaboration Diagram	43
5.22	Main Component Diagram	44
5.23	Login Component Diagram	45
5.24	Create Room Component Diagram	46
5.25	Screen Sharing Component Diagram	47
5.26	Mute Audio/Video Component Diagram	48
5.27	Logout Component Diagram	49
5.28	Deployment Diagram	50

CHAPTER-1

PROBLEM STATEMENT

The main purpose of system is to enable face-to-face communication between two or more people in different locations. Video conferencing boosts productivity, saves time, reduces travel expenses, and overall promotes collaboration. The advantage of video conferencing is the ability to facilitate all of those benefits without requiring constant travel for face-to-face communication. It is a popular alternative to phone conferencing for businesses and provides individual users with an inexpensive means of communication with distant friends and family. People from around the world can connect through many different ways. They can connect through one or two way audio, one or two way video, one or two way audio and video or through text. Basically, the purpose of making the video conference system project is to flow its main features and update the new features in it.

CHAPTER-2

PROJECT SCOPE

The future of video conferencing promises more than reliable, high-quality calls. While call quality is important and most vendors support high-definition video, video conferencing is more than just video. Video conferencing system primarily can be seen in business. No matter what kind of business one is running, unless one is a very localized group without much travel or remote working involved, video conferencing has probably at least come up as a point of discussion. It's a great way to get mobile workers involved, to cut down on expenses related to traveling, and to facilitate better communication between businesses, partners, and/or clients.

CHAPTER-3

SOFTWARE REQUIREMENT SPECIFICATION

1. Introduction

1.1 Purpose

The main purpose of system is to enable face-to-face communication between two or more people in different locations. Video conferencing boosts productivity, saves time, reduces travel expenses, and overall promotes collaboration. The advantage of video conferencing is the ability to facilitate all of those benefits without requiring constant travel for face-to-face communication. It is a popular alternative to phone conferencing for businesses and provides individual users with an inexpensive means of communication with distant friends and family. People from around the world can connect through many different ways. They can connect through one or two way audio, one or two way video, one or two way audio and video or through text. Basically, the purpose of making the video conference system project is to flow its main features and update the new features in it.

1.2 Document Conventions

The document is prepared using Microsoft Word 2013 and has used the font type Times New Roman'. The fixed font size that has been used to type this document is 12pt with 1.5 line spacing. It has used the bold property to set the headings of the document.

1.3 Intended Audience and Reading Suggestions

The document is intended for requirements engineer, domain expert, developer and project manager. The SRS document can be used in any case regarding the requirements of the project and the solutions that have been taken. The document would final provide a clear idea about the system that is building.

Brief outline of the document is,

1. Overall Description

2. System Features

3. External Interface Requirements

4. Non Functional Requirements

1.4 Product Scope

The future of video conferencing promises more than reliable, high-quality calls. While call quality is important and most vendors support high-definition video, video conferencing is more than just video.

Video conferencing system primarily can be seen in business. No matter what kind of business one is running, unless one is a very localized group without much travel or remote working involved, video conferencing has probably at least come up as a point of discussion. It's a great way to get mobile workers involved, to cut down on expenses related to traveling, and to facilitate better communication between businesses, partners, and/or clients.

Video conferencing is also making a name for itself in the world of education. Teachers are now able to leverage this technology to spread knowledge across much wider fields. Online lectures, interactive virtual classrooms, and other uses of this technology are taking the education industry by storm and truly improving on how we can spread information and promote higher learning.

It's also being used in the health care industry. Whether it's with remote doctors communicating with one another to determine strategies for treatment, or it's doctors treating patients remotely through video connectivity, video conferencing has increased options for health care around the world.

1.5 References

More information about the project, anomaly detection techniques used and improvement techniques is available

1. http://www.webconference.com
2. http://www.solutionzinc.com/videocommunication-basics/what-is-video-conferencing.html
3. http://www.businessdictionary.com/definition/video-conferencing.html#ixzz2Nt4vw8lL
4. http://vijaydev.wordpress.com/2009/02/13/using-jodconverter/
5. http://www.artofsolving.com/opensource/jodconverter

2. Overall Description

2.1 Product Perspective

The software product being developed is for providing video chat. Users can also exchange video messages, screen share as well as create conference calls. This system is available on Microsoft Windows, as well as Android and tablets. This is based on a premium model.

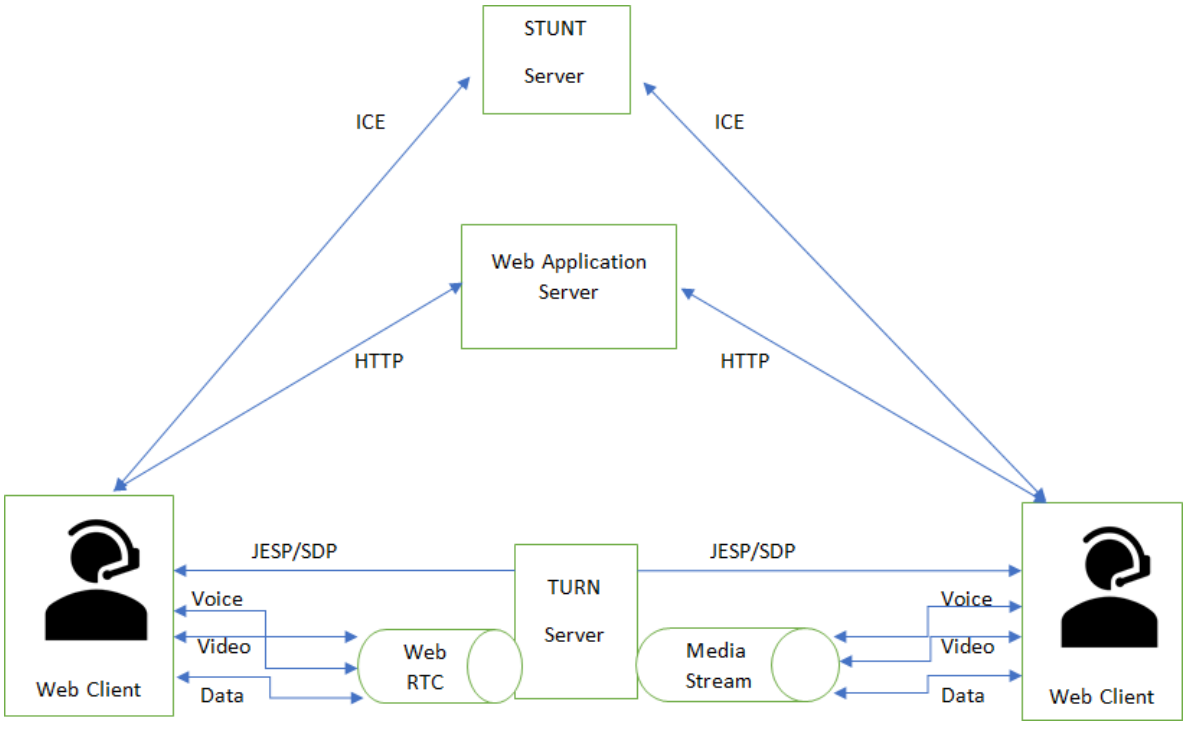


Fig 2.1 Block Diagram

2.2 Product Functions

- **Video calling :** Video Calling lets a person to talk with others and lets them see and hear you. You can make video calls to and receive calls from people.
- **Participating in group calls :** Allows users in different locations to hold face-to-face meetings.
- **Mute video :** One can turn off and on the video using this option.
- **Mute audio :** One can turn off and on the mic using this option.
- **Screen sharing :** The best video conferencing tools allow you to share your screen display, as well as any applications running on your computer, with other participants in your web presentation or webinar.

2.3 User Classes and Characteristics

This is especially true for video conferencing, which is saddled with a reputation for being difficult, cumbersome and ineffective. To overcome that false perception, new video technologies must sport interfaces that work at the touch of a button—every time and for every user. Launching a video conference should literally be as easy as placing a phone call or sending a text message. Finding and using features within the conferencing application should not require time or extra thought. And devices and applications should be designed in a way that makes people feel in command of the technology, and therefore want to use them even more.

2.4 Operating Environment

Environmental requirements of video conferencing for Business Server is a wider spectrum of things that need to be ready in your infrastructure before deploying Business Server. The target operating system of the application is Windows that support all webRTC in desktop and mobile browsers.

2.5 Design and Implementation Constraints

- Anti webrtc corporate or regulatory policies.
- Hardware limitations ie. no mic or camera.

- Software limitations ie. no or unsuitable chrome browser.
- RAM and memory requirements.
- Organization has blocked access to TFX signalling server or TURN server.

2.6 User Documentation

User manual and CD will be made available for troubleshooting and help. The user manual will contain detailed information about the usage of the product from a layman perspective to an expert network/system administrator. The manual shall also be made available online

2.7 Assumptions and Dependencies

The proposed solution will be designed to work in an enterprise environment. The target environment may consist of wired and wireless links inside the network.

- The users should have webrtc supported browser and allow VOIP communication in enterprise policies.
- User must give media permission for the camera and microphone access for video sessions.
- Permissions should be given to share screen when asked by screen sharing widget.

3. External Interface Requirements

3.1 User Interfaces

A graphical user interface is available providing following functionalits

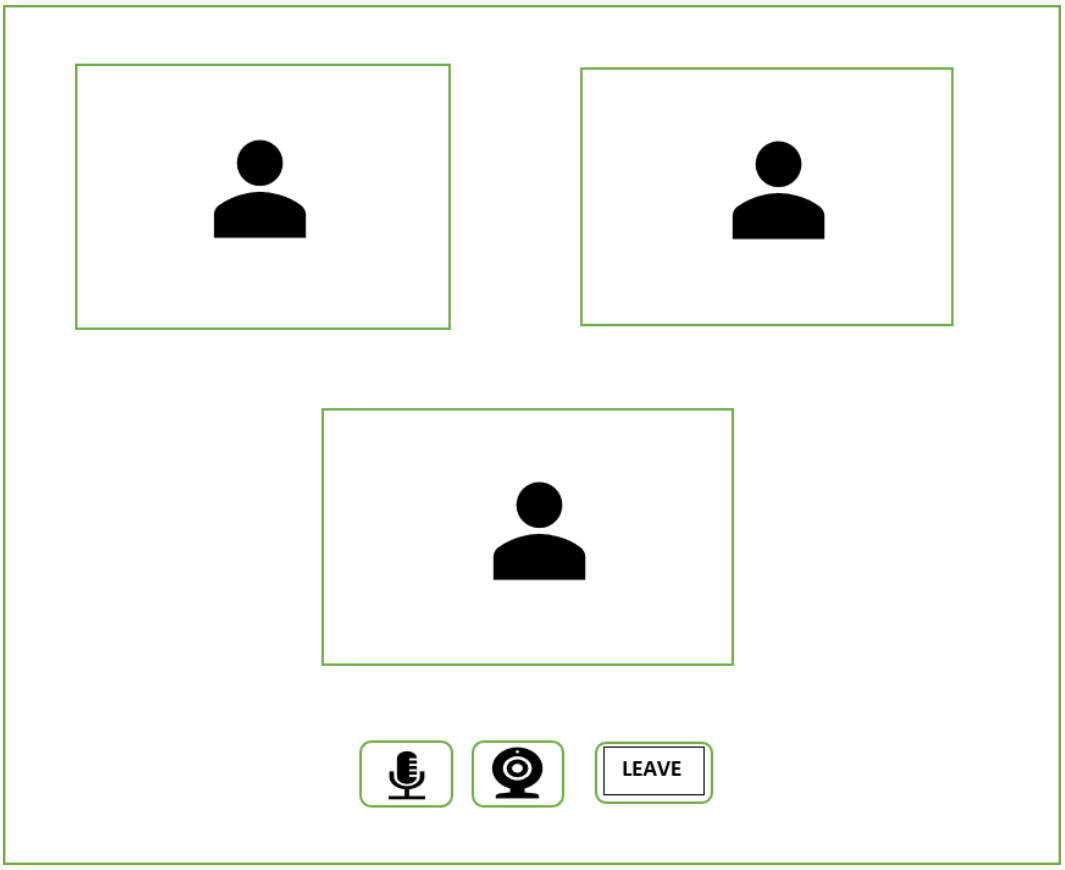


Fig 3.1 Video Conference

3.2 Hardware Interfaces

This application makes use of the following hardware devices :

- Network connections
- Webcams
- Microphones
- Monitors & Display Screen
- Routers
- Speakers
- Windows and Linux (any distribution) client computers
- Cell phones and PCs

3.3 Software Interfaces

- Client on Internet

- Client on Intranet
- Web Browser
- Web Server
- VS Code Editor

3.4 Communications Interfaces

- An internet connection (broadband is best - find out how much bandwidth you need).
- Speakers and a microphone (mobile phones, tablets and most computers have these built in).
- A camera to make video calls (again, most devices have theses).

4. System Features

4.1 System Feature 1

The main feature of the system is video conference between members which will be very helpful to communicate through online.

4.1.1 Description and Priority

This technology is particularly convenient for business users in different cities or even different countries because it saves time, expenses, and hassles associated with business travel. Uses for video conferencing include holding routine meetings, negotiating business deals, and interviewing job candidates.

4.1.2 Stimulus/Response Sequence

By clicking start call the user is very convenient in operating the calls by just one click. So it will also enhance market of video conferencing.

S.No	User Action	System Response
1.	Open home page	Home page will be opened

2.	Member conversation	Conversation will be shown
3.	Start call	Call will be connected
4.	Leave call	Call will be disconnected

4.1.3 Functional Requirements

Video calls

Online video conferencing can integrate these communications platforms through a set of telecommunication technologies which allow two or more locations to communicate by simultaneous two-way video and audio transmissions. This system can give you a competitive advantage that the ability to call through online by providing the username or user id and be able to connect to the members.

4.2 System Feature 2

The second main feature of the system is screen sharing which will be very useful to show demos or take lectures through online mode.

4.2.1 Description and Priority

Screen sharing is a common name for technologies and products that allow remote access and remote collaboration on a person's computer desktop through a graphical terminal emulator.

4.2.2 Stimulus/Response Sequence

By selecting screen share option the user is very convenient in sharing their screen. So it will useful for one to explain their presentations.

4.2.3 Functional Requirements

Screen Sharing

When you start sharing your screen, the meeting controls will move into a menu

that you can drag around your screen.

New Share: Start a new screen share. You will be prompted to select which screen you want to share again.

Pause Share: Pause your current shared screen.

Annotate: Display annotation tools for drawing, adding text, etc.

When the sharing process is started and 80% of devices acknowledge receiving the shared screen, you will see a notification banner stating Participants can now see your shared screen, or whatever you have chosen to share.

- Zoom will automatically switch to fullscreen to optimize the shared screen view.
- To exit full-screen, click Exit Full Screen in the top-right corner or press the Esc key.
- To disable automatic full screen when viewing a shared screen, adjust the Window size when screen sharing behaviour in your desktop client settings.

5. Other Non-functional Requirements

5.1 Performance Requirements

Image data transfer through internet connection and live streaming makes performance measures crucial. For desired performance, image capturing, transfered data size, speed of connection, response time, processing speed must be considered. System should work real-time which means there should be an acceptable time delay such as max 4-5 seconds between request and response. Wearable device should have wifi adapter which is fast enough to transfer live camera feed to the web server. Web server should be able to handle multiple device and user connection. Web server may stream video at least 20 user at the same time. Image processing should be optimized so it should not take time more than 2 seconds. Web server should not process every frame and should determine whether process or not the frame.

5.2 Safety Requirements

Wearable device has a battery to supply power to the device. Our device designed to wear on user's head, therefore, battery is a serious safety issue and cause severe result on unfortunate events. Battery of the device should be covered by case or it should be placed on a belt or bag. In addition to that voltage levels on device should be adjusted. Noise or heat produced by the device should be minimized. In case of malfunction, system should shutdown itself and reboot in order to prevent unpredicted results

5.3 Security Requirements

Streaming the device camera on web makes security measurements crucial. Accessing and interacting with the streaming web server should be controlled and any misusage should be prevented. User authorization and data encryption are important security requirements of the project. User stream should not be available to anyone who is not authorized by the user of the device. System should store user data on database securely and set access permissions to the these datas carefully.

5.4 Software Quality Attributes

Video conferencing is used increasingly in many telemedicine applications, including medical personnel education, peer consultation, patient education, and direct patient care. Advances in technology and changes in medical care delivery have enhanced the ability to develop effective telemedicine video conferencing systems. Measures of effectiveness for technology systems rely on identified requirements for system quality. In this research, we propose a comprehensive model of quality attributes for telemedicine video conferencing systems. The quality attribute model is developed from an extensive literature review, direct observations of telemedicine encounters, and structured interviews with telemedicine experts. The model contains four quality attribute groups: Technical, Usability, Physical Environment, and Human Element. Interview citations are used to justify the importance of these individual quality attributes. Both researchers and practitioners can make use of the model to understand, design, and evaluate telemedicine video conferencing systems.

5.5 Business Rules

Check the tech such as WiFi, your video calling platform of choice, your specific settings, and any hardware in play.

- ➢ First, check your Internet. WiFi signal not strong enough for a stable video stream? Connect the Ethernet cable or mute your video to avoid distracting teammates with your background.
- ➢ Second, check the platform. Familiarize yourself with your settings to make sure your camera and microphone are accessible and functional.
- ➢ Third, check your specific tech. Video calls take a lot of CPU, so don't think about it—plug everything in. If you use headphones, headsets, or a speaker, make sure they're on, connected, and selected for audio input/output. It doesn't hurt to turn off Bluetooth on other devices for the meeting to prevent connection interference.

CHAPTER-4

SYSTEM ANALYSIS

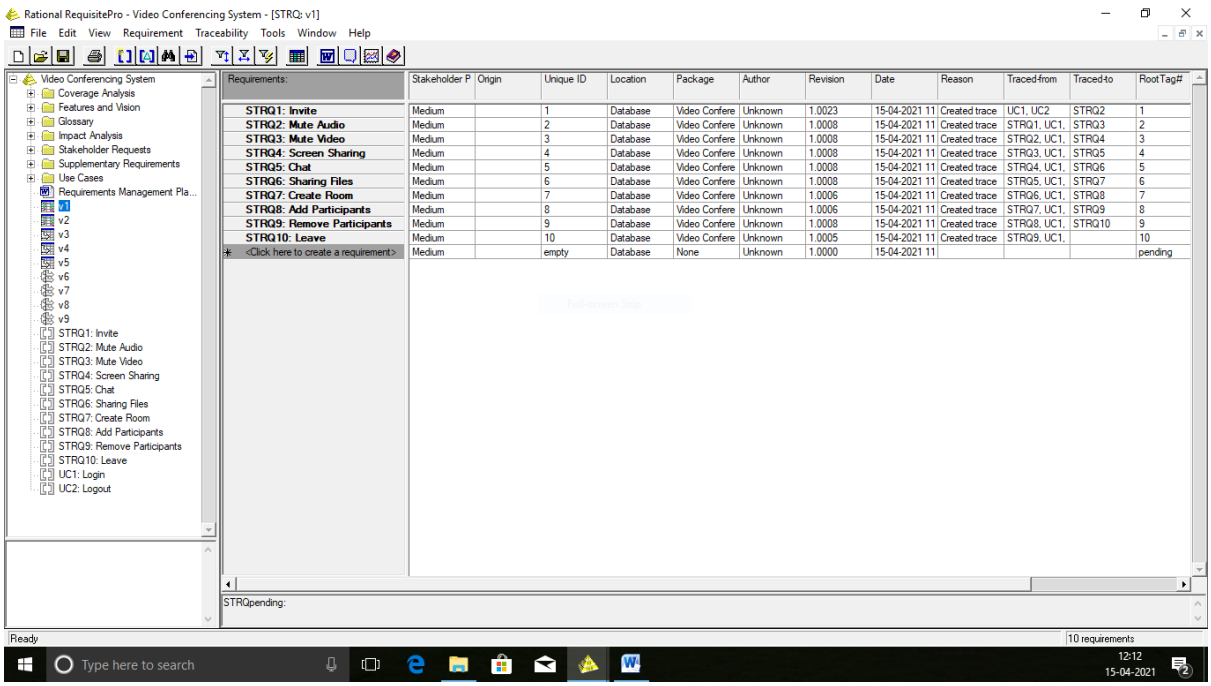
The next workflow in the RUP is the Analysis of the requirements which have been specified in the SRS. The Analysis is done with the help of Rational Requisite Pro. The three views or reports which form the basis for analysis are

1. Attribute Matrix
2. Traceability Matrix
3. Traceability Tree

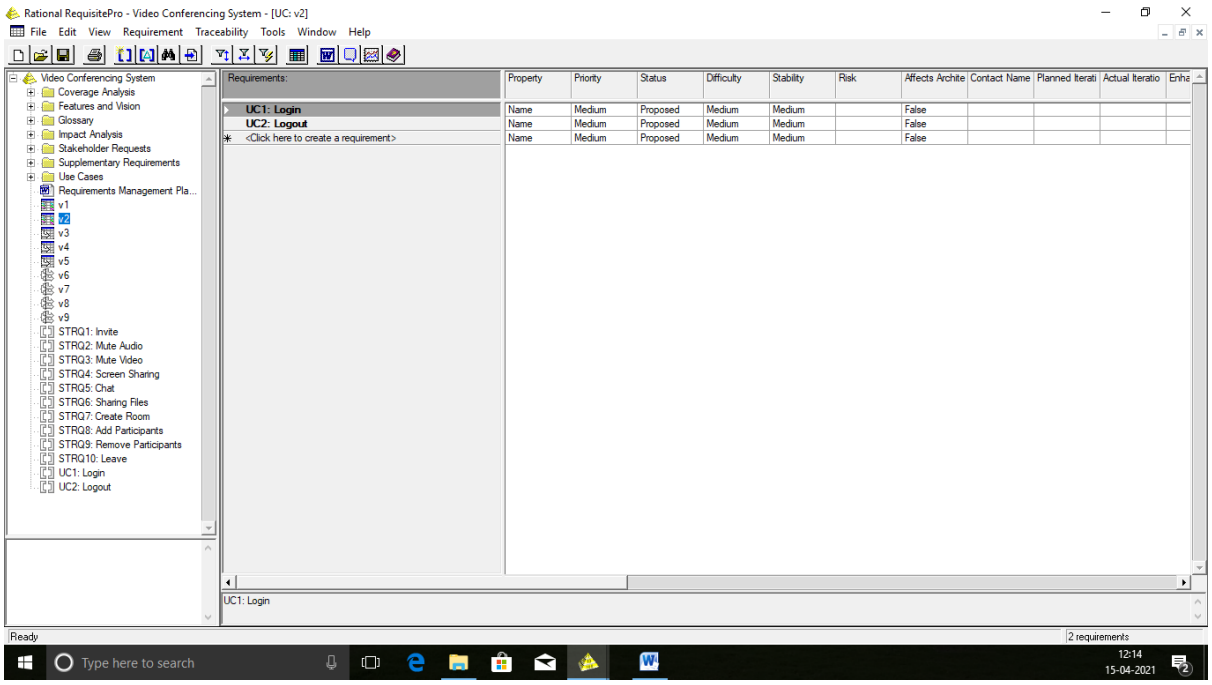
4.1 ATTRIBUTE MATRIX

The Attribute Matrix view is a spreadsheet like display that lists the requirements of a specific requirements type and their attributes. Requirements are arranged in rows, listed by tag number and followed by requirement name. Attributes are arranged in columns.

4.1.1 STACKHOLDER REQUIREMENT SPECIFICATION



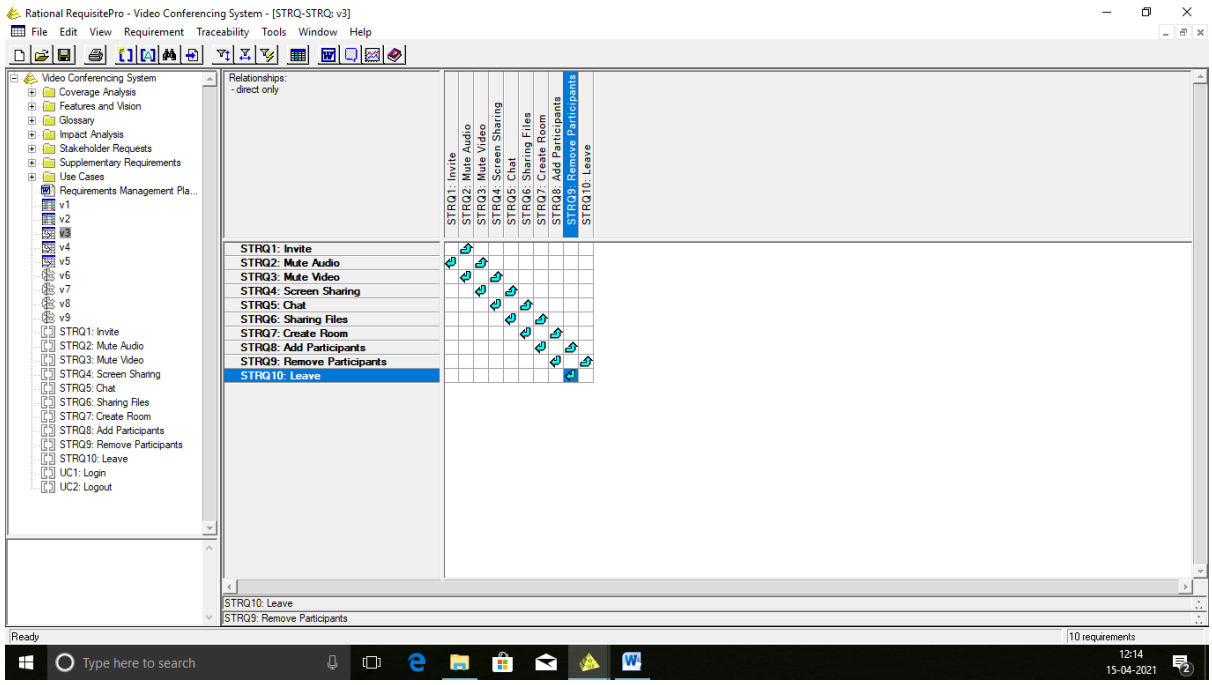
4.1.2 USECASE REQUIREMENT SPECIFICATION



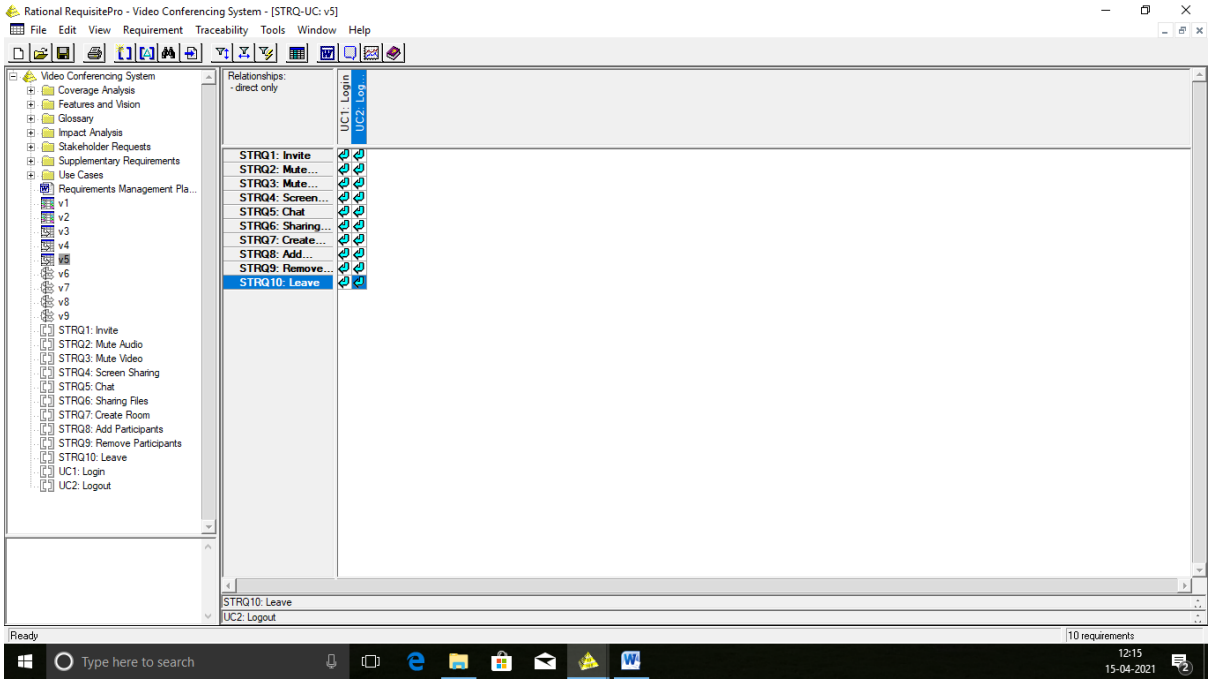
4.2 TRACEABILITY MATRIX

Traceability Matrix is a view that illustrated the relationships between requirements of the same or different types. We can use this matrix to create, modify and delete traceability relationships and view indirect relationships and view direct relationships and traceability relationships with a suspect state. We can also use the traceability matrix to filter and sort the requirements and columns requirements separately.

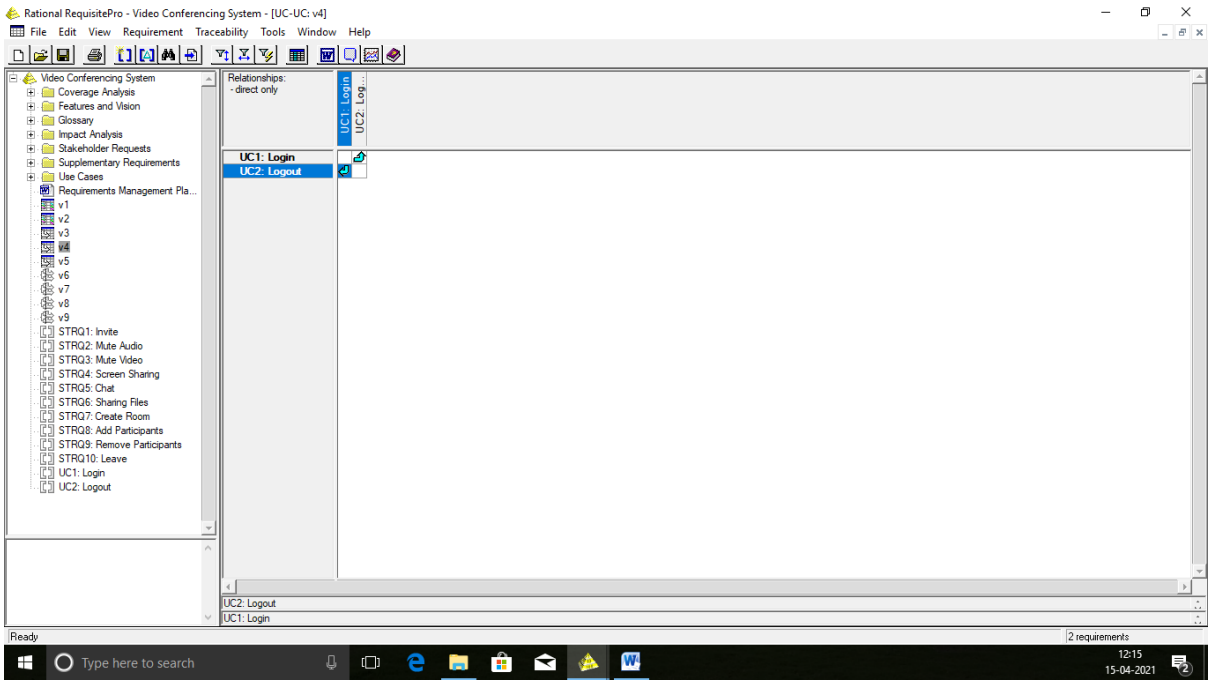
4.2.1 STACKHOLDER VS STACKHOLDER REQUIREMENT TYPE



4.2.2 STACKHOLDER VS USECASE REQUIREMENT TYPE



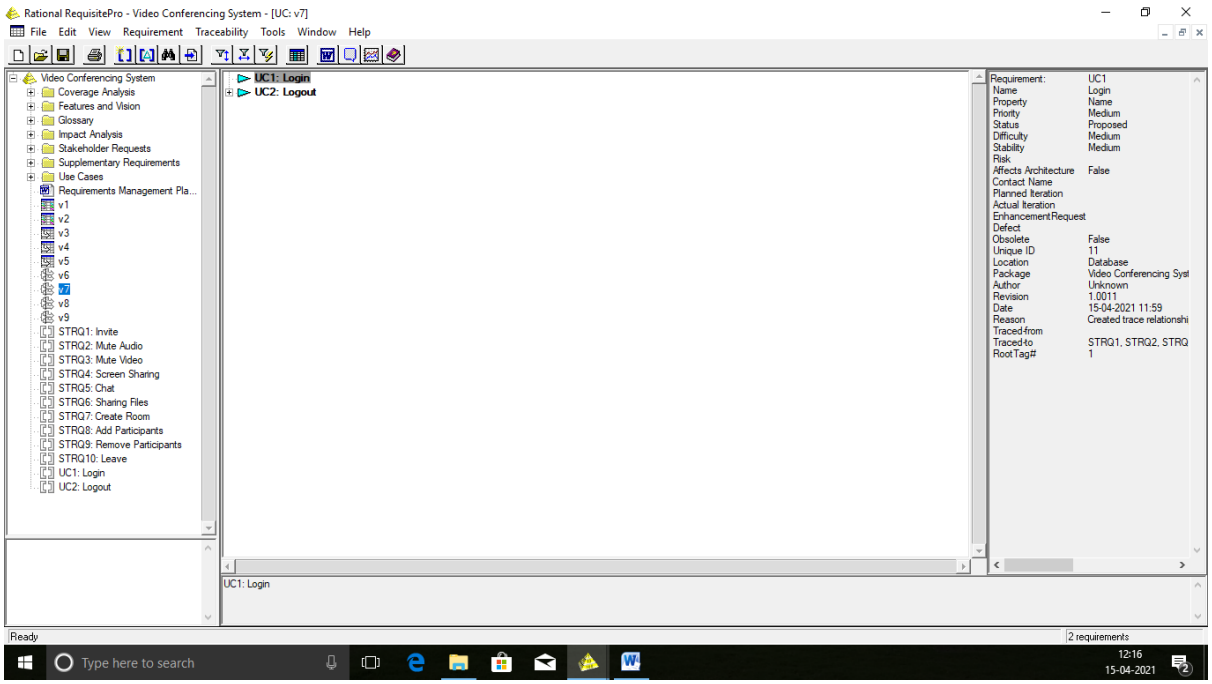
4.2.3 USECASE VS USECASE REQUIREMENT TYPE



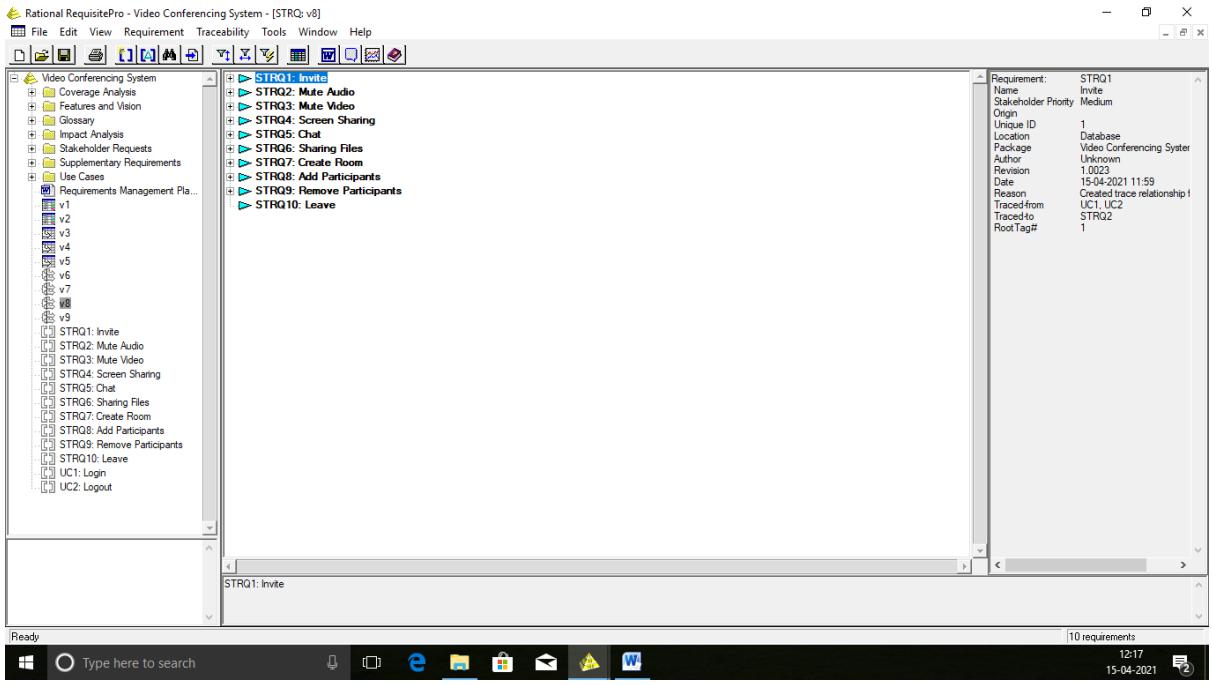
4.3 TRACEABILITY TREE

A view that displays all internal and external requirements traced to or from a requirement. The traceability tree only displays the first level project traceability.

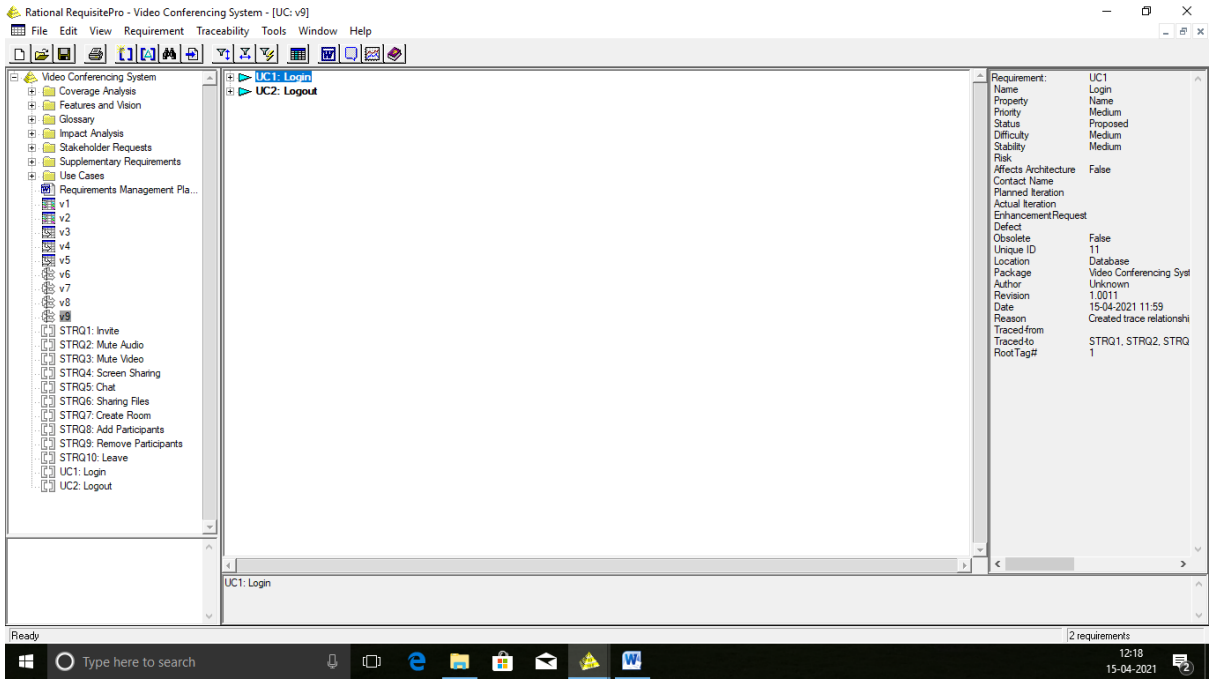
4.3.1 TRACEABILITYTREE(TRACEDINTO)-USECASE REQUIREMENT TYPE



4.3.2 TRACEABILITY TREE(TRACEDINTO)-STACKHOLDER REQUIREMENT TYPE



4.3.3 TRACEABILITY TREE(TRACEDOUTOFF)-STACKHOLDER REQUIREMENT TYPE



CHAPTER-5

SYSTEM DESIGN

5.1 USECASE DIAGRAM

A Use Case Diagram is a graph of actors, a set of use cases enclosed by a system boundary, communication association between the actors and the use cases and generalization among the use cases. A use case corresponds to a sequence of transactions, in which each transaction is invoked from outside the system and engages internal objects to interact with each other. An actor is anything that interacts with the use case.

5.1.1 USECASE DIAGRAM FOR VIDEO CONFERENCING SYSTEM

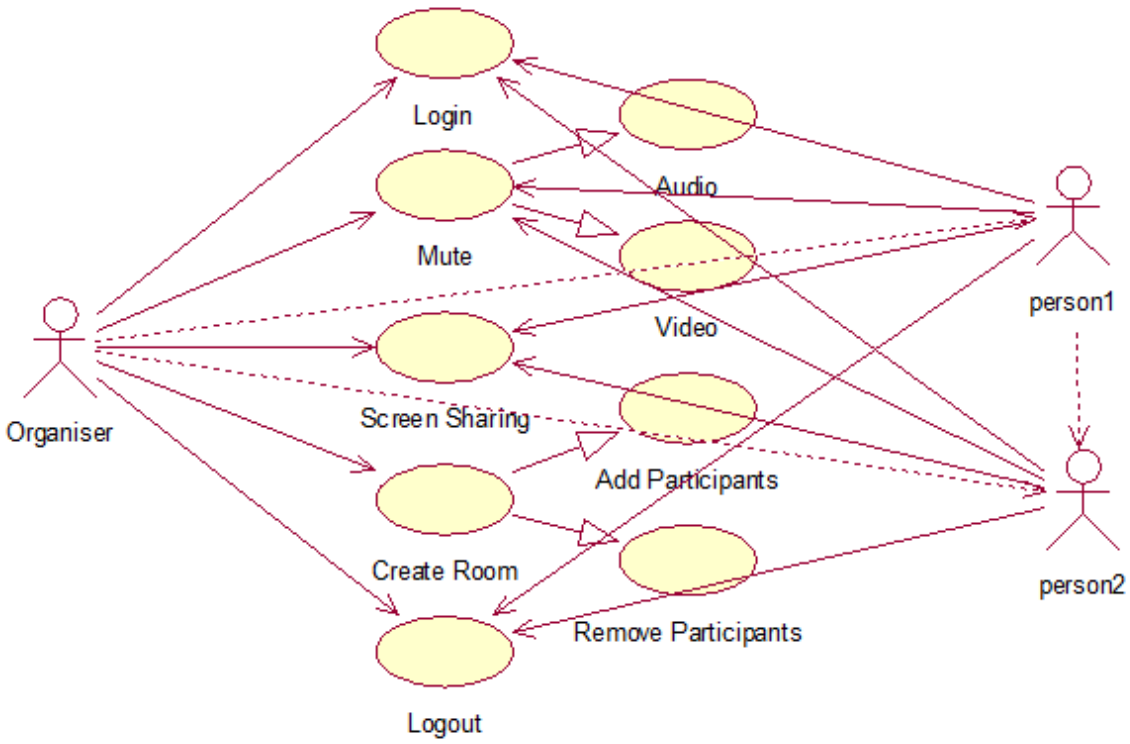


Fig 5.1 Use Case Diagram

5.2 ACTIVITY DIAGRAM

Activity Diagrams illustrate the dynamic nature of a system by modeling the flow of control from activity to activity. An activity represents an operation on some class in the system that results in a change in the state of the system. Typically, activity diagrams are used to model workflow or business processes and internal operation.

5.2.1 LOGIN

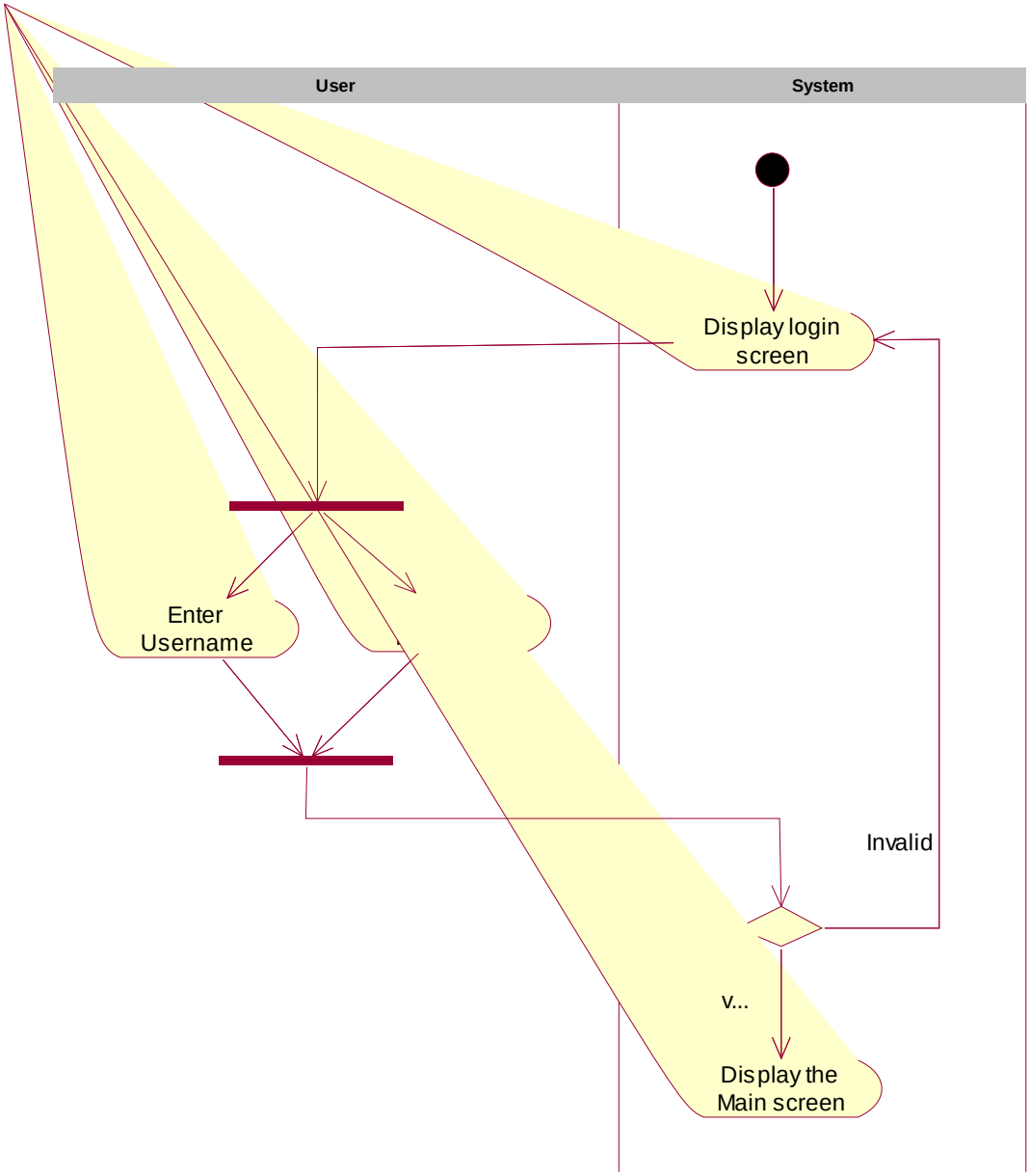


Fig 5.2 Login Activity Diagram

5.2.2 CREATE ROOM

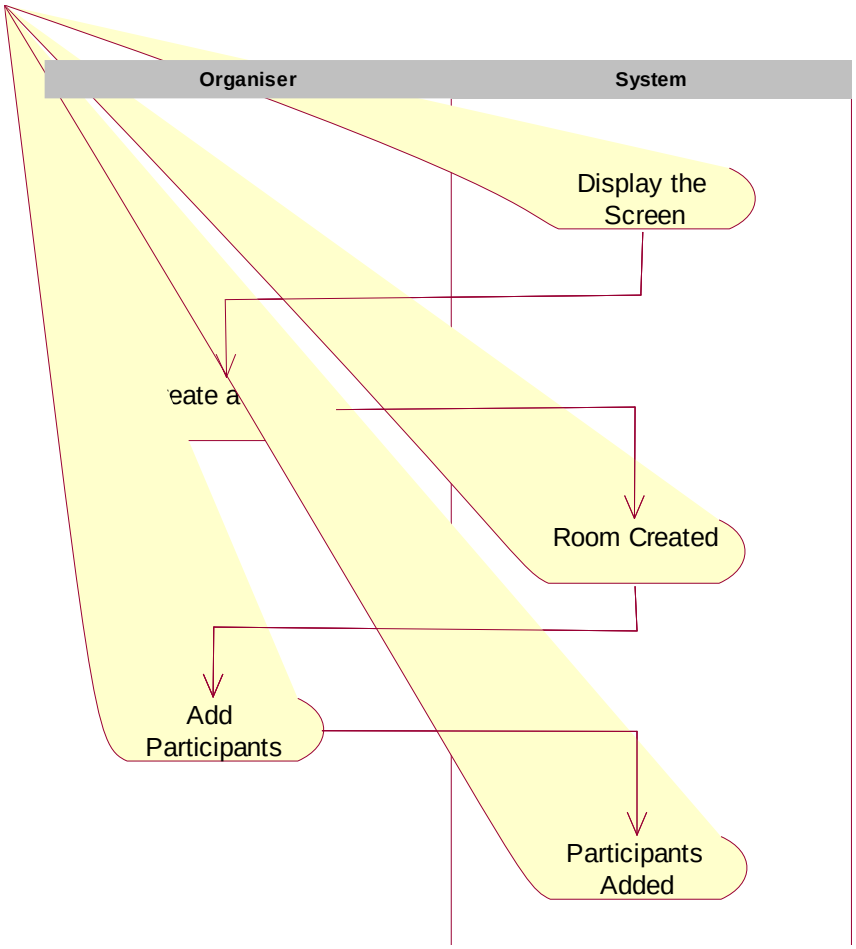


Fig 5.3 Create Room Activity Diagram

5.2.3 SCREEN SHARING

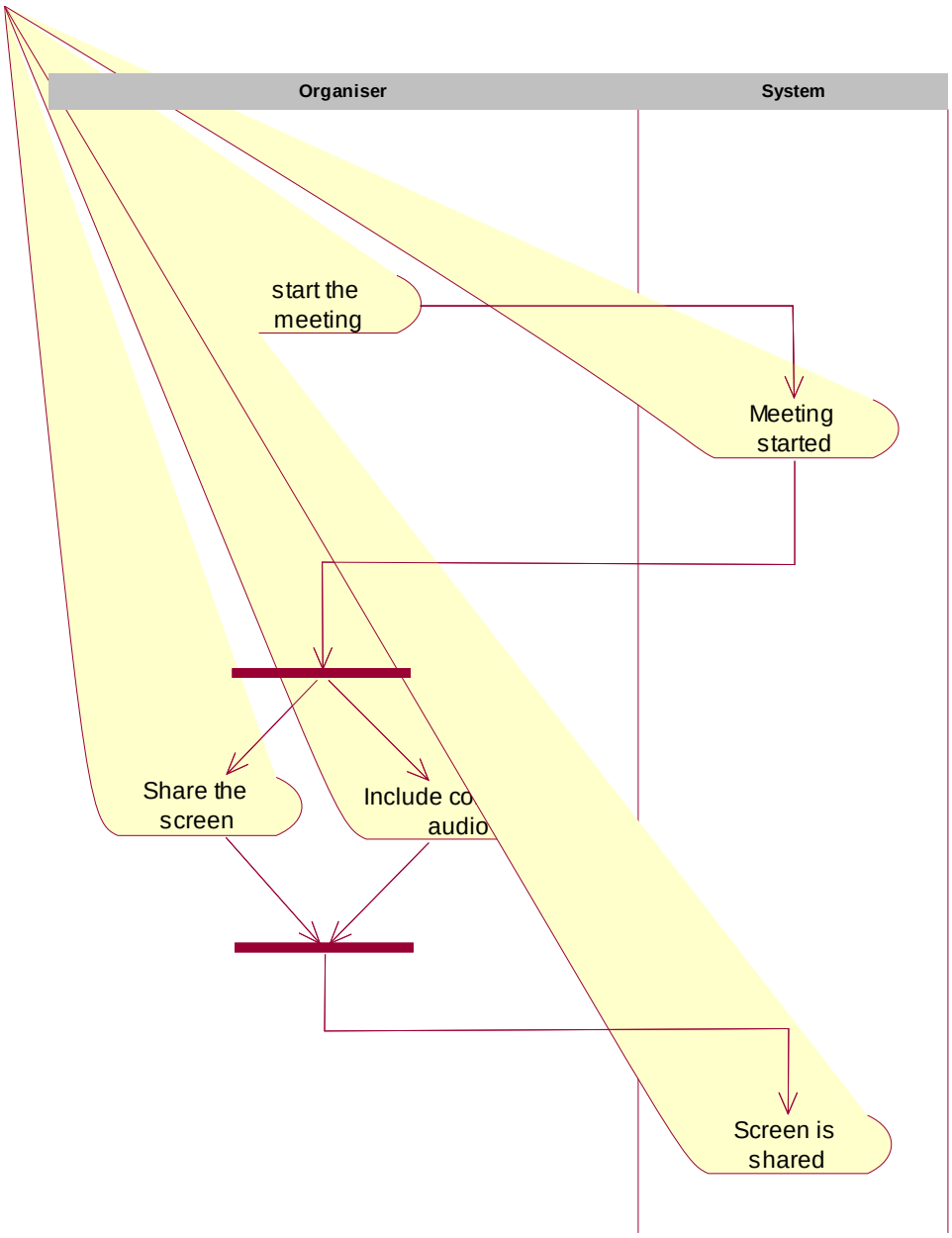


Fig 5.4 Screen Sharing Activity Diagram

5.2.4 MUTE AUDIO/VIDEO

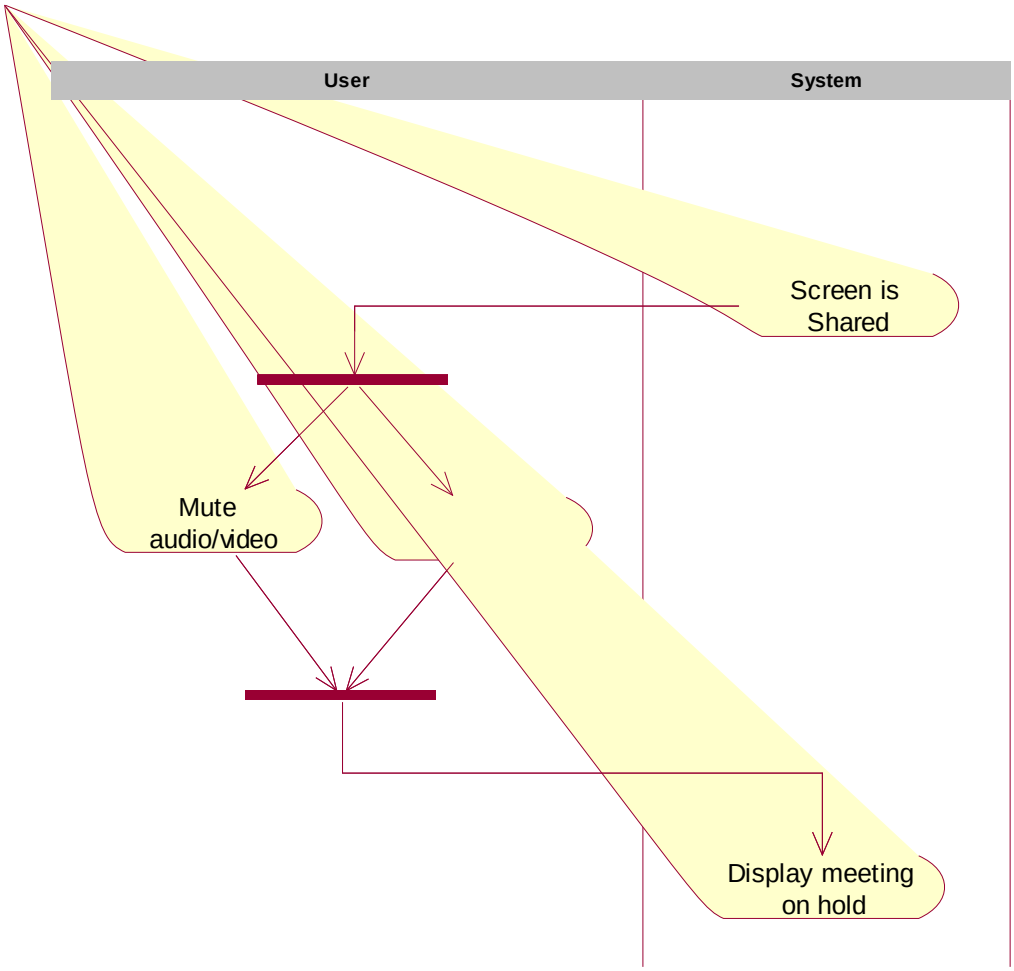


Fig 5.5 Mute Audio/Video Activity Diagram

5.2.5 LOGOUT

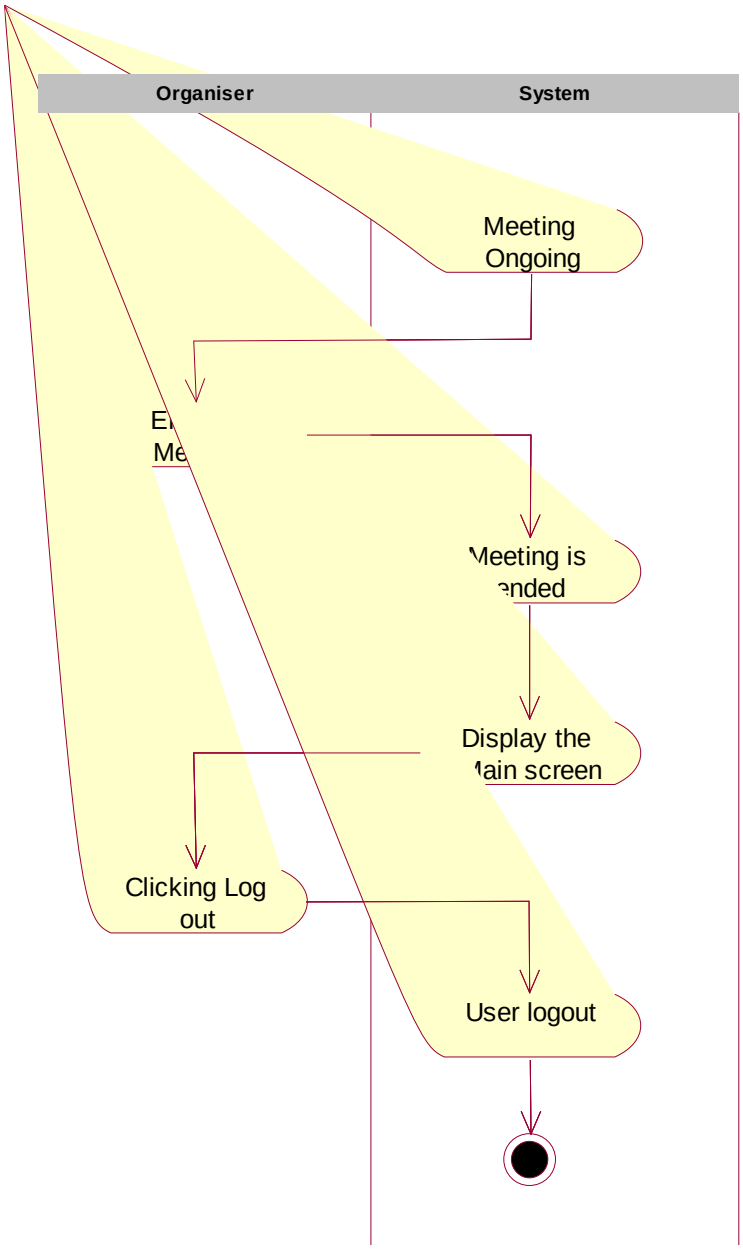


Fig 5.6 Logout Activity Diagram

5.3 CLASS DIAGRAM

A Class Diagram is a collection of static modeling elements such as classes and their relationships, connected as a graph to each other and to their contents. These diagrams show the static structures of the model.

5.3.1 LOGIN

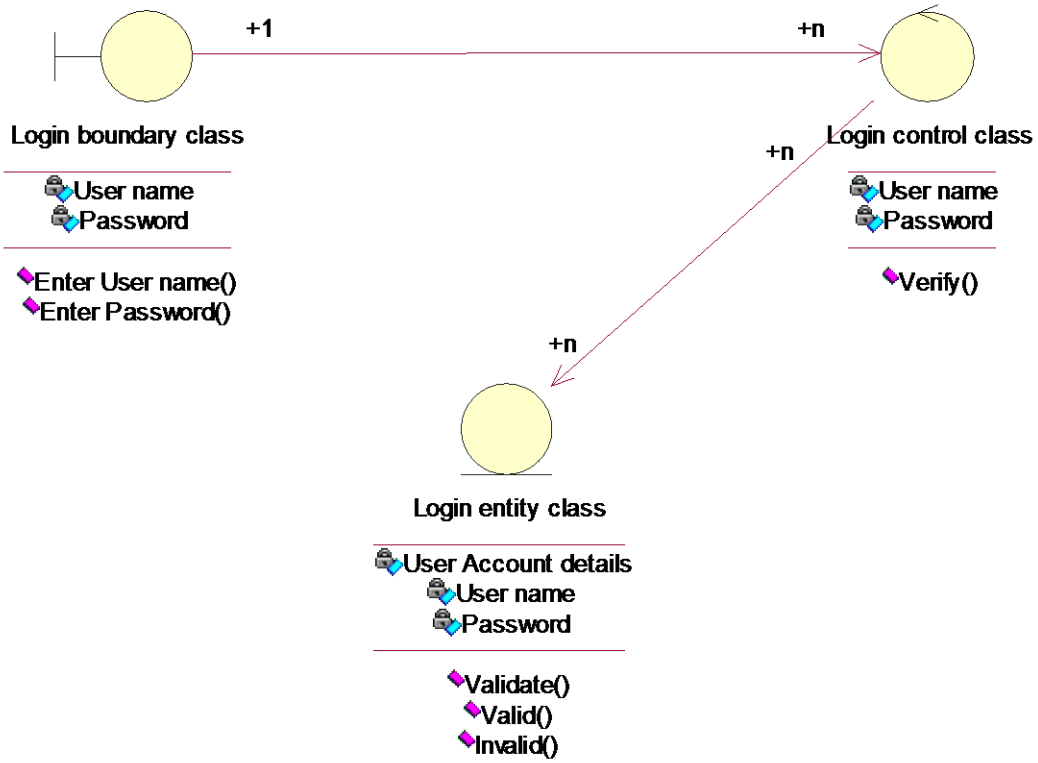


Fig 5.7 Login Class Diagram

5.3.2 CREATE ROOM

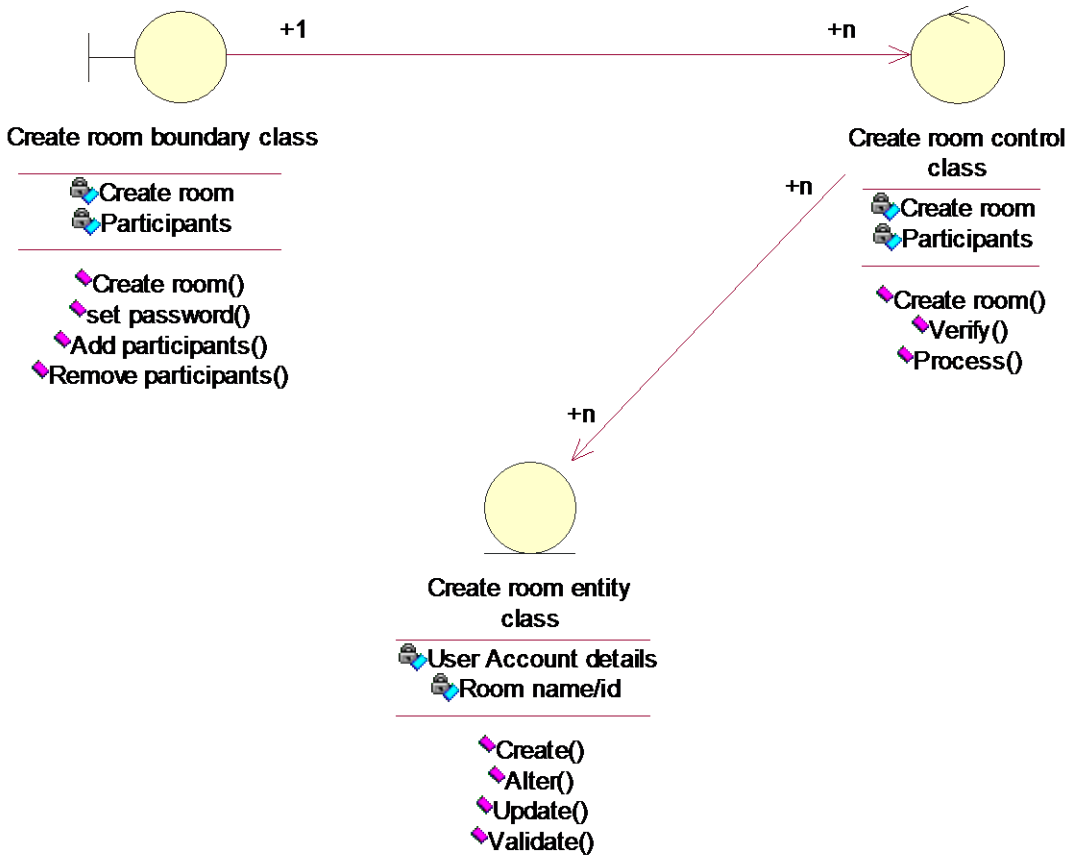


Fig 5.8 Create Room Class Diagram

5.3.3 SCREEN SHARING

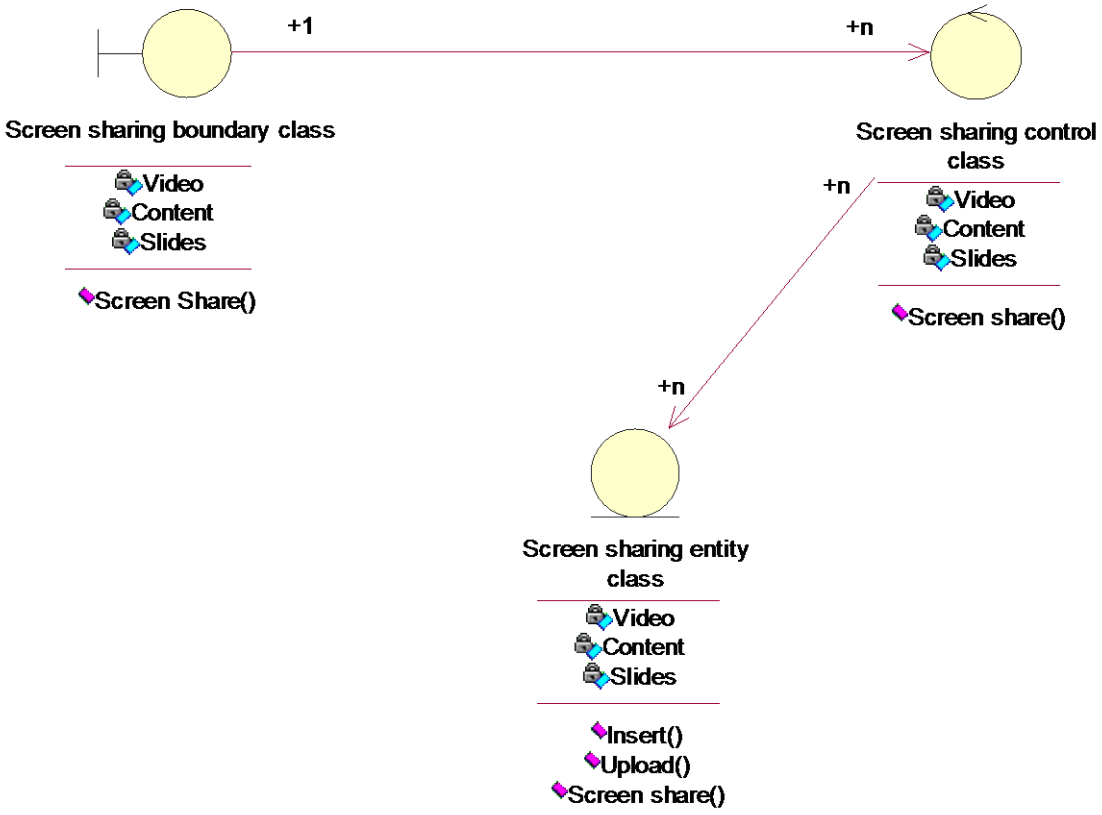


Fig 5.9 Screen Sharing Class Diagram

5.3.4 MUTE AUDIO/VIDEO

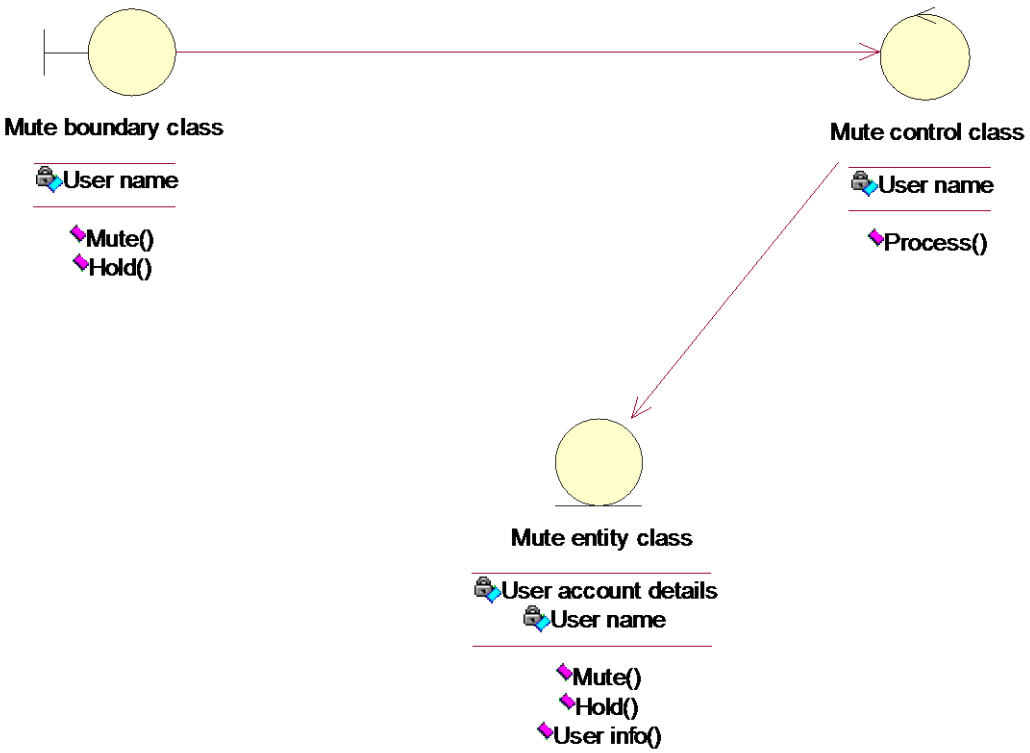


Fig 5.10 Mute Audio/Video Class Diagram

5.3.5 LOGOUT

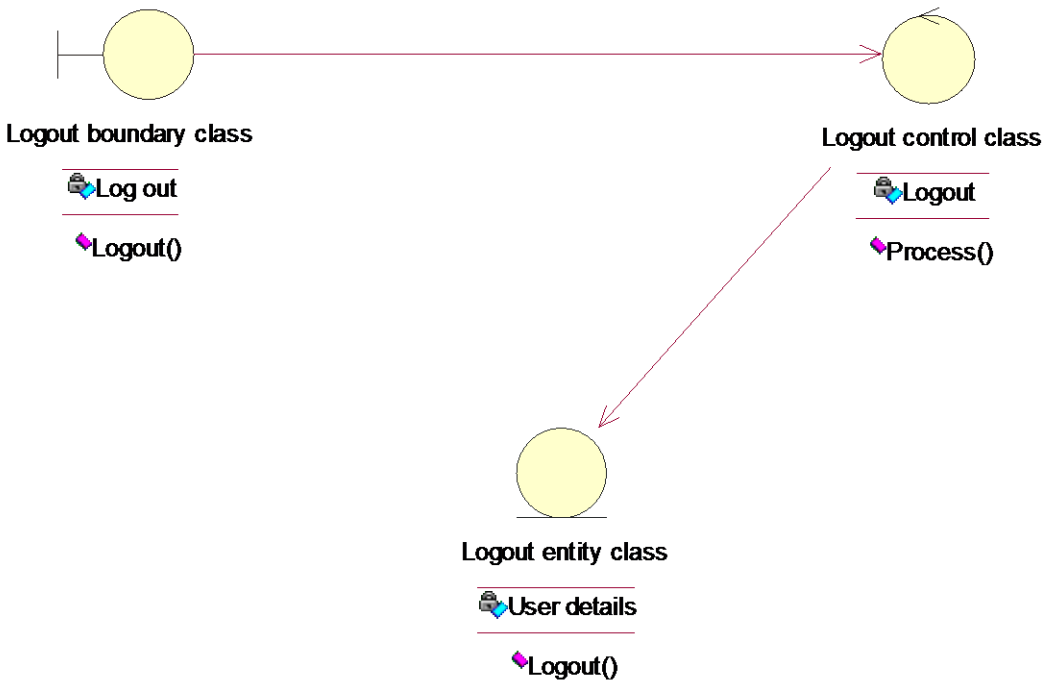


Fig 5.11 Logout Class Diagram

5.4 SEQUENCE DIAGRAM

Sequence Diagrams are easy and intuitive way of describing the behavior of a system by viewing the interaction between the system and its environment. A sequence diagram shows the interaction arranged in a time sequence. It shows the object participating by their life lines and the messages they exchange, arranged in a time sequence.

5.4.1 LOGIN

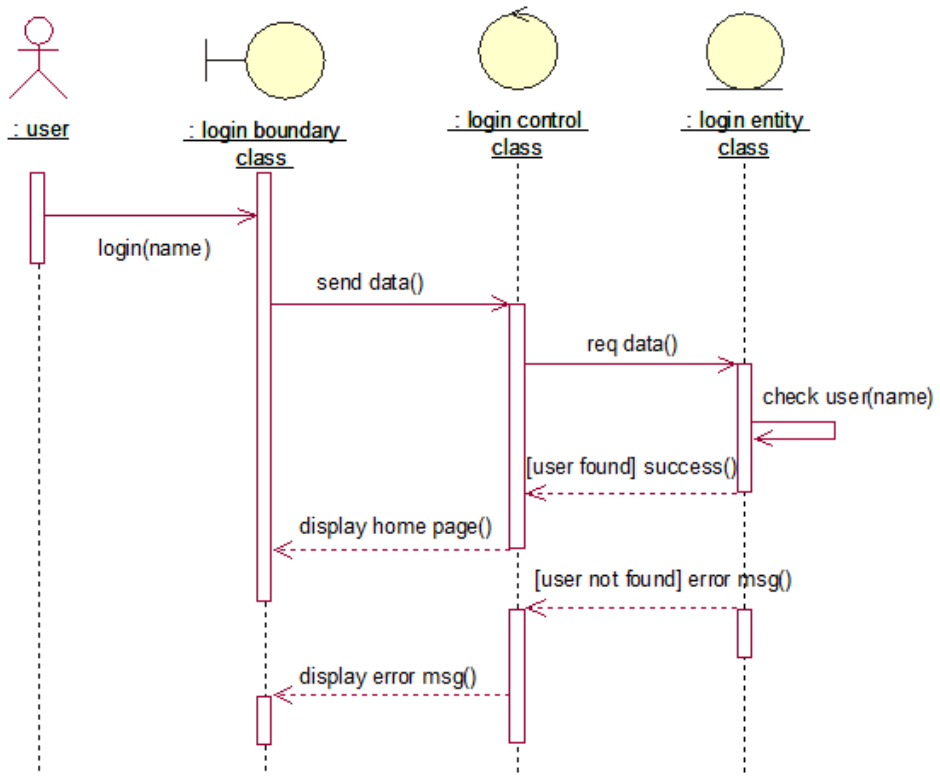


Fig 5.12 Login Sequence Diagram

5.4.2 CREATE ROOM

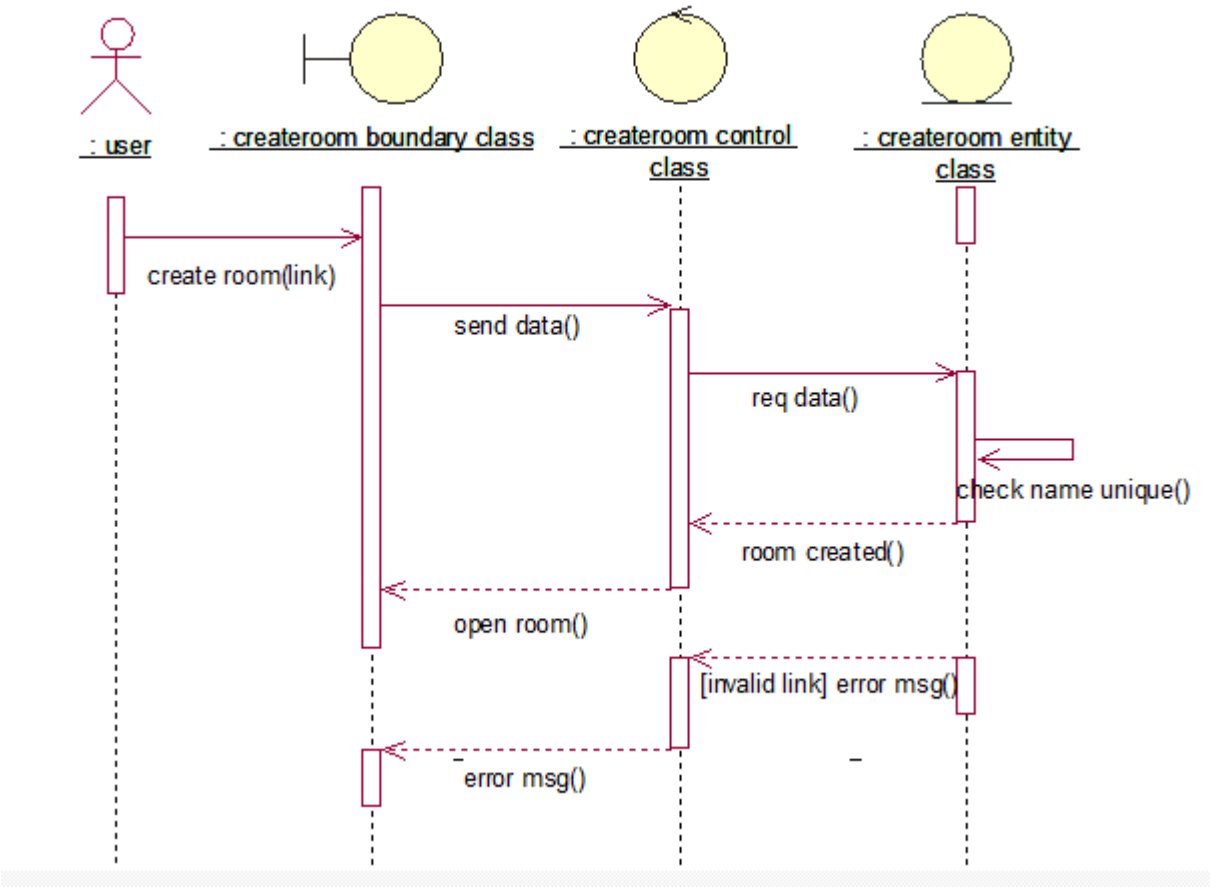


Fig 5.13 Create Room Sequence Diagram

5.4.3 SCREEN SHARING

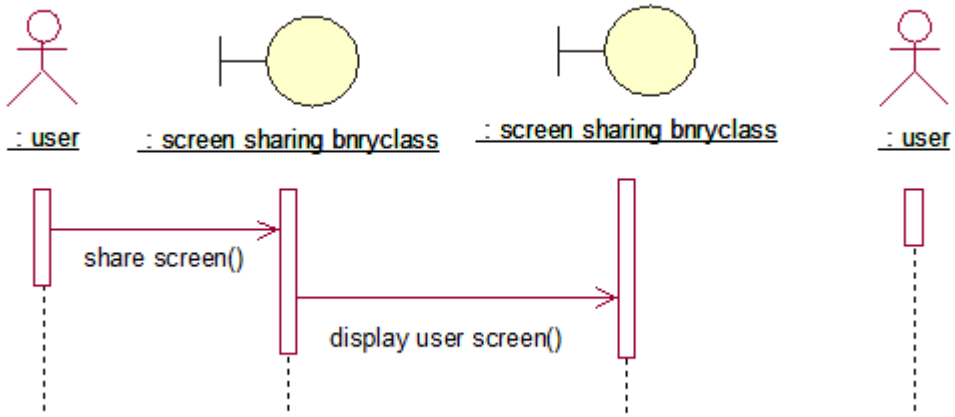


Fig 5.14 Screen Sharing Sequence Diagram

5.4.4 MUTE AUDIO/VIDEO

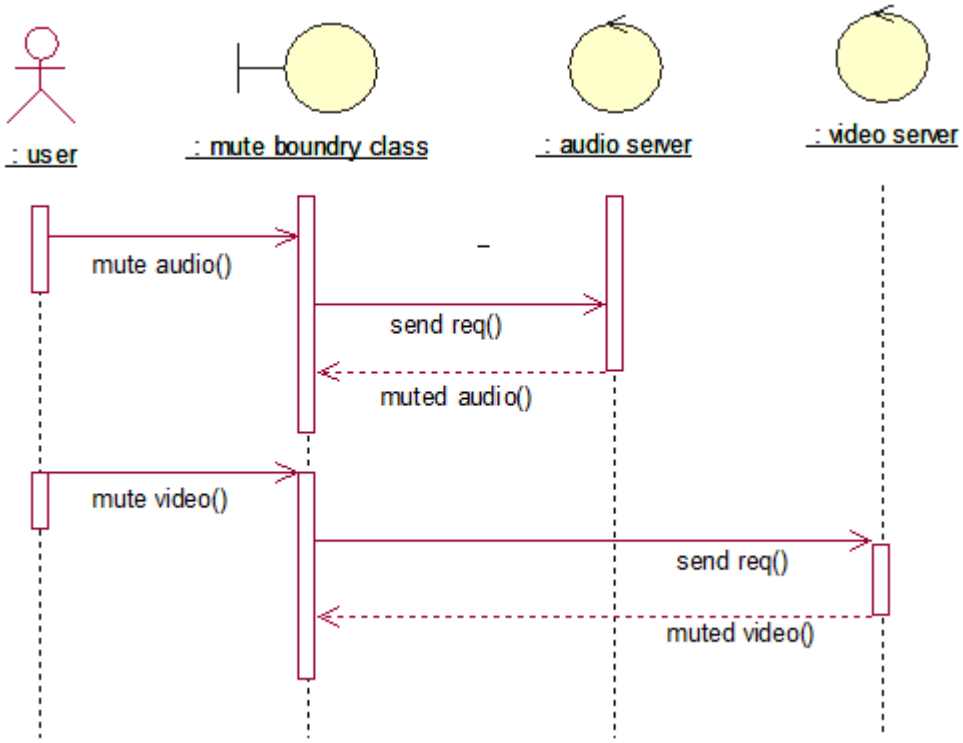


Fig 5.15 Mute Audio/Video Sequence Diagram

5.4.5 LOGOUT

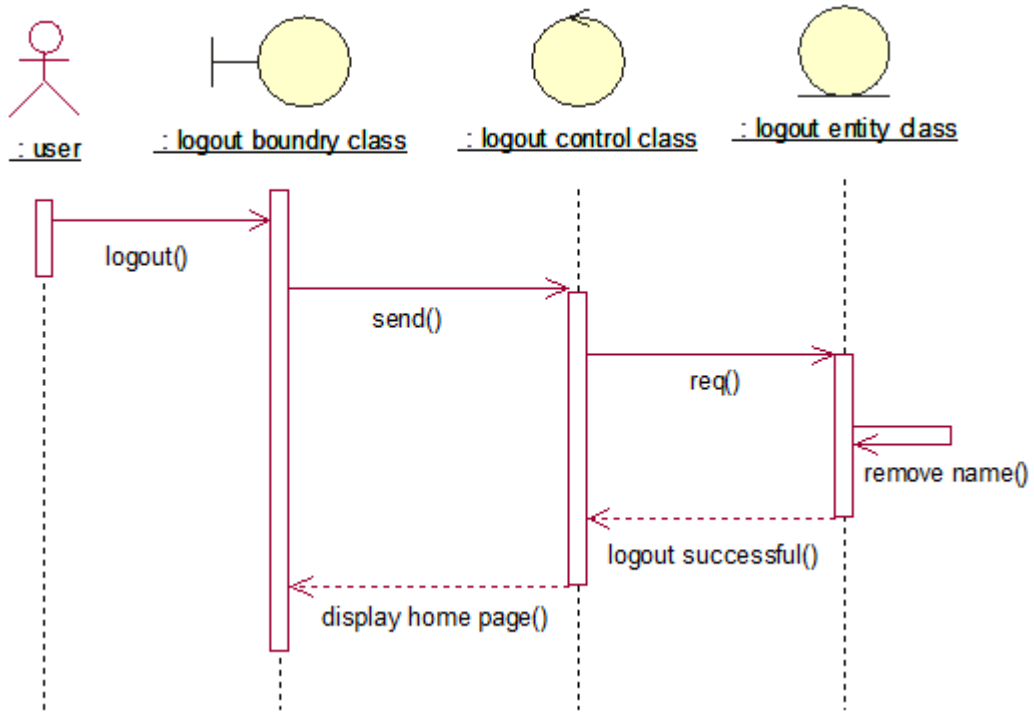


Fig 5.16 Logout Sequence Diagram

5.5 COLLABORATION DIAGRAM

A collaboration diagram represents a collaboration, which is a set of objects related in a particular context, and interaction, which is a set of messages exchanged among the objects within the collaboration to achieve a desire outcome.

5.5.1 LOGIN

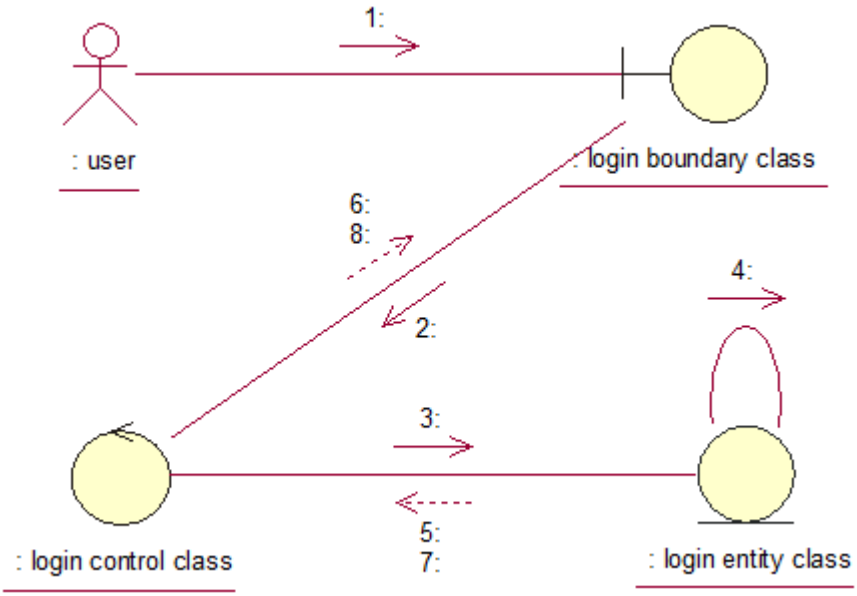


Fig 5.17 Login Collaboration Diagram

5.5.2 CREATE ROOM

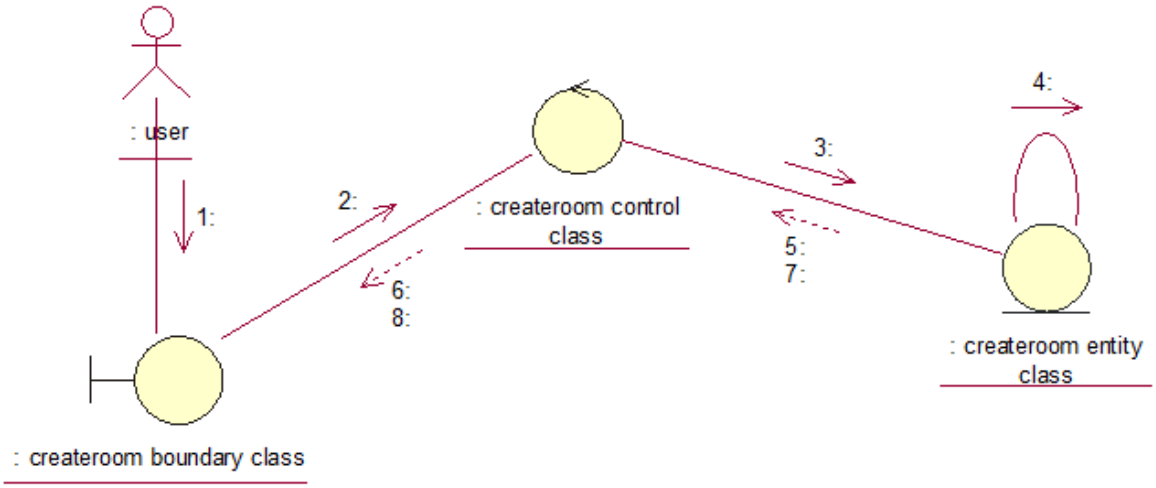


Fig 5.18 Create Room Collaboration Diagram

5.5.3 SCREEN SHARING

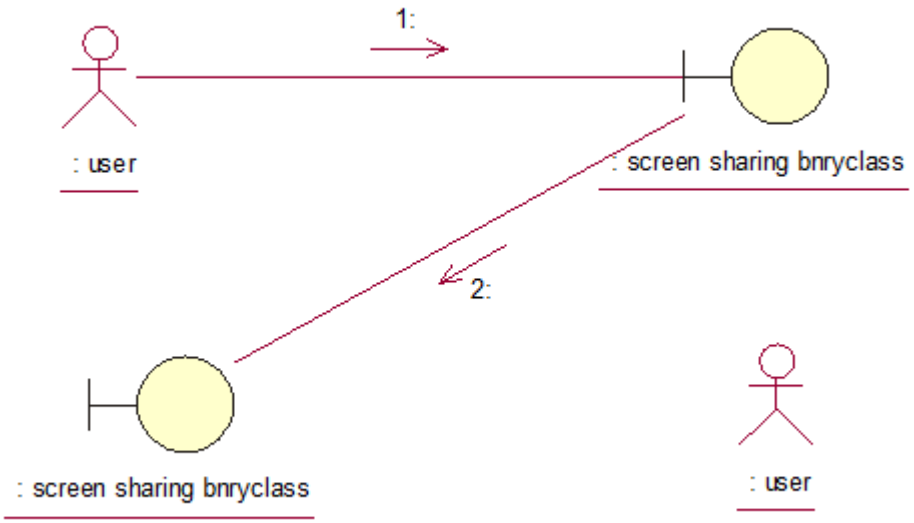


Fig 5.19 Screen Sharing Collaboration Diagram

5.5.4 MUTE AUDIO/VIDEO

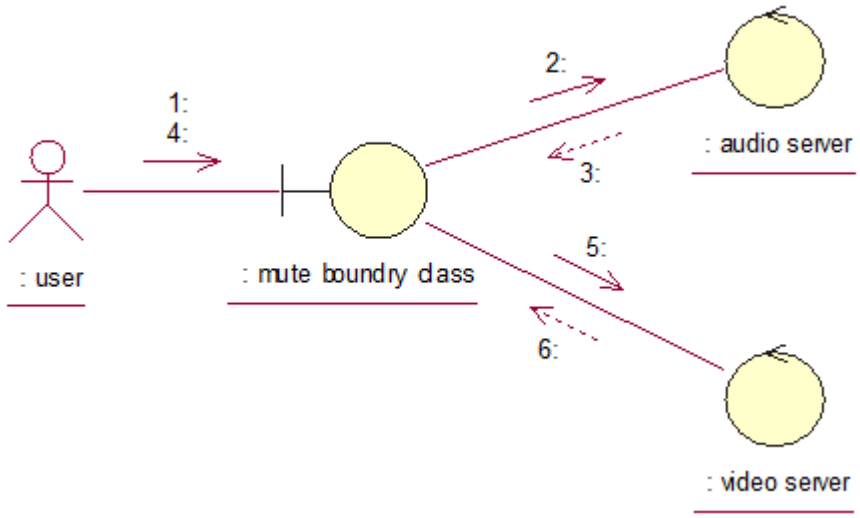


Fig 5.20 Mute Audio/Video Collaboration Diagram

5.5.5 LOGOUT

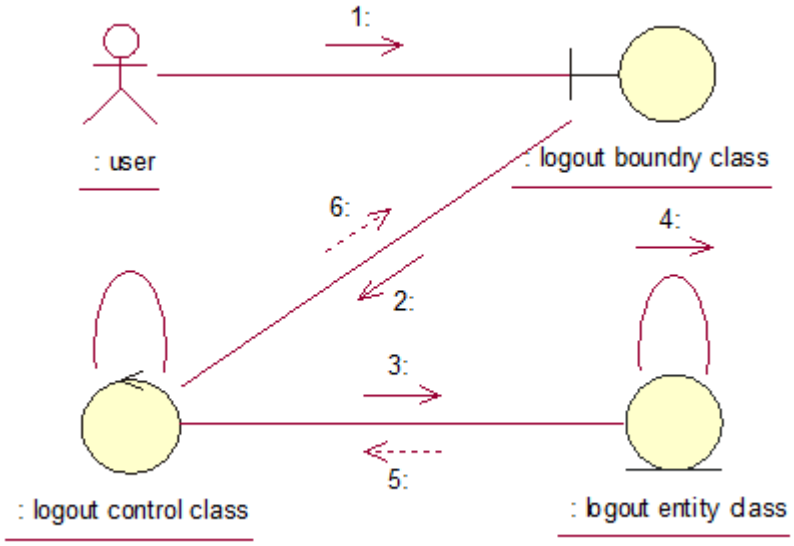


Fig 5.21 Logout Collaboration Diagram

5.6 COMPONENT DIAGRAM

Component diagram models the physical components such as source codes, executable programs, user interface in a design. These high level physical components may or may not be equivalent to many smaller components we use in creation of application.

5.6.1 MAIN COMPONENT DIAGRAM

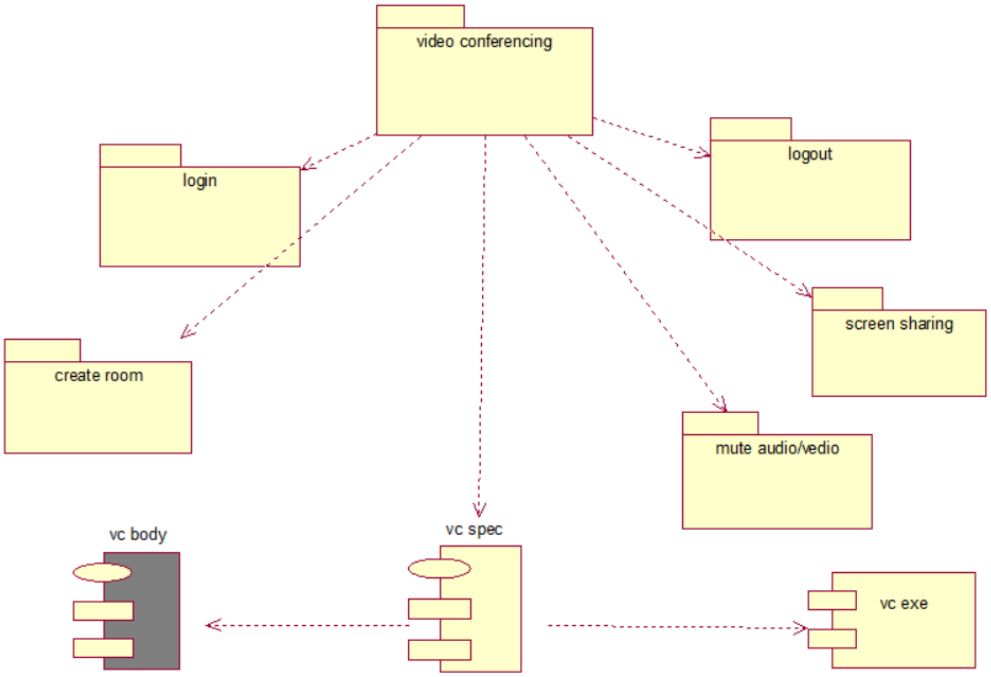


Fig 5.22 Main Component Diagram

5.6.2 LOGIN

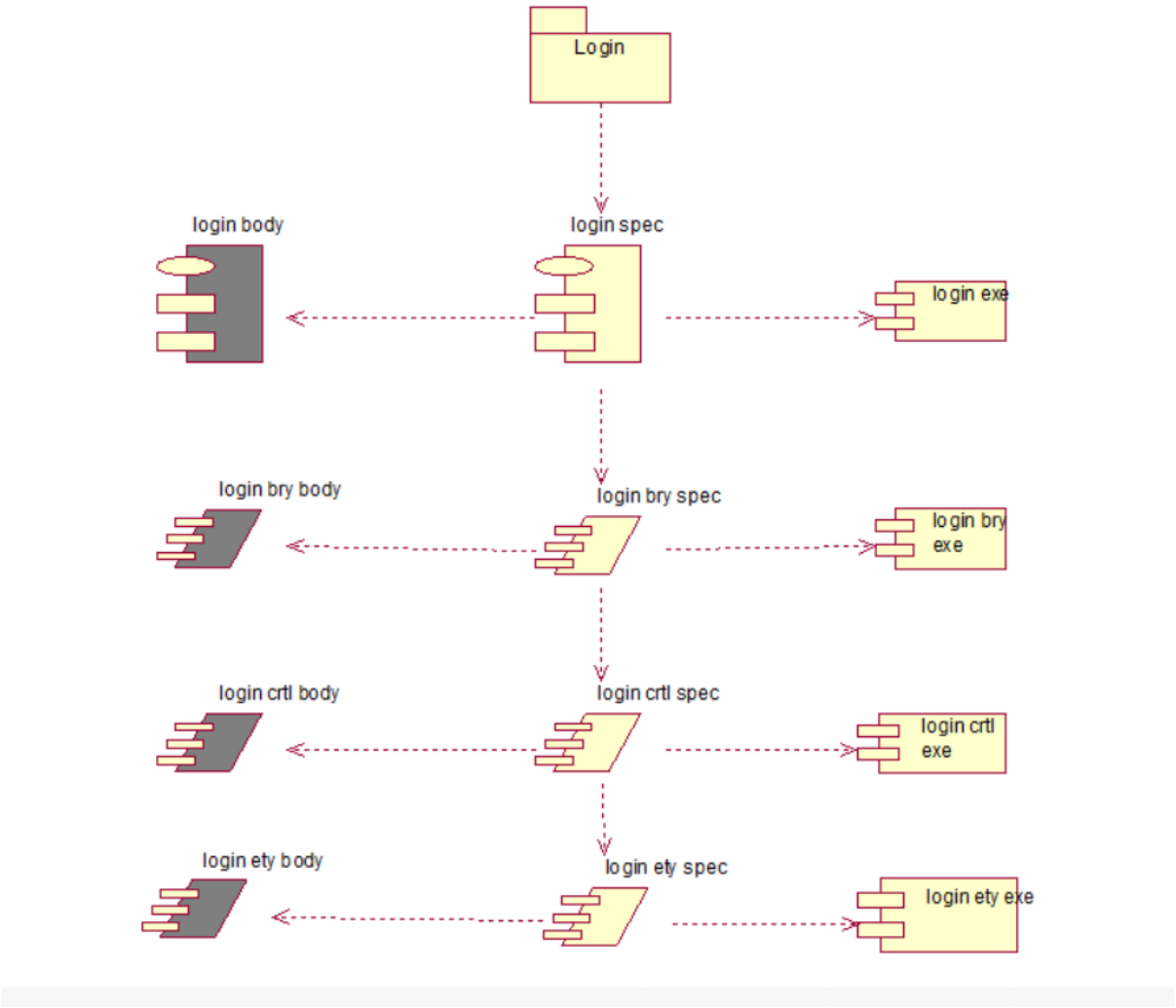


Fig 5.23 Login Component Diagram

5.6.3 CREATE ROOM

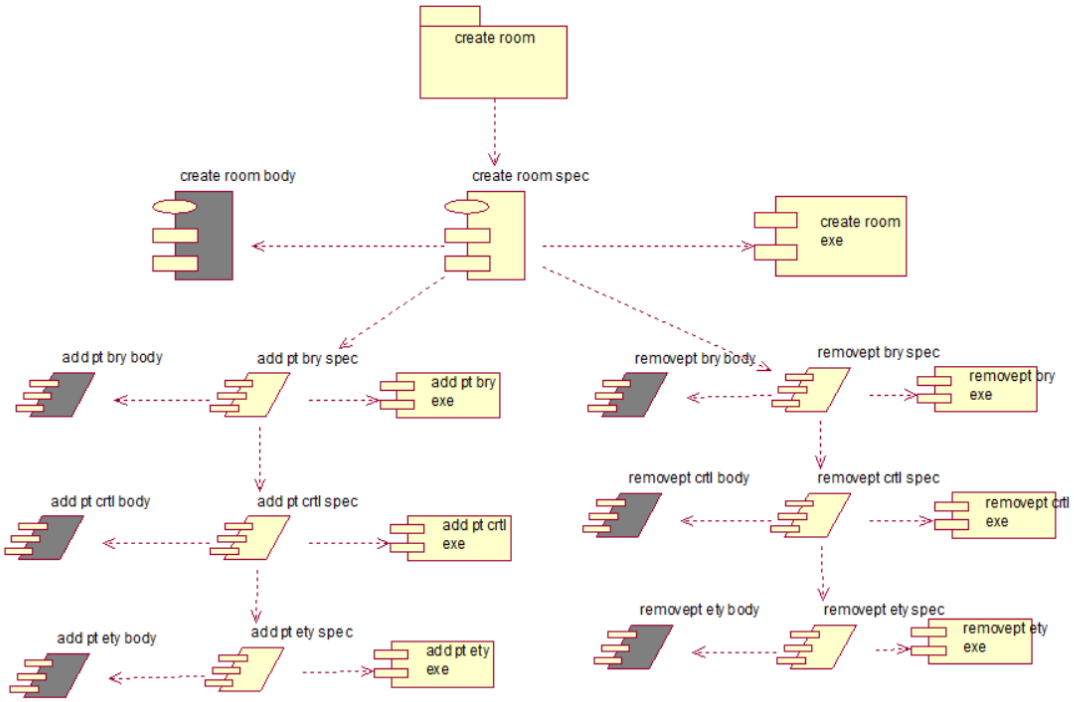


Fig 5.24 Create Room Component Diagram

5.6.4 SCREEN SHARING

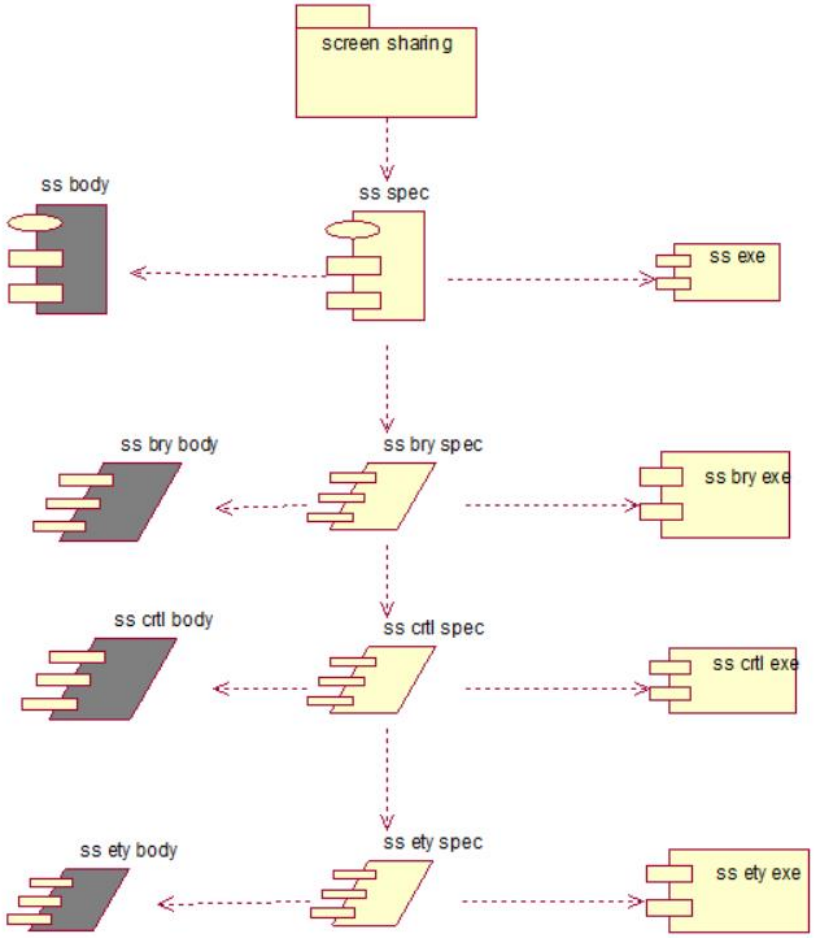


Fig 5.25 Screen Sharing Component Diagram

5.6.5 MUTE AUDIO/VIDEO

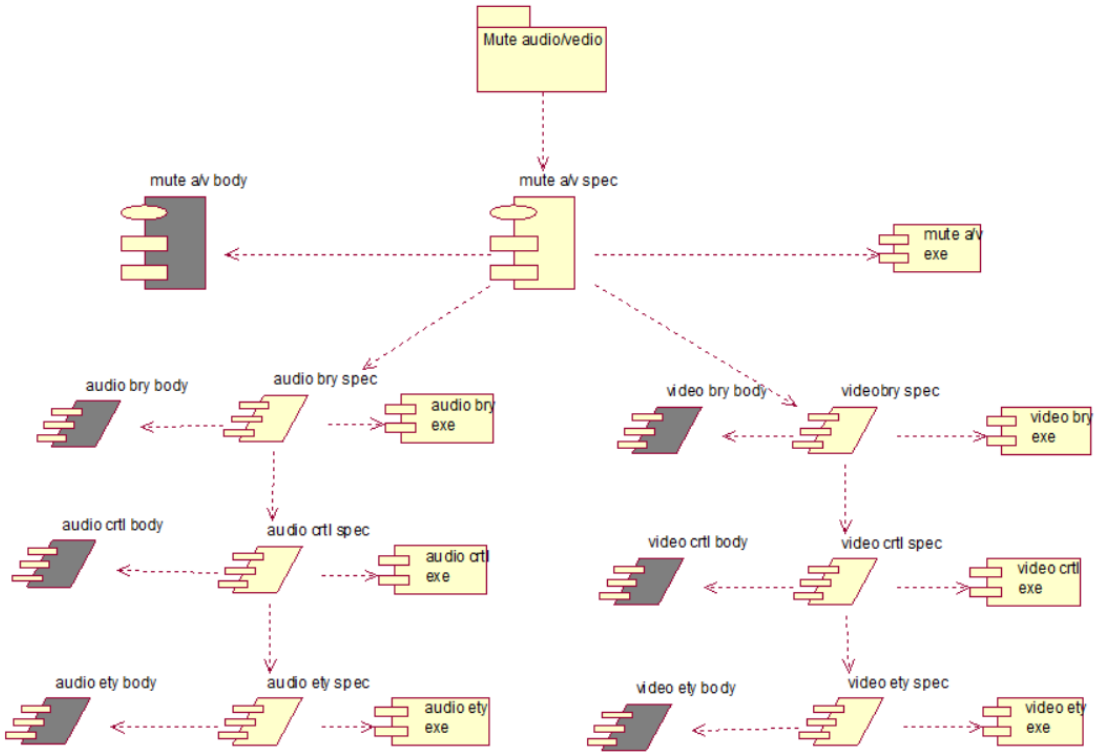


Fig 5.26 Mute Audio/Video Component Diagram

5.6.6 LOGOUT

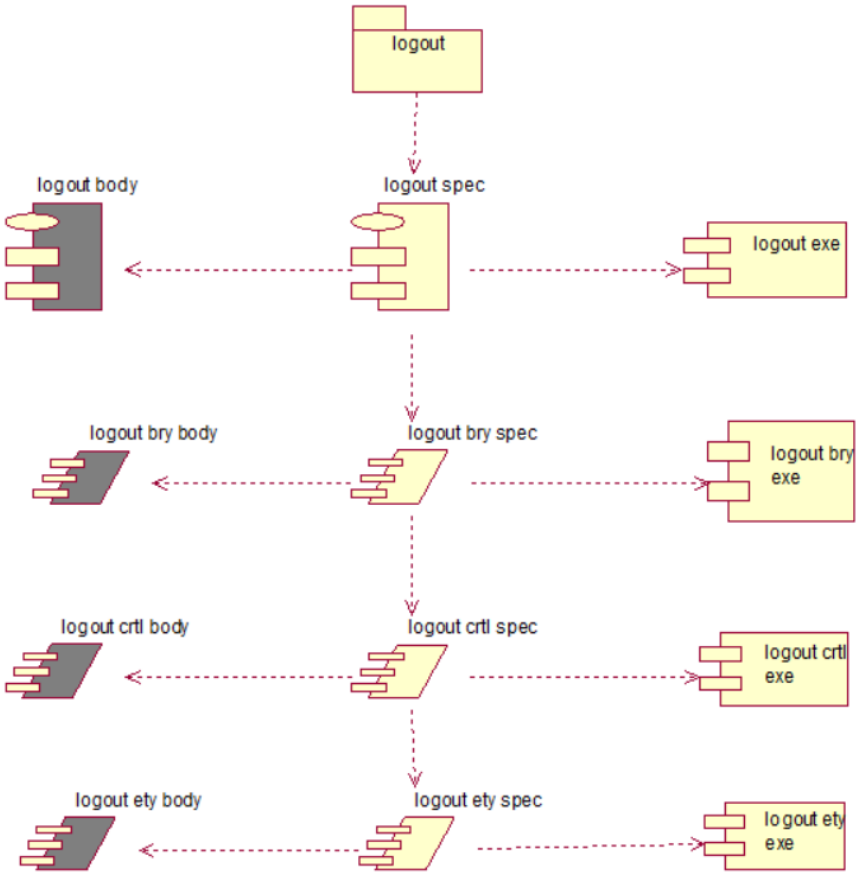


Fig 5.27 Logout Component Diagram

5.6 DEPLOYMENT DIAGRAM

Deployment diagram shows the configuration of run time processing elements and the software components, processes and objects that live in them.

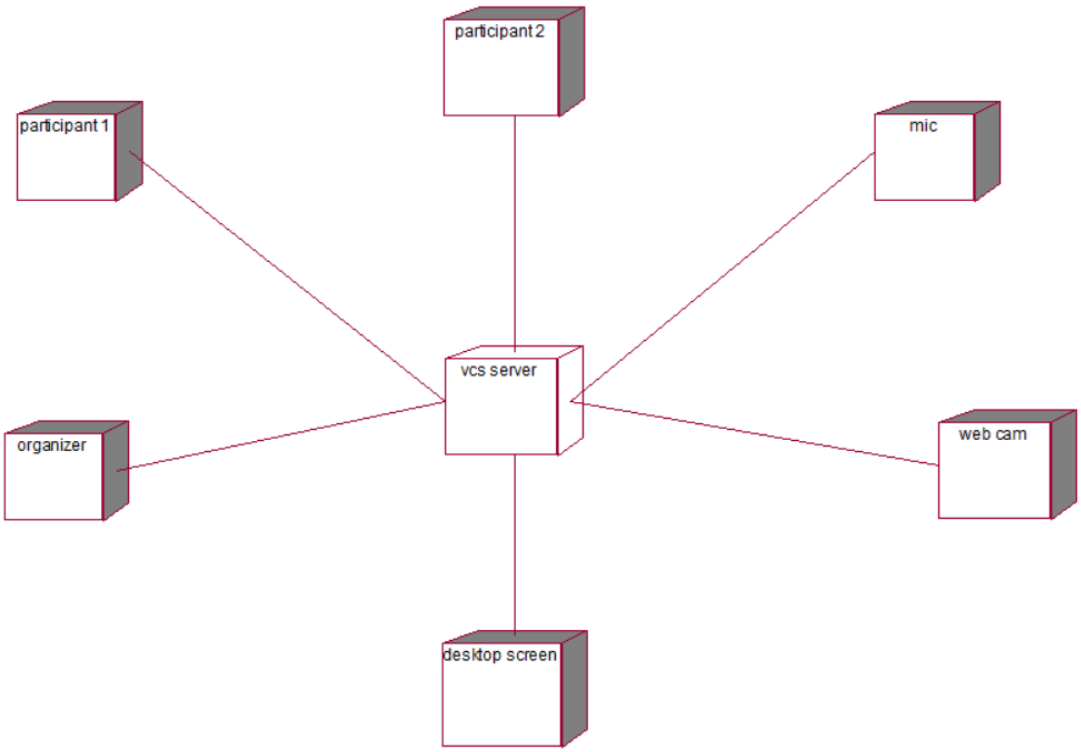


Fig 5.28 Deployment diagram

CHAPTER-6
SYSTEM DEVELOPMENT

6.1 VIDEO CALLS

Online video conferencing can integrate these communications platforms through a set of telecommunication technologies which allow two or more locations to communicate by simultaneous two-way video and audio transmissions

6.2 CREATE ROOM

Create room helps in connecting to the members of the meeting. This is done by generating a meeting link and can be shared with the members. This is an easier way to join a meeting.

6.3 CHAT

Chat option helps to send messages and transfer the files to all the members in meeting so that everyone can have a view and access to it.

6.4 SCREEN SHARING

When you start sharing your screen, the meeting controls will move into a menu
that you can drag around your screen.

6.5 LEAVE MEETING

Leave option in the page helps the member to leave from the meeting in case if meeting has finished or any emergency.

CHAPTER-7

SYSTEM IMPLEMENTATION

7.1 SYSTEM PREREQUISITE

HARDWARE REQUIREMENT

Operating System – Windows 10

RAM – 4 GB

Hard disk – 1 TB

SOFTWARE REQUIREMENT

OS – Windows 10

Editor – VS Code

Application – Chrome

Design – NodeJS, socket.io, WebRTC, HTML, CSS, JavaScript

CHAPTER-8
SYSTEM TESTING

8.1. TEST PLAN

8.1.1 PROJECT DESCRIPTION

The four basic steps in project are

1. Extract the problem domain statement from the user.
2. Classify into nouns and verbs based on the Noun Phrase Approach.
3. Generate the classes.
4. Place the operations in appropriate classes.

8.1.2 RELATED DOCUMENTS

The related documents in our project are project proposal and plan, Software Requirements Specification, Software Design Specification and test plan.

8.1.3 TESTING STRATEGY

Unit test and Functional test is used in each phase of testing.

8.1.4 TESTING LEVEL PLAN

White Box Testing is given higher priority than Black Box Testing.

8.1.5 UNIT TEST

Unit test is generally accomplished by the person who wrote the code. Generally white box testing is used at this level, since errors in logic are easier to find when testing each path through the code.

8.1.6 MODULE TEST

Module testing is usually done by a single programmer or a small group of programmers writing units that work together in a single module. Test cases are often used for the module testing, especially if the module is self contained.

8.1.7 INTEGRATION TEST

Usually the development team handles integration test. Integration test is best accomplished incrementally, by adding one module at a time to isolate errors.

8.1.8 ALPHA TEST

In this test, “Internal”testers run live data through the system to shake out bugs not found in integration test, the customer may wish to observe the alpha test or to provide some o the “Real” data.

8.1.9 BETA TEST

Beta test is the next step, where the programmers actually released to the customer with the understanding that the program is still being tested. The customer agrees to stress the application and to report any discovered bugs or problems to the development team. The team agrees to be a “Friendly” user but to really put the system through its paces, trying to break it.

8.1.10 REGRESSION TEST

After the product is released, errors may be found or enhancements suggested by the customers in the field. As these are corrected or implemented, the rest of the system must also be tested again to make sure that the new fixes did not break any of the old code. Regression test is usually an automated script that runs a set of test cases known to exercise the entire system.

8.1.11 USABILITY TEST

Usability testing is a special form of testing that looks for bugs not in the functionality of the program, but in the layout and utility of the user interface. This step is often a prototype before the actual system code is written, so it is easy to change if needed.

CHAPTER-9

CONCLUSION AND FUTURE ENHANCEMENTS

we present our redeveloped real-time web browser-based video conference. The proposed video conferencing system architecture based on a environments that supports single or multiple participants in a video conferencing session using a single connection. Compared with existing commercial video conference software, the proposed system is web browser-based and it is a cross-platform application which can be run on different devices such as desktop computers, smartphone, tablets, etc. The video streaming has been securing two level of authorization Gust and Member to allow only to authorized people to enter videoconference room and makes Live video chat with friends. The test results show that the video conference system has worked perfectly. The tests approved that the quality and streaming speed of video conference is highly dependable on speed of internet of clients and streaming bandwidth of the server, this means it is independent of the number of conference members

In future study, several areas could be improved in this application. Firstly, also is the most vital one is the limitations of number of users in one video calling session. If the number of user increases for video conferencing purpose, it will encourage more users to use this application. There were still a lot more function can be implemented with WebRTC technology like video broadcast which enable lecturer to teach online. Thus, it is more advisable for future improvement is more towards video broadcast and increase more file type that support in file sharing

APPENDIX 1

SAMPLE CODE

Script.js

```
const socket = io("/");
const chatInputBox = document.getElementById("chat_message");
const all_messages = document.getElementById("all_messages");
const main__chat__window =
document.getElementById("main__chat__window");
const videoGrid = document.getElementById("video-grid");
const myVideo = document.createElement("video");
myVideo.muted = true;

var peer = new Peer(undefined, {
  path: "/peerjs",
  host: "/",
  port: "3030",
});

let myVideoStream;

var getUserMedia =
  navigator.getUserMedia ||
  navigator.webkitGetUserMedia ||
  navigator.mozGetUserMedia;

navigator.mediaDevices
  .getUserMedia({
    video: true,
    audio: true,
  })
  .then((stream) => {
    myVideoStream = stream;
    addVideoStream(myVideo, stream);

    peer.on("call", (call) => {
      call.answer(stream);
      const video = document.createElement("video");

      call.on("stream", (userVideoStream) => {
```

```
        addVideoStream(video, userVideoStream);
      });
    });

    socket.on("user-connected", (userId) => {
      connectToNewUser(userId, stream);
    });

    document.addEventListener("keydown", (e) => {
      if (e.which === 13 && chatInputBox.value != "") {
        socket.emit("message", chatInputBox.value);
        chatInputBox.value = "";
      }
    });

    socket.on("createMessage", (msg) => {
      console.log(msg);
      let li = document.createElement("li");
      li.innerHTML = msg;
      all_messages.append(li);
      main__chat__window.scrollTop = main__chat__window.scrollHeight;
    });
  });

peer.on("call", function (call) {
  getUserMedia(
    { video: true, audio: true },
    function (stream) {
      call.answer(stream); // Answer the call with an A/V stream.
      const video = document.createElement("video");
      call.on("stream", function (remoteStream) {
        addVideoStream(video, remoteStream);
      });
    },
    function (err) {
      console.log("Failed to get local stream", err);
    }
  );
});

peer.on("open", (id) => {
  socket.emit("join-room", ROOM_ID, id);
});
```

```
// CHAT

const connectToNewUser = (userId, streams) => {
  var call = peer.call(userId, streams);
  console.log(call);
  var video = document.createElement("video");
  call.on("stream", (userVideoStream) => {
    console.log(userVideoStream);
    addVideoStream(video, userVideoStream);
  });
};

const addVideoStream = (videoEl, stream) => {
  videoEl.srcObject = stream;
  videoEl.addEventListener("loadedmetadata", () => {
    videoEl.play();
  });

  videoGrid.append(videoEl);
  let totalUsers = document.getElementsByTagName("video").length;
  if (totalUsers > 1) {
    for (let index = 0; index < totalUsers; index++) {
      document.getElementsByTagName("video")[index].style.width =
        100 / totalUsers + "%";
    }
  }
};

const playStop = () => {
  let enabled = myVideoStream.getVideoTracks()[0].enabled;
  if (enabled) {
    myVideoStream.getVideoTracks()[0].enabled = false;
    setPlayVideo();
  } else {
    setStopVideo();
    myVideoStream.getVideoTracks()[0].enabled = true;
  }
};

const muteUnmute = () => {
  const enabled = myVideoStream.getAudioTracks()[0].enabled;
  if (enabled) {
    myVideoStream.getAudioTracks()[0].enabled = false;
    setUnmuteButton();
```

```
  } else {
    setMuteButton();
    myVideoStream.getAudioTracks()[0].enabled = true;
  }
};

const setPlayVideo = () => {
  const html = `<i class="unmute fa fa-pause-circle"></i>
  <span class="unmute">Resume Video</span>`;
  document.getElementById("playPauseVideo").innerHTML = html;
};

const setStopVideo = () => {
  const html = `<i class=" fa fa-video-camera"></i>
  <span class="">Pause Video</span>`;
  document.getElementById("playPauseVideo").innerHTML = html;
};

const setUnmuteButton = () => {
  const html = `<i class="unmute fa fa-microphone-slash"></i>
  <span class="unmute">Unmute</span>`;
  document.getElementById("muteButton").innerHTML = html;
};
const setMuteButton = () => {
  const html = `<i class="fa fa-microphone"></i>
  <span>Mute</span>`;
  document.getElementById("muteButton").innerHTML = html;
};
```

Style.css

```
@import
url("https://fonts.googleapis.com/css2?family=Roboto&display=swap");
* {
  margin: 0;
  padding: 0;
  box-sizing: border-box;
}
html,
body {
  height: 100%;
  font-family: "Roboto", sans-serif;
```

```
}
#video-grid {
  display: flex;
  justify-content: center;
  height: 100%;
  width: 100%;
  align-items: center;
  flex-wrap: wrap;
  overflow-y: auto;
}
video {
  display: block;
  flex: 1;
  object-fit: cover;
  border: 5px solid #000;
  max-width: 600px;
}
.main {
  height: 100%;
  display: flex;
}
.main__left {
  flex: 0.8;
  display: flex;
  flex-direction: column;
}
.main__right {
  flex: 0.2;
  display: flex;
  flex-direction: column;
  background-color: #242324;
  border-left: 1px solid #3d3d42;
}
.main__videos {
  flex-grow: 1;
  background-color: #000;
  display: flex;
  align-items: center;
  justify-content: center;
}
.main__controls {
  display: flex;
  background-color: #1c1e20;
  color: #d2d2d2;
```

```
  padding: 5px;
  justify-content: space-between;
}
.main__controls_block {
  display: flex;
}
.main__controls_button {
  display: flex;
  flex-direction: column;
  cursor: pointer;
  padding: 10px;
  justify-content: center;
  align-items: center;
  min-width: 80px;
  transition: all 0.3s ease-in-out;
  border-radius: 10px;
  margin: 5px;
}
.main__controls_button span {
  margin-top: 10px;
  display: block;
}
.main__controls_button:hover {
  background-color: #34383b;
}
.main__controls_button i {
  font-size: 25px;
}
.leaveMeeting {
  background-color: red;
  color: #fff;
}
.main__header {
  color: #f5f5f5;
  text-align: center;
  padding: 20px;
  border-bottom: 2px solid #3d3d42;
}
.main_chat_window {
  flex-grow: 1;
  overflow: auto;
  padding: 20px 20px 0 20px;
}
.main__message_container {
```

```
  padding: 22px 11px;
  display: flex;
}
.main__message_container input {
  flex-grow: 1;
  background-color: transparent;
  border: none;
  color: #f5f5f5;
  user-select: none;
  outline: none;
}

#all_messages li {
  color: #fff;
  list-style: none;
  border-bottom: 1px solid #3d3d42;
  padding: 10px 0;
}
.unmute {
  color: red;
}
```

room.ejs

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-
scale=1.0" />
    <title>HOME</title>
    <link
      rel="stylesheet"
      href="https://stackpath.bootstrapcdn.com/font-
awesome/4.7.0/css/font-awesome.min.css"
    />
    <link rel="stylesheet" href="style.css" />
    <script
src="https://unpkg.com/peerjs@1.3.1/dist/peerjs.min.js"></script>
    <script
src="https://cdnjs.cloudflare.com/ajax/libs/peerjs/1.3.1/peerjs.min.js
.map"></script>
```

```
  <script src="/socket.io/socket.io.js"></script>
  <script>
    const ROOM_ID = "<%= roomId %>";
  </script>
</head>
<body>
  <div class="main">
    <div class="main__left">
      <div class="main__videos">
        <div id="video-grid"></div>
      </div>
      <div class="main__controls">
        <div class="main__controls_block">
          <div
            class="main__controls_button"
            id="muteButton"
            onclick="muteUnmute()"
          >
            <i class="fa fa-microphone"></i>
            <span>Mute</span>
          </div>
          <div
            class="main__controls_button"
            id="playPauseVideo"
            onclick="playStop()"
          >
            <i class="fa fa-video-camera"></i>
            <span>Pause Video</span>
          </div>
        </div>

        <div class="main__controls_block">
          <div class="main__controls_button">
            <i class="fa fa-shield"></i>
            <span>Security</span>
          </div>
          <div class="main__controls_button">
            <i class="fa fa-users"></i>
            <span>Participants</span>
          </div>
          <div class="main__controls_button">
            <i class="fa fa-comment"></i>
            <span>Chat</span>
          </div>
```

```
          </div>

          <div class="main__controls_block">
            <div class="main__controls_button leaveMeeting" id="leave-
meeting">
              <i class="fa fa-times"></i>
              <span class="">Leave Meeting</span>
            </div>
          </div>
        </div>
      </div>
      <div class="main__right">
        <div class="main__header">
          <h6>Chat</h6>
        </div>
        <div class="main_chatwindow" id="mainchat_window">
          <ul class="messages" id="all_messages"></ul>
        </div>
        <div class="main__message_container">
          <input
            type="text"
            id="chat_message"
            placeholder="Type message here.."
          />
        </div>
      </div>
    </div>
    <script src="script.js"></script>
  </body>
</html>
```

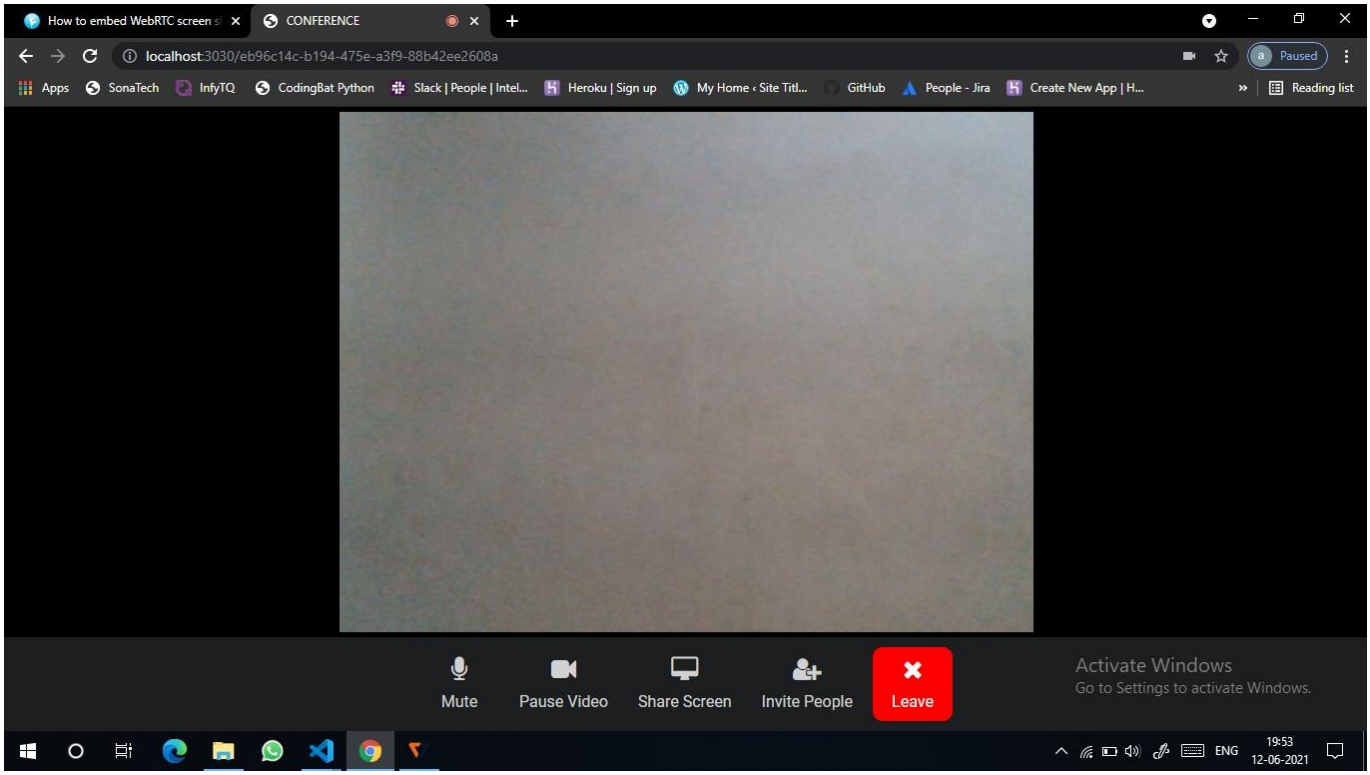
server.js

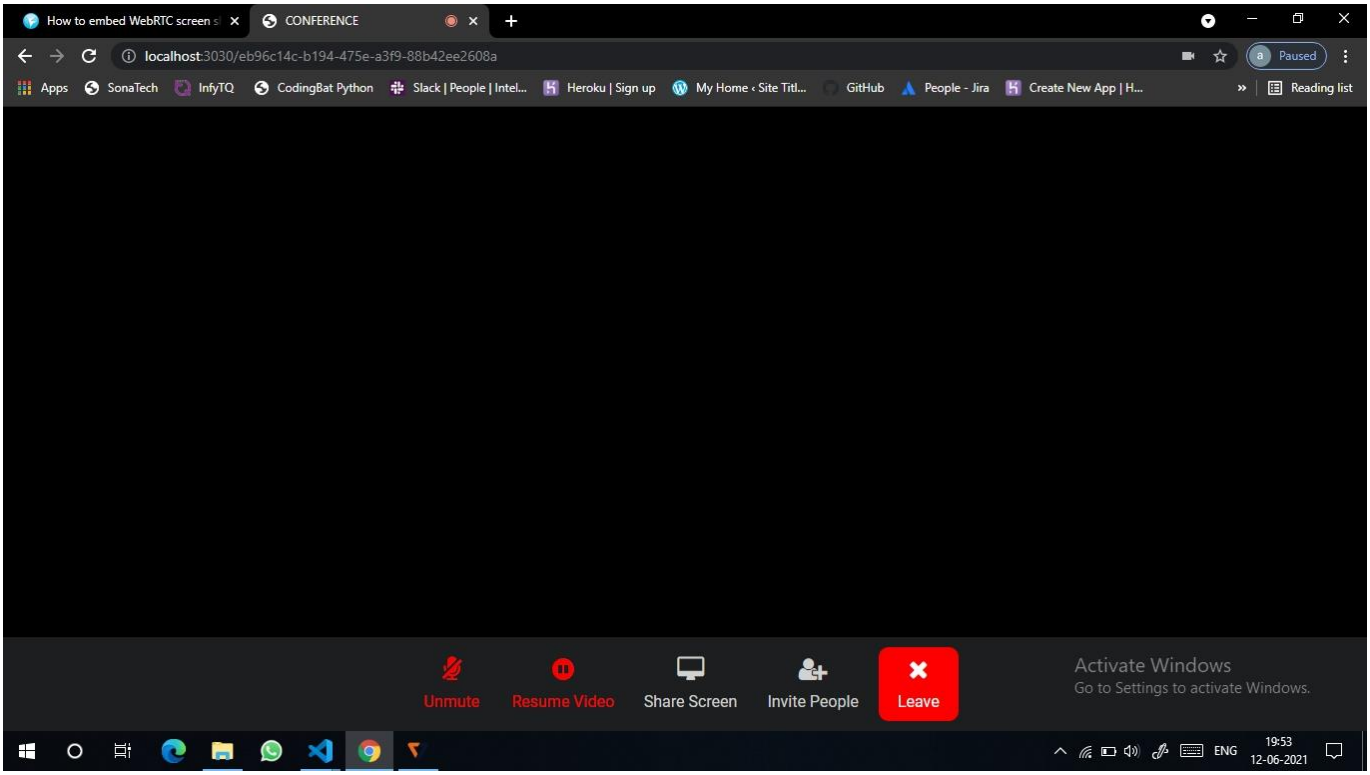
```
const express = require("express");
const app = express();
const server = require("http").Server(app);
const { v4: uuidv4 } = require("uuid");
const io = require("socket.io")(server);
// Peer

const { ExpressPeerServer } = require("peer");
const peerServer = ExpressPeerServer(server, {
```

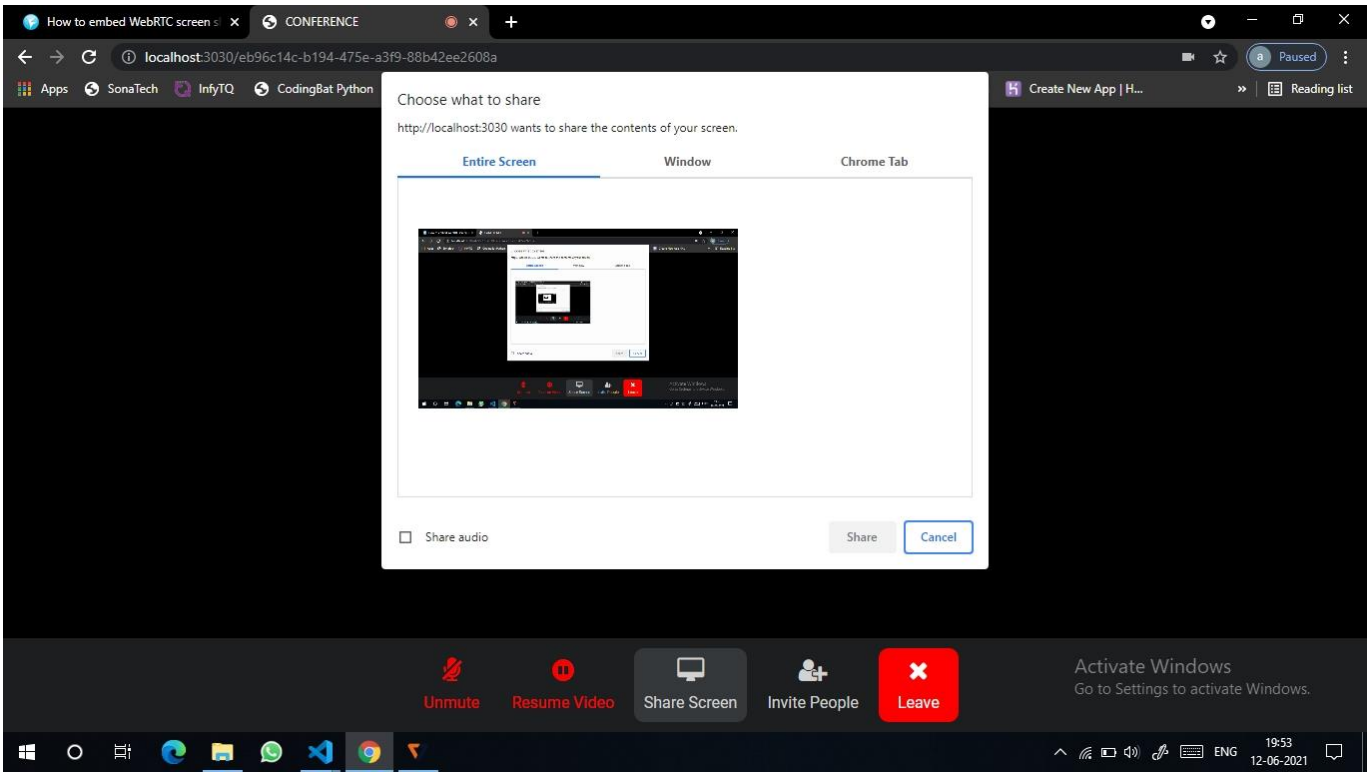
```
  debug: true,
});

app.set("view engine", "ejs");
app.use(express.static("public"));
app.use("/peerjs", peerServer);

app.get("/", (req, rsp) => {
  rsp.redirect(`/${uuidv4()}`);
});

app.get("/:room", (req, res) => {
  res.render("room", { roomId: req.params.room });
});

io.on("connection", (socket) => {
  socket.on("join-room", (roomId, userId) => {
    socket.join(roomId);
    socket.to(roomId).broadcast.emit("user-connected", userId);

    socket.on("message", (message) => {
      io.to(roomId).emit("createMessage", message);
    });
  });
});

server.listen(process.env.PORT || 3030);
```

APPENDIX 2

SAMPLE SCREEN SHOTS




How to embed WebRTC screen s
CONFERENCE
localhost:3030/eb96c14c-b194-475e-a3f9-88b42ee2608a
Paused
Apps
SonaTech
InfyTQ
CodingBat Python
Slack | People | Intel...
Heroku | Sign up
My Home ‹ Site Titl...
GitHub
People - Jira
Create New App | H...
Reading list
Unmute
Resume Video
Share Screen
Invite People
Leave
Activate Windows
Go to Settings to activate Windows.
ENG
19:53
12-06-2021


How to embed WebRTC screen s
CONFERENCE
localhost:3030/eb96c14c-b194-475e-a3f9-88b42ee2608a
Paused
Apps
SonaTech
InfyTQ
CodingBat Python
Create New App | H...
Reading list
Choose what to share
http://localhost:3030 wants to share the contents of your screen.
Entire Screen
Window
Chrome Tab
Share audio
Share
Cancel
Unmute
Resume Video
Share Screen
Invite People
Leave
Activate Windows
Go to Settings to activate Windows.
ENG
19:53
12-06-2021