Audio Video Class Room Interaction on Aakash

Summer Internship 2013

Submitted in fulfillment of internship project

By
AVCRIOA Team

Under the Guidance of
Prof. D. B. Phatak

Department of Computer Science and Engineering
Indian Institute of Technology, Powai
Bombay

Acknowledgement

We, the summer interns of the team AVCRIOA, are overwhelmed in all humbleness and gratefulness to acknowledge our deep gratitude to all those who have helped us put our ideas to perfection and have assigned tasks well above the level of simplicity and into something concrete and unique .

We, whole heartedly thank Prof. D.B. Phatak for having faith in us, selecting us to be a part of his valuable project and for constantly motivating us to do better.

We are very thankful to our mentors Mr. Rajesh Kushalkar, Mr.Rajavel and Mr Gobinath for their valuable suggestions. They were and are always there to show us the right track when needed help. With help of their brilliant guidance and encouragement, we all were able to complete our tasks properly and were up to the mark in all the tasks assigned. During the process, we got a chance to see the stronger side of our technical and non-technical aspects and also strengthen our concepts.

Last but not the least, we whole heartedly thank all our other colleagues working in different projects under Prof. D.B Phatak for helping us evolve better with their critical advice.

Team Members

Ashwini Sharnagat (VNIT Nagpur): Worked on broadcasting and stored the audio requests to database.

Asula Venkata Ravi Kumar (RGU-IIIT Nuzvid): Worked on audio-video live streaming, text messaging and GUI.

Chiranjeevi Ankamreddy (MGIT Hyderabad): Worked on instructor login part and validation.

Chiru varun Mooraboyina (RGU-IIIT Nuzvid): Worked on Audio recording and GUI.

Satyanarayana IVVV (RGU-IIIT Nuzvid): Worked on audio-video live streaming, GUI and back-end database.

Mentors

Mr. Rajesh Kushalkar: A person who proposed the basic layout of our project and introduced us to the new technology of webRTC.

Mr. Rajavel: A very nice and helpful person, who supervised our full project and guided as in every logical and technical aspects of our project.

Mr. Gobinath Mani: A good person, who helped us at every point in our project.

Mr.Tushar Kambli: Our guide for all problems and a Linux OS mastermind.

Abstract

Audio Video Class Room Interaction on Aakash- is a MHRD sponsored project aimed to establish interactive class room environment among students and instructor. It is a technology aimed mainly to reduce the manual labour like circulating microphone in a big hall. It is an affordable interactive software used for asking questions in live environment, record your questions, broadcast a particular session and ask text questions. It enhances the interaction between teachers and students.

The software part is a computational and analytical tool running on an instructor machine which uses MySQL for database storage, HTML5, JAVA, JSP, Java-Script, J-query etc. for designing and utilizing the Web-Based application.

Contents

1. Introduction 7

1.1 Purpose 7

1.2 Scope 7

1.3 Definitions, Abbreviations 8

1.4 Motivation 8

2. Objective 9

3. Design Considerations 10

3.1 Assumptions and Dependencies 10

3.2 General Constraints 10

3.3 Goals and Guidelines 11

3.4 Technology used 11

4. Architecture 13

5. Design and Implementation 14

5.1 System Flow Diagram 14

5.2 Schema Diagram 16

5.3 Use Case Diagram 17

5.3.1 Overall 17

5.3.2 Student Use Case Diagram 18

5.3.3 Instructor Use Case Diagram 19

5.4 Sequence Diagram 20

5.4.1 Student login/logout 20

5.4.2 Instructor login/logout 20

5.4.3 Video/Text Doubts 21

5.4.4 Audio Doubts 22

5.4.5 Video Broadcast 23

5.5 Data Flow Diagram 24

5.5.1 Level 0 DFD 24

5.5.2 Level 1 DFD 24

6. Detailed Description 27

6.1 Student module 27

6.1.1 Student registration 27

6.1.2 Student login 27

6.1.3 Choose a course 27

6.1.4 Ask video doubts 27
6.1.5 Ask textual doubts 27
6.1.6 Ask audio doubts 28
6.1.7 Video broadcasting 28
6.1.8 Student logout 28
6.2 Instructor module 28
6.2.1 Instructor registration 28
6.2.2 Instructor login 29
6.2.3 Enable/disable video and text doubts 29
6.2.4 View video requests 29
6.2.5 View saved audio doubts 29
6.2.6 View saved text doubts 29
6.2.7 Video broadcasting 29
6.2.8 Instructor logout 29
7. Technical Details 30
7.1 Student module 30
7.2 Instructor module 31
7.3 Broadcast module 31
7.4 Audio recording 32
7.5 Database and server files 32
8. Challenges 33
9. Conclusion 33
10.Appendix A 34
11.Appendix B 39
11.Bibliography 44

Chapter 1

Introduction

Audio Video Class Room Interaction on Aakash is an educational software promoting interactive classroom environment among students and teachers, being developed at IIT Bombay as a part of National Mission on Education through ICT (NMEICT), MHRD. It mainly establishes interactive session in classroom. There are two aspects of the software. One of them is instructor, who initiates the session and has facility of listening video doubts of a particular student in between the session, see recorded audio and text doubts and broadcasting his session to whole class. Another face is student, who can ask doubts either in current session or save his doubts. The student has facility of choosing any instructor to ask his question and able to join a particular broadcast. It is an effective software which can be utilised in big classrooms to full extent where a student gets a nice opportunity to ask questions without hesitation.

We developed a Web Based Application such that instructor can access the utility directly from internet without downloading anything. Similarly, the student can easily interact with the instructor in any session.

1.1 Purpose

Web based development for this software has provided an efficient and easier way to extract and manage the software. It does not require any installation and hence is user friendly. This Web Based Application is made with the purpose of establishing interactive session between students and teacher. We are using simple database to store student and instructor information. It can be accessed in any classroom by any instructor. It also provides an efficient way of learning, as it increases interaction between student and teacher. It is a good software for students who hesitate in the class for asking questions and do not get their concepts clear. It increases the level of understanding of each student. The main purpose of making this document is to describe all the features and its working to make any enhancements in future. To increase the students interest in the classroom session and also increase their understanding.

1.2 Scope

This web based application makes the class room session interesting by increasing interaction between students and teacher. It provides the facility of asking questions in live environment and storing student doubts. This feature helps in developing good understanding among students. It can be used in every institutes during online or offline session (whenever teacher is offline student can still store his doubt). Its area of use mainly comprises of big classrooms where a lot of time is wasted in doing manual labour to circulate mic in the big hall. In such big halls, it proves to be worth.

This software can also be used to provide lectures to large number of students existing at different places in institute simultaneously by broadcasting feature. Overall it is more interactive environment over and above our normal classes.

1.3 Definitions, Abbreviations

1. WebRTC: Web Real Time Communication
2. RTP: Real-time Transfer Protocol
3. RTCP: RTP Control Protocol
4. SDP: Session Description Protocol
5. ICE: Interactive Connectivity Establishment
6. XML : eXtensible Markup Language
7. JSP : Java Server Pages
8. SQL: Structured Query Language
9. XLS: Excel File
10. GUI : Graphical User Interface
11. UML : Unified Modeling Language
12. HTTP : Hyper Text Transfer Protocol
13. JRE : Java Runtime Environment
14. HTML : Hyper Text Markup Language
15. CSS : Cascading Style Sheet
16. Ajax : Asynchronous JavaScript and XML
17. AVCRIOA: Audio Video Class Room Interaction on Aakash
18. Real-time communication: Real-time communications (RTC) is any mode of telecommunications in which all users can exchange information instantly or with negligible latency. The term "real-time" is synonymous with "live."
19. LOC: Lines of Code
20. MAC address: Media Access Control address

1.4 Motivation

The motivation behind developing this software is mainly to make interaction between students and professors in a classroom more efficient and less time consuming. The motivation for the project was to provide an interface between student and teacher which help the instructor to conduct the classroom sessions more effectively. Also to create an interface which will allow a large number of students to participate and learn in a single classroom session and also allows those students to participate who are out of classroom.

It facilitates live questioning. In a big classroom, if a instructor is giving any lecture then student can ask the questions if he doesn't understand any point. Also, due to lack of time if session is over, student has the facility to store his doubt which will be answered by instructor in next session. He can store his question in his own voice by utilizing the feature of audio recording or in the form of text. An instructor has been provided with the facility of broadcasting his session. All these features of the software overcome the problems like consuming large time, less interaction among students and teachers, availability of a person at a particular place at a time.

Chapter 2

Objective

The main objective is to design and implement interactive class room application for audio and video communication in aakash tablet using HTML 5.It provides an interactive class room environment between student and teacher for sharing their doubts / queries through audio and video in the tablets. It enables the feature of audio and video in a table using HTML 5, and stream the data from tablet to communicate with web server.

The objective of each assigned task can be described as:

Live Questioning: This facility allows the student to ask his doubts in live session. It is attained through live audio video streaming between instructor's desktop pc and student's tablets over IITB local server. Server stores the data in database, which is received from instructor and student module and whenever required by client can be processed. Instructor has privilege to select a particular students from the list of those students who are eager to ask questions.

Audio Recording: It aims mainly for a student to save his doubts in his own voice using audio recording. An instructor is able to listen those doubts and save those doubts in his record. It increases better understanding of concepts in each and every student by getting his doubts clear.

Broadcasting: It aims at providing a instructor with the facility of broadcasting his session to everyone. At a time a student can select only one instructor and he has the options for selecting the instructor to which he wants to join, from the list of live broadcasters.

Text Doubts: The student has privilege to ask his doubts in text format. An instructor can answer those doubts according to his wish.

Overall main objective of this software is making a classroom environment more interactive and interesting to enhance student's knowledge.

Chapter 3

Design Considerations

3.1 Assumptions and Dependencies

Assumptions

1. An instructor uses desktop pc and student uses tablet.
2. The student who is looking forward to use this software should have a mild experience of using tablet. It is very simple to use and handle, just it requires little bit of experience of using tablet.
3. Any instructor who wants to use this software, he should be already registered.

Dependencies

AVCRIOA software works on platforms like windows, linux and android. We have used following softwares to develop AVCRIOA software:

1. Eclipse-Indigo for JEE developers
2. MySQL Administrator
3. PHP MyAdmin
4. Google Chrome 27 and above version
5. Apache Tomcat 7.0
6. Aakash Tablet
7. Node js server
8. We need to enable two flags in chrome browsers: getUserMedia and webRTC.

All these softwares are open source softwares and are available freely.

3.2 General Constraints

AVCRIOA has some general constraints or limitations which should be known for better working of software:

1. Web Audio API is not supported in chrome 27 in android. Upcoming versions of chrome will support this feature which is required for audio recording.
2. We are dealing with one way streaming only i.e; from student to instructor in live questioning and from instructor to student in broadcasting.
3. Time limit for audio doubt is 5 minutes only.
4. WebRTC supports 10 RTC Peer Connections simultaneously which restricts the number of clients for any broadcasters to be 10.
5. Text doubts is one way. A student can ask question but an instructor cannot respond to that student individually. He needs to answer it in next session for the whole class.

3.3 Goals and Guidelines

1. The main goal behind this project is to provide an efficient environment for instructor and student to have an interactive session in the classroom.
2. To implement new technology of webRTC and HTML5.
3. To strengthen the student and teachers relationship by giving facility of live questioning.
4. This technology supports real-time communication without the use of plug-ins in browser.
5. This technology enables more effective, more efficient, and more engaging education.

3.4 Technology Used

1. **WebRTC:** WebRTC is an API definition that enables web browsers with real-time capabilities using simple javascript and HTML5. It enables browser to browser applications like voice calling, video chat and P2P file sharing without the use of plug-ins.
2. **HTML5:** It is a markup language for designing web pages and presenting the contents of World Wide Web.
3. **Nodej.js:** Node.js is a server-side software system designed for writing scalable Internet applications, notably web servers. Programs are written on the server side in JavaScript.
4. **CSS:** (CSS) is a style sheet language used for describing the presentation semantics (the look and formatting) of a document written in a markup language.
5. **JavaScript:** It is a prototype-based scripting language that is dynamic, weakly typed and has first-class functions and is mainly used for validation etc.
6. **Ajax:** AJAX is the art of exchanging data with a server, and updating parts of a web page without reloading the whole page.
7. **JQuery:** It is a cross-browser JavaScript library designed to simplify the client-side scripting of HTML. It also includes the functionality of Ajax.
8. **JSP:** JSP provide excellent server side scripting support for creating database driven web applications. JSP enable the developers to directly insert java code into jsp file, this makes the development process very simple and its maintenance also becomes very easy.
9. **MySQL:** It is the world's most used open source relational database management that runs as a server providing multi-user access to a number of databases.
10. **Servlet:** A servlet is a Java programming language class used to extend the capabilities of servers that host applications access via a request-response programming model.
11. **XML:** The web.xml file defines each servlet and JSP page within a Web Application. It also enumerates enterprise beans referenced in the Web application. The file goes into the WEB-INF directory under the document root of a web application.

12. **Java:** It is a technology used to design several types on application on different platforms like android, windows etc.

Detailed description of new technologies and uses of above listed technologies in our software is as follows:

To acquire and communicate streaming data WebRTC implements following APIs.

1. MediaStream: Media Stream API allows browser to get access of user's camera and microphone to take stream from it. In our software we are using predefined function getUserMedia( ) to take local stream through user's camera.
2. RTCPeerConnection: RTCPeerConnection is the WebRTC component that handles stable and efficient communication of streaming data between peers. We are using RTCPeerConnection() object to exchange their respective streams between two browsers.
3. RTCDataChannel: The RTCDataChannel API will enable peer-to-peer exchange of arbitrary data, with low latency and high throughput. We are not using this API in our software.

There are number of protocols that is used in application layer of webRTC. The Real-time Transport Protocol (RTP) is required to be implemented as the media transport protocol for WebRTC. RTP itself comprises two parts: the RTP data transfer protocol, and the RTP control protocol (RTCP). RTCP is a fundamental and integral part of RTP, and must be implemented in all WebRTC applications.

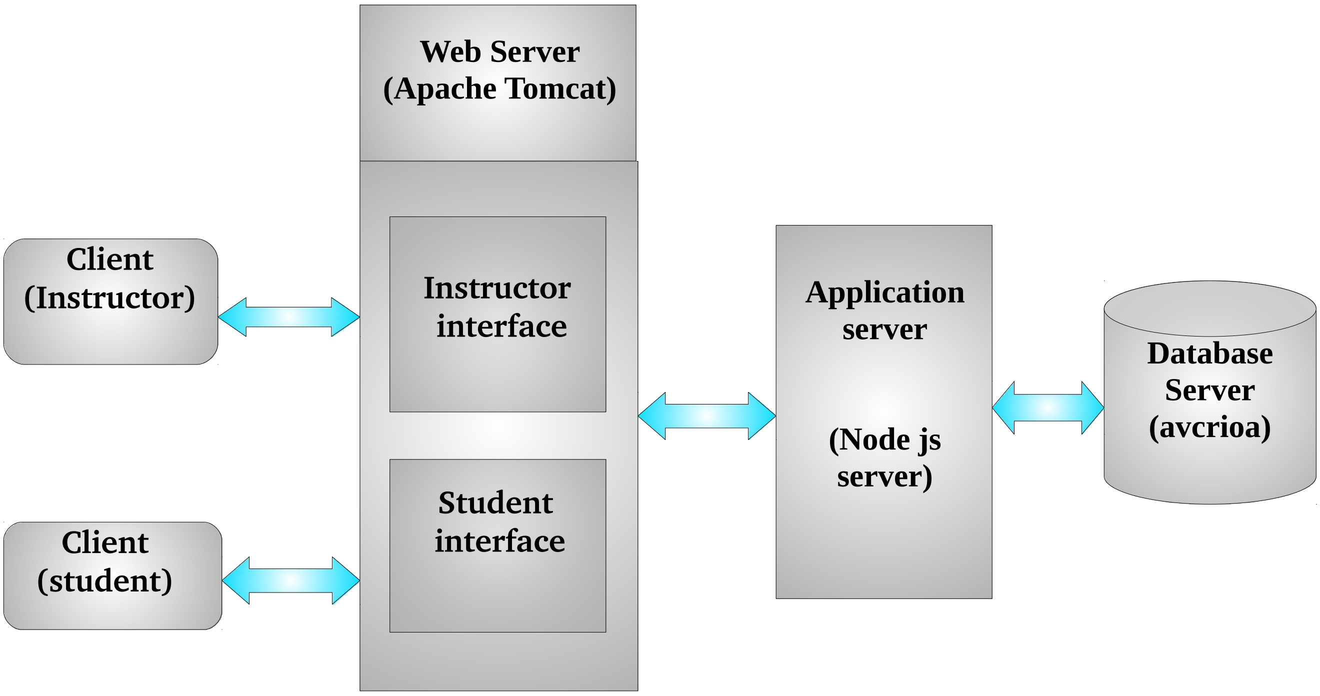
It also uses Session Description Protocol (SDP). When initiating multimedia teleconferences, voice-over-IP calls, streaming video, or other sessions, there is a requirement to convey media details, transport addresses, and other session description metadata to the participants. SDP provides a standard representation for such information, irrespective of how that information is transported.

It also uses ICE Protocol. ICE is a framework for connecting peers, such as two video chat clients. Initially, ICE tries to connect peers *directly*, with the lowest possible latency, via UDP. If UDP fails, ICE tries TCP: first HTTP, then HTTPS.

- Node.js: For our software, this technology is used to create server program.
- HTML5: This is the fifth revision of the HTML standard. This provide features for supporting multimedia. It is come up with audio and video elements which are used in our software to display the streams on browsers.
- Java-Script: We are using this technology to do socket programming to create the signaling channel to transfer the streams.
- JSP: We are using this technology in creating dynamic web pages in our software.
- MySQL: For storing student and instructor information.

Chapter 4

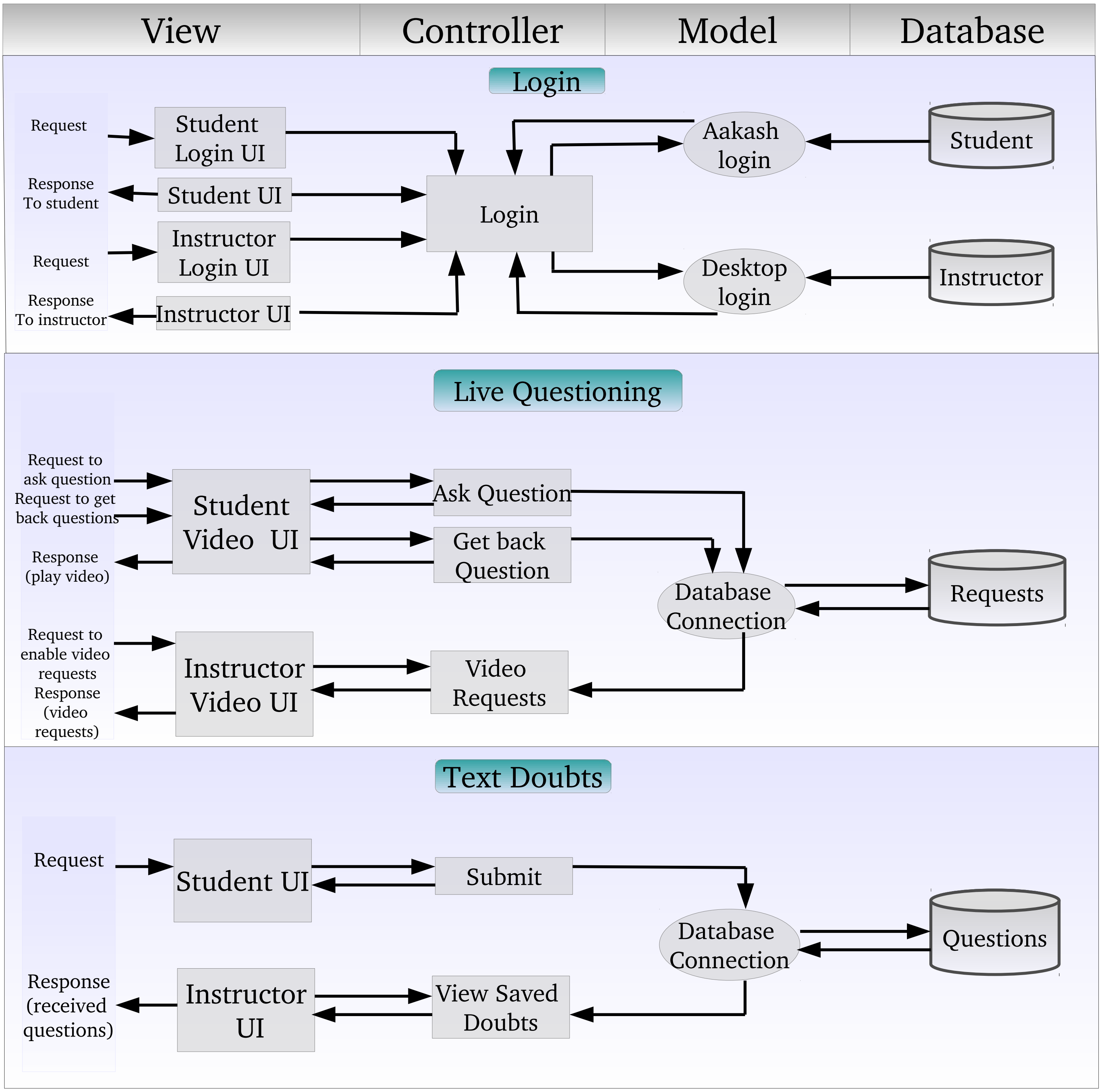
Architecture

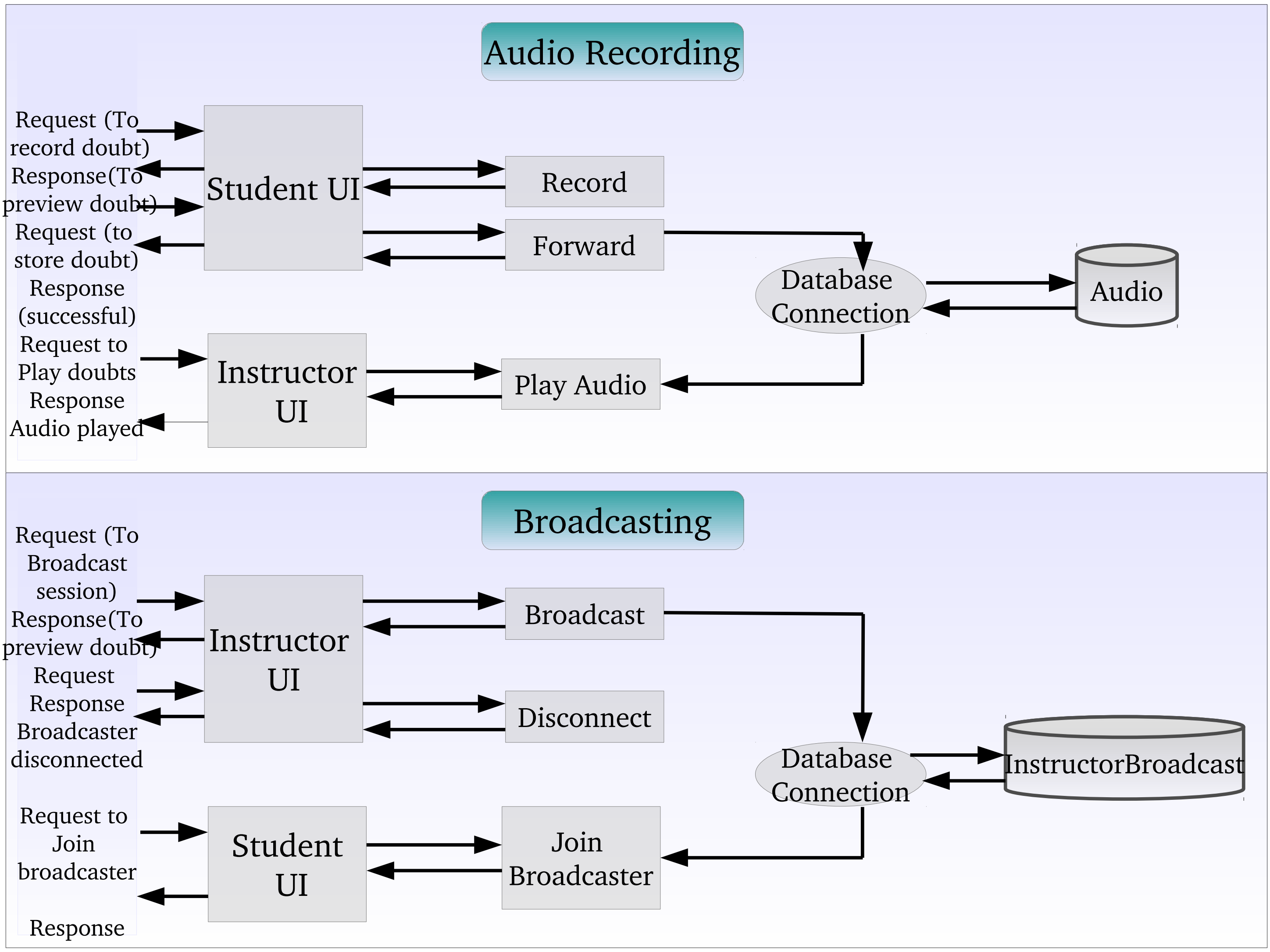


Chapter 5

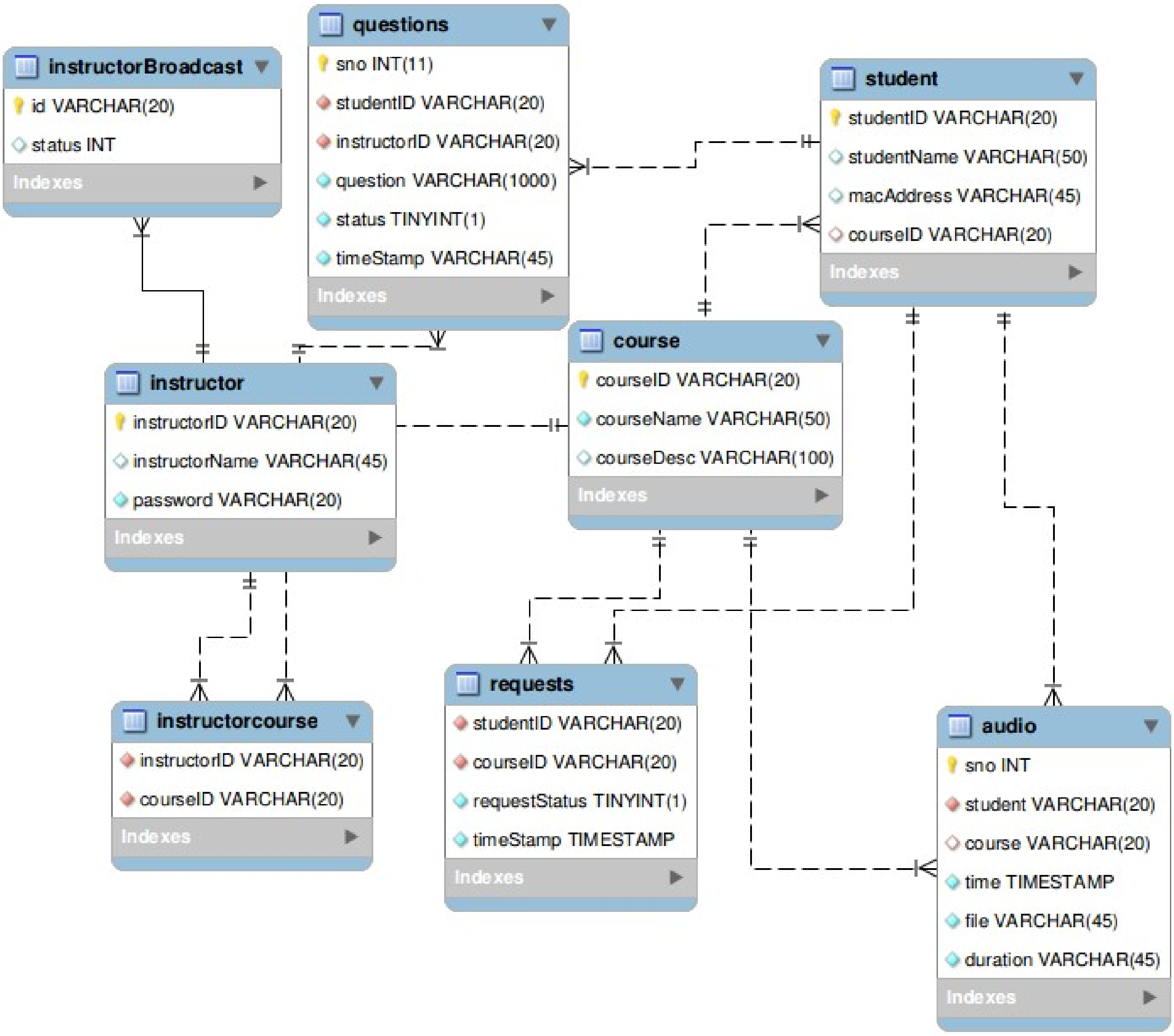
Design and Implementation

5.1 System Flow Diagram



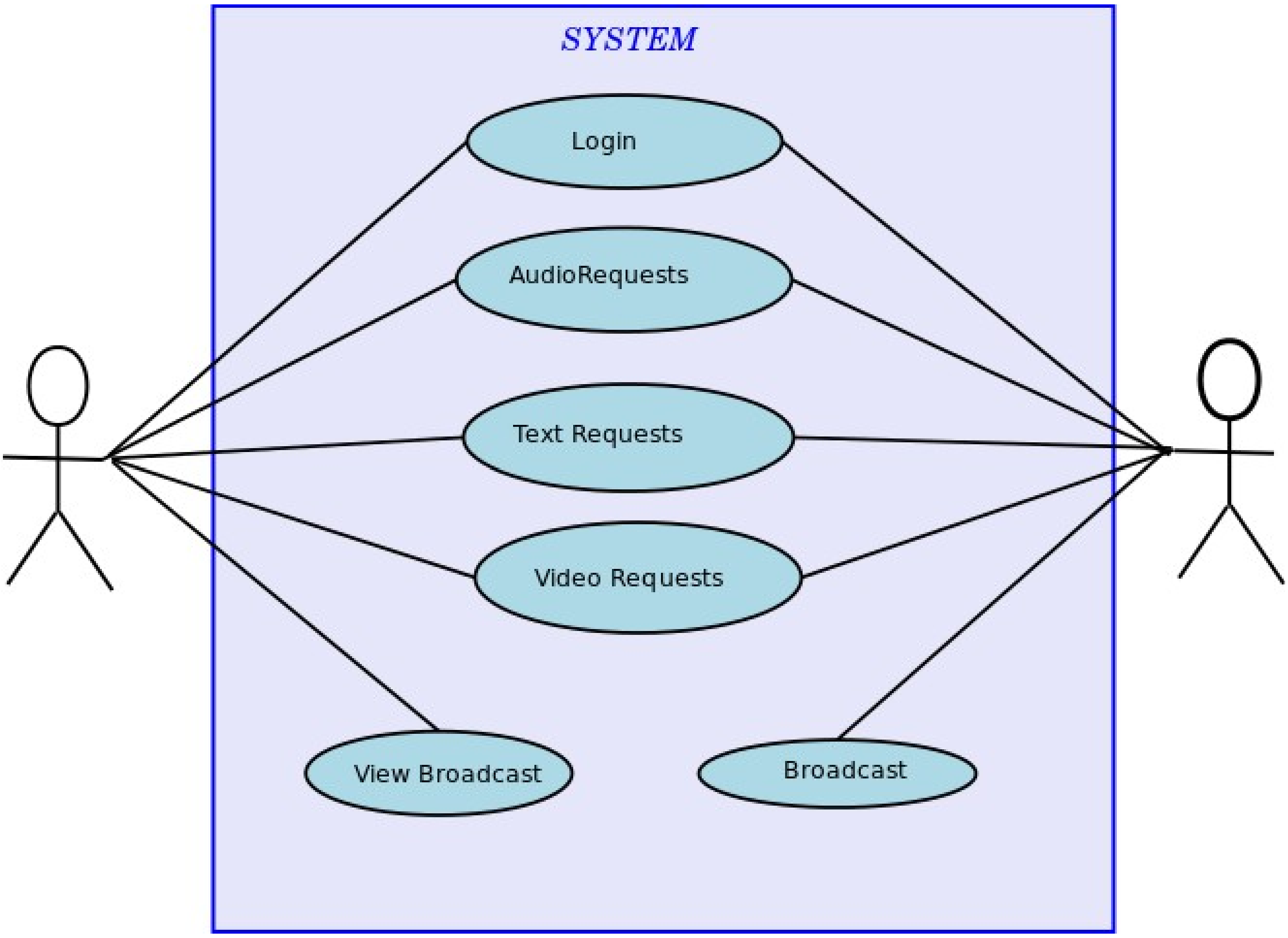

Audio Recording
Request (To record doubt)
Response(To preview doubt)
Request (to store doubt)
Response (successful)
Student UI
Record
Forward
Database Connection
Audio
Request to Play doubts
Response Audio played
Instructor UI
Play Audio
Broadcasting
Request (To Broadcast session)
Response(To preview doubt)
Request
Response Broadcaster disconnected
Instructor UI
Broadcast
Disconnect
Database Connection
InstructorBroadcast
Request to Join broadcaster
Response
Student UI
Join Broadcaster

5.2 Schema Diagram

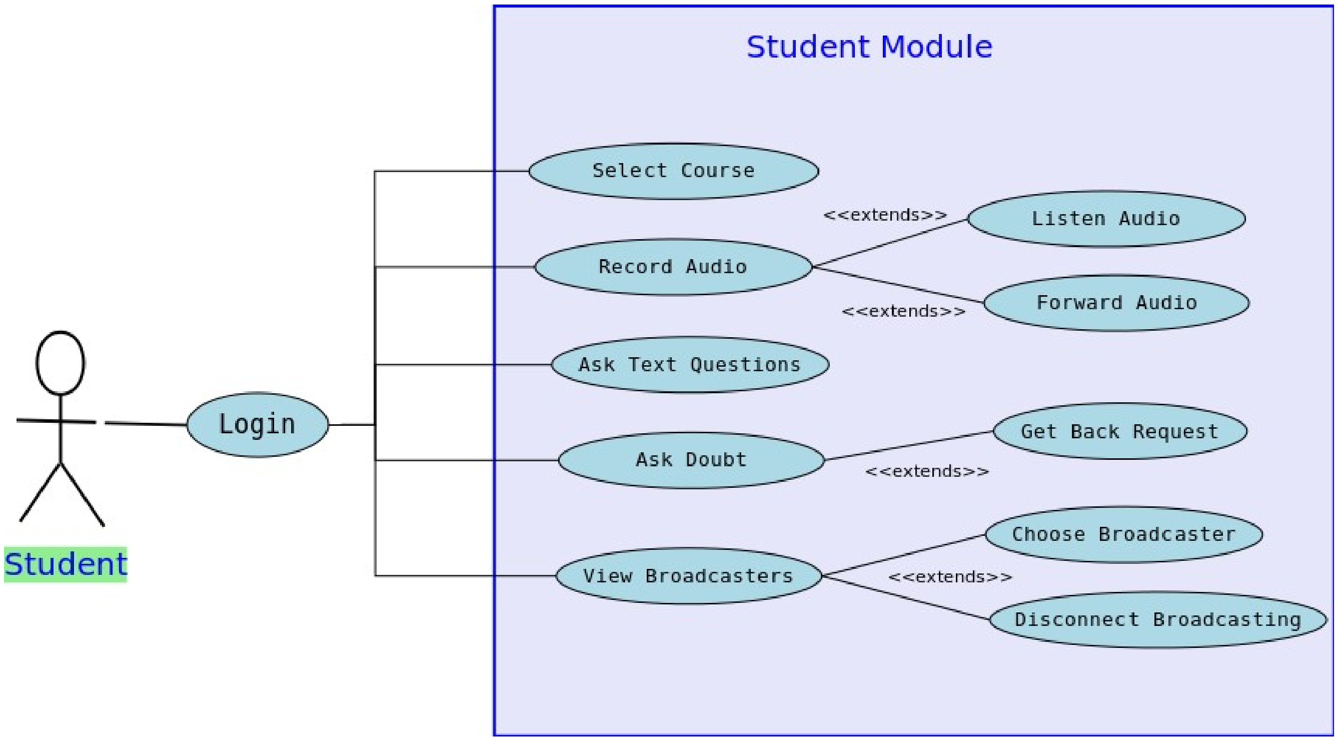


5.3 Use Case Diagram

5.3.1 Overall

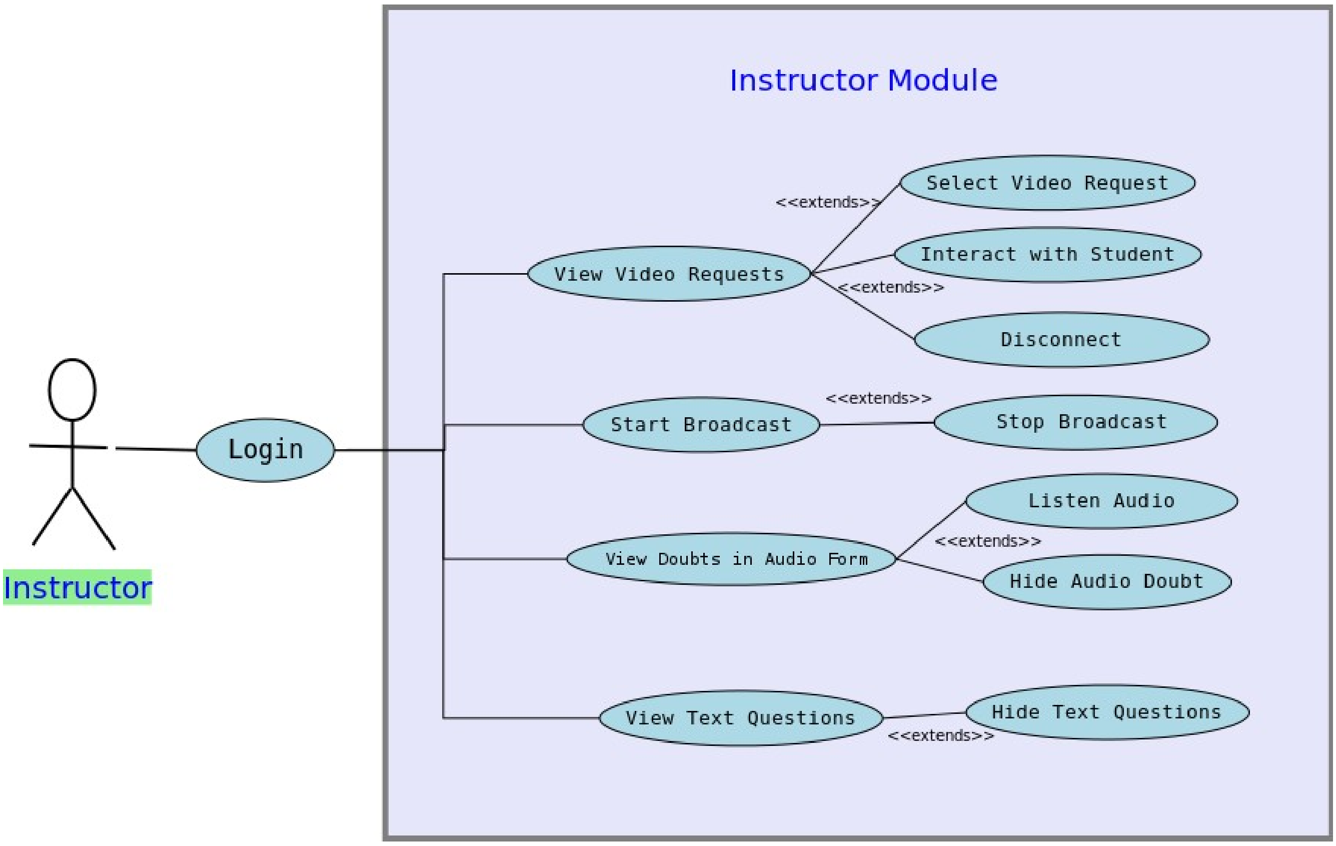


5.3.2 Student Use Case Diagram



Use case	***Description***
Login	Student must login with a valid user name from aakash tablet
Select Course	Student should select course in order to avail any functions
Record Audio	Student can record audio from microphone
Listen Audio	Student can listen audio what he has recorded
Forward Audio	Student can forward audio doubt to instructor
Ask Text Questions	Student can submit text questions to the instructor
Ask Doubt	Student sends the video request to Professor
Get Back Request	Student can cancel the video request what he has sent
View Broadcasters	Student can view the list of broadcastings going on
Choose Broadcaster	Student can choose the broadcaster and can view broadcasting
Disconnect Broadcasters	Student can disconnect from broadcasting

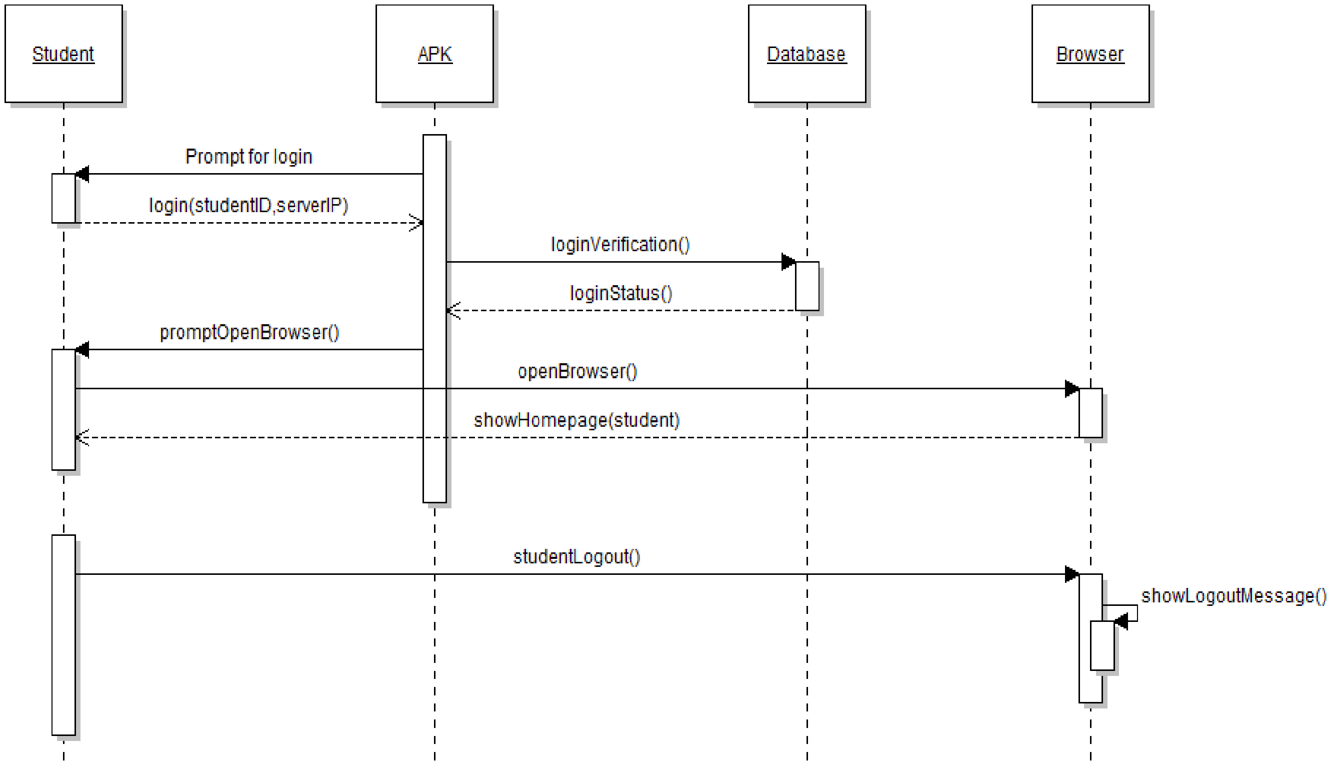
5.3.3 Instructor Use Case Diagram



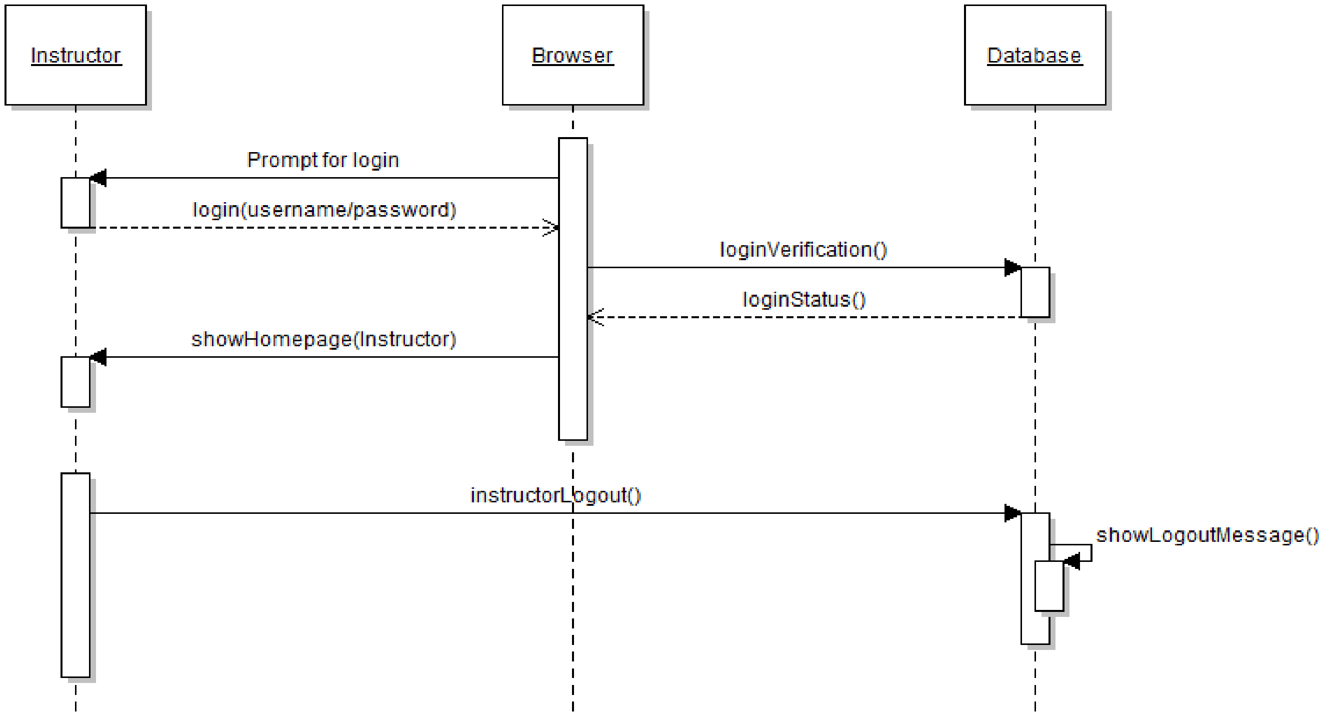
Use Case	Description
Login	Instructor must login with valid user name and password.
View Video Requests	Instructor can see video requests from the users to connect
Select Video Request	Instructor selects the video request.
Interact with student	Instructor can see student and listen to student's doubt.
Disconnect	Instructor can disconnect from the video request
Start Broadcast	Instructor can start broadcasting his lecture.
Stop Broadcast	Instructor can stop broadcasting at any time.
View Doubts in Audio Form	Instructor can view the doubts which are in audio form from student
Listen Audio	Instructor can listen the student's doubts
Hide Audio Doubt	Instructor can hide the audio doubts from future listings
View Text Questions	Instructor can view text questions posed by students
Hide Text Questions	Instructor can hide text questions from future listings.

5.4 Sequence Diagram

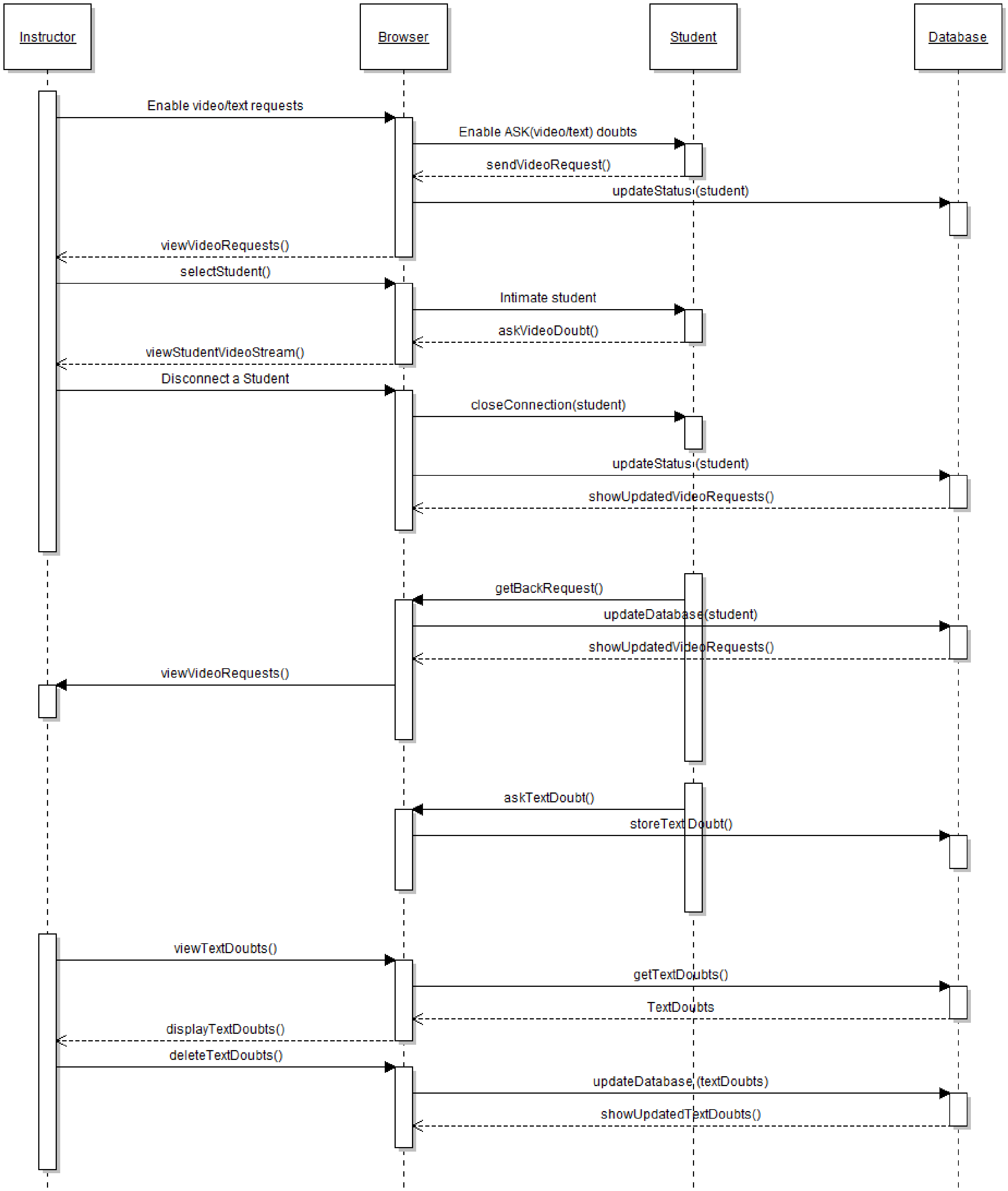
5.4.1 Student login/logout



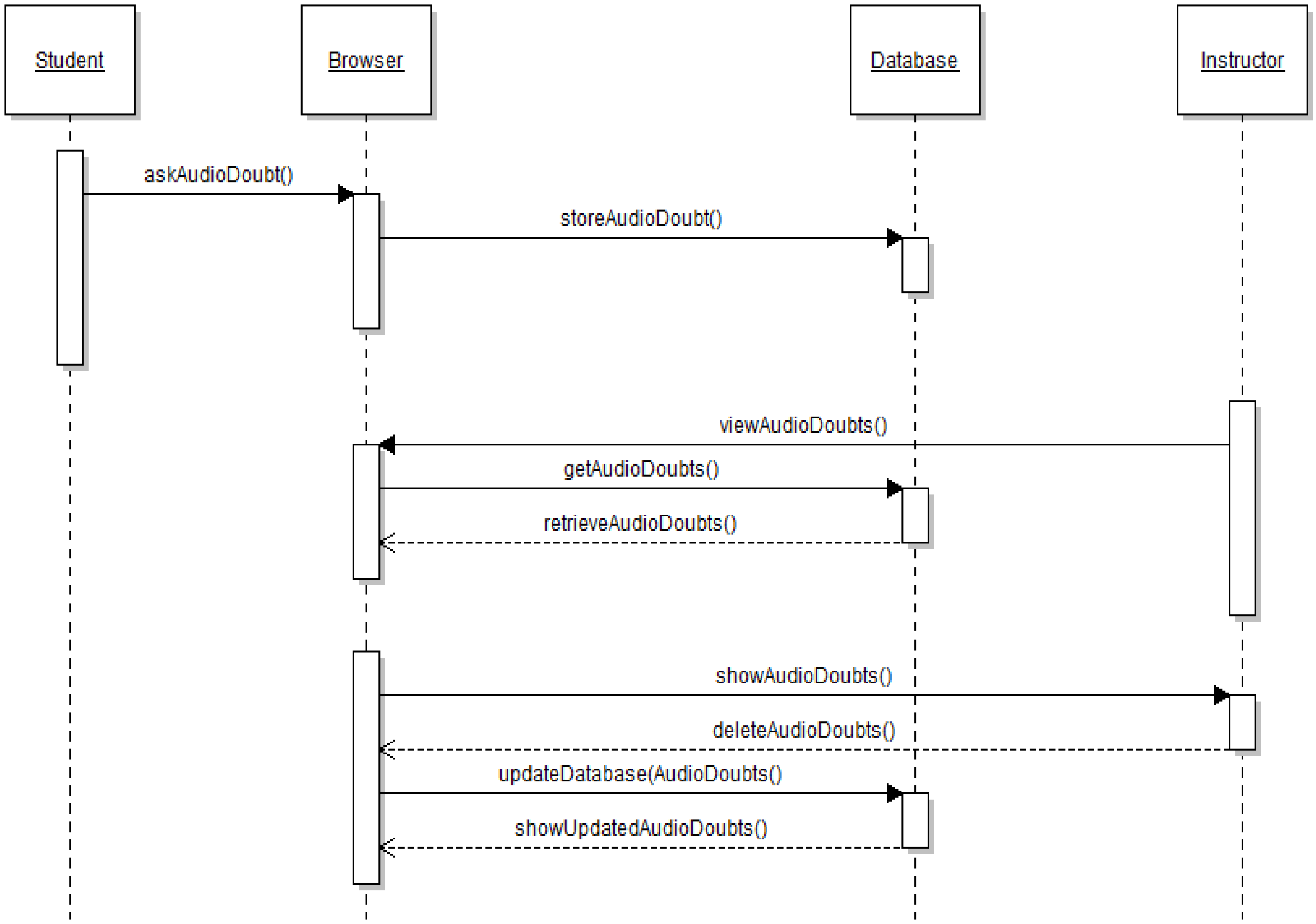
5.4.2 Instructor login/logout



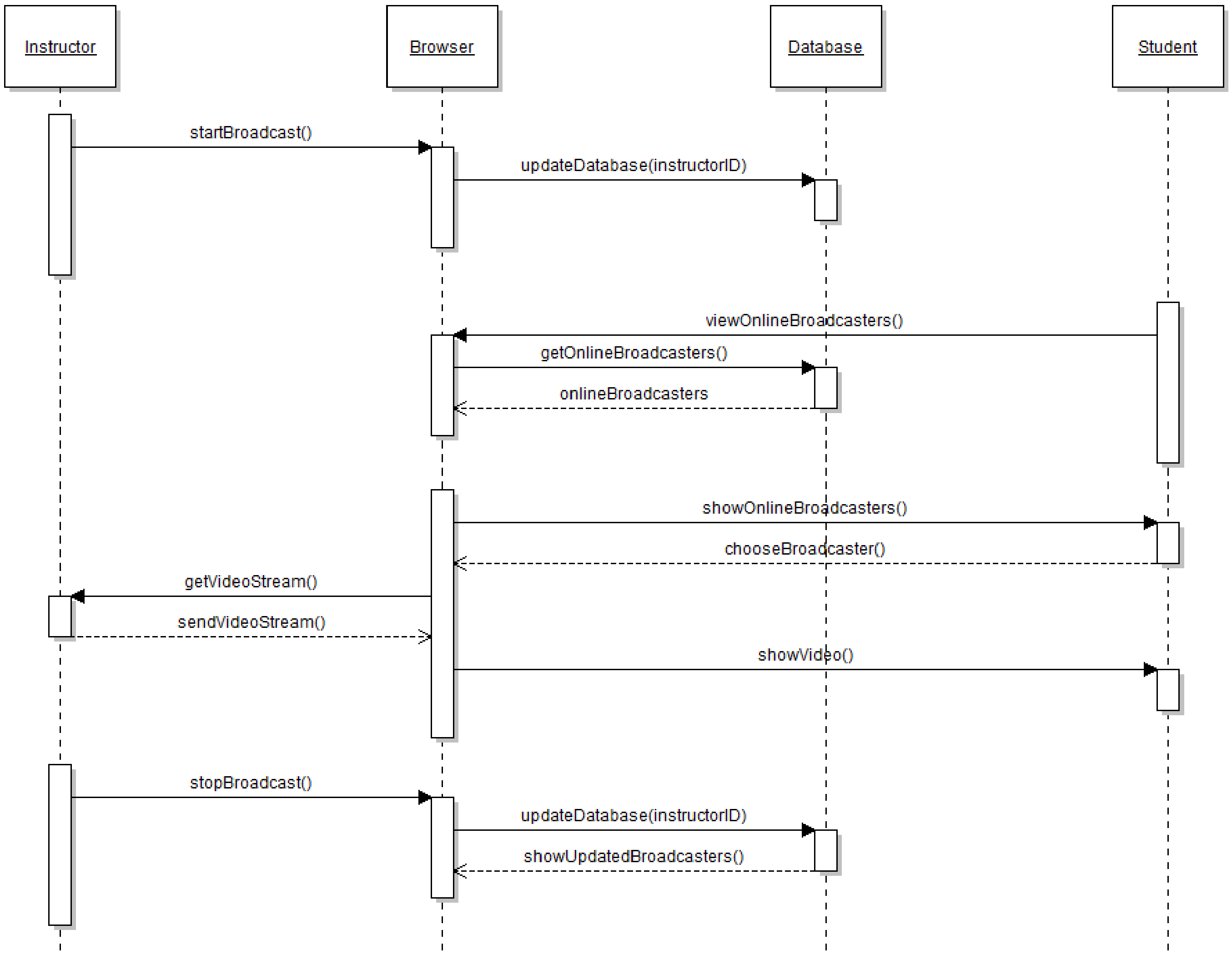
5.4.3 Video/Text doubts



5.4.4 Audio doubts

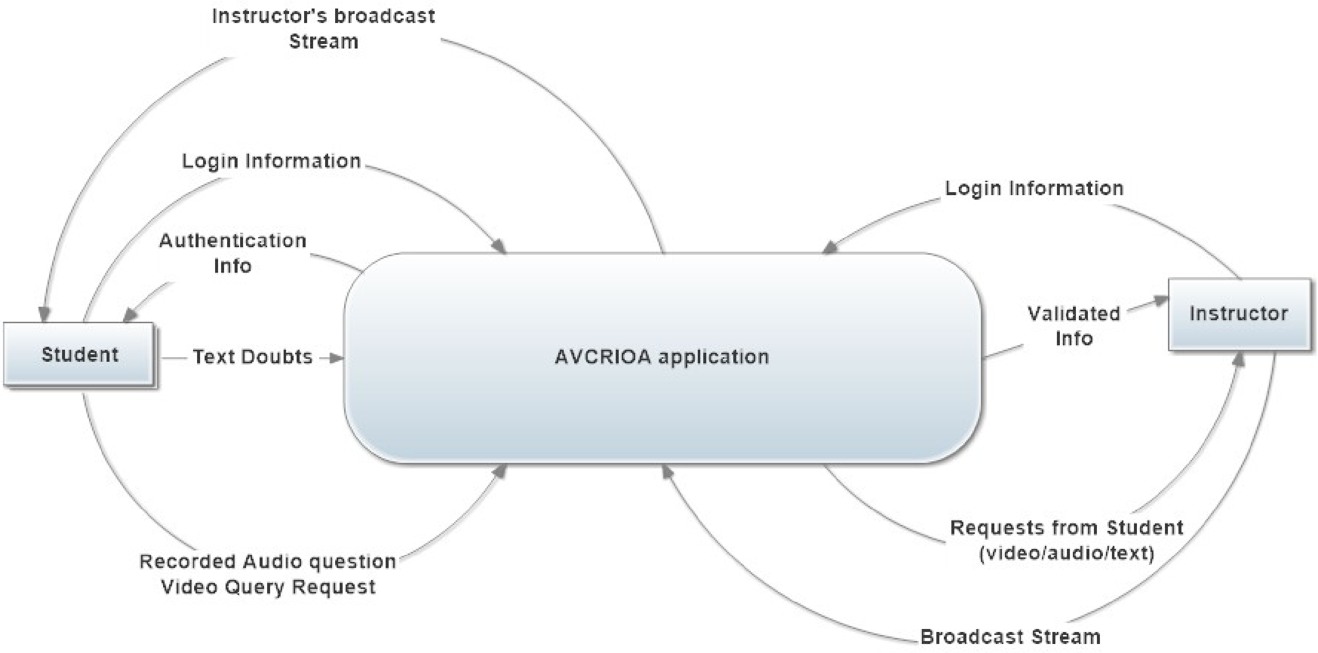


5.4.5 Video Broadcast



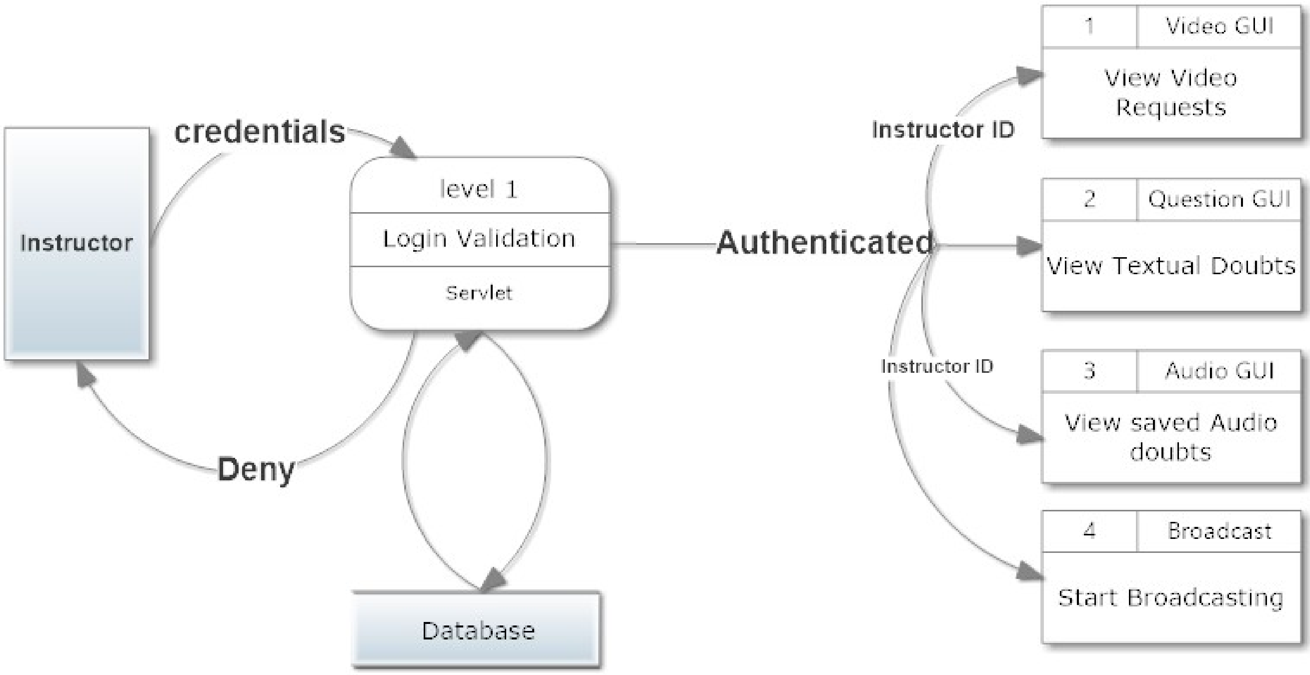
5.5 Data Flow Diagrams

5.5.1 Level 0 DFD

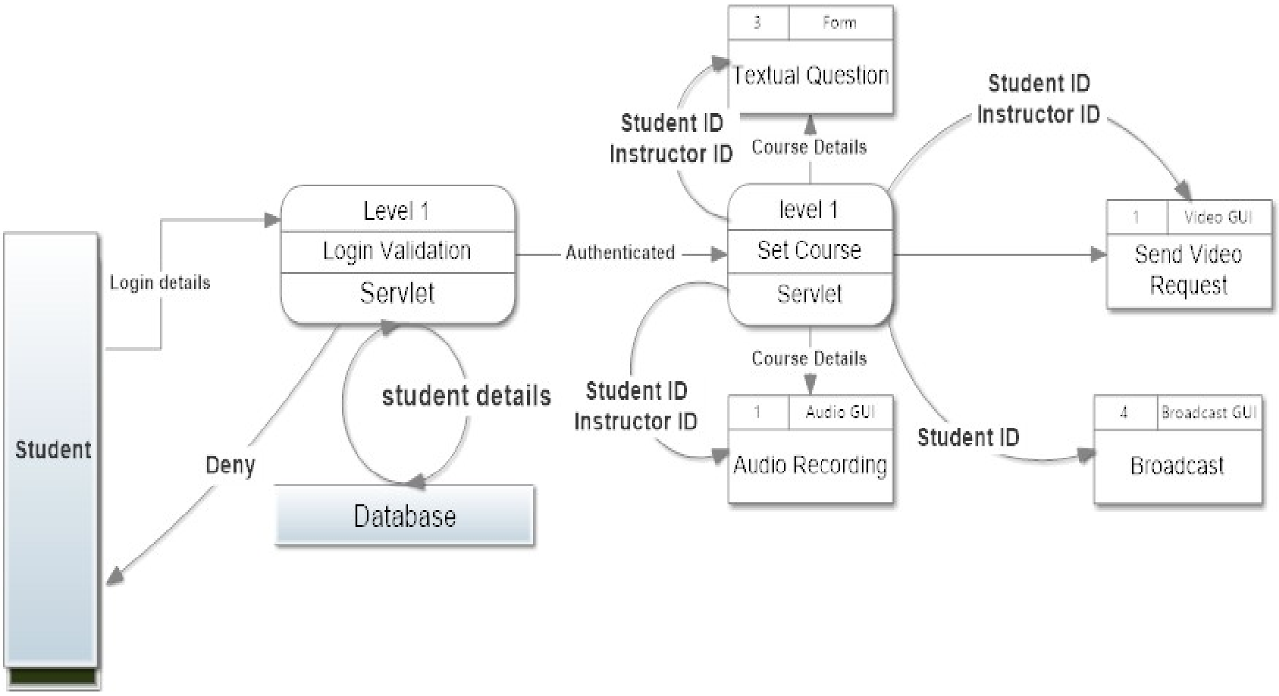


5.5.2 Level 1 DFD

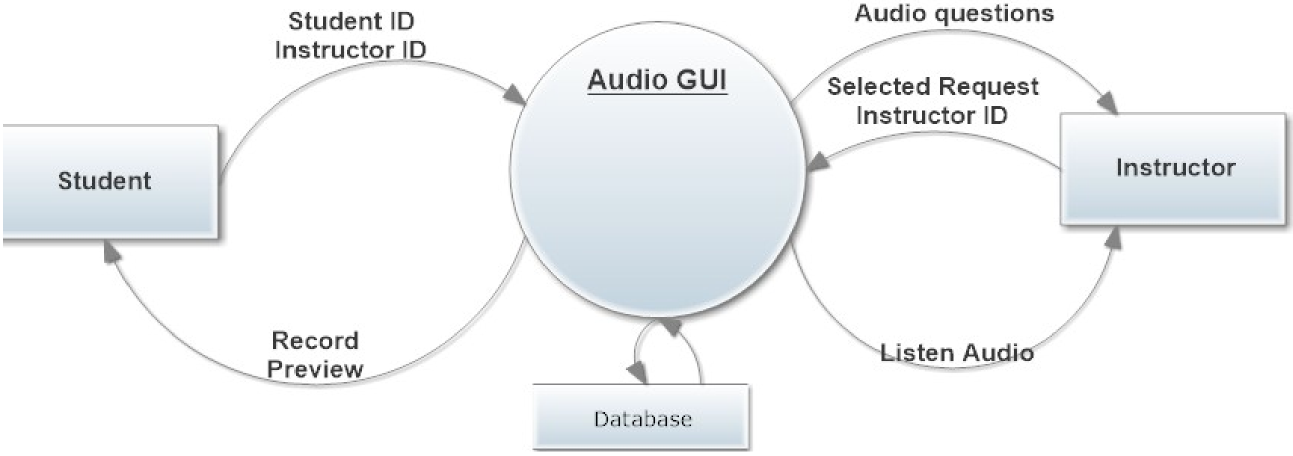
Instructor



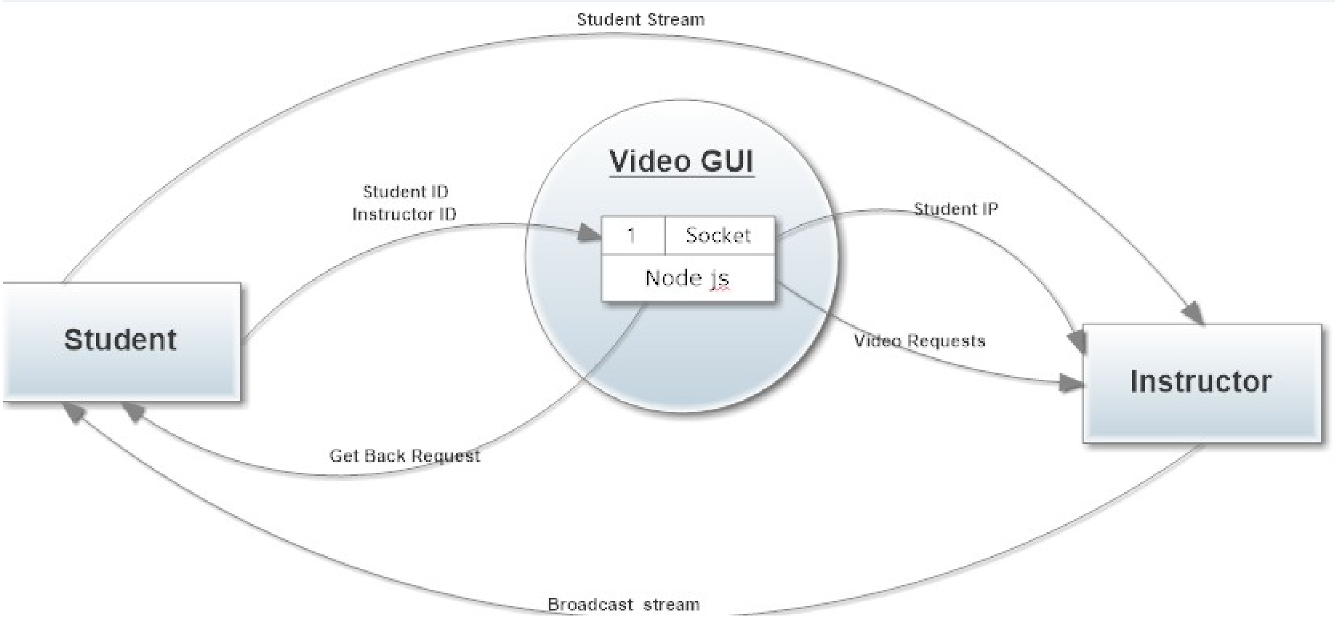
Student



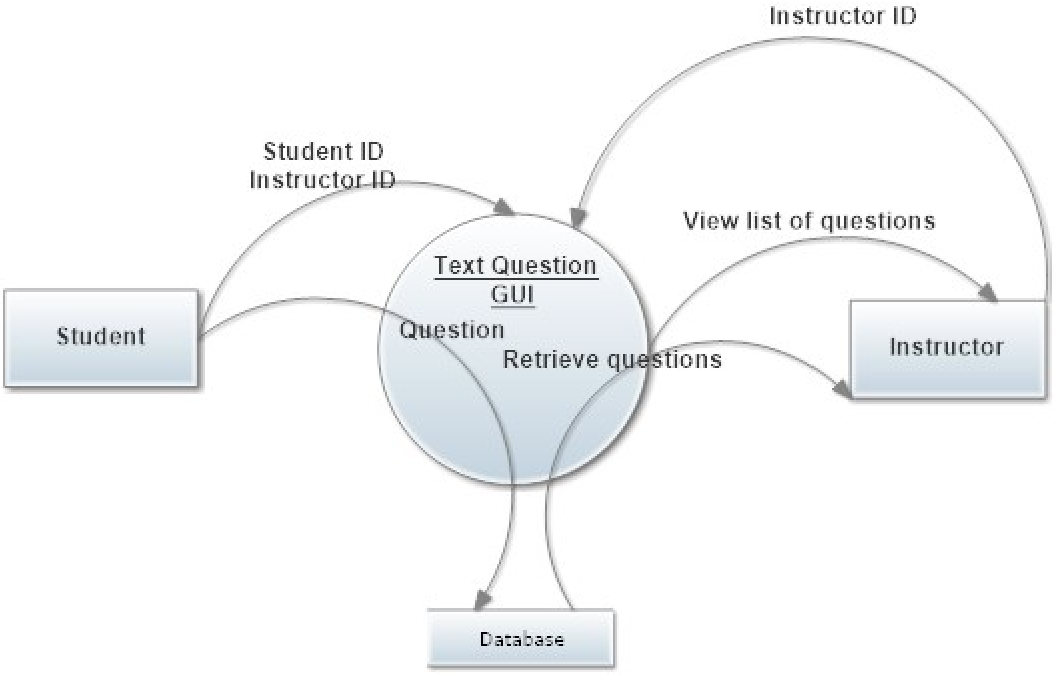
Audio GUI



Video GUI



Text GUI



Chapter 6
Detailed Description

6.1 Student Module

6.1.1 Student Registration

A student have to register for AVCRIOA before using its functionality. This is done for the first time he accesses AVCRIOA. The student can register for multiple courses preferably to the courses of his current semester. Every student will be given a unique 'StudentID' and also a 'Server IP', which are to be used for logging in to the AVCRIOA application. Once a student is successfully registered for AVCRIOA, he will be free to access the application.

6.1.2 Student Login

For any student, login is mandatory in order to use AVCRIOA. A student has to login through the apk of AVCRIOA, which asks for a student's StudentID and Server IP. The StudentID is given to the student at the time of his prior registration. The server IP is the address of the server on which the application is hosted. The actual interaction between the student and instructor goes over the mentioned server ip. Once a student enters his StudentID and Server IP and clicks 'CONNECT', he will be redirected to the browser (preferably Chrome 27 or above), from which the actual interaction occurs. The above mentioned procedure is followed only once, at the first time a student connects to a server. From the next time, the previously entered details, which are stored by the apk file are automatically used, unless a student explicitly wants to change the server ip. From the second time of student login, only 'CONNECT' option is visible to the student and once he clicks the button, he will be automatically connected to the server. What actually happens when the student clicks the CONNECT option is that the MAC address of the aakash tablet will be extracted by the application and is stored in the server's database. This occurs for the first login. From then on, the MAC address sent by the application is compared with the already stored MAC address in the database for the authentication purposes.

6.1.3 Choose a Course

Once a student successfully logs in to AVCRIOA, he is free to choose between a list of courses, under which he is currently registered. A student can choose a particular course and post a doubt to the mentioned instructor of that particular course.

6.1.4 Ask Video Doubts

A student is provided with several modes for conveying his doubts to the instructor. Once a student logs in, he will be in the 'Ask Video Doubts' section, in which he will be provided with an 'ASK' option, through which he can send a video request to the student. Once he clicks 'ASK' option, AVCRIOA asks the student permissions to access the camera and microphone of the aakash tablet. Once the student clicks 'Allow', the student's video request will be sent to the instructor and the instructor can choose a student among the several video requests he receives. The student can 'Get Back Request' of his already sent request. The 'Get Back' option is visible to the student after he sends a video request to the instructor. This feature gets back the already sent video request to the instructor i.e., the student's request will not be visible on the instructor side. But once the instructor chooses a student to listen to his doubt, the student can't Get Back the request. This feature of asking video requests will be the choice of instructor. If the instructor enables this feature, the student is allowed to send the video requests, else he can't.

6.1.5 Ask Textual Doubts

As the name itself suggests, a student can send his doubt in the form of text, just by typing his doubt in the 'Textual Doubts' section. Once a student types his doubt and posts it to the instructor, it will be stored in the database and is directly visible to the instructor. A student can't edit or delete the textual doubt already sent to the instructor. Similar to the video doubts, a student can text his doubt to the instructor only when the instructor enables it. If the instructor disables, then the student can't forward a text doubt to the instructor.

6.1.6 Audio Audio Doubts

A student can record his audio doubt and post it to a particular instructor. These audio doubts are stored in the database and are available to the instructor anytime, in the class or outside the class. This feature is mainly important if a student wants to ask his doubt but the class time is over. In such situations, a student can just drop his doubt in an audio form to the instructor, which the instructor can listen to after the class. In this section, a student is given several options like 'Start Recording', 'Stop Recording', 'Preview' and 'Forward the Doubt'. Once the student forwards an audio doubt, he can't delete it. For that purpose, every student is requested to 'Preview' the doubt before forwarding. A student can post his audio doubts not only when he is in the class or not only when the instructor is online. It's completely the choice of student on when to send the audio doubts.

6.1.7 Video Broadcasting

Every student will have an option of 'Video Broadcasting', in which the student is able to see the currently running video lectures by instructor. A list of all the currently running video broadcasts of the instructors are shown to the student and the student can choose any one of the lectures to attend.
A student can comfortably switch between all the fields like Video Doubts, Audio Doubts, Text Doubts and Video Broadcasting.

6.1.8 Student Logout

A student is provided with the 'logout' option if he wants to navigate away from AVCRIOA. Once a student logs out, his session will be completely destroyed and he has to login again from the application in case he wants to use AVCRIOA again.

6.2 Instructor Module

6.2.1 Instructor Registration

Every instructor will be registered by the database administrator of the institute and each instructor is given a unique username and password, by using which an instructor can login and use the features of AVCRIOA. Once an instructor is successfully registered for AVCRIOA, he will be free to access the application.

6.2.2 Instructor Login

An instructor will have to enter his credentials like his username and password in order to login for AVCRIOA. Once an instructor logs in, his session will be initiated and he can view various requests of the students in various forms like video, audio and text doubts.

6.2.3 Enable/Disable Video and Text doubts

Every instructor will be provided with an Enable/Disable option for the Video and Text doubt requests. A student can send video requests and text requests only if the instructor enables this field. If an instructor disables this option, any student can't send video/text request to that particular instructor.

6.2.4 View Video Requests

Once the instructor enables Video/Text requests, he will be able to see a list of video requests sent by the students to him. The instructor can choose a student to listen to him, he can disconnect a student and also switch between students. The instructor can listen to the student doubt and can clear it to the entire class.

6.2.5 View Saved Audio Doubts

In this field, an instructor can view the saved audio doubts sent by the students to him. He can respond to those doubts in the next class. The full details of the student who sent the audio doubt are visible to the instructor along with his audio doubt. These details include student ID, student name, time stamp etc. The instructor can delete the doubt after he listens to it.

6.2.6 View Saved Text Doubts

An instructor can view the saved text doubts sent to him by the students and then respond and clear their doubts in the next phase of the class. The full details regarding the student ID, student name, the time stamp etc are visible to the instructor. The instructor is provided with the option of 'DONE' to delete the text doubt from his view once he sees it.

6.2.7 Video Broadcasting

An instructor can start broadcasting of his video. The instructor will be given an option of 'Start Broadcasting' and once the instructor starts broadcasting, all the students can see that this instructor is online and can view his video.

6.2.8 Instructor Logout

The instructor is provided with the 'logout' option if he wants to navigate away from AVCRIOA. The instructor can logout from his session whenever he wants to. Once he clicks 'logout' option, he has to login again in order to use AVCRIOA.

Chapter 7

Technical details

7.1 Student Module

File Name	Description	LOC
index.jsp	This is the Welcome page for student where student has to select his course and then can see the GUI which contains options to ask a question through video, audio or in textual form.	53
broadcast.jsp	This page is linked with index.jsp where student can see online broadcasters and can join to get broadcasting stream	39
loadData.java	This servlet stores the video request sent by the student in database. This request contains the information like studentID, courseID, timestamp and status	56
setCourse.java	This servlet assigns the course selected by student in index.jsp so that the requests (video/audio/text) can forwarded to the respective instructor.	55
textDoubts.java	This servlet store the text question asked by the student with corresponding course and instructor	51
StudentLoginVerification.java	It validates the login data coming from apk in Aakash and validates the student by sending response	153
Student_connection.js	This file contains the functions to establish socket connection, capturing stream, peer connection and all other related stuff.	143
Student.js	JavaScript functions that are required by student side GUI	34
textForm.js	All the JavaScript functions that related to textual question submission	52

Total Lines of Code = 636

7.2 Instructor Module

File Name	Description	LOC
instructor.jsp	This is instructors welcome page which shows the well structured GUI for controlling and viewing questions (text/audio/video).	51
broadcast.jsp	When the instructor navigates to this page instructor has the options to start the broadcasting or disconnecting.	53
listRequests.java	Servlet that lists the video requests sent by the students. As soon as the student sent the request it will be updated at instructor side	57
loginValidation.java	Servlet that validates the instructor credentials comparing from database and redirects to appropriate page	59
questionStatus.java	Servlet that updates the status of textual question whether it is answered by the instructor or not	45
textRetrievalInstructor.java	Servlet that retrieves the textual doubts from database and lists at instructor side	51
updateRequests.java	This servlet updates the video requests whenever instructor disconnected the student, switching the student requests and also whenever student getback his requests	53
videoRequests.js	This file contains the functions for socket connection, peer connection and all other that corresponds to instructor side student video requests.	126
index.jsp	Login page for instructor which contains the basic information about the project	51

Total Lines of Code = 546

7.3 Broadcast Module

File Name	Description	LOC
broadcastLoad.java	Servlet that loads the list of online broadcasters at student side	50
broadcastUpdate.java	Servlet that updates the status of broadcaster whether broadcaster left the session or refreshed the page.	51
broadcastStore.java	Servlet that stores the broadcaster information in database	57
broadcast.js	All the functions for socket connection, peer connection, taking stream and then broadcasting to clients (students) related to broadcasting.	306

Total Lines of Code = 464

7.4 Audio Recording

File Name	**Description**	**LOC**
GetAudio.java	Servlet that fetches the audio blob which was stored in database when student sent the request and plays at instructor side	95
GetAudioQuest.java	Servlet that lists all the stored audio doubts with timestamp	84
UploadAudio.java	After recording the audio question from student side this servlet stores it in the database in the form of a blob object	106
record.js	Sends the appropriate headers and messages for recording of student audio to the file recorderWorker.js	71
recorderWorker.js	Records the audio in parallel with the other functions	101
recordstudent.js	Captures the audio using getUserMedia API	100

Total Lines of Code = 557

7.5 Database and Server files

File Name	**Description**	**LOC**
dbconnection.java	Returns a connection object to establish connection with database	23
server.js	Node js server that acts as intermediates while exchanging the session description between peers.	40
style.css	All styles for all modules	358

Total Lines of Code = 421

Total Lines of Code in entire project = 2624

Challenges & Solutions achieved

1) Choosing technologies for live video streaming.

In our search for live video streaming technologies, we came to know about the new technology WebRTC, being developed by Google, which offers browser to browser real time communications without the need of any plugins and any intermediate video streaming servers.

2) Search for WebRTC supporting browsers.

As WebRTC is a new technology, still under development, all the browsers doesn’t support it. We searched very extensively by testing sample codes in both desktop and android browsers and came to know that WebRTC support starts from Chrome 26 and firefox 21.

3) Selection of suitable server for signaling purpose.

WebRTC doesn’t specify any signaling mechanism and is completely left for the developer to choose the signaling procedure. For maintaining concurrent requests and broadcasting the messages between peers, we chose socket.io module and node.js server, which enables the server side scripting to be done in javascript.

4) Establishing multipeer connections for video broadcasting.

We succeeded so early in establishing single peer connections, but for broadcasting, we need to establish and maintain multiple peer connections from a single browser. The problem we faced was that when a second peer connection is established, the previous connection was terminated. This problem was overcomed by establishing a separate private channel for each peer connection.

5) Recording audio.

We faced a problem in uploading the recorded audio file directly to mysql database, without saving the recorded file locally or on server. We have uploaded the audio file using AJAX without saving it in a blob format.

Conclusion

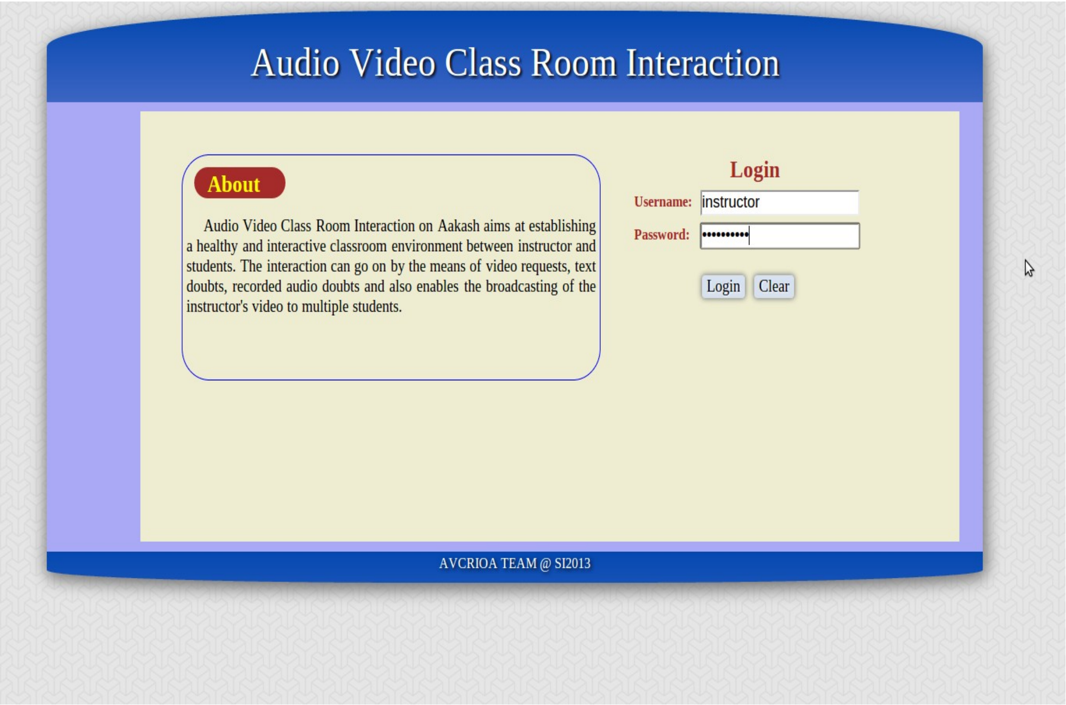
The basic aim of the project is to establish interactive class room environment by replacing mi at student side. All the functionalities mentioned above are successfully completed. The future work regarding AVCRIOA can be

1. Extending the functionality of broadcasting to more than 10 peers.
2. Adding the functionality of getting instructor side stream which is presently one way from student to instructor.

Appendix A

1.Instructor module

1. **Instructor login page**



Audio Video Class Room Interaction

About

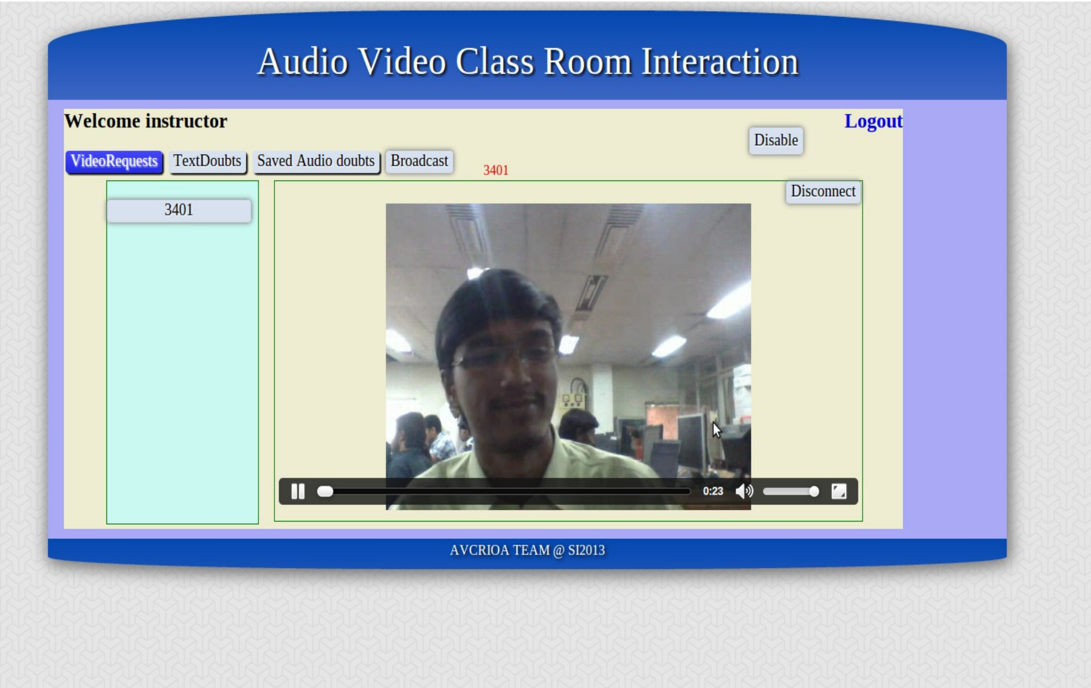
Audio Video Class Room Interaction on Aakash aims at establishing a healthy and interactive classroom environment between instructor and students. The interaction can go on by the means of video requests, text doubts, recorded audio doubts and also enables the broadcasting of the instructor's video to multiple students.

Login

Username: instructor

Password: ••••••••••

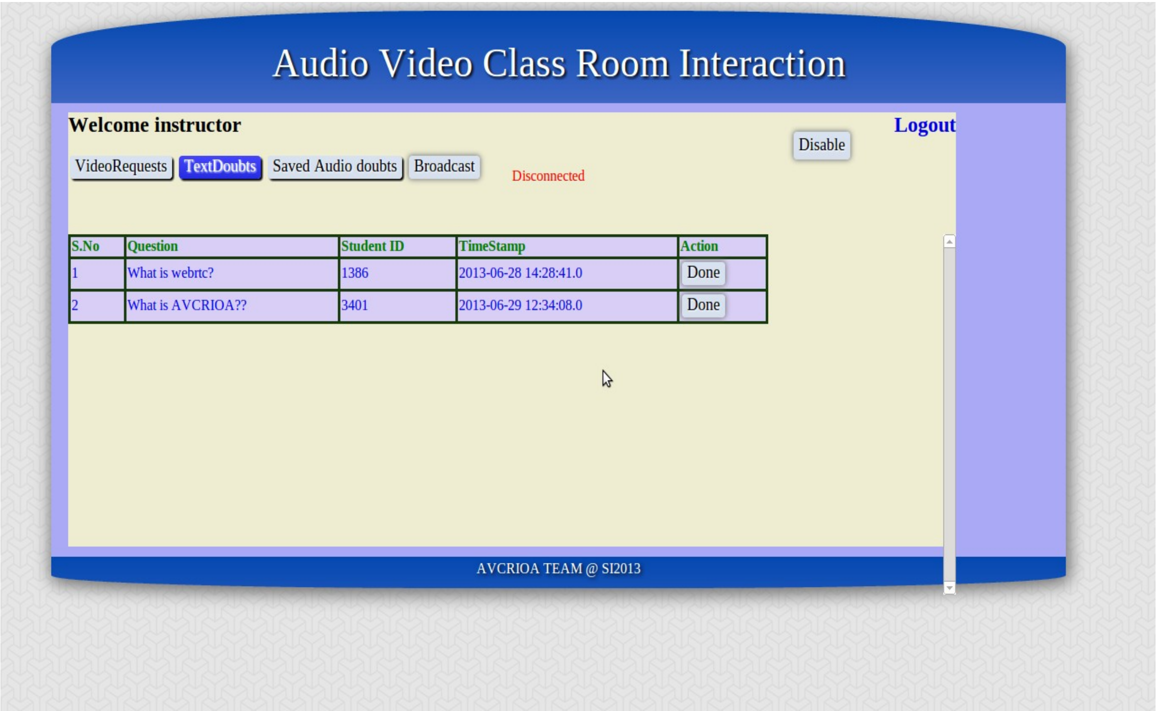
2. **Video Requests**



Audio Video Class Room Interaction

Welcome instructor

3. Text Doubts



Audio Video Class Room Interaction

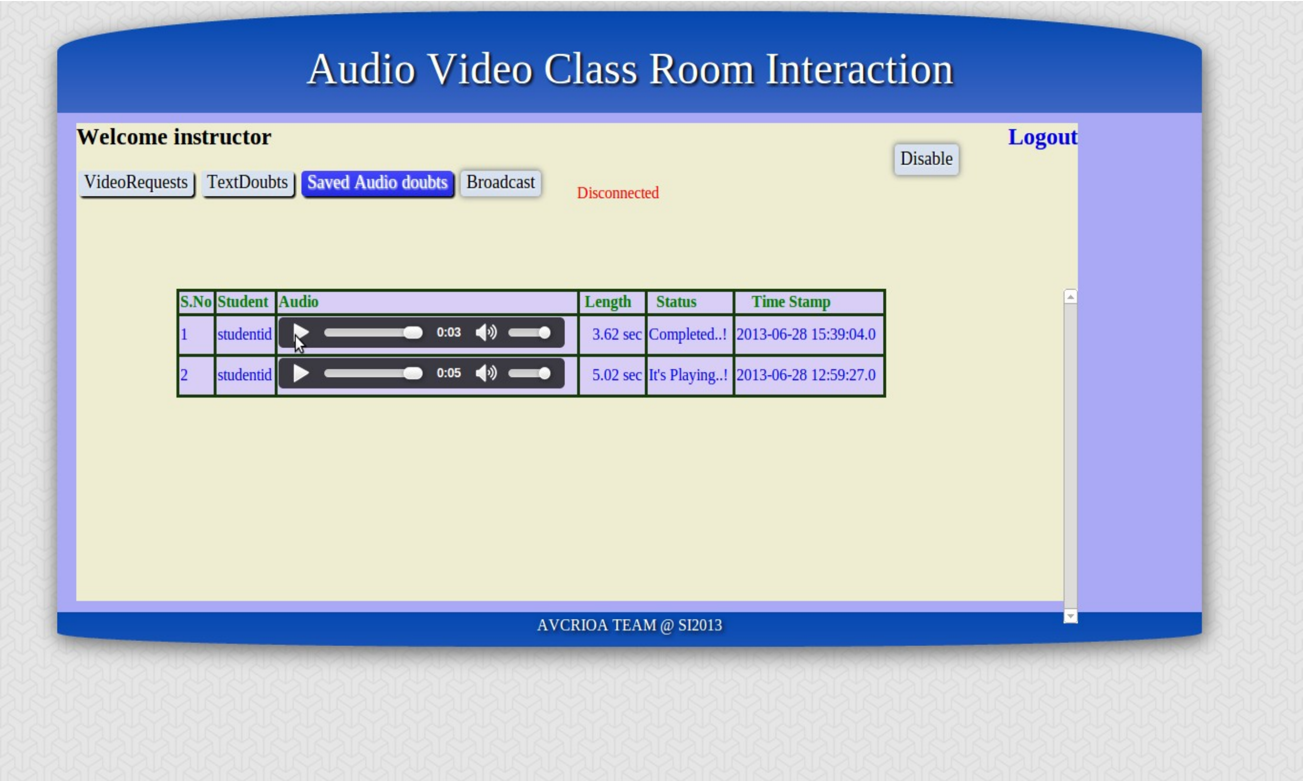
Welcome instructor

Disconnected

S.No	Question	Student ID	TimeStamp	Action
1	What is webrtc?	1386	2013-06-28 14:28:41.0	Done
2	What is AVCRIOA??	3401	2013-06-29 12:34:08.0	Done

AVCRIOA TEAM @ SI2013

4. Saved Audio Doubts



Audio Video Class Room Interaction

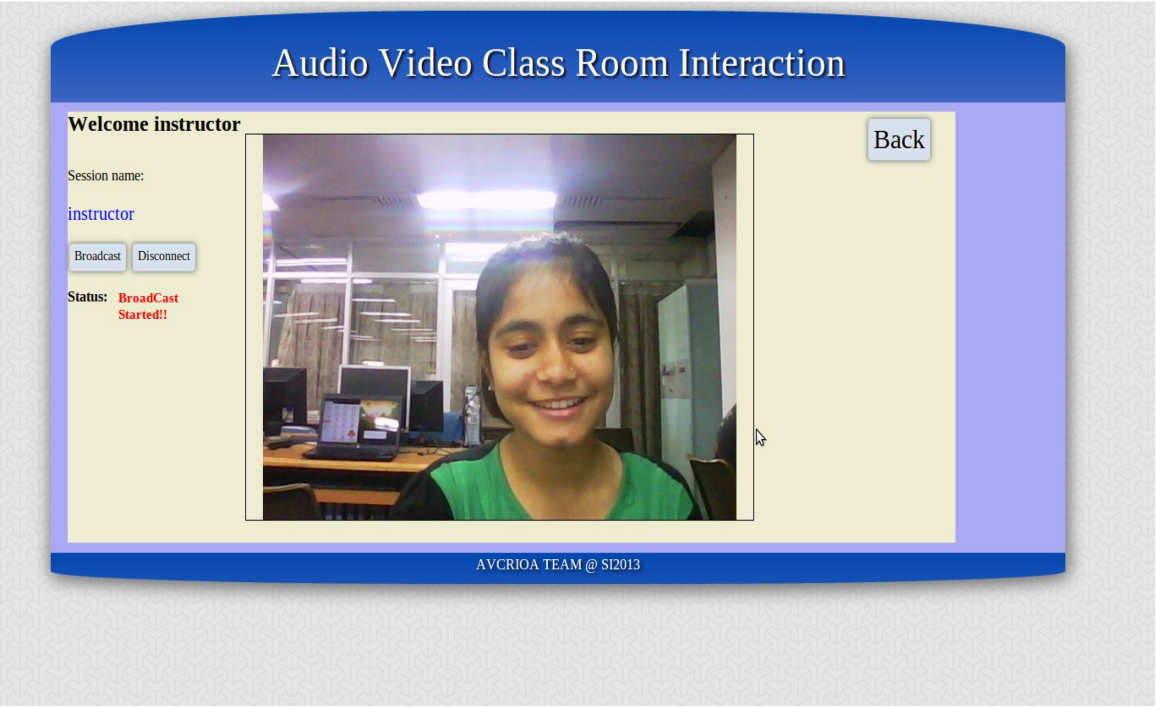
Welcome instructor

Disconnected

S.No	Student	Audio	Length	Status	Time Stamp
1	studentid	0:03	3.62 sec	Completed..!	2013-06-28 15:39:04.0
2	studentid	0:05	5.02 sec	It's Playing..!	2013-06-28 12:59:27.0

AVCRIOA TEAM @ SI2013

5. Broadcasting



Audio Video Class Room Interaction

Welcome instructor

Session name:

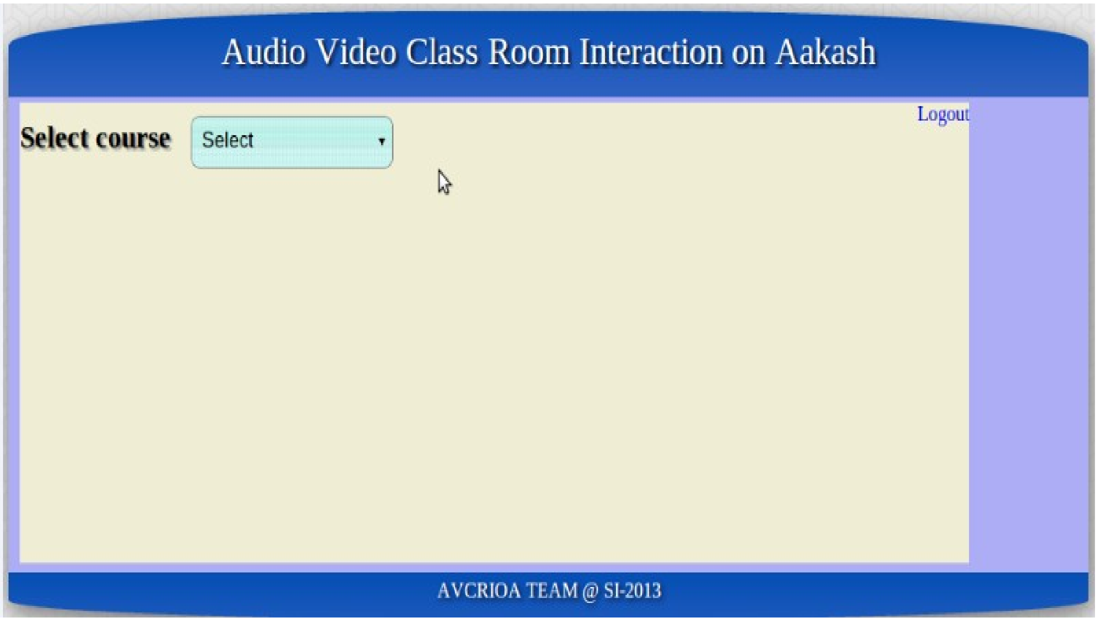
instructor

Status: BroadCast Started!!

AVCRIOA TEAM @ SI2013

2.Student module

a. Student Starting page

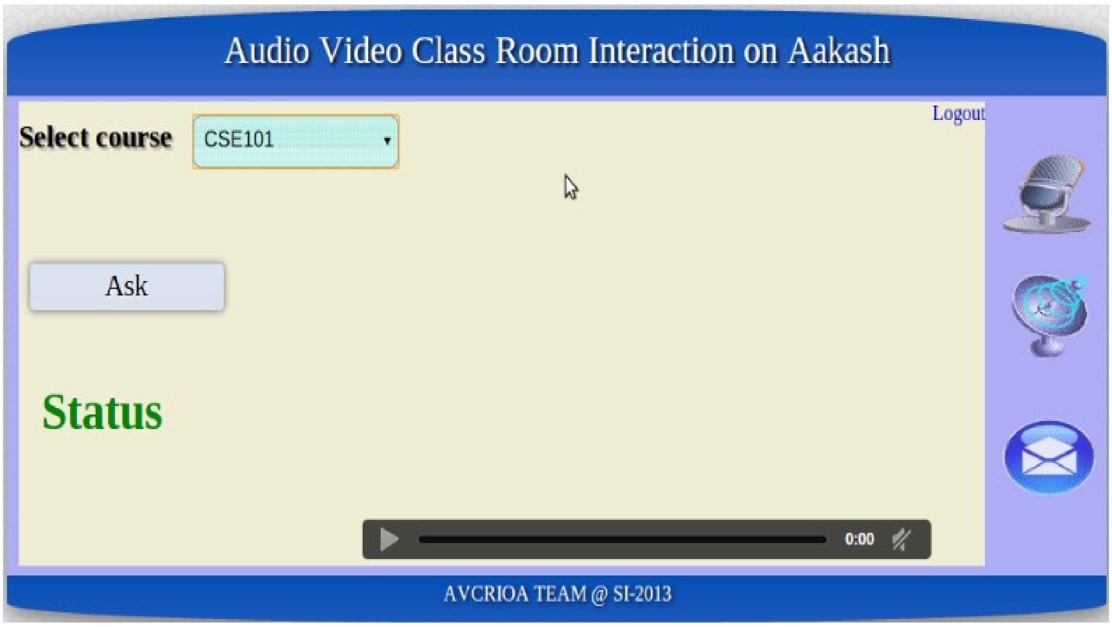


Audio Video Class Room Interaction on Aakash

Select course: Select

AVCRIOA TEAM @ SI-2013

b. Video Query Module



Audio Video Class Room Interaction on Aakash

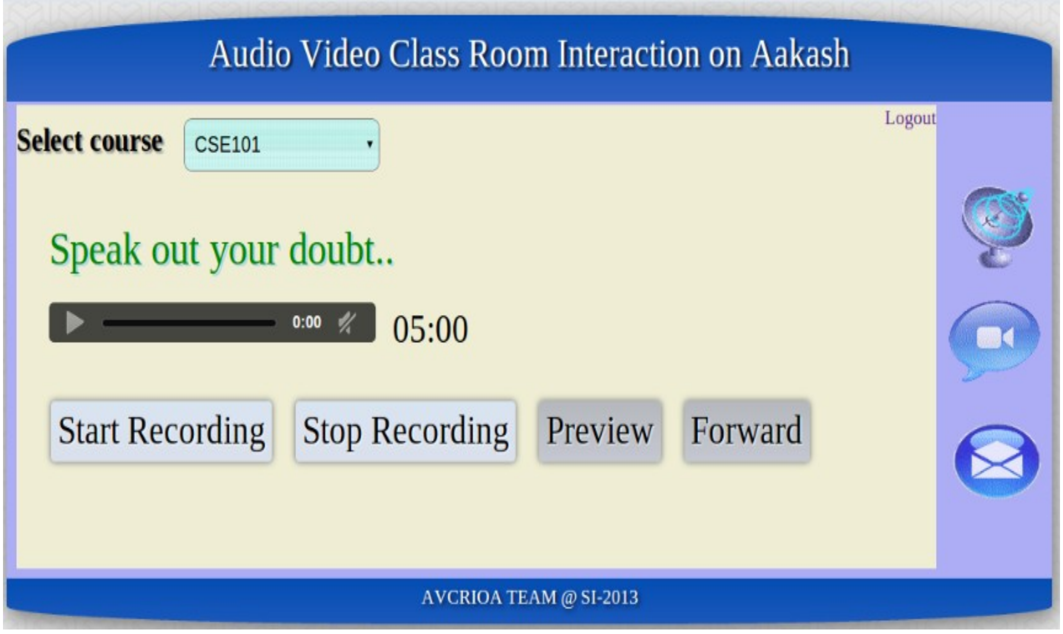
Select course CSE101

Logout

Status

AVCRIOA TEAM @ SI-2013

c. Audio Query Module



Audio Video Class Room Interaction on Aakash

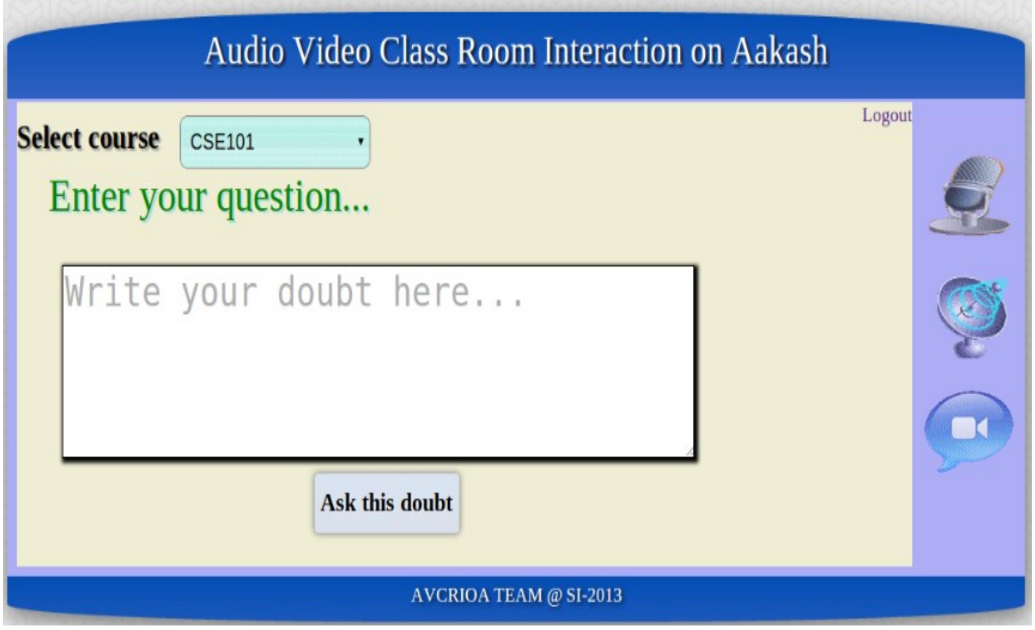
Select course CSE101

Logout

Speak out your doubt..

AVCRIOA TEAM @ SI-2013

d. Text Question Form


Audio Video Class Room Interaction on Aakash

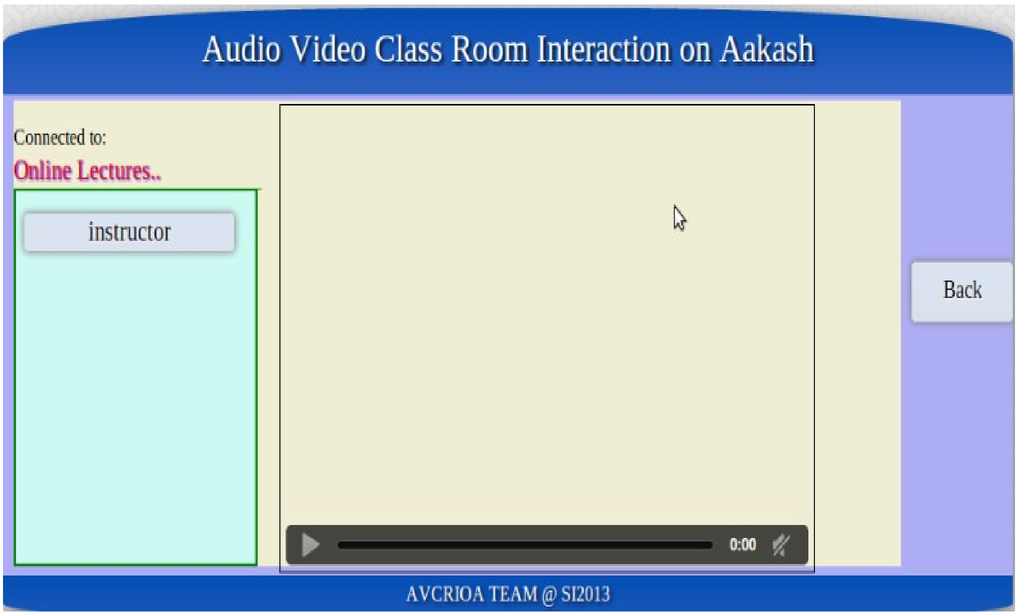
Select course CSE101

Enter your question...

Write your doubt here...

AVCRIOA TEAM @ SI-2013

e. Broadcast Lectures


Audio Video Class Room Interaction on Aakash

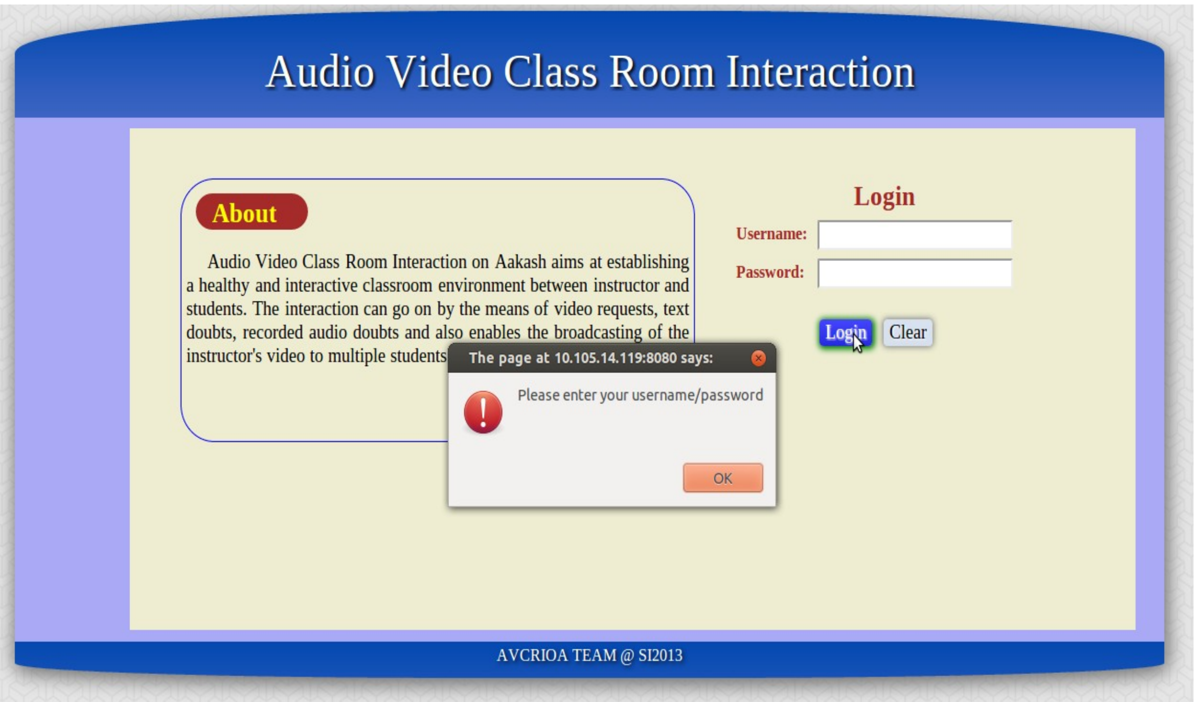
Connected to:
Online Lectures..

AVCRIOA TEAM @ SI2013

Appendix-B

Test Cases:

Test Case 1
Test Case Name: Unit Testing
Description: If the instructor clicks on login, without entering anything in the user name and password fields, he will be intimated by a pop up saying that enter user name and password correctly.
Output: A dialog box that shows “Please enter username/password?”



Audio Video Class Room Interaction

About

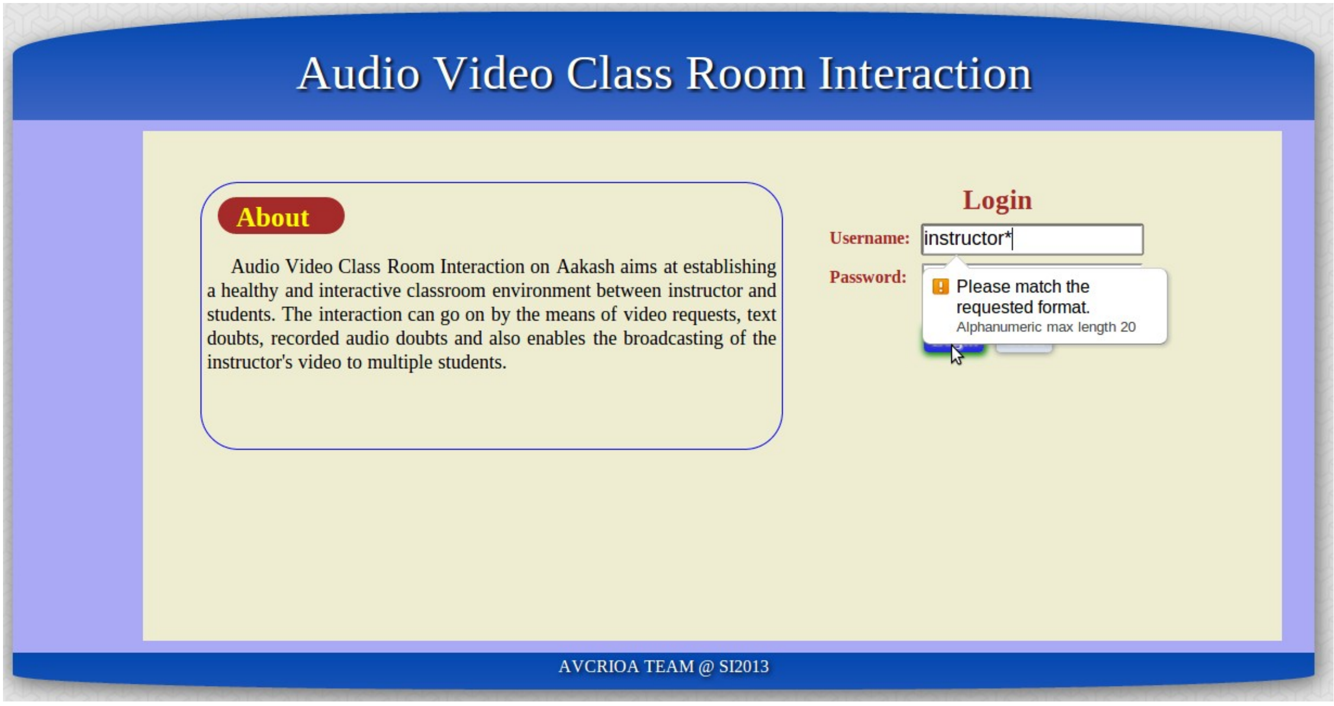
Audio Video Class Room Interaction on Aakash aims at establishing a healthy and interactive classroom environment between instructor and students. The interaction can go on by the means of video requests, text doubts, recorded audio doubts and also enables the broadcasting of the instructor's video to multiple students

The page at 10.105.14.119:8080 says:

Please enter your username/password

AVCRIOA TEAM @ SI2013

Test Case 2
Test Case Name: Unit Testing
Description: If the instructor types non alpha numeric characters in user name field or more than 20 characters, he will be intimated to type alpha numeric characters less than 20 characters by showing small pop up box
Output: A pop up box that shows “Please match the requested format.”



Audio Video Class Room Interaction

About

Audio Video Class Room Interaction on Aakash aims at establishing a healthy and interactive classroom environment between instructor and students. The interaction can go on by the means of video requests, text doubts, recorded audio doubts and also enables the broadcasting of the instructor's video to multiple students.

Login

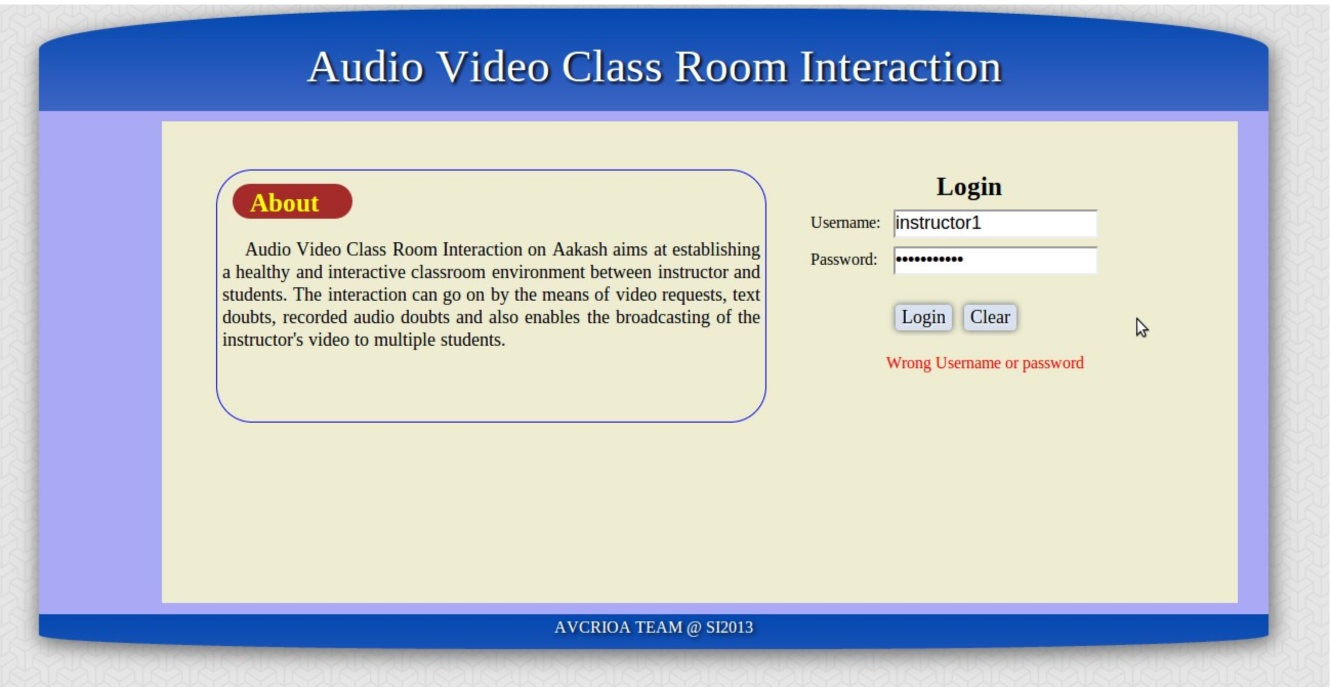
Username: instructor*

Password:

Please match the requested format.
Alphanumeric max length 20

AVCRIOA TEAM @ SI2013

Test Case 3
Test Case Name: Unit Testing
Description: If the instructor enters invalid user name or password in the login, message will be displayed saying that he entered wrong name or password..
Output: A message is shown on under the login that is “Wrong User Name or password”.



Audio Video Class Room Interaction

About

Audio Video Class Room Interaction on Aakash aims at establishing a healthy and interactive classroom environment between instructor and students. The interaction can go on by the means of video requests, text doubts, recorded audio doubts and also enables the broadcasting of the instructor's video to multiple students.

Login

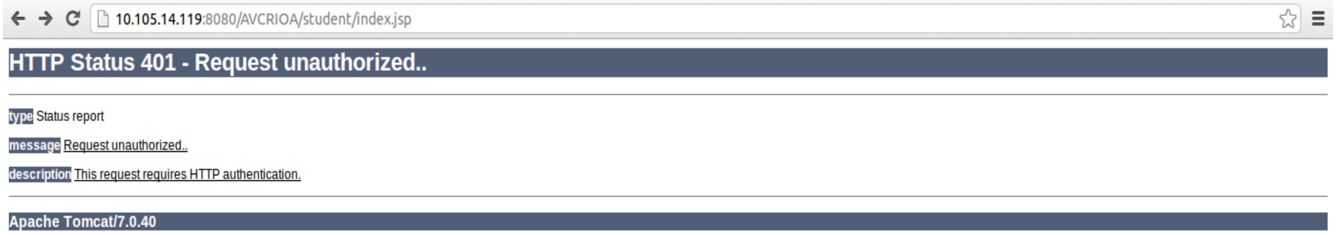
Username: instructor1

Password: •••••••••••

Wrong Username or password

AVCRIOA TEAM @ SI2013

Test Case 4
Test Case Name: Unit Testing
Description: Student should login from the aakash tablet. If he open page directly, request will not be entertained and error is shown by the server.
Output: "Request unauthorized" error will be shown on the browser.



10.105.14.119:8080/AVCRIOA/student/index.jsp

HTTP Status 401 - Request unauthorized..

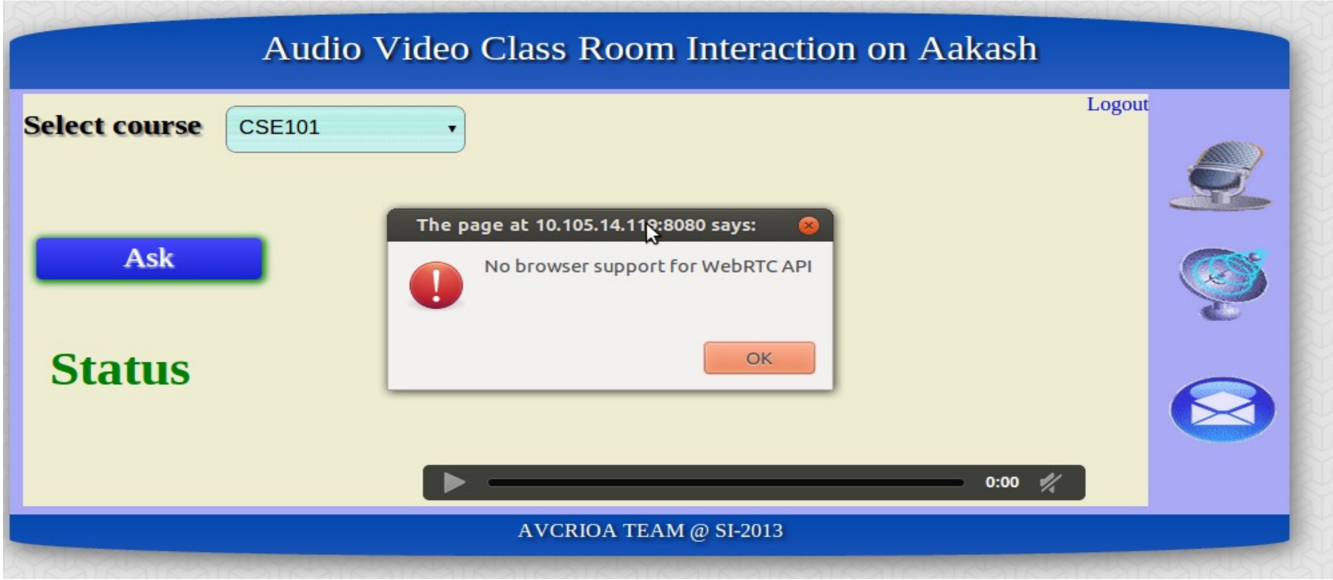
type Status report

message Request unauthorized..

description This request requires HTTP authentication.

Apache Tomcat/7.0.40

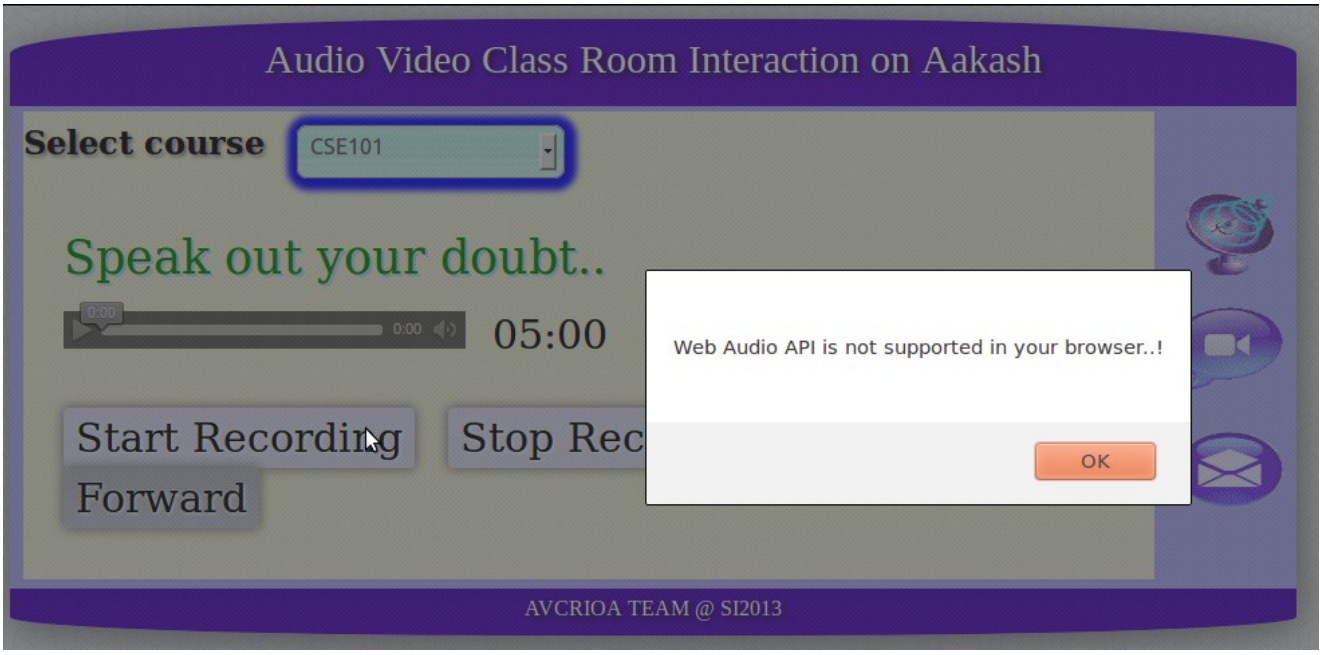
Test Case 5
Test Case Name: Unit Testing
Description: When the student clicks on ask button, if browser doesn't support webRTC, it will be notified through dialog box.
Output: Dialog box will be opened with message, “No browser support for webRTC API”



The page at 10.105.14.119:8080 says:

No browser support for WebRTC API

Test Case 6
Test Case Name: Unit Testing
Description: When the student clicks on “Start Recording” button, if browser doesn't support Web Audio API, it will be notified through a dialog box.
Output: Dialog box will be opened with message, “Web Audio API is not supported in your browser”



Web Audio API is not supported in your browser..!

Installation Manual

Student tablet should install Chrome browser 27 or above and AVCRIOA.apk . There is no need of installing plug-in.
At server side AVCRIOA needs a basic set up described below.

Linux Operating System

1. Apache Tomcat Version 7.0
2. MySQL database
3. Node.js
    a. Install node package manager (npm) with the following command in terminal

    ```
    $sudo apt-get install npm
    ```

    b. Install socket.io module using installed npm

    ```
    $npm install socket.io
    ```

After successful installation type the following commands for knowing about versions

```
$node -v
$npm -v
```

Running Node.js server

a. Navigate to the Directory where "*server.js*" is placed

b.Type thecommand in terminal : `$node server.js`

After the execution you will find the message as "*info - socket.io started*" which means nodejs server is ready to listen messages from socket.

Frequently Asked Questions:

Q) Which browsers are suitable for AVCRIOA?

A) All the latest browsers support AVCRIOA. We recommend chrome 26 and above or firefox 22 and above for convenient use of AVCRIOA.

Q) What flags have to be enabled in the browser?

A) For desktop versions, no need to enable any flag. For android chrome browser, we need to enable just one flag, 'Enable WebRTC' by going to the url 'chrome://flags'.

Q) Can students communicate with each other through AVCRIOA?

A) No, AVCRIOA doesn't allow communication between students. AVCRIOA is developed for building an interactive session between the student and the instructor, not for communication between students.

Q) Why do the student sometimes not able to see the 'Send Video Requests' option?

A) This option will be available only if the instructor wants it to. The instructor can enable/disable video requests from the student. Whenever the instructor enables video requests from students, then only a student can send a live video request. In case the instructor disables the video requests, then all the options related to the video requests on student side will be disabled.

Q) How can a student ask his doubt when the instructor is not available in the classroom?

A) A student can convey his doubt to the instructor in the form of audio or text when the instructor is not available in the classroom or whenever he disables the video requests.

Q) Can a student 'Get Back' his video request which was already sent to the instructor?

A) Yes, a student can successfully 'Get Back' his video request prior to his selection by the instructor. But once the instructor selects a particular student's video request, a student can't get it back. The student will not have any option to disconnect to the instructor once he starts speaking to the instructor. Once connected, only the instructor will have full control over disconnecting.

Q) Why is the video broadcasting limited to atmost 10 people?

A) The current WebRTC API(working draft) supports atmost 10 multipeer connections from a single browser. The work is going on to extend this facility to more than 10 peers.

Q) Why can't a student send his audio doubt from the aakash tablet?

A) For recording audio from the browser, the 'Web Audio API' is needed. Currently, android chrome browsers upto version 27 doesn't support it. It was announced by google that it will be enabled from android chrome 28 version onwards. But it works fine on desktop.

Bibliography

http://webrtc-experiment.appspot.com
http://www.github.com
http://www.w3.org/html/wg/drafts/html/master/
http://www.nodejs.org
http://www.html5rocks.com/en/tutorials/webrtc/basics/
http://www.w3schools.com/
http://www.eclipse.org/downloads/
http://www.stackoverflow.com/
http://www.jquery.com/
http://www.tutorialspoint.com/jsp/