



Virtual Classroom

Prepared by

Jafar Yasin and Hussam Salameh

Submitted to

Dr. Sofyan Samara

Virtual class room is an e-learning browser-based project aims to facilitate distance education by simulate the traditional classroom

Department of Computer Engineering

Faculty of Engineering

An-Najah National University

Tuesday, December 20, 2011

Table of Contents

Abstract:.....	1
Introduction	2
1.1 Overview	2
1.2 Problem description.....	4
1.3 Used Technologies	5
1.4 Report organization	5
Design.....	6
2.1 Use-case Diagrams	6
2.1.1 Administrator’s Use-case Diagram.....	7
2.1.2 Instructor’s Use-case diagram	8
2.1.3 Student’s Use-case diagram.....	9
2.2 Sequence Diagrams.....	10
2.2.1 Administrator’s sequence diagram.....	11
2.2.2 Instructors sequence diagrams.....	16
2.2.3 Student’s Sequence diagrams.....	28
2.2.4 Other sequence diagrams.....	35
2.3 database design	36
2.4 Conclusion.....	37
References	38

Table of Figures

Figure 1: Admin use-case diagram.....	7
Figure 2: Instructor use-case diagram	8
Figure 3: Student use-case diagram.....	9
Figure 4 : Admin login sequence diagram.....	11
Figure 5: redirect to admin page sequence diagram	11
Figure 6: initialize admin page with users.....	12
Figure 7: insert user sequence diagram.....	12
Figure 8: Update user sequence diagram	13
Figure 9: Insert user sequence diagram.....	13
Figure 10: initialize classes sequence diagram	14
Figure 11: add classroom sequence diagram	14
Figure 12: update class sequence diagram	15
Figure 13: delete class sequence diagram	15
Figure 14: Instructor login sequence diagram	16
Figure 15: redirect instructor.....	16
Figure 16: initialize instructor classes	17
Figure 17: add student sequence diagram	17
Figure 18: remove student sequence diagram	18
Figure 19: start lecture sequence diagram	18
Figure 20: start lecture timer sequence diagram	19
Figure 21: set question time sequence diagram.....	19
Figure 22: pause lecture timer sequence diagram	20
Figure 23: accept camera request sequence diagram	21
Figure 24: accept mic request sequence diagram	21
Figure 25: accept board sequence diagram.....	22
Figure 26: accept chat sequence diagram	22
Figure 27: deny student camera sequence diagram.....	23
Figure 28: deny student microphone.....	23
Figure 29: deny student from writing on board	24
Figure 30: deny student chat sequence diagram.....	24
Figure 31: open instructor camera sequence diagram	25
Figure 32: open instructor mic sequence diagram	25
Figure 33: instructor chat sequence diagram	26
Figure 34: instructor private chat sequence diagram.....	26
Figure 35: instructor writing on board sequence diagram	27
Figure 36: instructor class logout sequence diagram	27
Figure 37: student login sequence diagram.....	28
Figure 38: get student classes sequence diagram	28
Figure 39: student class join sequence diagram.....	29
Figure 40: student camera request sequence diagram	30

Figure 41: student mic request sequence diagram	30
Figure 42: student board request sequence diagram.....	31
Figure 43: student chat request sequence diagram	31
Figure 45: student writing board sequence diagram.....	32
Figure 44: student cam publish sequence diagram	32
Figure 46: student private chat sequence diagram	33
Figure 47: student class logout sequence diagram.....	33
Figure 48: student system logout sequence diagram.....	34
Figure 49: board history sequence diagram	35
Figure 50: chat history sequence diagram.....	35
Figure 51: database tables of virtual classroom	36

Abstract:

Virtual class room is an e-learning project that enables the traditional teaching process to be done using a website through internet.

The virtual class room helps students to attend their lectures from anywhere in the world, and provides more efficient technologies for instructors to communicate with the students and provide information that can be easily understood by the students.

Our project is a website based on java and flash actionscript languages, which provides a virtual class room for registered student and instructors, which can interact with each other through video and audio conferences, chatting, and whiteboard that are controlled by the instructor.

Introduction

1.1 Overview

Concept of E-learning

With the ever increasing popularity and accessibility of the internet, it is only natural that the educational community should want to make use this huge resource. Use of the internet and web are leading to significant changes in the educational models. Effective exploitation of these changes requires adequate attention to understanding the technology, the educational process and issues, student characteristics, etc

The e-learning can be defined an approach to facilitate and enhance learning by means of personal computer, and internet. It may be simple as teacher may simply post their material on the internet; student can read it online or can download it for further access.

Since student won't to be in classroom with instructor and classmates he will need to be capable of independent learning. Instructor will provide him with syllabus, course documents, and required reading. The interaction between instructor and student will happen via email, Discussion board, forums, etc. since the class doesn't meet in physical space at scheduled time, and the student will have to learn independently. He will be responsible for keeping up with assigned reading and completing assignments according to the time on syllabus.

The growing popularity of e-learning has introduced a new term to education, as **virtual classroom** where students will be present with his instructor and fellow learners in a classroom. They will not present physically in the classroom but connected to the classroom via internet. Virtual classroom aims to simulate the experience of attending class over the web, so everyone is able to see other participant virtually.

Concept of virtual classroom

Just as the term virtual means simulation of the real thing, virtual classroom is a simulated classroom via internet, which provides a convenient communication environment for distance learners just like face-to-face traditional classrooms. A virtual class room aims to providing a learning experience that is similar to a real classroom.

When we go to college, we have a schedule of lectures which we must attend, and when we enter a classroom we find teachers fellow learners, whiteboard, and LCD projector, likewise virtual class room is a scheduled online teacher-led training session were teacher and students interact with each other using computers linked to a network such as the internet.

Virtual classroom enables to bring learners from around the world together in a highly interactive virtual class where greatly reducing the travel, time, and expense of onsite learning programs. It can be used as a solution for live delivery and interaction that addresses the entire process of creating and managing our teaching-learning process. It facilitates instructor and student in teaching-learning events, such as a seminar, online discussion or a live training for employees in company.

As a traditional classroom, virtual classroom contains instructor followed by students, instructor can talk using MIC, open its camera to students, use whiteboard to explain special topics, also the instructor has full control over all the class like the traditional class room by give permission for anyone to ask, talk with microphone, open camera, write on board, and chatting.

1.2 Problem description

In our virtual classroom we try to solve the limitation of traditional class room. Here we provide some disadvantages of traditional classrooms and how our system solved them.

- Scheduling a lecture using traditional system will need a physical location with special equipments like chairs, whiteboard, whiteboard pens, and projector. These equipments have a very high cost. But in our system to schedule a class all what you have to do to tell the administrator to add a new class with a specific time.
- In traditional system students must travel, may long distance, to attend their lectures, which increasing the learning cost on students. But in our system all what you need is an internet connection, and personal computer to attend your lecture.
- In traditional system it's difficult to provide one-to-one communication between instructor and student that may decrease the understanding level for the student. But in our system you can provide one-to-one communication with any student and explain unclear ideas.
- In traditional system the understanding level for the student may decrease by many factors such as the place where the student sits in the class and how much the instructor voice is audible. But in our system each student has its own virtual class on his screen and can hear instructor without any noise
- Traditional system may suffer from the noise in the environment of class such as the transportation noise and noise from the students outside the class

1.3 Used Technologies

- Red5 server

It's an open source flash server, which provides a powerful video streaming and multi user solution to the adobe flash player and other exciting client technologies. Red5 include support for the latest multi-user API's including Net Connection, Net Stream, and shared object, while providing o powerful RTMP/Servlet implementation. In addition to support for the RTMP protocol red5 has an embedded tomcat Servlet container for JEE web applications.
- Flex

a highly productive, open source framework based on flash, for building and maintaining expressive web applications that deploy consistently on all major browsers, desktops, and operating systems. The GUI of flex is XML and the backend code is actionscript, the output of flex is SWF file that runs on flash player, which is available as plug-in in most browsers.
- MYSQL

A relational database management system (RDBMS) that runs as a server providing multi-user access to a number of databases
- Hibernate

It's a powerful high performance Object-Relational persistence and query service for any java application. Hibernate maps java classes to database tables and from java data types to SQL data types.

1.4 Report organization

The following chapter of this report will describes the design of virtual classroom project, in the first section of the following chapter we will provide the use-case diagrams of our project, the second section will describes the sequence diagrams of the project and explain the implementation idea for each process, the third section will examine the database design of the project, and the last section will contains conclusion.

Design

2.1 Use-case Diagrams

A use-case is a technique for capturing potential requirements of a new system or software change. Each use-case provides one or more scenarios that convey how the system should interact with the end user or another system to achieve a specific business goal.

Use-case description:

- Use-case: describes a sequence of actions that provides something of measurable value to an actor and is drawn as horizontal ellipse
- Actor: is a person, organization or external system that plays a role in one or more interactions with the system. Actors are drawn as stick figure
- Associations: Associations are indicated in the use-case diagrams by solid lines. An Association Exists whenever an actor is involved with an interaction described by a use-case

Virtual Class Room Use-case Diagram:

Virtual Class Room website has three major actors that interacts with the system

- Administrator: the one who responsible for the overall system management like adding new Class or User
- Instructor: the one who interacts directly with the student through Virtual Class Room, and he is responsible for teaching process through the Virtual Class Room
- Student: the one who interacts directly with the instructor, and he is responsible for the learning process through Virtual Class Room.

2.1.1 Administrator's Use-case Diagram

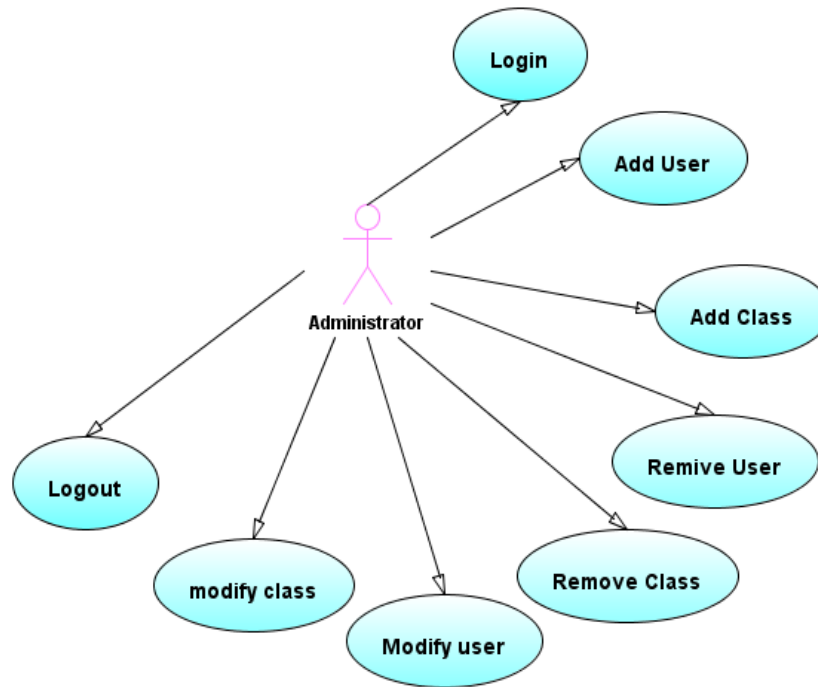


Figure 1: Admin use-case diagram

As we see in the above figure the activities of the administrator are:

- Login to the system by providing a user name and password
- Adding users of any type (Administrators, Instructors, and students) to the system
- Adding Classes to the system and specify the name of class, the lecturer for this class, days, and time of the class
- Removing Users from the system
- Modifying system users
- Removing classes from the system
- Modifying system classes
- Logout from the system

2.1.2 Instructor's Use-case diagram

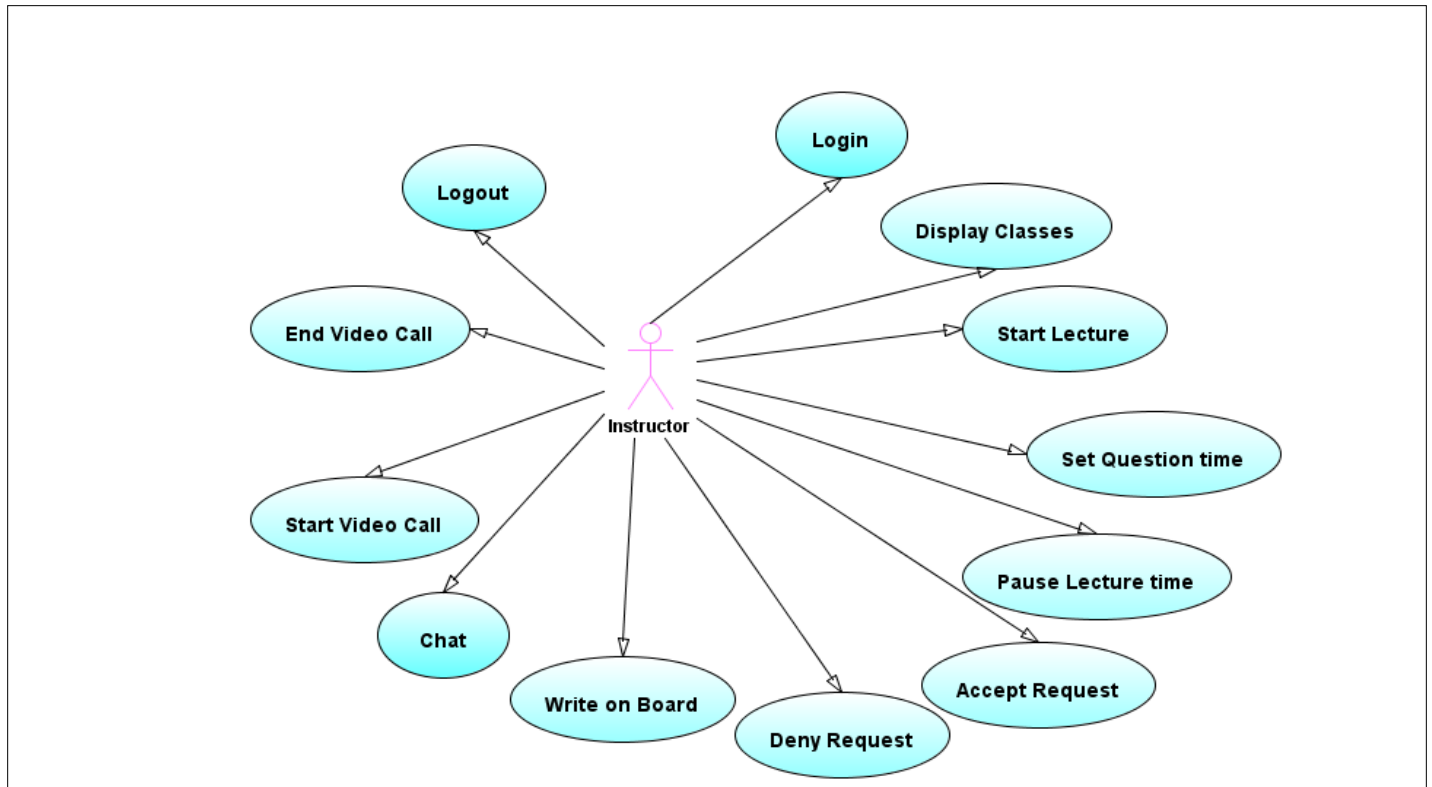


Figure 2: Instructor use-case diagram

As we see in the figure the instructor activities are:

- Login to the system by providing user name and password
- Display information about his own classes like starting time, duration, days, and registered students
- Start the lecture
- Set question time to determine when the student can ask questions, and open a discussion
- Pause lecture time if he want to give a break in the lecture
- Accepts requests from students like video conference, and writing on board
- Remove permissions from the students at any time like stop video call for a student, or prevent him from writing on board
- Writing on board
- Chatting with the students in public or private chat
- Starting a video or audio calls with students at any time he want
- Stop video or audio calls
- Logout from the system

2.1.3 Student's Use-case diagram

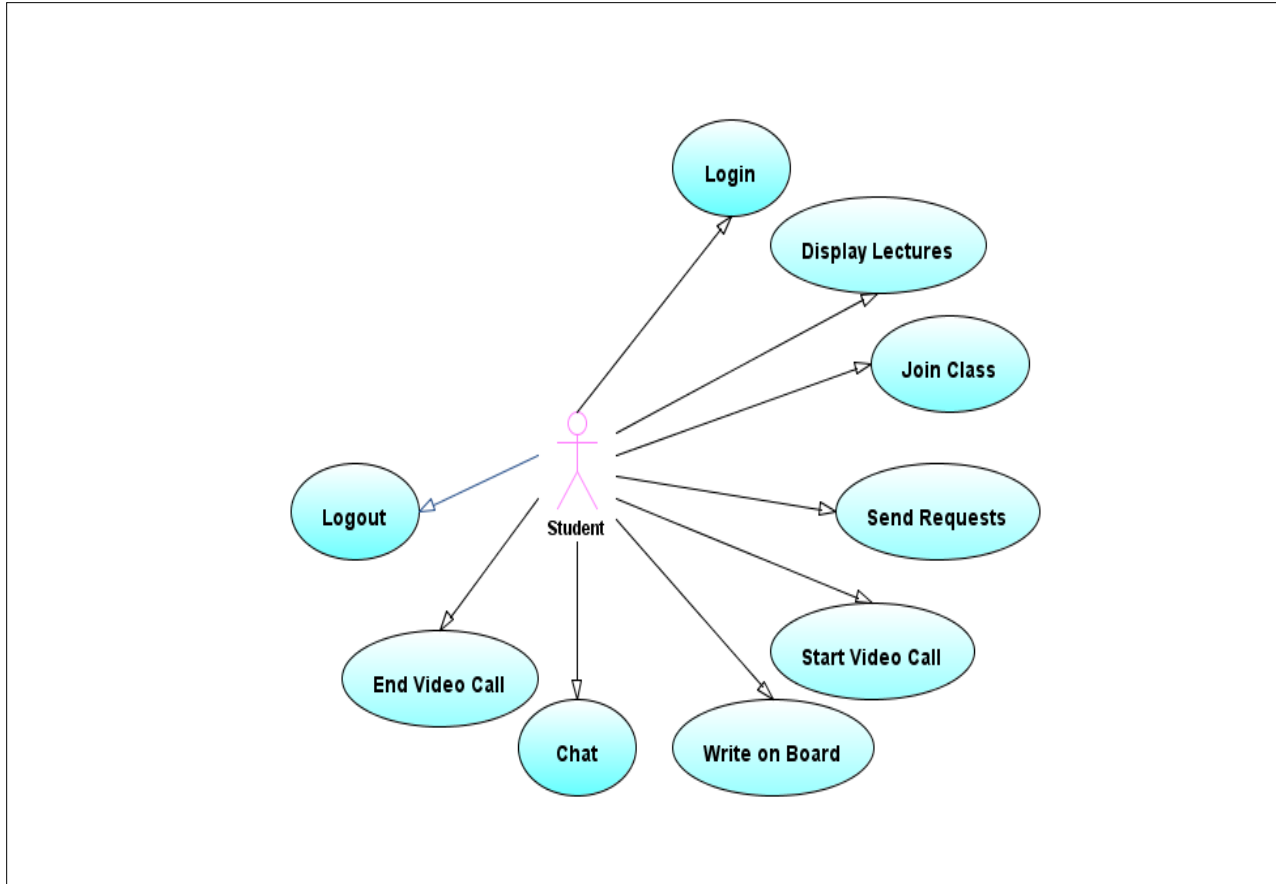


Figure 3: Student use-case diagram

As we see from the above figure the activities of the student are:

- Login to the system by providing user name and password
- Display information about his lectures and the instructor of each lecture
- Join classes that have been registered by the instructor
- Send requests like video calling, write on board, and chatting
- Start a video call when response is received from the instructor
- Write on board when permission is allowed
- Chatting on public chat or on private chat with the instructor
- End video call
- Logout from the system

2.2 Sequence Diagrams

A sequence diagram depicts the sequence of actions that occur in system, the invocation of methods in each object, and the order in which the invocation occurred is captured by the sequence diagram. This make the sequence diagram a very useful tool to easily represent the dynamic behavior of the system.

A sequence diagram is two-dimensional in nature. On the horizontal axis it shows the life of the object that it represents, while on the vertical axis it shows the sequence of the creation or invocations of these objects.

Elements of sequence diagrams:

- Object which represented by a named rectangle
- Message which shows the interactions between objects.

2.2.1 Administrator's sequence diagram

Administrator login sequence diagram:

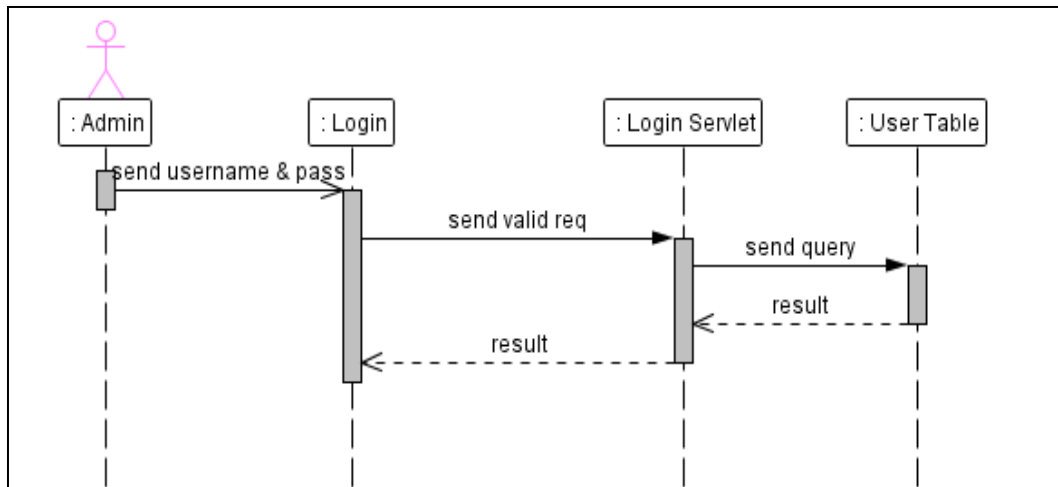


Figure 4 : Admin login sequence diagram

The figure above describes the sequence of the login process for the administrator, the administrator enter the system by providing a username and password to the login page, then login page sends HTTP request for the login Servlet to validate information, then login Servlet get users from the User table in the database and check if username and password are correct, if they are correct login Servlet opens a session for the user and sends response to the login page.



Figure 5: redirect to admin page sequence diagram

After response is received from the login Servlet then login page discusses how to redirect the page. In the case of administrator the login Servlet tells the login page that this user is admin, as a result of this the login page redirect to the instructor page

Initialize users sequence diagram:

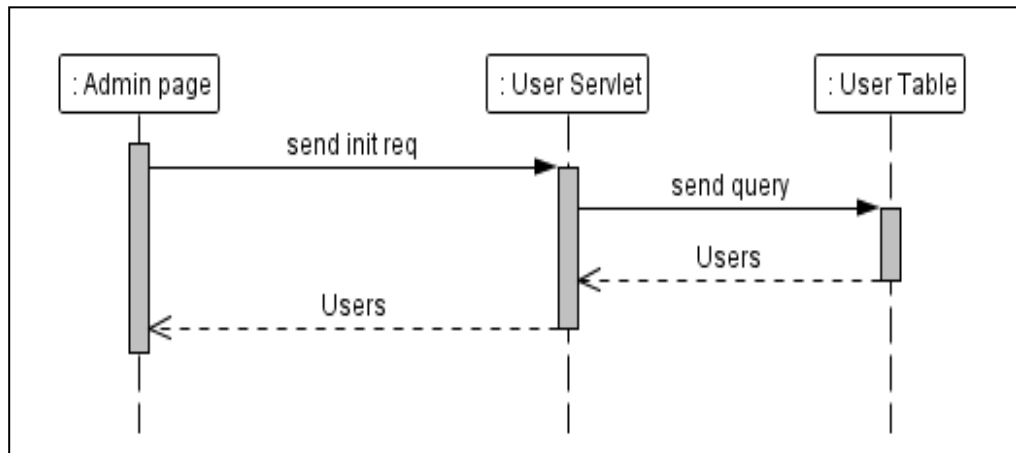


Figure 6: initialize admin page with users

This figure describes the process after the admin logged to the system, and wants to see all users registered in the system. The process starts by sending an HTTP request to the User Servlet to get all users, then the User Servlet get all users from the User table in the database and send response to the Admin Page.

Add User sequence diagram:

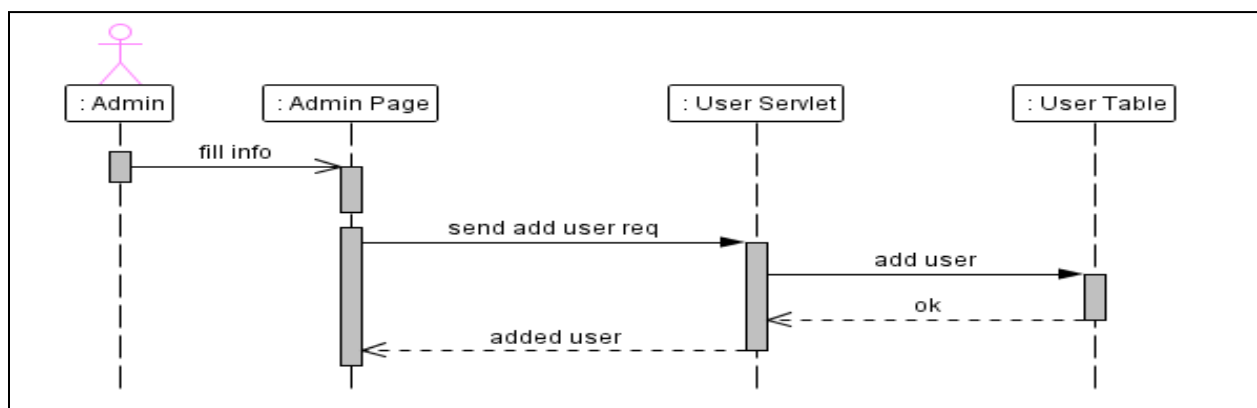


Figure 7: insert user sequence diagram

This sequence diagram shows us the process of add user performed by the Admin, the Admin fill information of student in Add User form, and click add then the Admin page sends HTTP request for the User Servlet, user Servlet get information and validate if the username is already exists, if not it adds the new user to the database, else send a response to the Admin page says that username is already exists in the system.

Update User sequence diagram:

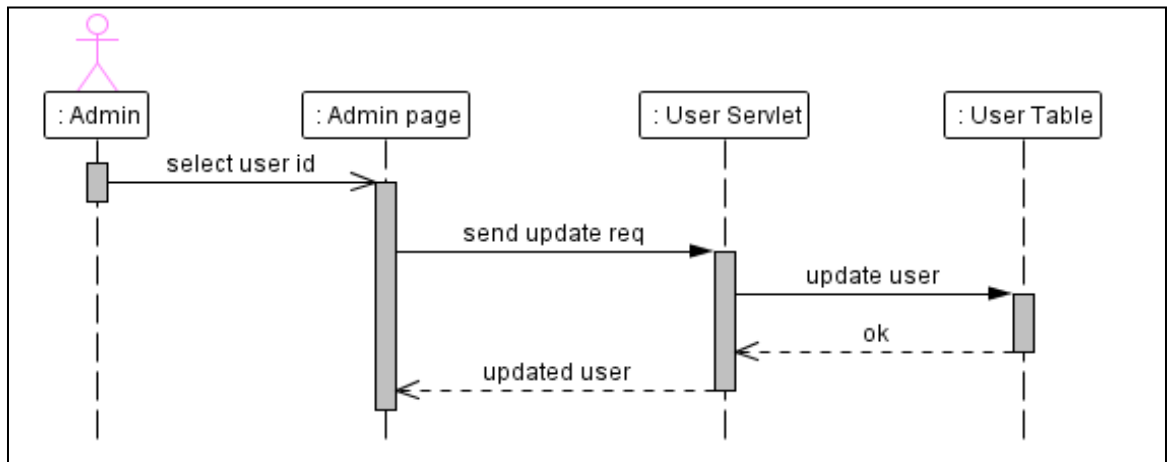


Figure 8: Update user sequence diagram

The update user process begins by selecting a user that the admin want to update its information, after the Admin press the update button on the Admin Page, it sends HTTP request with the new information of the user to the User Servlet, then the User Servlet validate the new information if the username doesn't exists in the system then the User Servlet updates user in the database, else it sends to the Admin page that this user name is already exists.

Delete User sequence diagram:

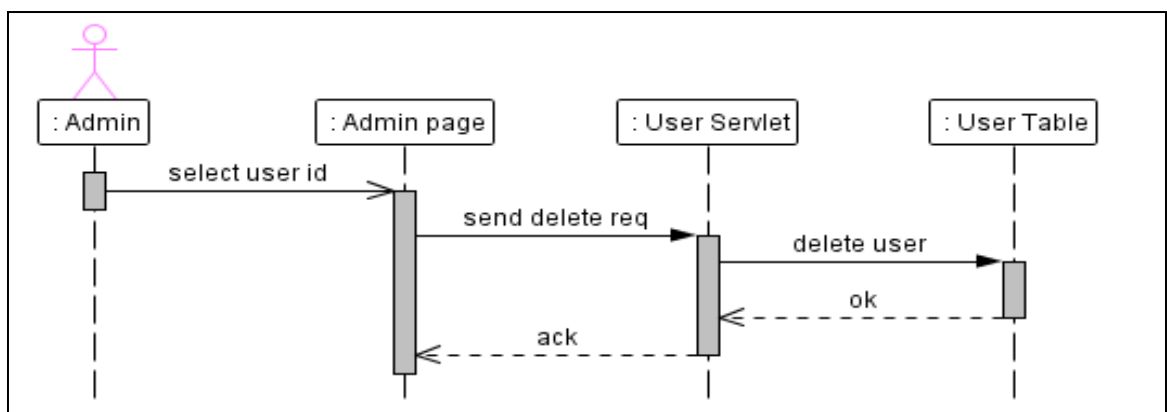


Figure 9: Insert user sequence diagram

As we see in the figure, when the Admin want to delete a user from the system, he is select the user and sends HTTP request to the User Servlet to delete this user, then the User Servlet delete the user from the database.

Initialize Classes sequence diagram:

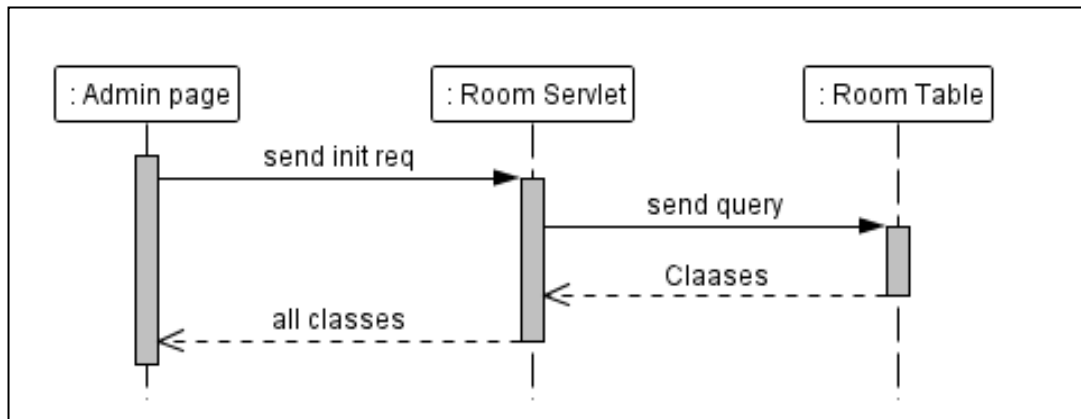


Figure 10: initialize classes sequence diagram

One of the activities of the instructor is to display all classes in the system, the initializing process of classes in the Admin page is Similar to the initializing user process, the Admin page sends HTTP request to the Room Servlet, and room Servlet get classes from the room table in the database, after that room Servlet sends response to the Admin Page it displays all classes.

Add Class sequence diagram:

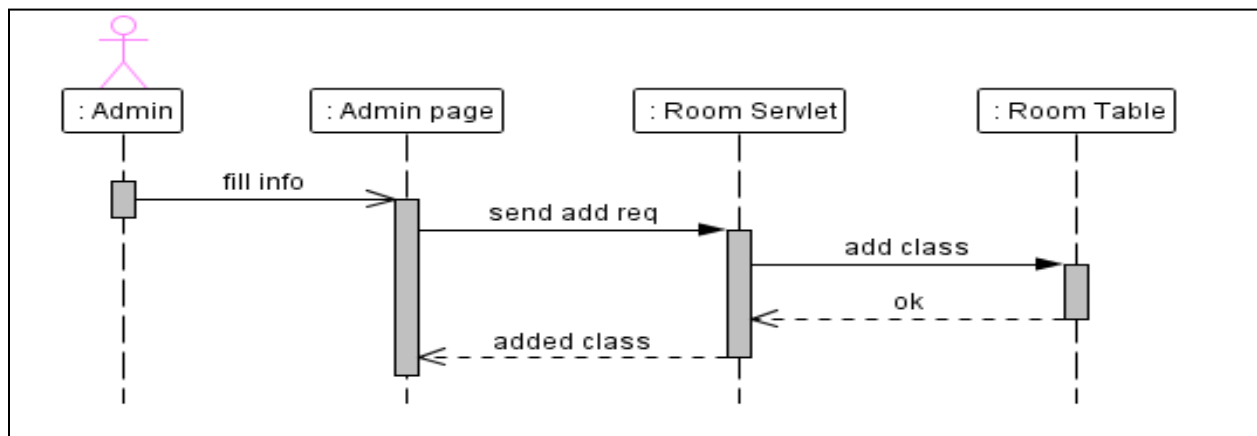


Figure 11: add classroom sequence diagram

The adding class process is similar to adding user process, that the information of class filled by the Administrator, and sends HTTP request to the Room Servlet, then the room Servlet validate information, and ensure that the time of class doesn't conflict with other classes for the same instructor, if a conflict occurs, it sends a response to the Admin page to tell that the Instructor has a lecture in this time.

Update Class sequence diagram:

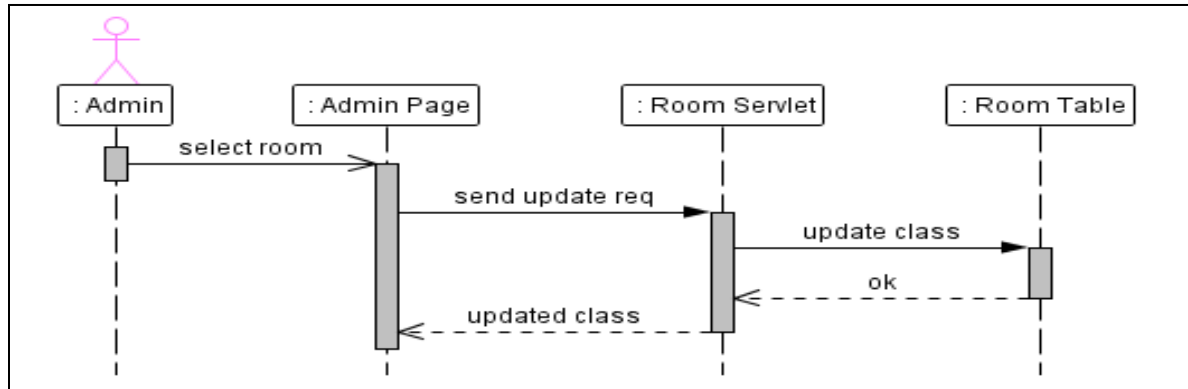


Figure 12: update class sequence diagram

In this diagram we explain the update class process, the Admin select a specific Class and fill the new information, after that the admin Page sends HTTP request to the room Servlet which get the new information, validate them and update the class in the database if it's valid.

Delete Class Sequence Diagram:

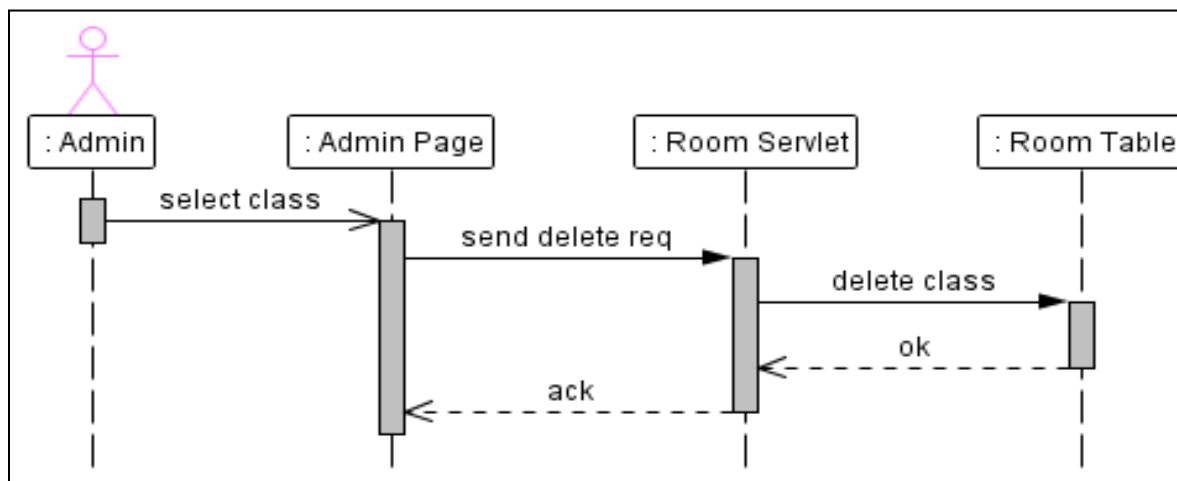


Figure 13: delete class sequence diagram

Deleting class process is similar to the deleting user process, the Admin select class and the Admin page sends HTTP request to the Room Servlet, then the Servlet delete the class from the database.

2.2.2 Instructors sequence diagrams

Instructor Login Sequence Diagram

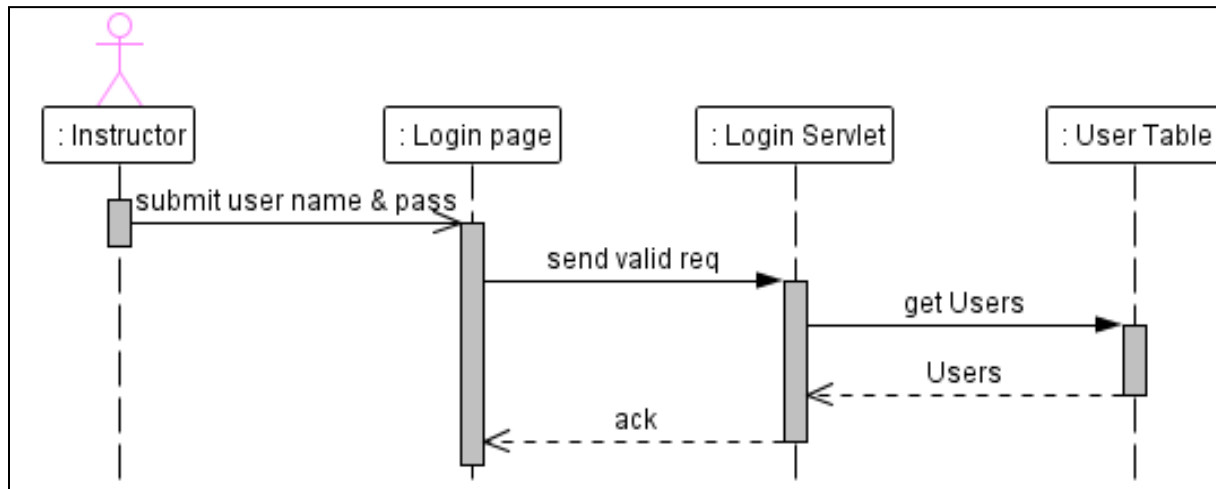


Figure 14: Instructor login sequence diagram

This figure describes how the instructor access the system, the instructor enters a username and password in the login page, then the login page send HTTP request to the Login Servlet, validate information, if the information are valid, the Login Servlet opens a session in the server, and sends the information of the user to the Login Page, if the information are invalid, the Login Servlet sends an error message to the Login Page.

After the Login Servlet sends information about the Instructor to the login page, the login page redirect Instructor to the Instructor Page

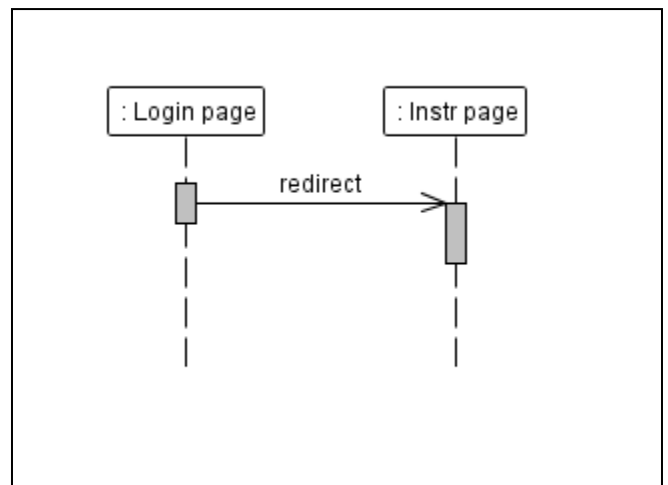


Figure 15: redirect instructor

Get Instructor Classes Sequence Diagram:

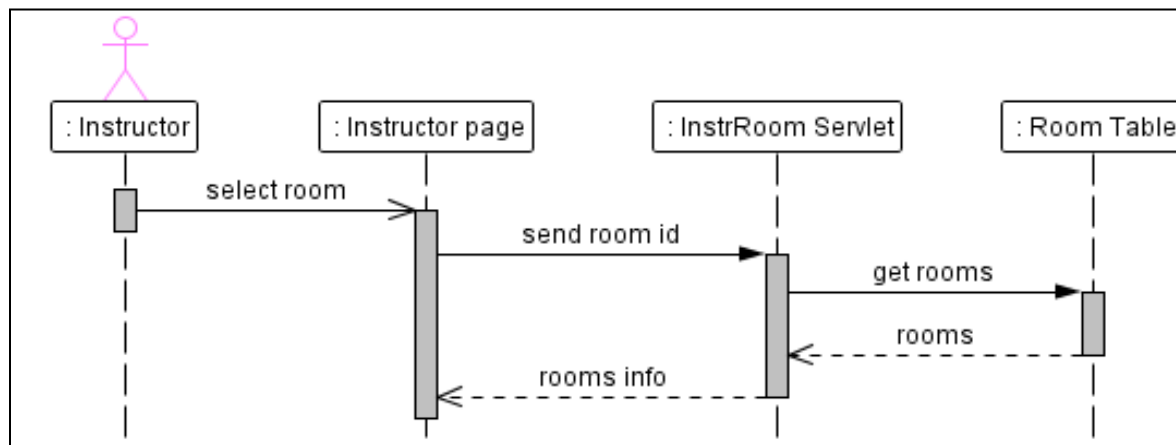


Figure 16: initialize instructor classes

After the instructor login to the system, the Instructor page displays all Classes that registered to the instructor, the Instructor Page sends HTTP request for the InstrRoom Servlet, which get all classes from the Room table in the database, after that the InstrRoom Servlet sends the rooms information to the Instructor Page.

Add Student Sequence Diagram:

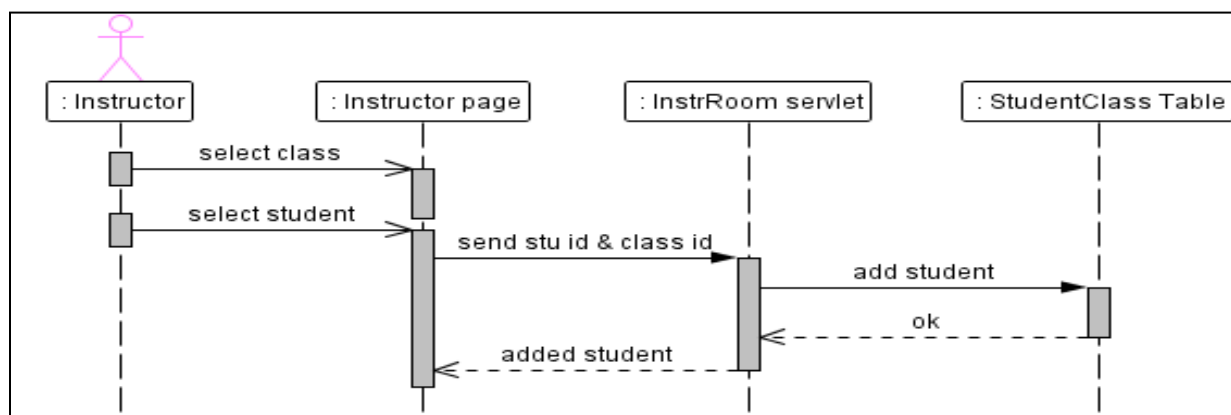


Figure 17: add student sequence diagram

One of the responsibilities of the instructor is to add students to its own classes, the instructor select a specific class and click add button, a form appear that contains all students that not registered in the class, the instructor select the student, then the instructor page sends HTTP request to the InstrRoom Servlet, the InstrRoom Servlet get parameter and validate information to avoid time conflict between student lectures, if everything is ok the InstrRoom Servlet add student to the Student Class table In the database, else send a response to the Instructor page to tell that the student has a lecture in this time.

Remove Student Sequence Diagram:

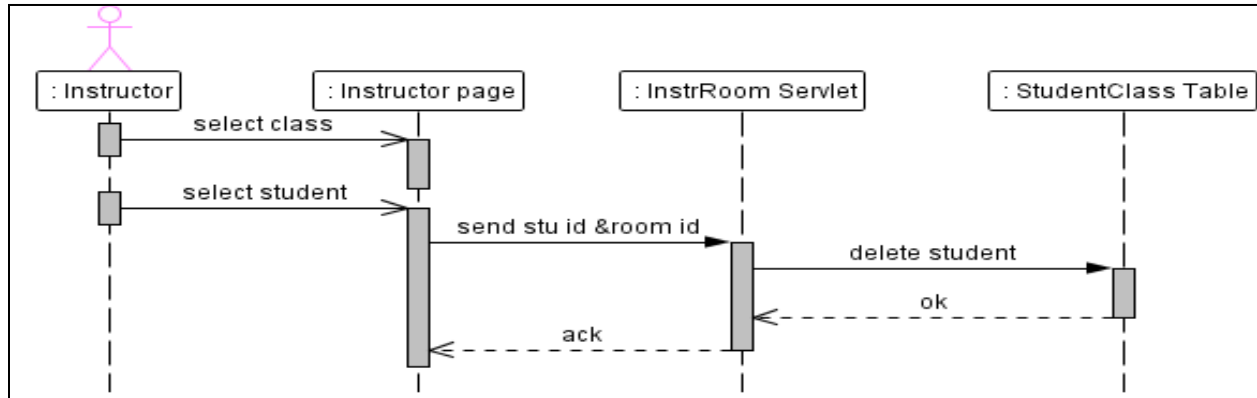


Figure 18: remove student sequence diagram

The removing student process is similar to the adding process, that the instructor select a class, then click remove, a form of all registered student is appeared, then the Student Page Sends HTTP request to the InstrRoom Servlet with the student id and class id parameter, then InstrRoom Servlet remove the student from the Student Class table

Start Lecture Sequence Diagram:

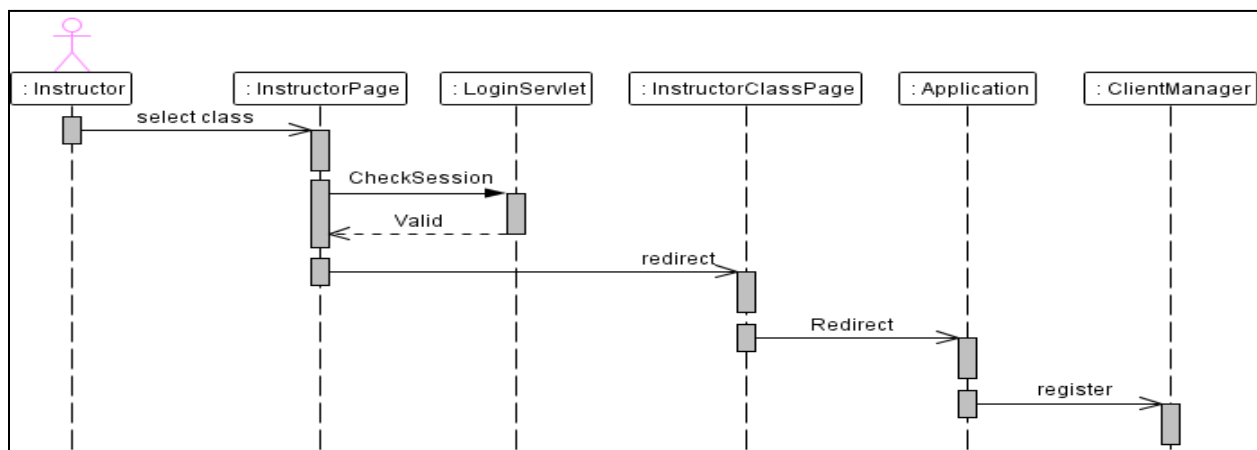


Figure 19: start lecture sequence diagram

This figure explains when the instructor needs to start the lecture, first the instructor selects a class that want to enter, then the instructor page sends HTTP request to the Login Servlet to check if session is valid, then the Login Servlet return the result to the Instructor Page, After that the instructor page redirect to the Instructor Class Page (Virtual Class Room) , and sends Instructor username and Class name with the URL variables.

After the page has been redirect to the Instructor Class page, Instructor Class Page sends RTMP connection to the Application in the server, the Application get the Instructor username and Class name, and register client in ClientManager Class. ClientManager class notifies other clients after the instructor has been registered.

Start Lecture Timer Sequence Diagram:

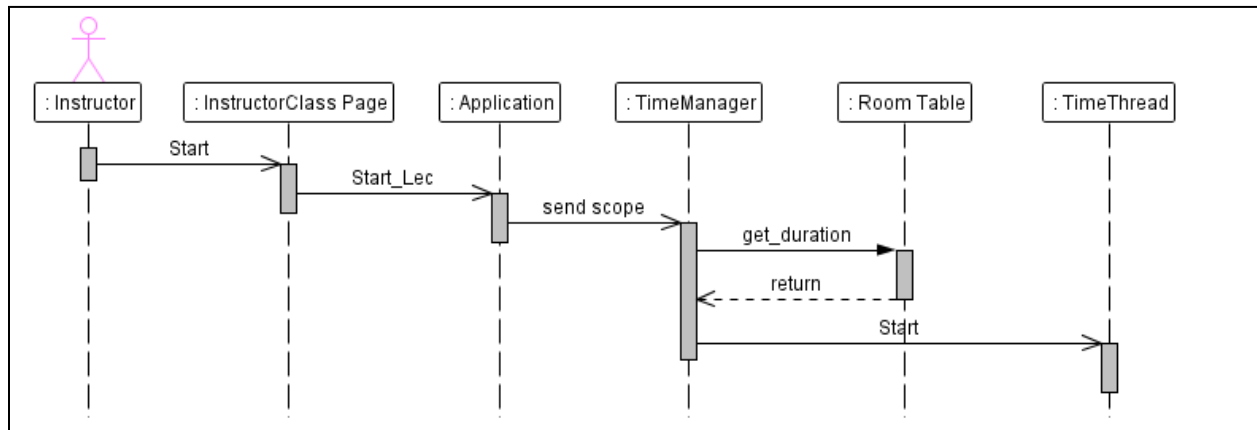


Figure 20: start lecture timer sequence diagram

This diagram shows the starting timer process for the lecture. Each Class has a specific start and end time, when the instructor clicks on the start button, the Instructor Class Page calls a method in the server Application to start the lesson time, the Application sends the scope of the class to the Time Manager class, TimeManager class is responsible for maintain a thread for each class, and it gets the duration of the Class from the Room Table in the database, after get the start and time from the table TimeManager class calculate the duration of the lecture and start the time thread. Time thread gets the time from the system and informs all clients about the global time.

Set Question Time Sequence Diagram:

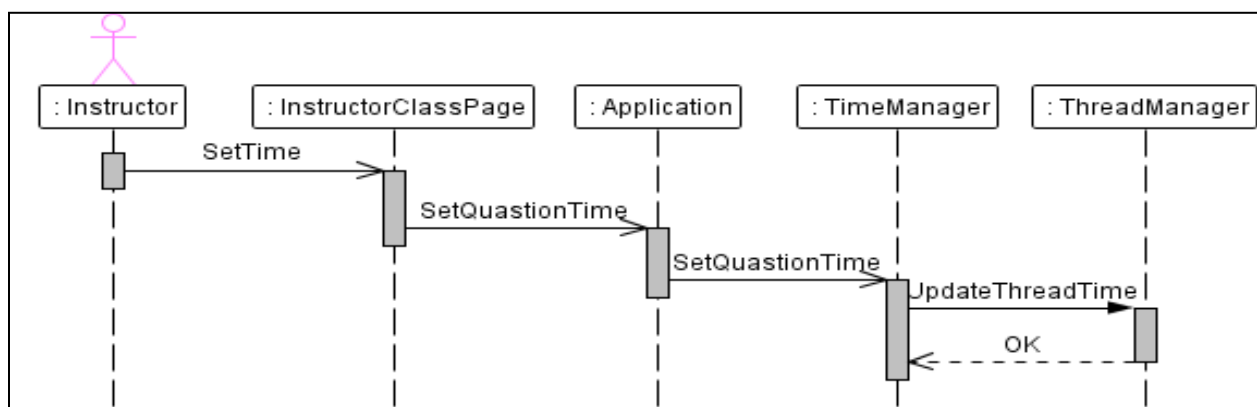


Figure 21: set question time sequence diagram

Before starting the lecture, the instructor can set the time when the students can ask questions, to the instructor, the instructor enter the time in Instructor Class Page, then the Page call a method in the Application to set question time, then the application call a method in the TimeManager class and sends the time and scope of the class, the TimeManager get the thread for that scope and update the question time. When the question time is reached the threads sends notification for student telling them that they can ask questions.

Pause Lecture Timer Sequence Diagram:

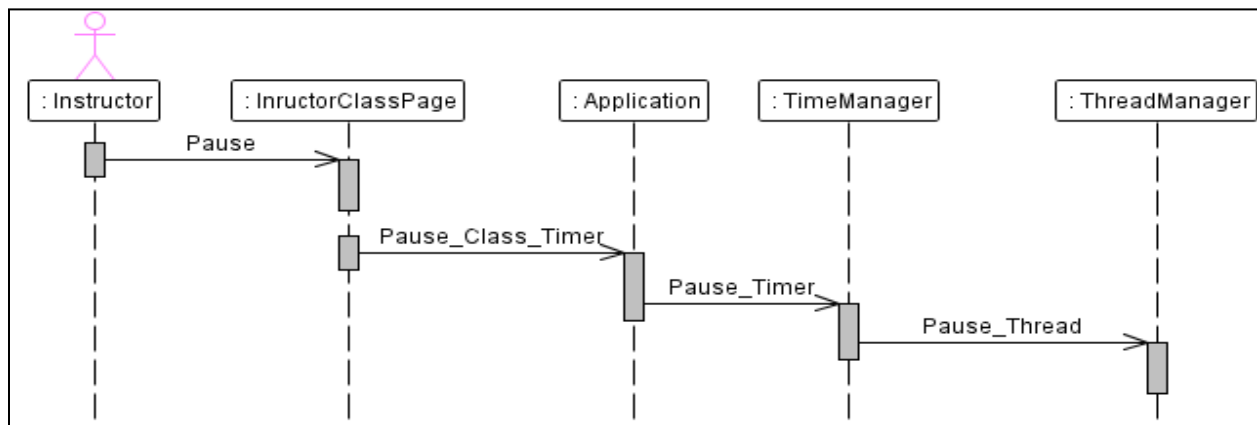


Figure 22: pause lecture timer sequence diagram

This diagram shows when the lecture needs to pause timer for any reason such as giving break to the students , when the instructor clicks on the pause button the Instructo9r Class Page call a method in the Application to pause the timer. Time Manager get the thread for this Class and suspends it.

Requests Acceptance sequence Diagrams:

When the student needs to perform an action in the lecture, he must request for the permission from the instructor, for example if the student needs to write on the board, he must send a request to the instructor, and the instructor verify it.

There are four types of requests that can be received from the students:

- Open Camera
- Talking with Microphone
- Writing on board
- Chatting

Accept Student Camera Requests Sequence Diagram:

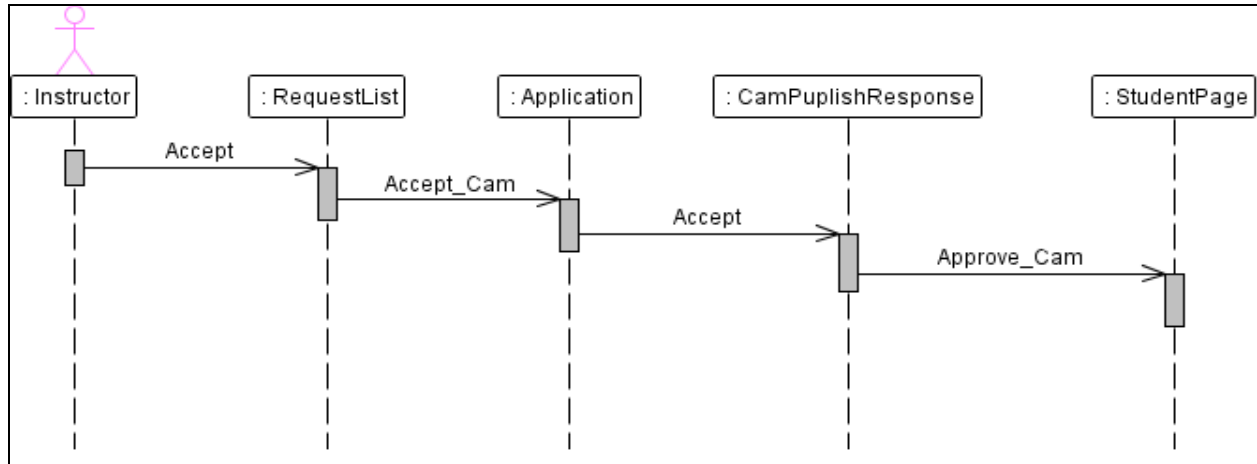


Figure 23: accept camera request sequence diagram

When a student sends request for the instructor to open his camera, the request received for the instructor in the request list, when the instructor approve the request then the Instructor Page call a method in the Application to approve cam request, then the application call a method in the CamPublishresponse class to inform the student that the instructor accepts his request.

Accept Students Microphone Requests Sequence Diagram:

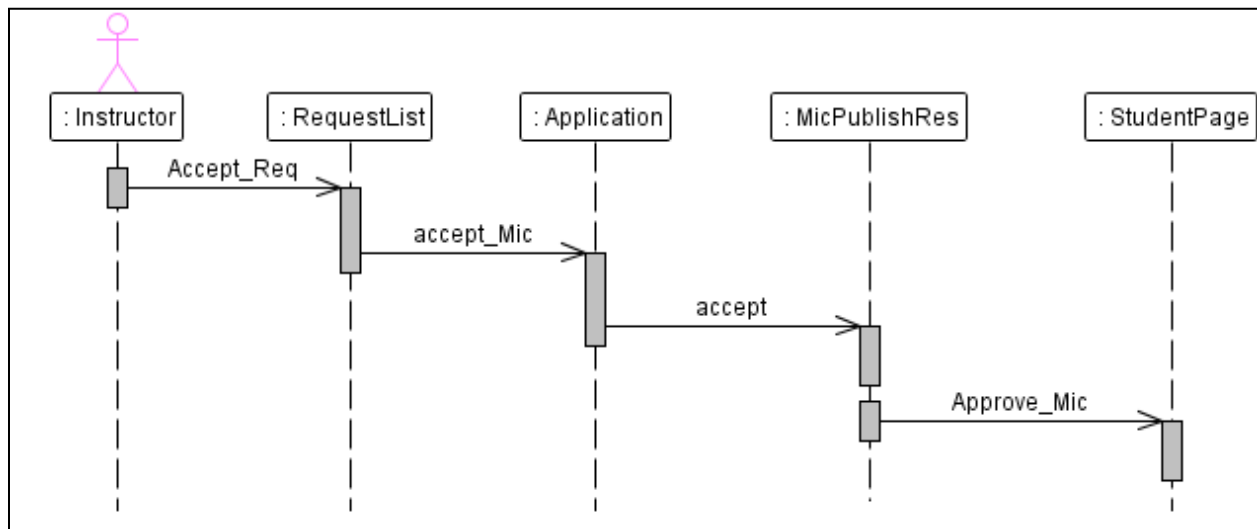


Figure 24: accept mic request sequence diagram

Accept Students Board Requests Sequence Diagram:

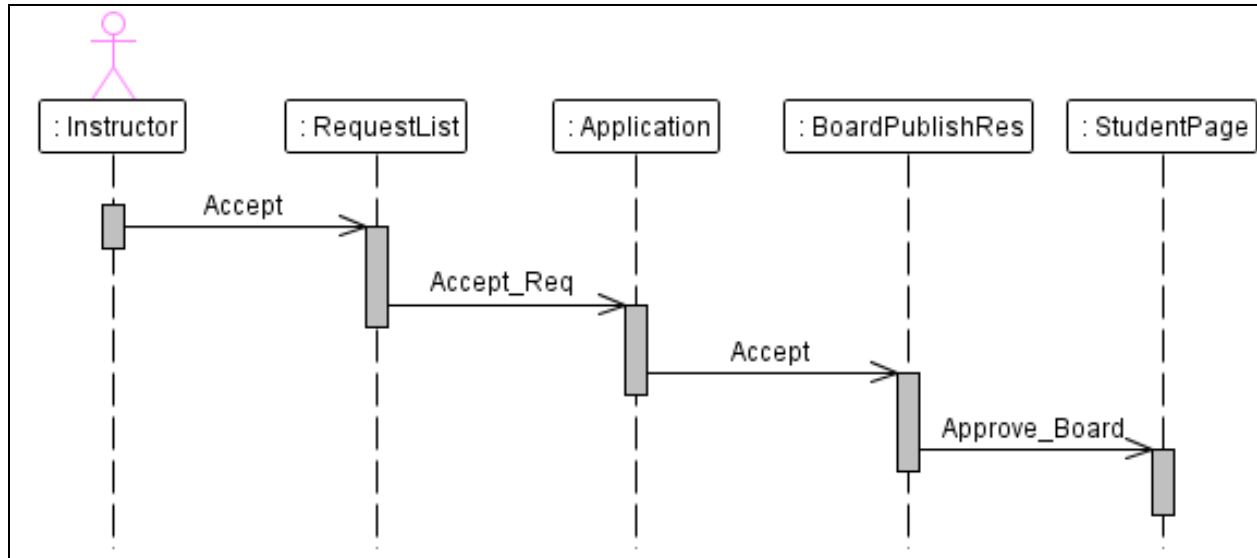


Figure 25: accept board sequence diagram

Accept Student Chat Request Sequence Diagram:

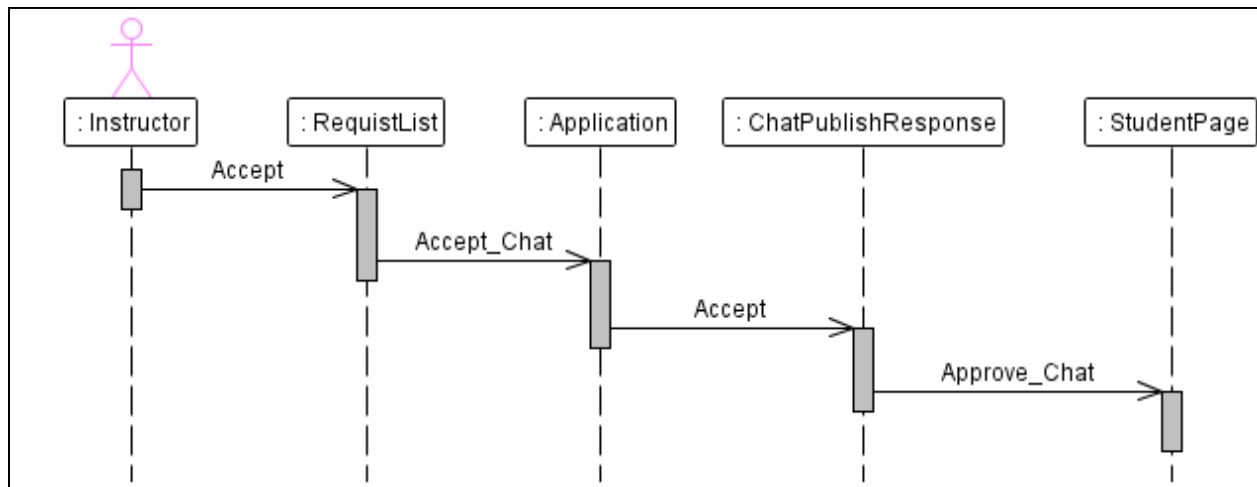


Figure 26: accept chat sequence diagram

Request Denied Sequence Diagrams:

The instructor can deny any action can be performed by the students, such as close camera and microphone, prevent them to write no board or chat.

Deny Student Camera Sequence Diagram:

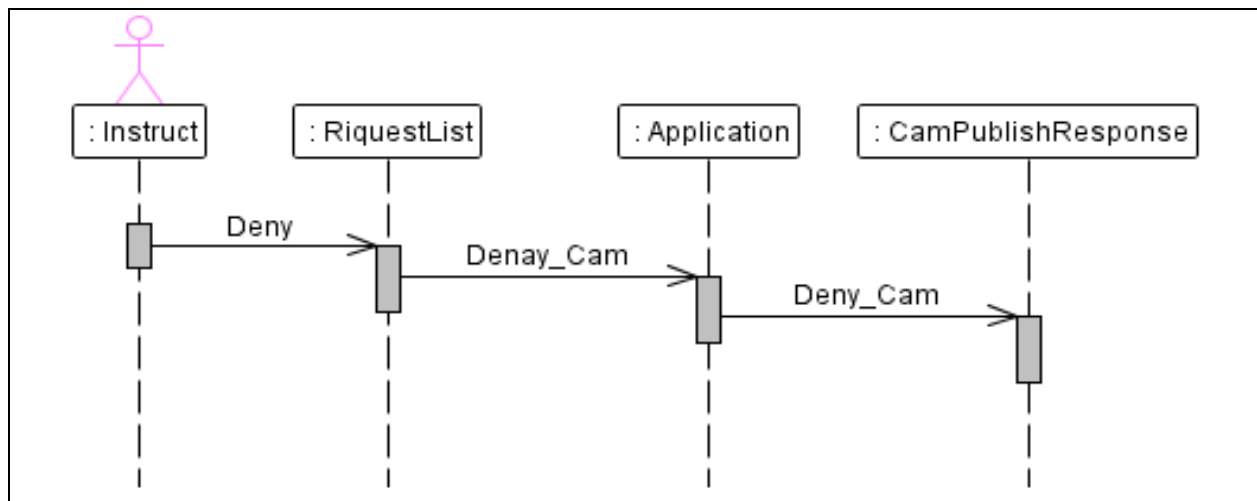


Figure 27: deny student camera sequence diagram

When the instructor close the camera that the student opens, from the request list, the Instructor Page sends a denied camera request to the Application in the server, then the Application call a method in the CamPublishResponse class to close the student camera, after that CamPublishResponse send a command for the Student Class Page to close the Camera

Deny Student Microphone Sequence Diagram:

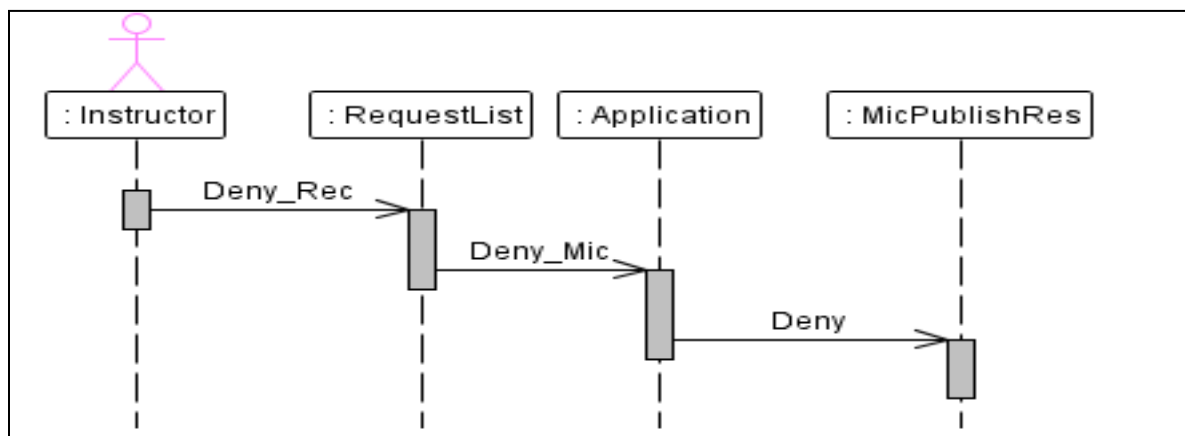


Figure 28: deny student microphone

The process of denying student microphone is similar to the denying camera process, but in the case of student the Application class in the server calls a method in the MicPublishResponse that send a command to the Student Cam Publisher class to close the microphone

Deny Board Request Sequence Diagram:

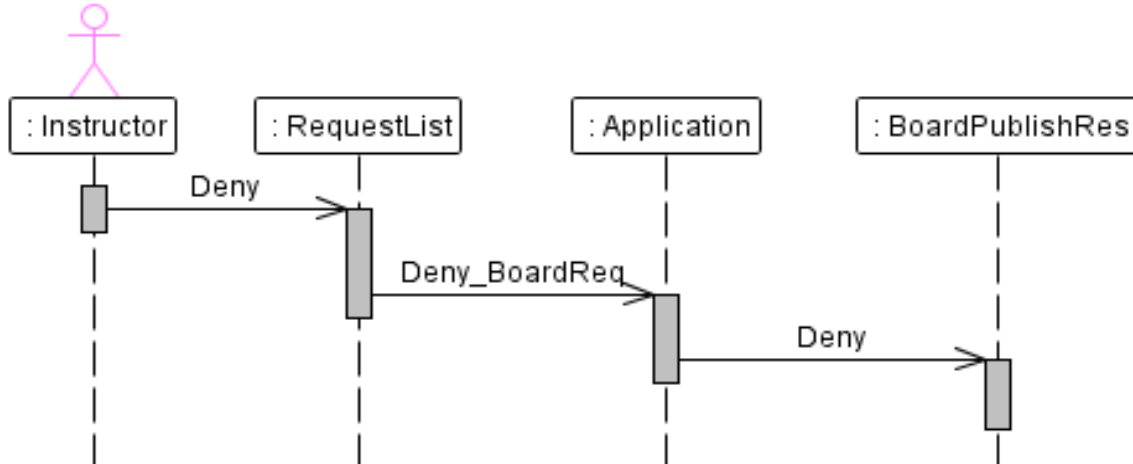


Figure 29: deny student from writing on board

As we see in the figure, the Request List in the Instructor Class Page sends a denied board request to the Application in the server, the application called a method in the BoardPublishResponse class which sends a command to the student page to disable Board control panel.

Deny Chat Request Sequence Diagrams:

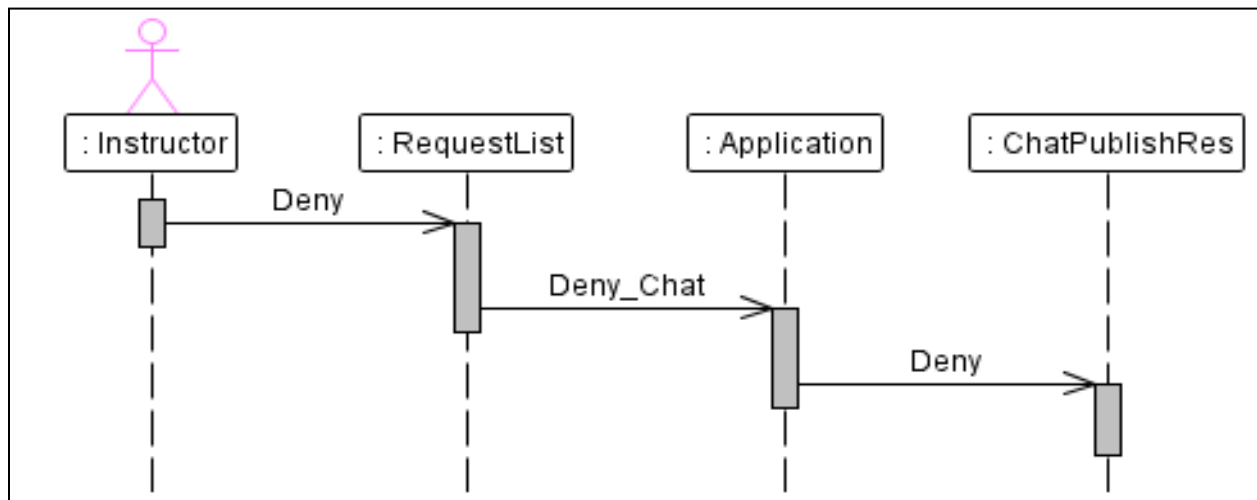


Figure 30: deny student chat sequence diagram

When the instructor prevent the student from writing on chat, the request list in the InstructorClass page sends a request to the Application class in the server, which calls a method from the ChatPublishResponse class to send a command to the student page to disable chat in his page.

Start Video Conference Sequence Diagram:

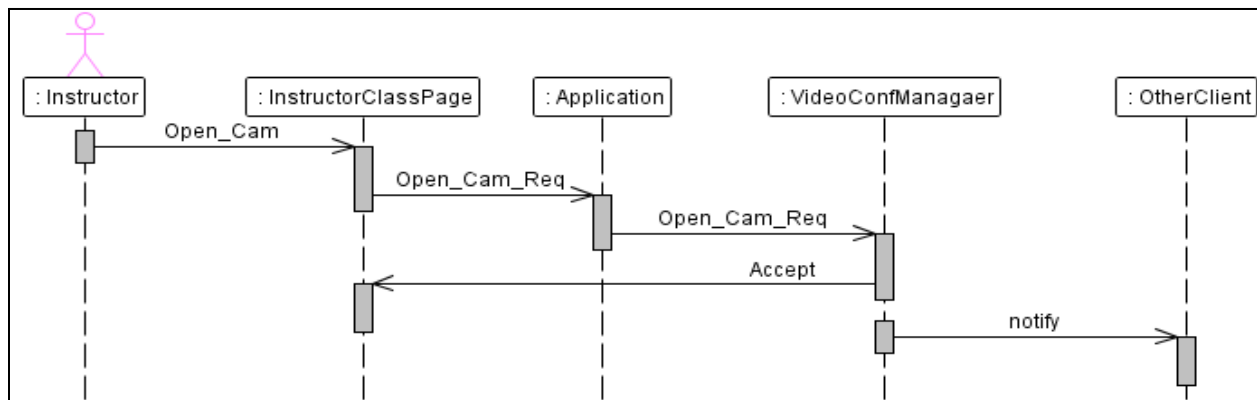


Figure 31: open instructor camera sequence diagram

When the instructor wants to open his camera, first he must select the camera device from the Instructor Class page, when the user clicks publish the Instructor Class Page sends streams to the server and the server broadcast streams to other clients, now we need to inform students that the instructor open its camera, so the Instructor Page call a method in the Application which calls another method from Video Conference Manager to register the instructor who opens camera, after that the Video Conference Manager notifies other clients that the instructor opens its camera.

Start Audio Conference Sequence Diagram:

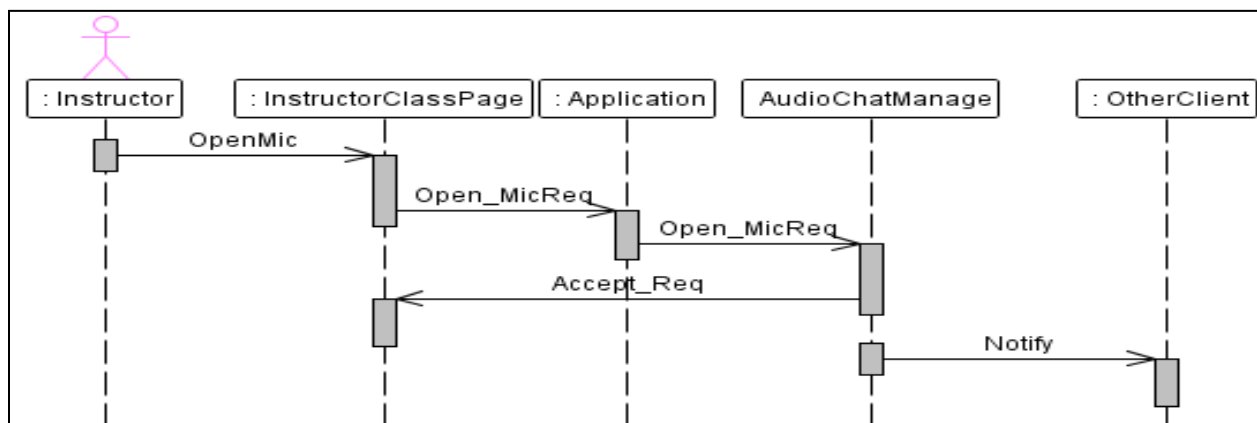


Figure 32: open instructor mic sequence diagram

The process of starting Audio conference is similar to the video conference process, since the instructor selects the microphone device from the Instructor Class Page and clicks publish, the flash player sends stream from the Instructor Class page to the server, where the server broadcast streams to other clients. The Audio Chat Manager is responsible to register the clients who open their microphones, and inform other clients to receive the streams from the server.

Public Chat Sequence Diagram:

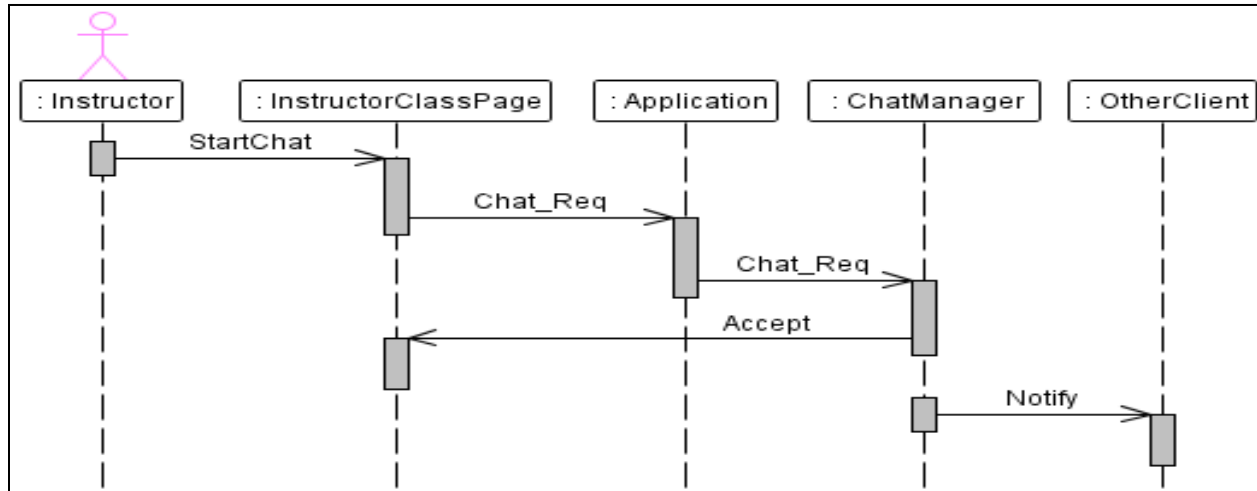


Figure 33: instructor chat sequence diagram

This figure describes the process when the instructor writes on public chat, the Instructor Class Page call a method from Application and the Application sends data to the Chat Manager which is responsible to publish data to other clients in the same class.

Private chat sequence diagram:

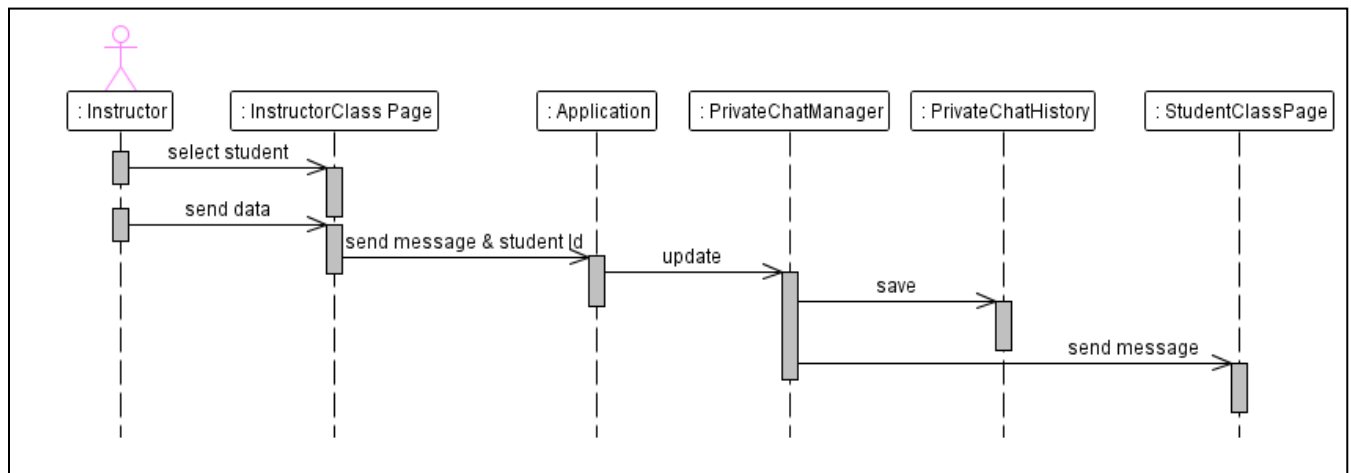


Figure 34: instructor private chat sequence diagram

In private chat the instructor can send private messages to any student in the class, the process starts when the instructor selects students and sends a message, the InstructorClass Page sends the message and the receiver Id to the Application class in the server, the Application class calls a method from the privateChatManager class to update the private chat shared object and save data in the history table in the database, then the shared object sends the message to the receiver.

Write On Board Sequence Diagram:

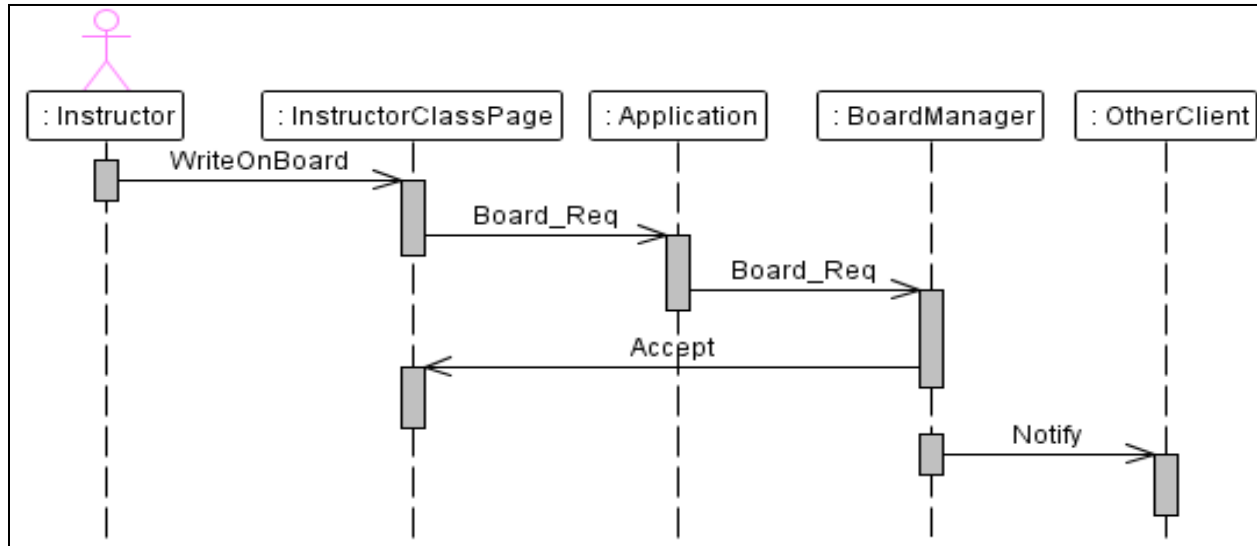


Figure 35: instructor writing on board sequence diagram

When the instructor writes on the whiteboard, the whiteboard class sends data to the application in the server, then the application update shared object in the board manager class, after updating shared object board manager broadcast data to all clients within the same class, to draw it on the board.

Logout from the Class sequence diagram:

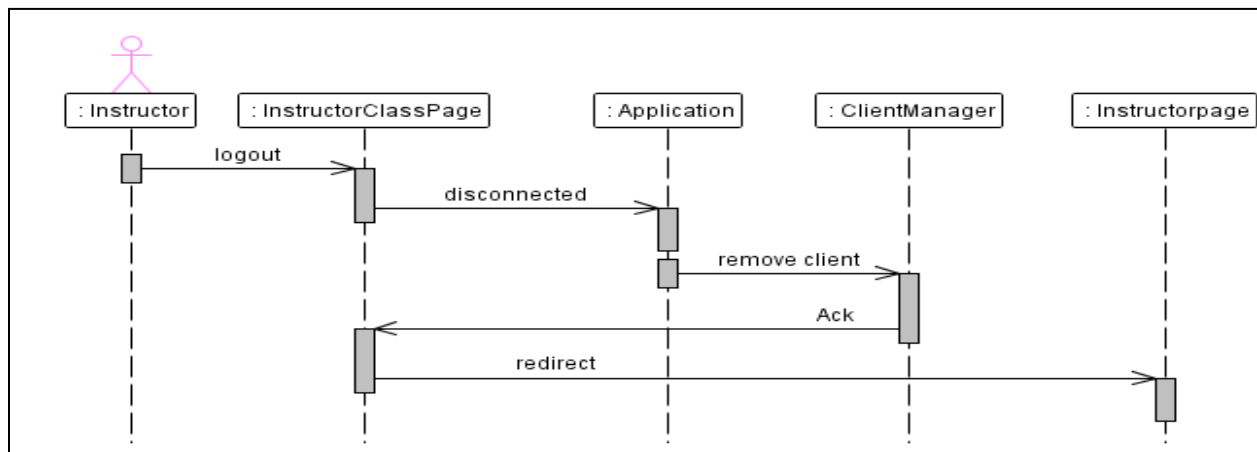


Figure 36: instructor class logout sequence diagram

When the instructor logout from the class, the Instructor Class Page disconnects from the Application in the server, then the Application sends the Instructor Id to Client Manager to remove it from the Shared Object. After that the client manager updates Shared Object to notify other clients in the class that the user logout and remove it from the user list, finally the Instructor Class Page redirects to the Instructor Page.

2.2.3 Student's Sequence diagrams

Student login sequence diagram:

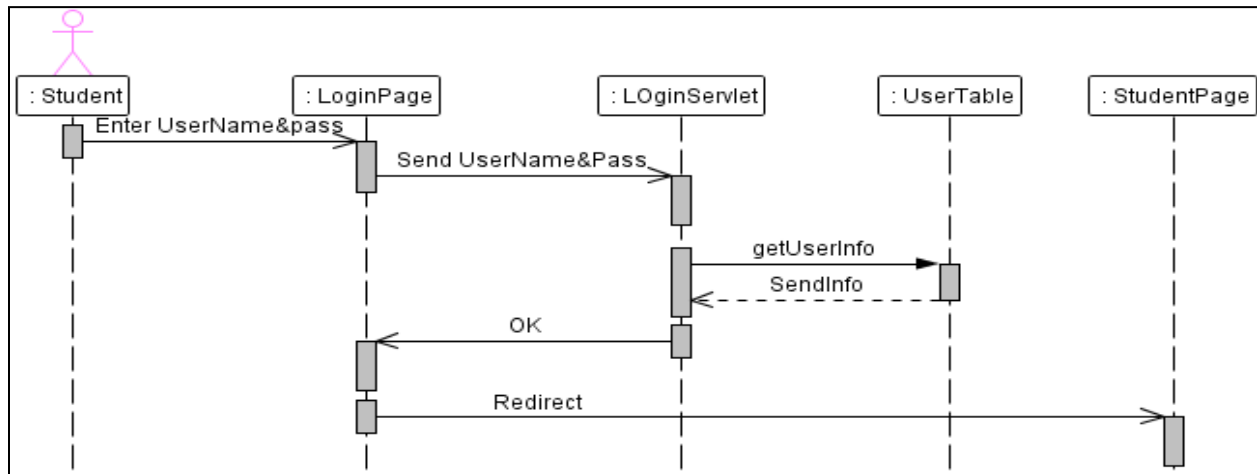


Figure 37: student login sequence diagram

This diagram shows the activity of the student when login to the system, first the student enter a username and password in the login page, then the login page sends information to the login Servlet to verify them, the login Servlet gets users from the User table in the database, if they are correct, the login Servlet opens a session and sends student information to the Login Page, then the login page redirect to the Student Page.

Initialize student page Sequence diagram:

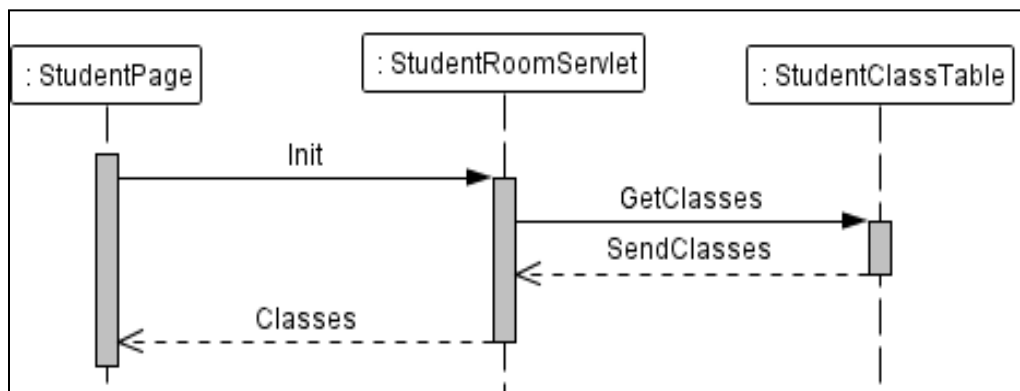


Figure 38: get student classes sequence diagram

After the login page redirect the Student to the Student Page, then the Student page sends HTTP request to the StudentRoom Servlet to get all classes that the student register, then the StudentRoom Servlet get all classes for the Student from the StudentClass Table, and return them to the Student Page to display them.

Student joining class sequence diagram:

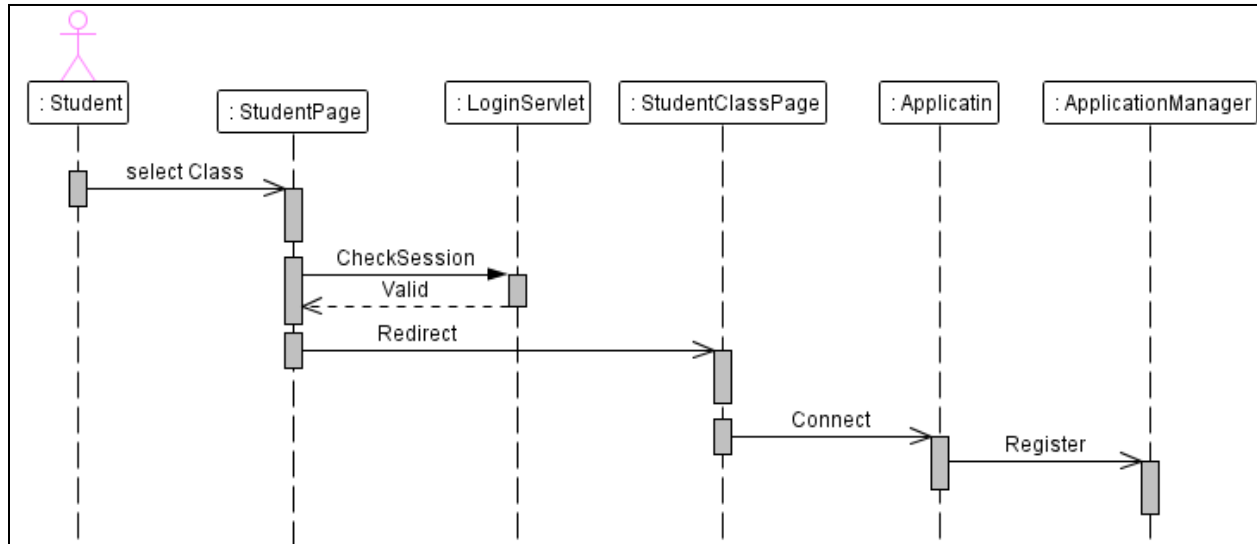


Figure 39: student class join sequence diagram

This process occurs when the student needs to join a class, first the student select a class from the Student Page and clicks enter. The student Page sends a request to the login Servlet to check if session is valid, if so the student page redirects instructor to the StudentClass Page, and sends the username and class name in the URL variables, then the StudentClass Page sends RTMP connection to the Application class in the server, the Application gets the username and password and sends them to the client manager, then the Client Manger update Shared Object which notifies other clients to update their user lists.

Student requests sequence diagrams:

The following diagram will show us how the process of the request will be done when the Student need to perform some action in the class such as opening video or audio conference, writing on board or chat.

Camera request sequence Diagram:

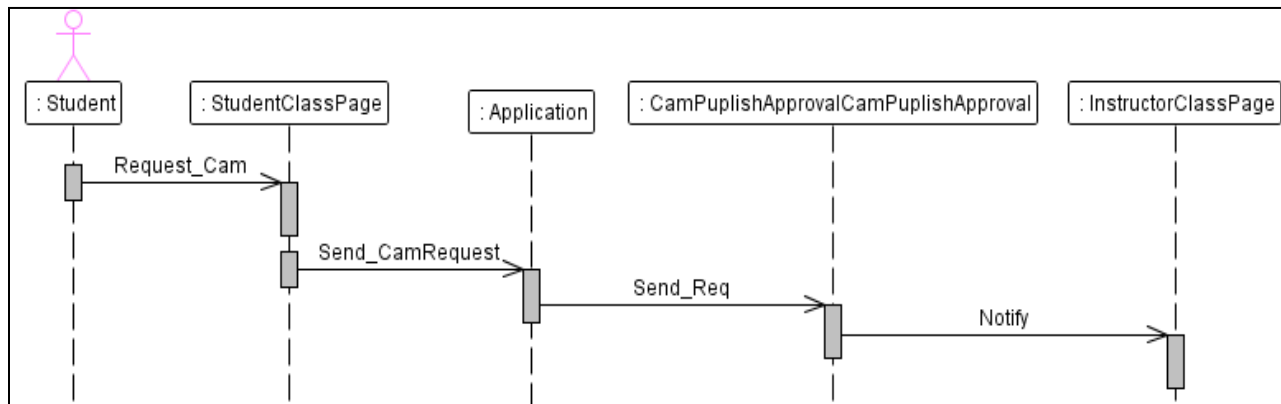


Figure 40: student camera request sequence diagram

When the student needs to open camera, he must select the camera device from the StudentClass Page, and when he clicks publish the StudentClass Page sends a request with the student Id to the Application class in the server, then the application call a method in the CamPublishApproval class to update shared object with the new camera request, then the shared object notify the Instructor page which adds a new request in the request list.

Microphone request sequence diagram:

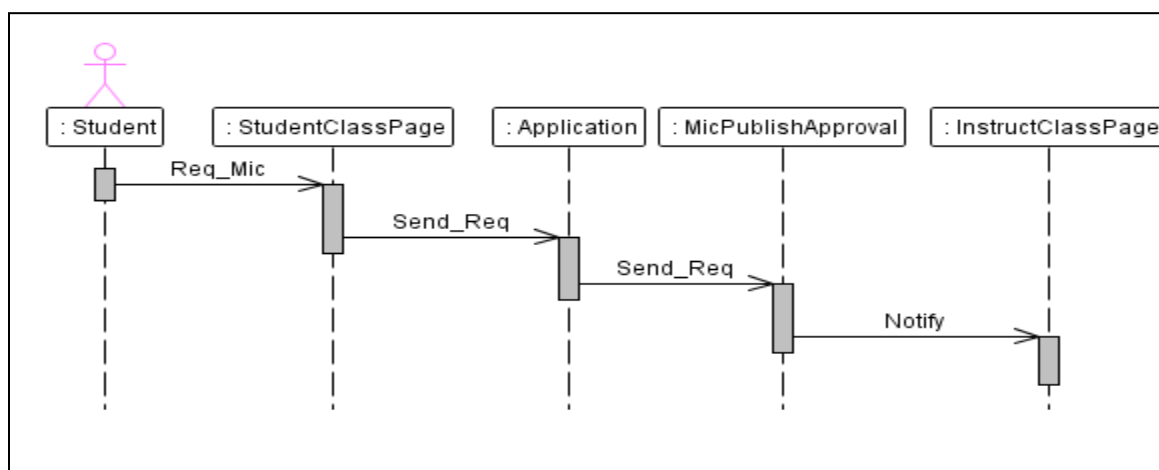


Figure 41: student mic request sequence diagram

Audio conference request is the same as Video Request, since Application class calls a method in the MicPublishApproval class and update shared object to notify InstructorClass Page and add new request for the client.

Board request sequence diagram:

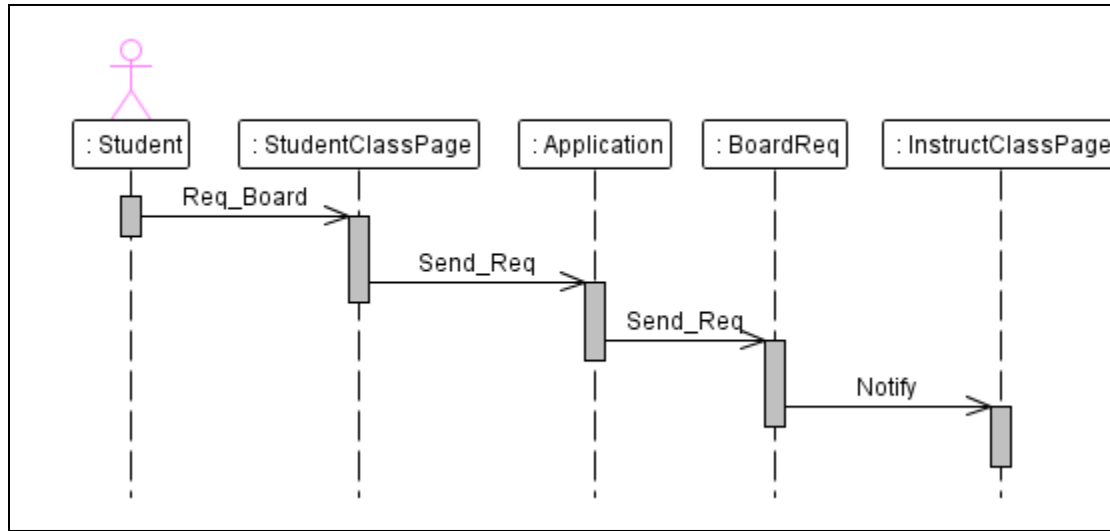


Figure 42: student board request sequence diagram

In the board request the when the student asks to write on the board, the StudentClass Page sends request to the Application class in the server , and the Application class calls a method in The BoardRequest class to update shared object with the new request and notify InstructorClass Page to add request to the request list.

Chat Request Sequence Diagram:

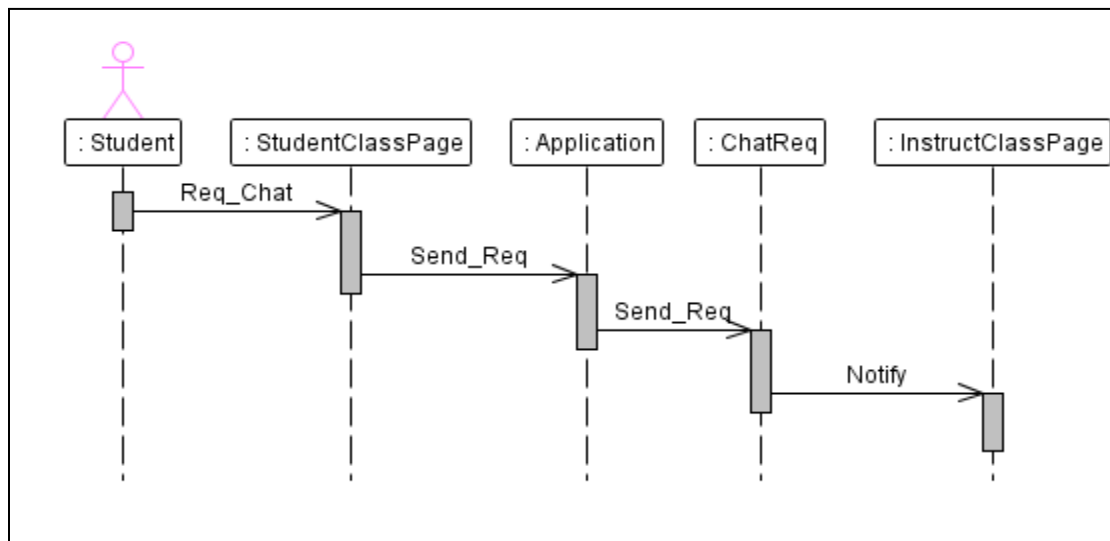


Figure 43: student chat request sequence diagram

Chat request is also similar to other requests since request received in the server from the StudentClass Page, and the application class update chat request shared object to notify instructor for the request.

Audio and video conference sequence diagrams:

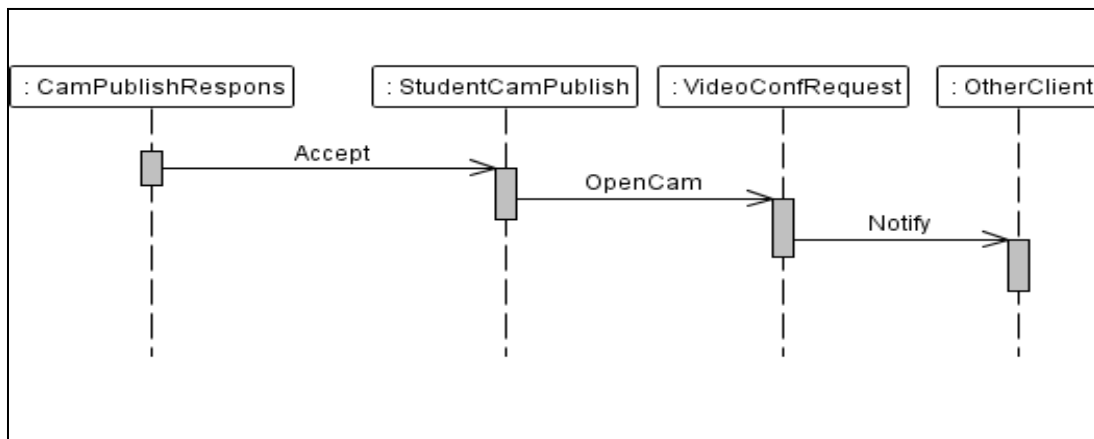


Figure 44: student cam publish sequence diagram

After the student send the request to open its camera, it has been verified by the instructor, the CamPublishResponse Class update camera response shared object to notify the Student that the request has been verified, start publish your streams, and inform other clients in the same class that the student will open its camera and everyone can see him. The audio conference process has the same scenario of the video conference process as we see in the above figures.

Write on board sequence diagram:

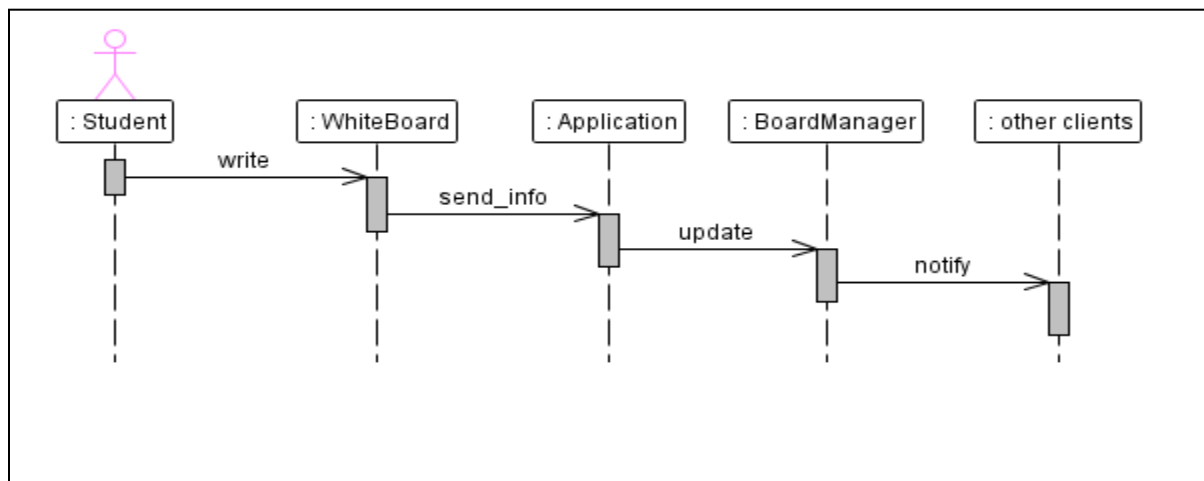


Figure 45: student writing board sequence diagram

When the student writes on Board after his request has been verified, the whiteboard sends coordination to the Application class in the server, and Application calls a method in the BoardManager class to update board shared object, and sends information to other clients within the class.

Student private chat sequence diagram:

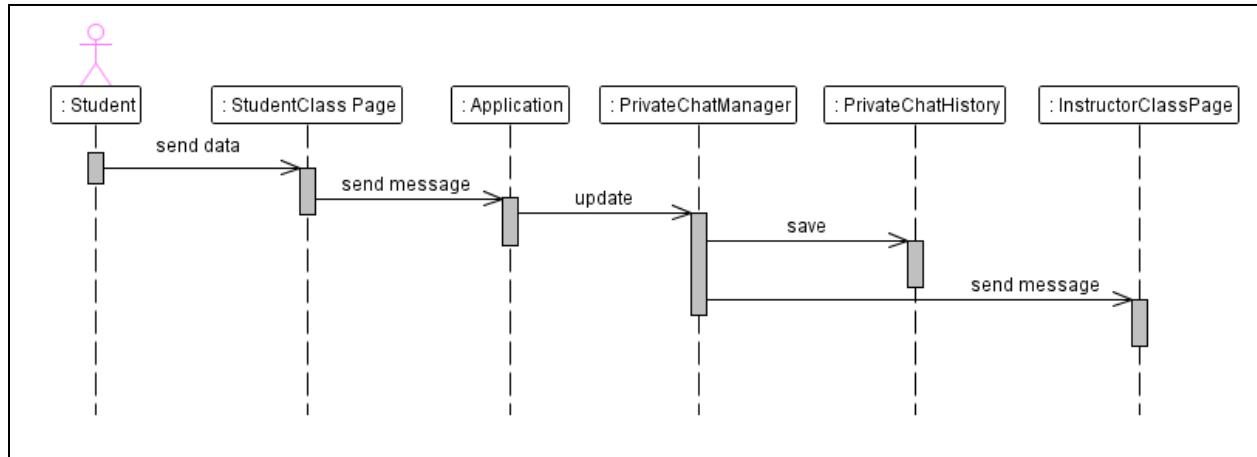


Figure 46: student private chat sequence diagram

In private chat the students can send private messages to the instructor only, the process starts by sending the message from the student page, then the StudentClass Page sends message to the Application class in the server, the Application calls a method in the PrivateChatManager class to update private shared object and save message in the database, after the private chat shared object has been updated, it sends the message to the instructor.

Logout from the class sequence diagram:

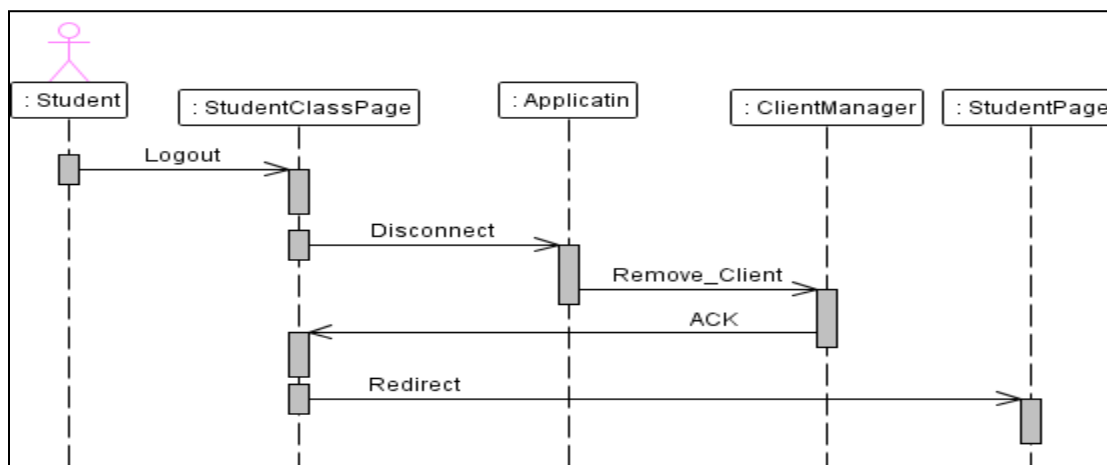


Figure 47: student class logout sequence diagram

When the student leaves the class, the StudentClass page disconnect from the Application, and the Application class sends the client Id to the ClientManager class to remove it from the clients list, and update shared object to notify other clients, that the student leave the class and update their lists.

Logout from the system sequence diagram:

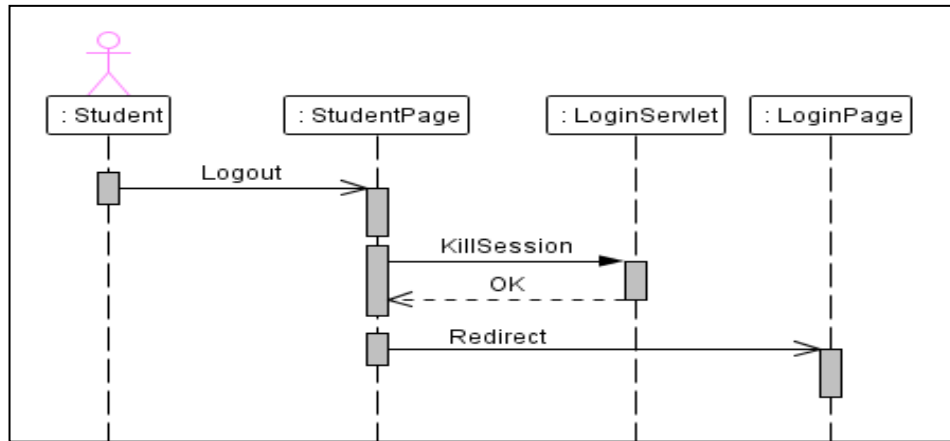


Figure 48: student system logout sequence diagram

After the student logout from the system, the Login Servlet kills the current session for the student, and sends acknowledgments to the Student page to redirect the student to the Login page.

2.2.4 Other sequence diagrams

Board History Shared Object Sequence Diagram:

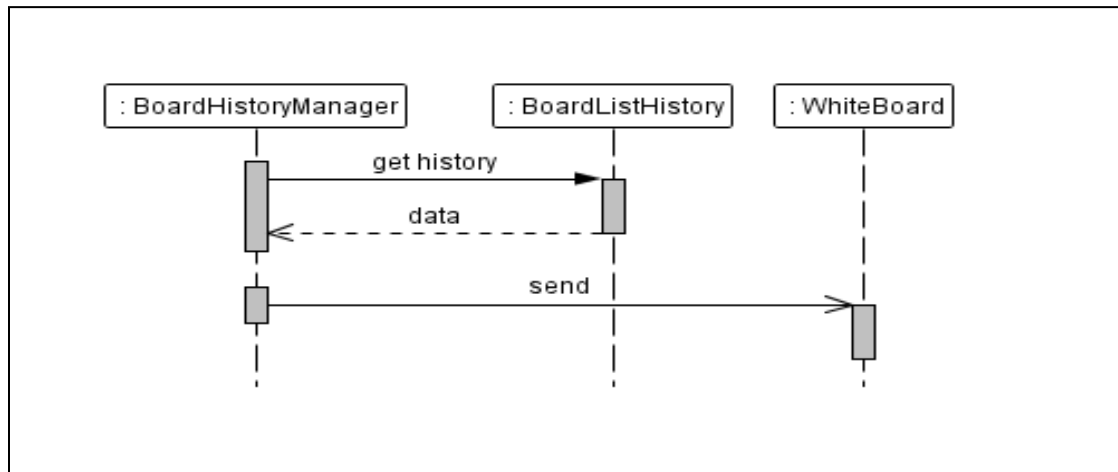


Figure 49: board history sequence diagram

If the user enters the room at a late time, or the connection failed during the lesson, the user is able to get all history of whiteboard when he logs in. When the user logs in, the `BoardHistoryManager` class in the server gets data from the history lists and sends them to the client that connected. The page of the client draws the data on the whiteboard.

Chat History Sequence Diagram:

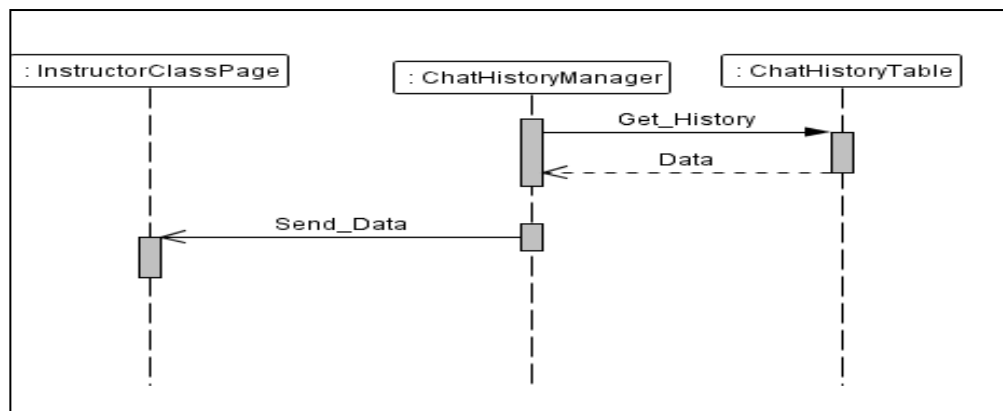


Figure 50: chat history sequence diagram

After the user has been logged to the class, the `HistoryChatManager` gets data from the history table in the database and updates the history chat shared object, which sends data to the connected user.

2.3 database design

In this section we will describe the tables of the database in our project according to the relational model. In our project we have a few numbers of tables that used to keep track of the registered users, created classes and history.

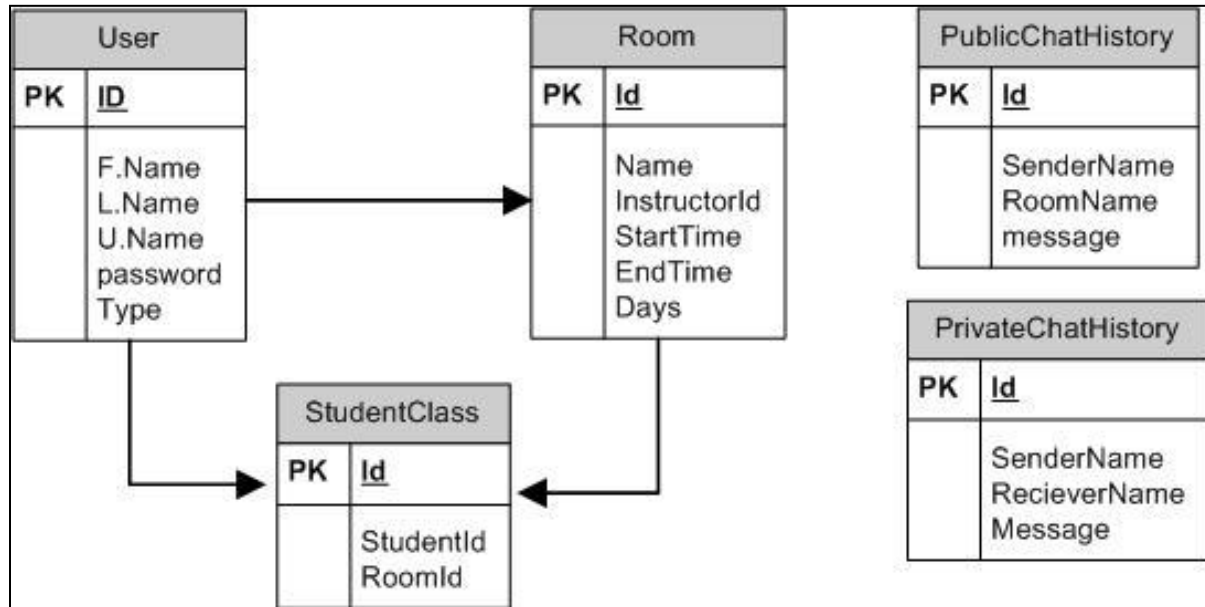


Figure 51: database tables of virtual classroom

In virtual classroom we have five tables, the first table is User which contains all users registered in the system (students, instructors, administrators).the room table contains the classes created by administrators in the system, and each class has only on instructor.

The Student class table contains students and classes they registered. The public chat table used to save history of chat during the lecture, this table is cleared when the lecture ends. The private chat history is like public chat history table but used to save history of private chat during the lesson, then it's cleared after the lesson ends.

2.4 Conclusion

At the end of this project we can say that we solve the most problems of traditional classroom that described before, so we have an online simple registration system, and virtual classroom that any instructor and student can access it from anywhere.

Because of our system is a website based, no special software needed to access the virtual classroom, all what we need in an internet connection with flash player plug-in in the browser, since the client side of our project base on flash actionscript.

Technically, we achieved a very good result for the quality of audio and video calls, we have a full real time whiteboard that can detect any written data and broadcast them to other clients, a full control in the instructor page hat give ability of the instructor to deny any interaction in the class, and we have a flexible graphical user interface that give user the ability to drag and close any component in the page.

References

- [1] Red5 server documentation
<http://www.red5.org>
- [2] Install Red5 on windows
<http://www.red5guide.com/>
- [3] Flex documentation
<http://www.adobe.com/devnet/flex/documentation.html>
- [4] Flex examples
<http://blog.flexexamples.com/>
- [5] Hibernate documentation
<http://www.hibernate.org/docs>