

LiveClass



Team Members

Hans (Wei-Han) Chang
weihanc@usc.edu

Zijian Hu
zijianhu@usc.edu

Jonathan Yao
jonathmy@usc.edu

Sophie Serrano
smserran@usc.edu

Beyza Bozbey
bozbey@usc.edu

Table of Contents

High-Level Requirements.....	3
Technical Specifications.....	4
Login Page.....	4
Home Page.....	4
Lectures Page.....	4
Streaming Page.....	5
User Access Level.....	5
Livestream.....	6
Database.....	6
Detailed Design.....	7
Hardware and System Requirements.....	7
Software Requirements.....	7
GUI: Front-end.....	8
Database Schema.....	15
ER - Diagram.....	17
Class Diagrams.....	18
State Diagrams.....	19
Testing	20
Login Test Cases	20
Lecture Page Test Cases.....	23
Streaming Page Test Cases.....	24
Access Level Test Cases.....	26
Camera Test Cases.....	34
Networking Test Cases.....	35
Stress Test Cases.....	36
Deployment.....	37
Files.....	37
Getting Started.....	37
Track Changes.....	38

High-Level Requirements

We will create a web application which will consist of a login page, a home page, a lectures page and a streaming page. Users will be required to login and once they have successfully done so, they will be redirected to the lectures page. Lectures page contains a list of all the available lectures. Once a lecture is clicked, the user will be redirected to the streaming page. This page will include a lecture live stream box, a live chat box, and a file/announcement section.

There will be three different types of users: guests, students, and course staffs. Each user will have different access levels. Guest can only view public live streams. Students will have the ability to view a live stream of a course staff's lecture as well as post questions through a live chat. Each time a live stream begins, students will be notified via email. Course staffs will have the same abilities as students in addition to being able to start and end a live stream and answer/delete student's chat questions. Also, course staffs will have the ability to control which users have access to the live streams through email invitations and access codes. Once a live stream ends, the chat will be saved to a file which can later be accessed by any valid user.

Technical Specifications

Login Page (2 hours)

The login page will contain an HTML form with username text field, password text field, and a login button. Users will be redirected to this page from the homepage using a login button that will be placed in the top corner of the home page.

Home Page (3 hours)

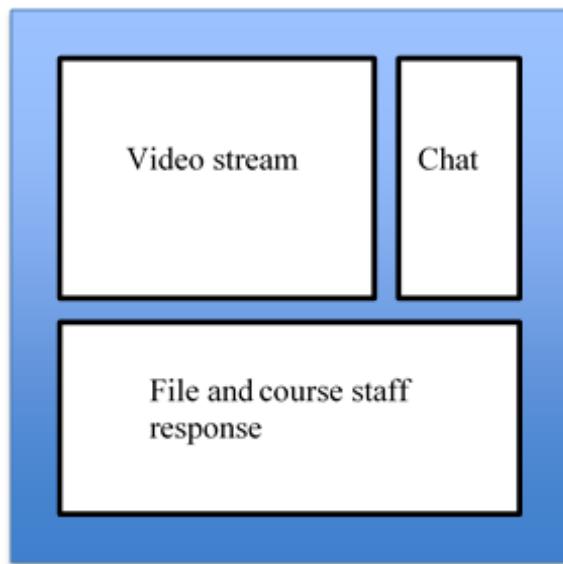
The home page will be a nicely designed page that provides an overview of the website's functionality. It will include a login button at the top.

Lectures Page (6 hours)

This page contains a list of all the public lecture live streams (live streams that can be accessed by anyone, including those without accounts). Once user signed in, this page will also display all the courses they are enrolled in.

Streaming Page (front end, 6 hours)

The main streaming page will have three major components; it will have the video stream, the chat feature on the side, as well as the file and comment section on the bottom. It will look something like this:



User Access Level (Business Logic, 7 hours)

There will be three types of users: guests, students, and course staff members. For each action, the access level of the user is checked against the database.

- A guest can only access public lectures and is not allowed to comment. Therefore, the chat section is disabled for all guests.

- Students will have the same ability as guests in addition to being able to send messages through the chat box and view lectures of enrolled classes. A student's homepage will have their enrolled class pinned at the top.
- Course staff members will have the ability to do all the above as well as the ability to control which users have access to the live streams through email invitations and access codes. Also, course staffs can start and end a live stream, answer/delete student questions and upload files.

Livestream (7 hours)

Livestream will consist of three parts – camera access, visual data parsing and serialization, and networking.

- The connection between course staff and the server is established when they start a live stream via WebSocket. The server will then distribute (broadcast) the video data to all validated viewers such as students in this class via WebSocket. Each viewer will have a socket that connects to the server.
- Camera access and visual data parsing and serialization will be implemented with JavaScript experimental technologies and third-party libraries

Database (4 hours)

- The database will consist of a user table, a course table, a course privilege table, a login token table, and a privilege table.
- The user table will be used for validating users. It consists of user ID, username, password, first name, last name and email.

- The course table will consist of course ID, course prefix, course number, and course name.
- The course privilege table will consist of course ID, user ID, and privilege ID. It is used to determine which users are course staffs of this course and which users are students.
- The privilege table contains privilege ID and privilege number. A privilege number is a 32-bit integer where each bit represents a privilege. A zero for a bit means this user doesn't have this privilege and a one means otherwise.

Detailed Design

Hardware and System Requirements

- Disk Space: Recommended 512MB
- Processor: Minimum Intel Core i3-2120 processor
- GPU: Minimum Intel HD Graphics 4000
- RAM: 512MB

Software Requirements

- Operating System:
 - Windows: Windows 7 and above (Windows 10 64-bit recommended)
 - Mac OS X: Intel-based Mac running Mac OS X 10.8.3+

- Linux: Ubuntu Linux 16.04 LTS and above
- Development:
 - JDK 8 and above
 - Gson
 - JSTL
 - Java EE 6 and above
 - An IDE that supports Java EE (Recommend Eclipse Oxygen)
 - MySQL database
- Production:
 - JRE 8 and above
 - Chrome Version 53 and above
 - Apache Tomcat 9.0 and above

GUI: Front-end

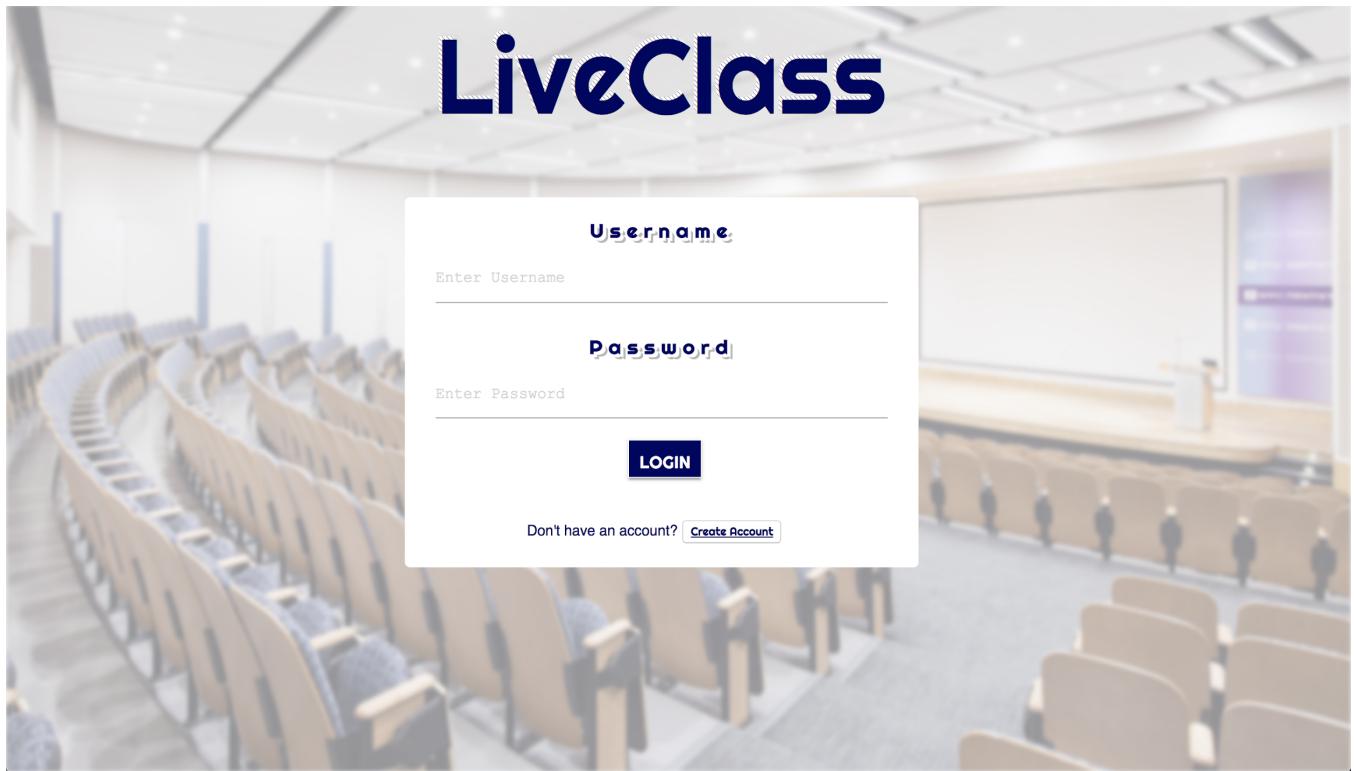
- Home Page:

The home page will include a home button, a login button, a brief summary of the website, and a View All Lectures button.



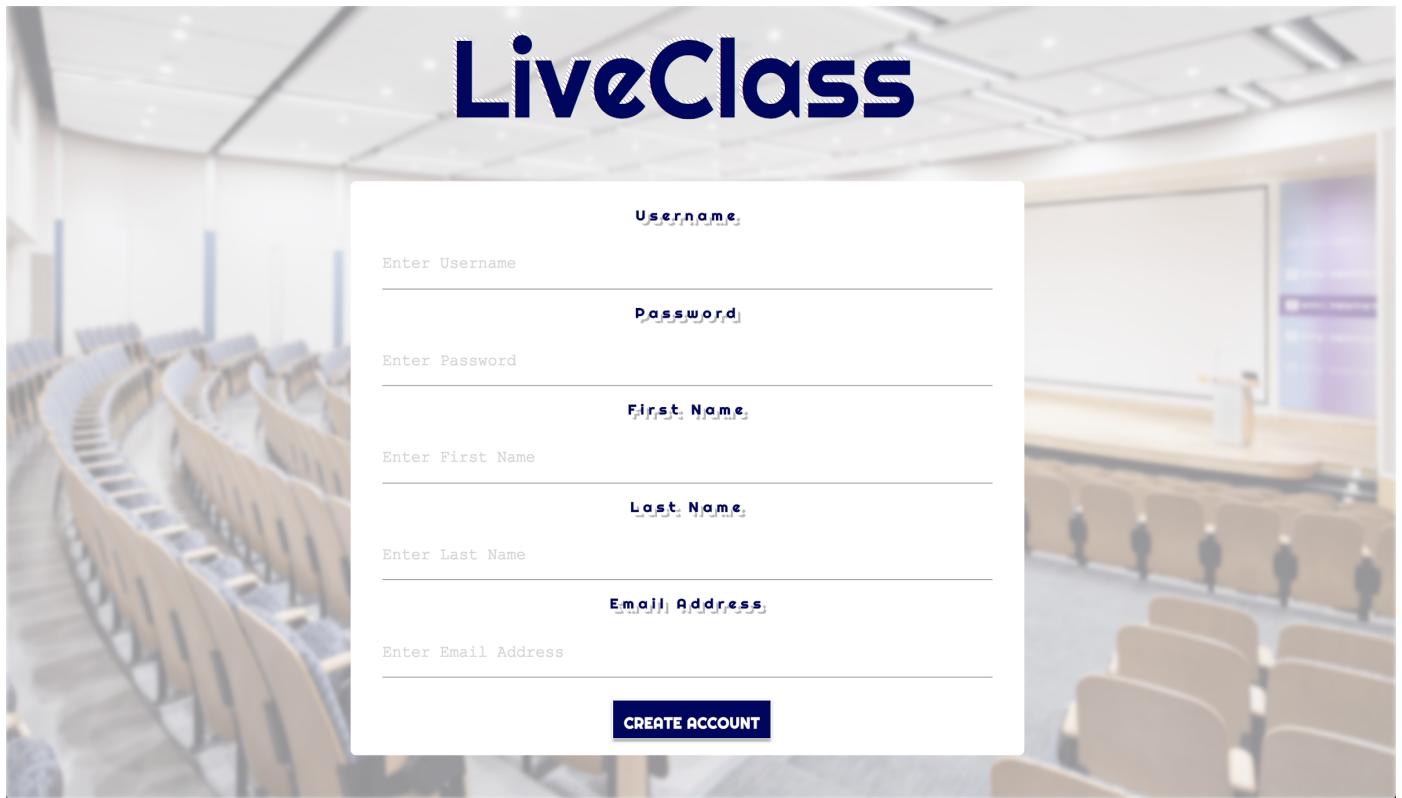
- **Login Page:**

The login page will simply contain a box where the user can enter their username and password in order to log them into their accounts. Logging in will give them access to the courses they are enrolled in, as well as all public lectures.



- **Create Account Page:**

The create account page will look like this for any user that does not already have an existing account and wants to make one. They will need to create a username and password and enter their first name, last name, and email address. Once they click create an account, they will have access to the courses they are enrolled in, as well as all public lectures.



- **Lectures Page:**

The lectures page will have three different looks according to the user who is logged in. The “Your Courses” box will contain all the courses that the user is enrolled in. The “All Public Lectures” box will contain all lectures that are available to users who do not have an account as well. When a user is not logged in, this page will look the same, just without having courses in the “Your Courses” box. If the user is a student, then this page will contain the “Enroll” button which will direct them to “Enroll Page”. If the user is an instructor, then this page will contain the “Access Code” button which will direct them to “Access Code Page”.

- ❖ Student view:

The screenshot shows a web application interface for "LiveClass". At the top, there is a navigation bar with links for "Home", "Enroll", and "Logout". The main content area has two main sections: "Your Courses:" and "All Public Lectures:". The "Your Courses:" section contains a list of three courses: "CSCI 103 - Intro to Programming", "CSCI 104 - Data Structures", and "CSCI 201L - Principles of Software Development". The "All Public Lectures:" section contains a list of four courses: "CSCI 270 - Algorithms", "CSCI 170 - Discrete Methods", "SOCI 150 - Social Problems", and "CSCI 356 - Intro to Computer Systems".

Your Courses:	
CSCI 103 - Intro to Programming	
CSCI 104 - Data Structures	
CSCI 201L - Principles of Software Development	

All Public Lectures:	
CSCI 270 - Algorithms	
CSCI 170 - Discrete Methods	
SOCI 150 - Social Problems	
CSCI 356 - Intro to Computer Systems	

❖ Instructor view:

The screenshot shows the 'LiveClass' application interface. At the top right, there are three buttons: 'Home', 'Access Code', and 'Logout'. The main content area has a blurred background of a university campus. On the left, there's a large blue header 'LiveClass'. Below it, the text 'Your Courses:' is centered above a list of course names in a light blue box. On the right, the text 'All Public Lectures:' is centered above another list of course names in a light blue box.

Your Courses:

- CSCI 103 - Intro to Programming

All Public Lectures:

- CSCI 270 - Algorithms
- CSCI 170 - Discrete Methods
- SOCI 150 - Social Problems
- CSCI 356 - Intro to Computer Systems

❖ Visitor view:

The screenshot shows the 'LiveClass' application interface for a visitor. At the top right, there are two buttons: 'Home' and 'Login'. The main content area has a blurred background of a university campus. On the left, there's a large blue header 'LiveClass'. Below it, the text 'Your Courses:' is centered above a message in a light blue box: 'Please log in to see your personal courses!'. On the right, the text 'All Public Lectures:' is centered above the same list of course names as the instructor view.

Your Courses:

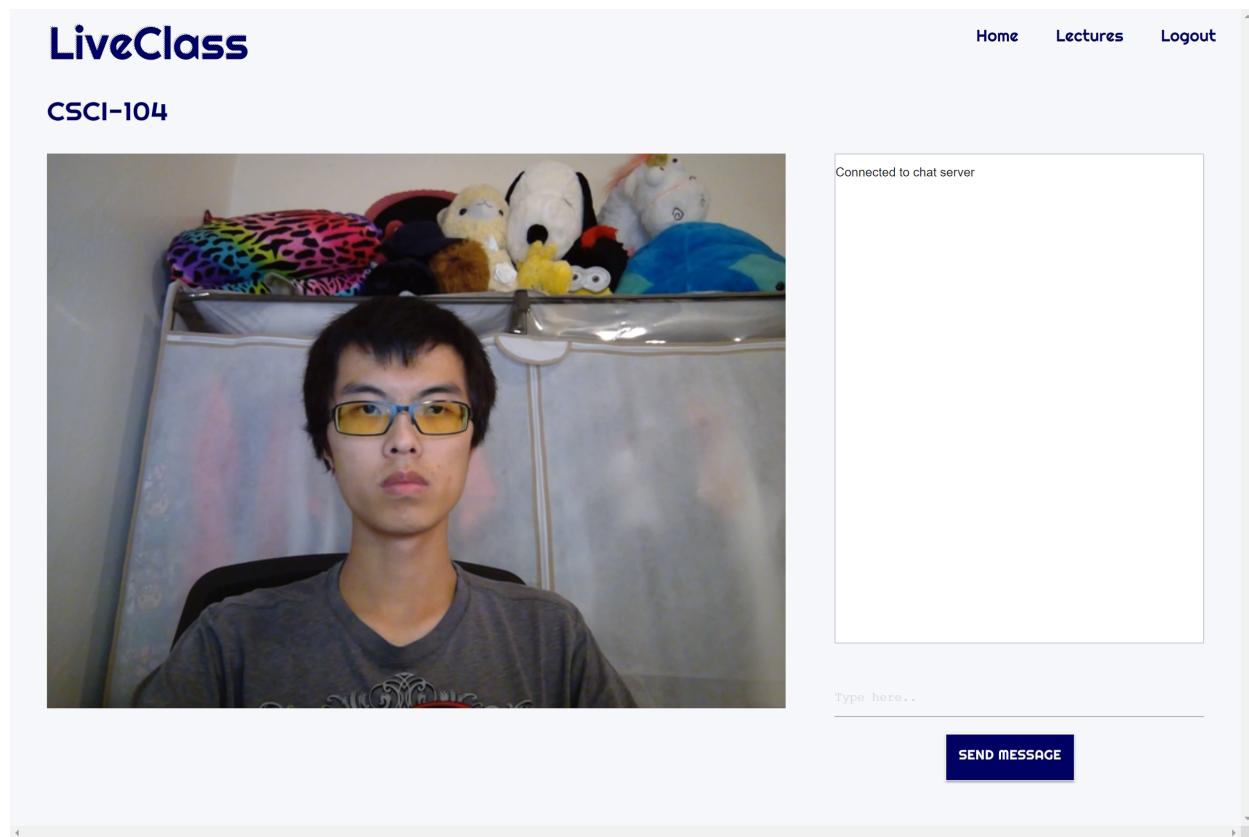
Please log in to see your personal courses!

All Public Lectures:

- CSCI 270 - Algorithms
- CSCI 170 - Discrete Methods
- SOCI 150 - Social Problems
- CSCI 356 - Intro to Computer Systems

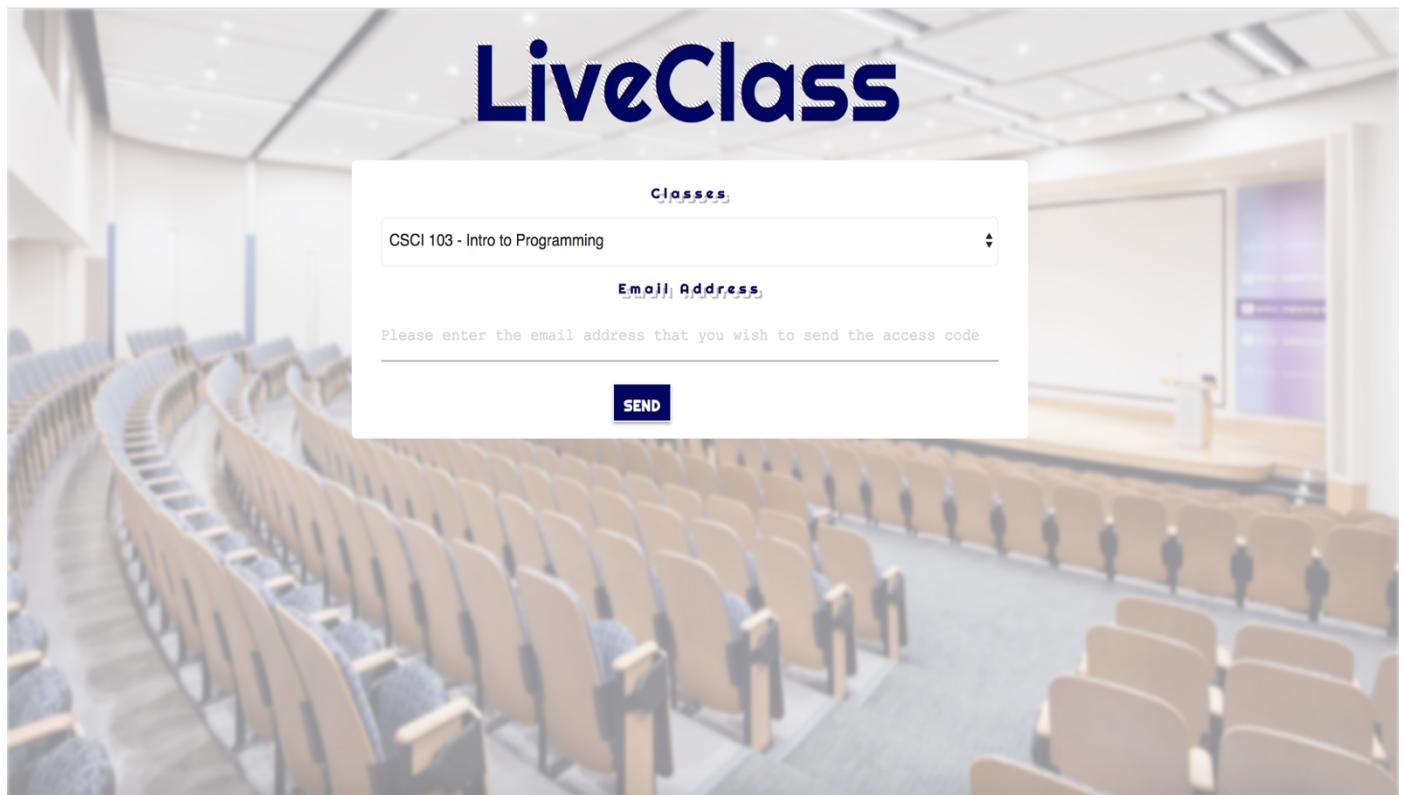
- **Streaming Page:**

The streaming page will contain a video window on the left, where the lecture livestream will be shown. Under the livestream will be a box where files and comments uploaded by the professor will be located. On the right will be the chat window where students can chat with other students as well as with professors/TA's.



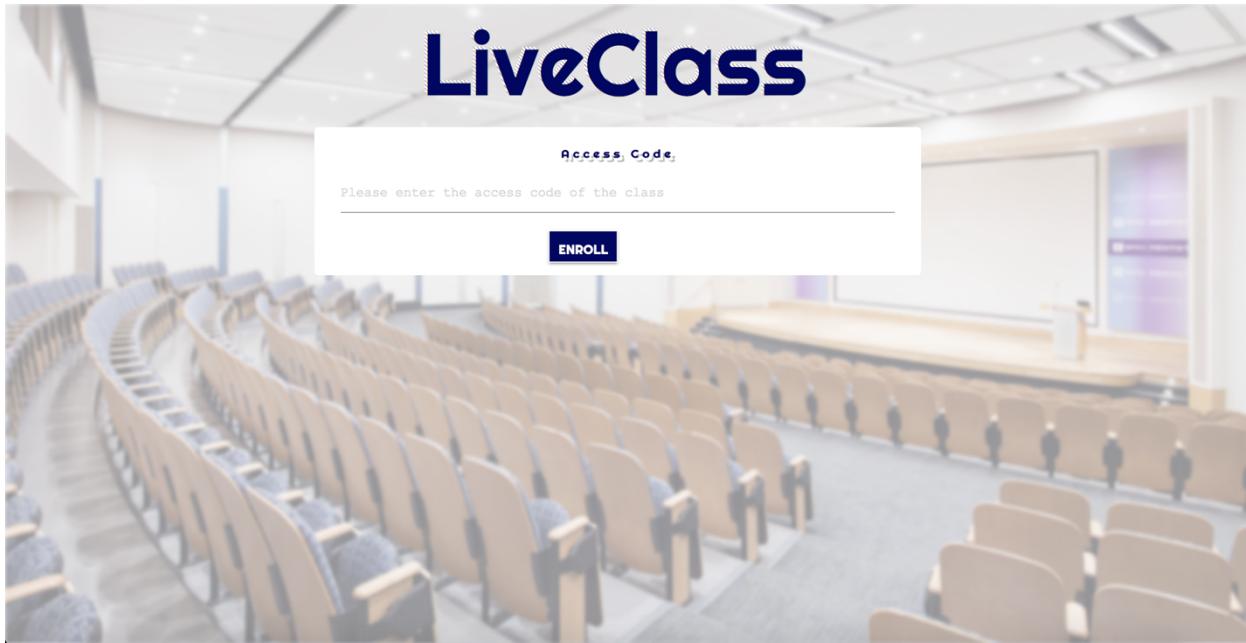
- **Access Code Page:**

This page will contain the “Courses” menu and the “Email Address” box. The user who is an instructor will be able to select the course that they would like to give access for and they will be able to enter the email address that they wish to send the access code of the course that they select from the menu.



- **Enroll Page:**

This page will contain the “Access Code” box that will allow the user to enter the access code of the course that they wish to enroll in.



Database Schema

Primary key

Foreign key

User Table

userID	username	password	firstName	lastName	email	salt
int	varchar(20)	varchar(15)	varchar(20)	varchar(20)	varchar(50)	varchar(100)
1	blue	pass1	Harry	Potter	hp@usc.edu	bcb64d3e824b 780a523a1d
2	green	pass2	Ron	Weasley	ronW@ucla. edu	dcb64d3e824b 480a433a1d
3	red	pass3	Hermione	Granger	herRed@usc. .edu	lds89d3e824b 9040a383a1d

Course Table

courseID	coursePrefix	courseNumber	courseName	accessCode
<i>int</i>	<i>varchar(10)</i>	<i>varchar(10)</i>	<i>varchar(100)</i>	<i>varchar(100)</i>
1	CSCI	103	Intro to Programming	2cbf77b3-c11c-4d81-bd9e-ab209e1f38
2	CSCI	201L	Principle of Software Development	2c009-c11c-4d81-bd9e-ab209e1f3810
3	EE	109	Intro to Embedded System	77b3-c11c-48u1-bd9e-a786f9-e1f381

Course Privilege Table

courseID	userID	privilegeID
<i>int</i>	<i>int</i>	<i>int</i>
1	1	2
2	2	3

privilegeID → “Privilege Table”.privilegeID

userID → “User Table”.userID

courseID → “Course Table”.courseID

Privilege Table

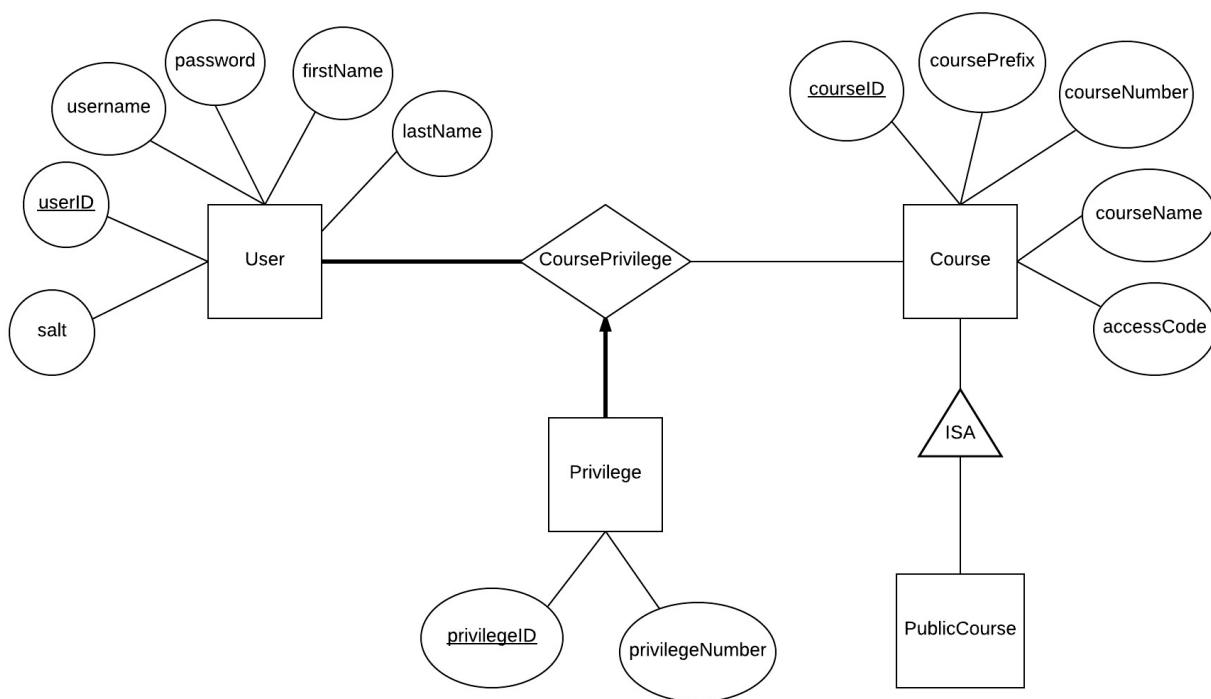
privilegeID	privilegeNumber
<i>int</i>	<i>int</i>
1	0
2	6
3	26

PublicCourse Table

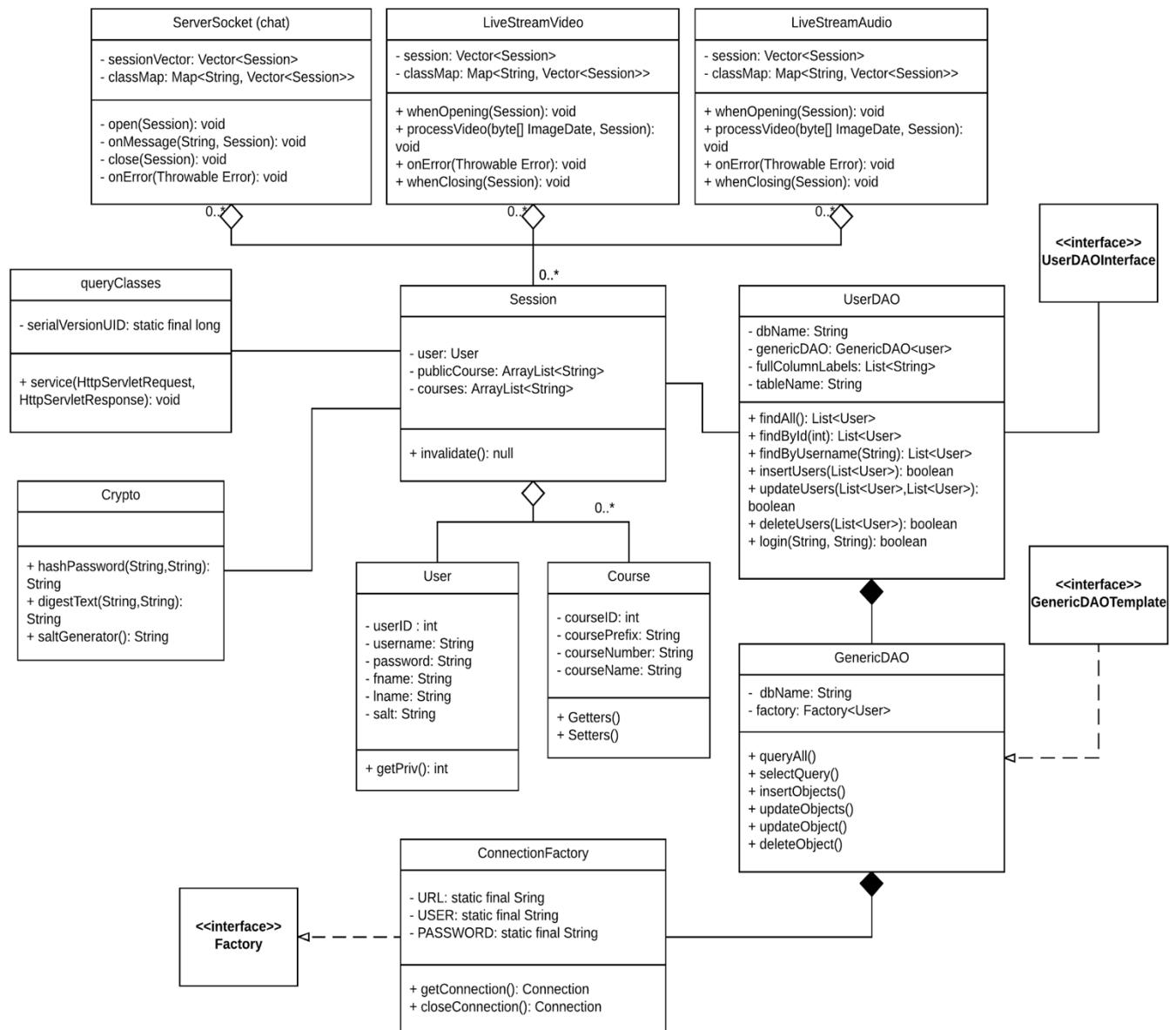
courseID
<i>int</i>
2
3

courseID → “Course Table”.courseID

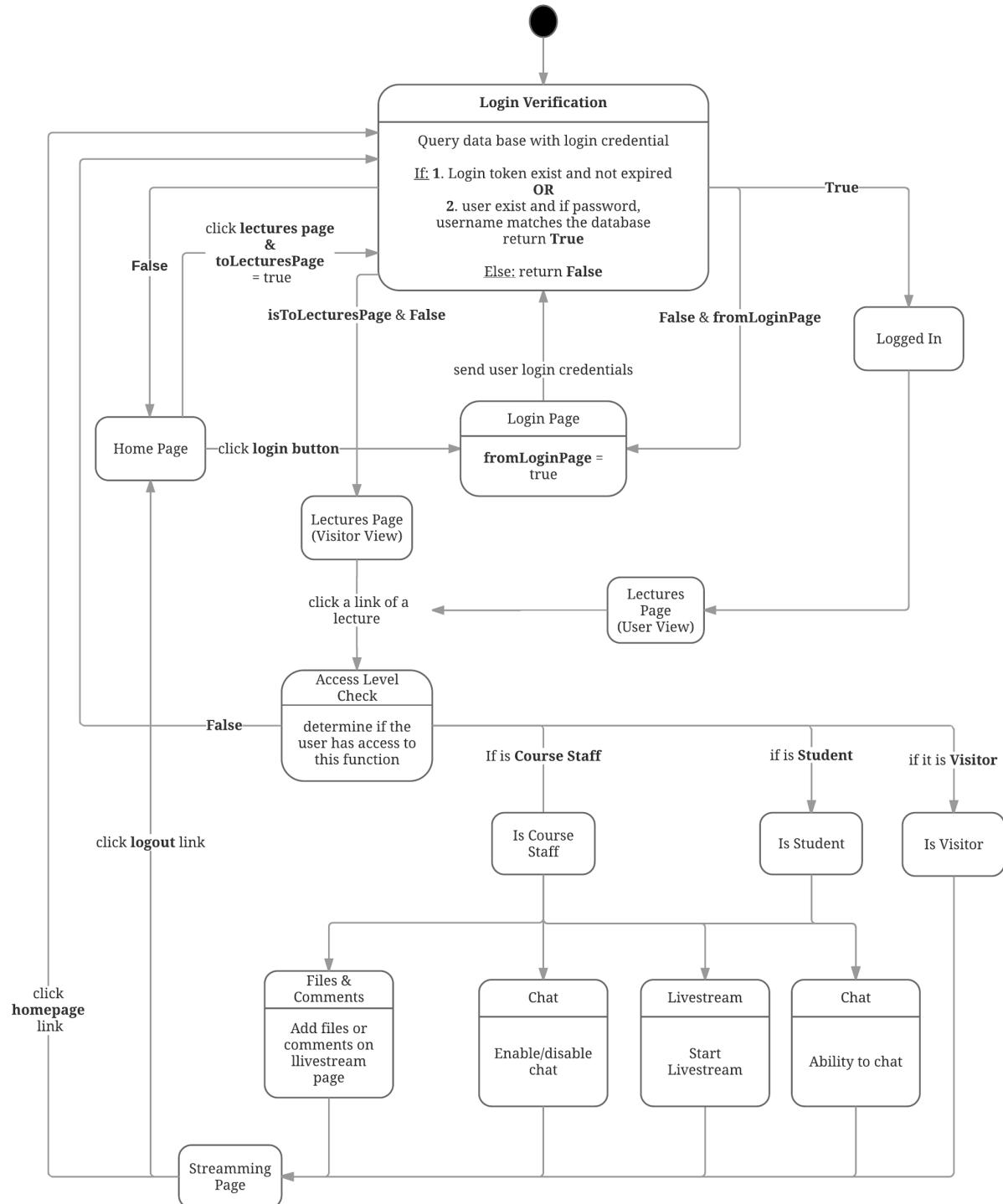
ER - Diagram



Class Diagrams



State Diagrams



Testing

Login Test Cases

White Box Test	
Test case	Test 1: Input no username and no password in login screen
Input/Steps	<ol style="list-style-type: none">1. Navigate to login page2. Without entering anything in the username and password fields, pressed the login button
Expected Output	Return an error message on the same page saying all fields must be filled

White Box Test	
Test case	Test 2: Input only username and no password to login screen
Input/Steps	<ol style="list-style-type: none">1. Navigate to login page2. Enter either a valid or invalid username3. Skip the password field4. Press login button
Expected Output	Return an error message on the same page saying all fields must be filled

White Box Test	
Test case	Test 3: Input only password and no username to login screen
Input/Steps	<ol style="list-style-type: none">1. Navigate to login page2. Enter a password but skip the username field3. Press login button
Expected Output	Return an error message on the same page saying all fields must be filled

White Box Test	
Test case	Test 4: Input a username that doesn't exist in database
Input/Steps	<ol style="list-style-type: none"> 1. Navigate to login page 2. Enter an invalid username that doesn't exist in database 3. Enter something into password field 4. Press login button
Expected Output	Return an error message saying invalid username/password

White Box Test	
Test case	Test 5: Input a username with incorrect password
Input/Steps	<ol style="list-style-type: none"> 1. Navigate to login page 2. Enter a valid username but the wrong password 3. Press login button
Expected Output	Return an error message saying invalid username/password

White Box Test	
Test case	Test 6: Input both valid username and valid password
Input/Steps	<ol style="list-style-type: none"> 1. Navigate to login page 2. Input a valid username and the password associated with it 3. Press login button
Expected Output	Redirect the user to the homepage, logged in as that user

Black Box Test	
Test case	Test 7: Check home button functionality
Input/Steps	<ol style="list-style-type: none"> 1. Navigate to login page 2. Press home button
Expected Output	Redirect the user to the homepage, without being logged in

Black Box Test	
Test case	Test 8: Check login button functionality
Input/Steps	<ol style="list-style-type: none"> 1. Navigate to login page 2. Press login button
Expected Output	Redirect the user to the login page again

Black Box Test	
Test case	Test 9: Check view all lectures button functionality
Input/Steps	<ol style="list-style-type: none"> 1. Navigate to home page 2. Press view all lectures button
Expected Output	Redirect the user to lectures page with lectures

Unit Test	
Test case	Test 10: Test SQL code in the username text space
Input/Steps	<ol style="list-style-type: none"> 1. Navigate to login page 2. Enter SQL injection code into username space
Expected Output	An error message is generated

Unit Test	
Test case	Test 11: Test SQL code in the password text space
Input/Steps	<ol style="list-style-type: none"> 1. Navigate to login page 2. Enter SQL injection code into password space
Expected Output	An error message is generated

Lecture Page Test Cases

Black Box Test	
Test case	Test 12: Check to see if “Your Courses” is removed when not logged in
Input/Steps	1. Navigate to lecture page when not logged in
Expected Output	The “Your Courses” section should not be visible

Black Box Test	
Test case	Test 13: Check to see if “Your Courses” is visible when logged in
Input/Steps	1. Navigate to lecture page without logging in, check that “Your Courses” is not visible 2. Log in through the login page 3. Navigate to lecture page again
Expected Output	The “Your courses” section should be visible

Black Box Test	
Test case	Test 14: Check home button functionality on this page
Input/Steps	1. Navigate to lecture page 2. Click the home button
Expected Output	The user should be redirected back to the homepage

Black Box Test	
Test case	Test 15: Check login button functionality on this page
Input/Steps	1. Navigate to lectures page 2. Click on the login button
Expected Output	The user should be redirected to the login page

Black Box Test	
Test case	Test 16: Test links to public courses when you're logged in
Input/Steps	<ol style="list-style-type: none"> 1. Navigate to lectures page when logged in 2. Click on a course under the “Public Course” section
Expected Output	The user should be sent to that course's streaming page

Black Box Test	
Test case	Test 17: Test links to user's courses when logged in
Input/Steps	<ol style="list-style-type: none"> 1. Navigate to lectures page when logged in 2. Click on course under “Your Courses” section
Expected Output	The user should be sent to that course's streaming page

Black Box Test	
Test case	Test 18: Test links to public courses when user isn't logged in
Input/Steps	<ol style="list-style-type: none"> 1. Navigate to lectures page when not logged in 2. Click on a course under “Public Course” section
Expected Output	The user should be sent to that course's streaming page

Streaming Page Test Cases

Black Box Test	
Test case	Test 19: Check files are added correctly
Input/Steps	<ol style="list-style-type: none"> 1. Navigate to Stream page 2. Add a file
Expected Output	File should appear in the file section

Black Box Test	
Test case	Test 20: Check files are accessible
Input/Steps	<ol style="list-style-type: none"> 1. Navigate to Stream page 2. Click on an arbitrary file
Expected Output	File should be able to be opened by all users

Black Box Test	
Test case	Test 21: Check if files are accessible when rejoin after closing the screen
Input/Steps	<ol style="list-style-type: none"> 1. Exit out of application 2. Navigate to Stream page and click arbitrary file
Expected Output	File should still be able to be accessed

Regression Test	
Test case	Test 22: Check if comments update correctly
Input/Steps	<ol style="list-style-type: none"> 1. Navigate to Stream page 2. Type in a comment and send
Expected Output	Message should show up in all current users' chat windows

Black Box Test	
Test case	Test 23: Check if comments are visible after rejoining
Input/Steps	<ol style="list-style-type: none"> 1. Leave the stream page 2. Rejoin stream page
Expected Output	Previous messages should still all be visible

Regression Test	
Test case	Test 24: Check if textarea updates the chat
Input/Steps	<ol style="list-style-type: none"> 1. Go to a streaming page as a student or course staff 2. Type something in chat box and press send 3. See if chat box updated on other viewer's computer
Expected Output	Chat updated on all users' computer

Regression Test	
Test case	Test 25: Check if textarea shows after quitting and rejoining screen
Input/Steps	<ol style="list-style-type: none"> 1. Click logout 2. Clear web browser cache 3. Go to stream page 4. Type something in textarea and press sent (if textarea shows) 5. See if chat box updated on other viewers' computer
Expected Output	Textarea shows and functional

Access Level Test Cases

Black Box Test	
Test case	Test 26: streaming page layout (no livestream started) as course staff
Input/Steps	<ol style="list-style-type: none"> 1. Login as course staff 2. Go to a course
Expected Output	<p>The following are visible</p> <ul style="list-style-type: none"> • Start livestream button • Add file button • Enable chat button

Black Box Test	
Test case	Test 27: streaming page layout (no livestream started) as student/guest
Input/Steps	<ol style="list-style-type: none"> 1. Login as student or not log in at all 2. Go to a course
Expected Output	<p>The following are disabled</p> <ul style="list-style-type: none"> • Video box • File box • Chatbox <p>A warning message is displayed in video box indicating livestream is not started</p>

Black Box Test	
Test case	Test 28: streaming page layout (livestream started) as course staff
Input/Steps	<ol style="list-style-type: none"> 1. Login as course staff 2. Go to a livestream
Expected Output	<p>The following are visible</p> <ul style="list-style-type: none"> • End livestream button • Add file button • Disable chat button • When right click on a chat message <ul style="list-style-type: none"> ◦ Delete ◦ Reply <p>The following are enabled</p> <ul style="list-style-type: none"> • Chatbox • Video box • File box

Black Box Test	
Test case	Test 29: streaming page layout (livestream started) as student
Input/Steps	<ol style="list-style-type: none"> 1. Login as student 2. Go to a livestream
Expected Output	<p>The following are visible</p> <ul style="list-style-type: none"> • When right click on a chat message <ul style="list-style-type: none"> ◦ Reply <p>The following are enabled</p> <ul style="list-style-type: none"> • Chatbox • File box • Video box

Black Box Test	
Test case	Test 30: streaming page layout (livestream started) as guest
Input/Steps	<ol style="list-style-type: none"> 1. Go to a livestream without logging in
Expected Output	<p>The following are disabled</p> <ul style="list-style-type: none"> • Chatbox <p>The following are enabled</p> <ul style="list-style-type: none"> • File box • Video box

White Box Test	
Test case	Test 31: Test starting stream as course staff
Input/Steps	<ol style="list-style-type: none"> 2. Login as course staff 3. Go to a course 4. Click start live stream 5. Check student's' mailbox 6. Login as student 7. Go to the livestream
Expected Output	Stream is shown on all clients connected. Email sent to enrolled students

White Box Test	
Test case	Test 32: Test starting stream as student/guest
Input/Steps	<ol style="list-style-type: none"> 1. Login as student or not log in at all 2. Go to an available livestream 3. Open browser console 4. Call start livestream javascript function directly in console
Expected Output	Display error in console. No livestream started

Black Box Test	
Test case	Test 33: Test ending stream as course staff
Input/Steps	<ol style="list-style-type: none"> 1. Login as course staff 2. Go to a course 3. Click start live stream 4. Open a different browser <ol style="list-style-type: none"> a. login as student or not log in at all b. Go to livestream page 5. Click end livestream
Expected Output	Stream ended shown on all clients' view

Regression Test	
Test case	Test 34: Test ending stream as course staff
Input/Steps	<ol style="list-style-type: none"> 1. Login as student or not log in at all 2. Go to a livestream available 3. Open browser console 4. Call end live stream function from console
Expected Output	Display an error. Livestream not affected

Black Box Test	
Test case	Test 35: Invite students as course staff
Input/Steps	<ol style="list-style-type: none"> 1. Login as course staff 2. Go to a course 3. Click invite user 4. Check invited user's mailbox
Expected Output	A link sent to target email address

Regression Test	
Test case	Test 36: Invite students as student/guest
Input/Steps	<ol style="list-style-type: none"> 1. Login as student or not log in at all 2. Go to an available course 3. Open console 4. Call invite user function from console
Expected Output	Display an error. No email sent

Black Box Test	
Test case	Test 37: Invite users who are already invited
Input/Steps	<ol style="list-style-type: none"> 1. Login as course staff 2. Go to a course 3. Click invite student 4. Open console 5. Invite an invited student by calling invite student function from console
Expected Output	All invited students are not visible when "invite student" is clicked Return an error saying the user is already enrolled

Regression Test	
Test case	Test 38: Test delete chat question as course staff
Input/Steps	<ol style="list-style-type: none"> 1. Login as course staff 2. Go to a livestream 3. Select a chat message 4. Click Delete 5. Log in using another account on a different browser 6. Go to the livestream
Expected Output	The chat message is deleted and other chat messages are not affected

White Box Test	
Test case	Test 39: Test delete chat question as student/guest
Input/Steps	<ol style="list-style-type: none"> 1. Login as student or not log in at all 2. Go to a livestream 3. Select a chat message 4. Open console 5. Call delete message function to delete the selected message from console
Expected Output	<p>Delete button should be disabled Display an error message in console</p>

Black Box Test	
Test case	Test 40: reply student chat message as course staff
Input/Steps	<ol style="list-style-type: none"> 1. Login as course staff 2. Go to a livestream 3. Right-click a message and click reply 4. Type something in chat box text area 5. Open a different browser 6. Login a different account 7. Go to the livestream page
Expected Output	<p>Course staff message should have different color The message should have @<user name> at the beginning</p>

Black Box Test	
Test case	Test 41: reply student chat message as student
Input/Steps	<ol style="list-style-type: none"> 1. Login as student 2. Go to a livestream 3. Right-click a message and click reply 4. Type something in chat box textarea 5. Open a different browser 6. Login a different account 7. Go to the livestream page
Expected Output	<p>Student message should have default color The message should have @<user name> at the beginning</p>

Regression Test	
Test case	Test 42: reply student chat message as guest
Input/Steps	<ol style="list-style-type: none"> 1. Go to a livestream without logging in 2. Open console 3. Call reply message function with an existing message from console 4. Open a different browser 5. Go to the livestream page
Expected Output	<p>Display an error in console No message sent</p>

White Box Test	
Test case	Test 43: Start more than one livestream at same time for a single user
Input/Steps	<ol style="list-style-type: none"> 1. Login as course staff 2. Go to a course 3. Click start livestream 4. Open a new tab 5. Go to a course 6. Open console 7. Call start livestream function
Expected Output	<p>Disable start livestream button once the user has started a live stream Display an error in console</p>

Black Box Test	
Test case	Test 44: Test deleting file as course staff
Input/Steps	<ol style="list-style-type: none"> 1. Login as course staff 2. Go to a livestream 3. Right-click a file in file box 4. Click Delete to delete the file 5. Refresh the page 6. Open a different browser 7. Login a different account 8. Go to the livestream
Expected Output	File should be deleted

White Box Test	
Test case	Test 45: Test deleting file as student/guest
Input/Steps	<ol style="list-style-type: none"> 1. Login as student or not log in at all 2. Go to a livestream 3. Open console 4. Call delete file function from console
Expected Output	Display an error in console

White Box Test	
Test case	Test 46: Test starting stream for a course when a livestream is already started for that course
Input/Steps	<ol style="list-style-type: none"> 1. Login as course staff 2. Go to a livestream 3. Open browser console 4. Call start livestream javascript function directly in console
Expected Output	Start livestream button should not be visible. Display an error in console. No additional livestream started

Camera Test Cases

Black Box Test	
Test case	Test 47: Camera unavailable but microphone available
Input/Steps	<ol style="list-style-type: none"> 1. Disable camera on a computer or use a computer that doesn't have camera 2. Login as course staff 3. Click start livestream 4. Open a different browser 5. Login a different account 6. Go to the live stream
Expected Output	<p>Live stream with sound only Display warning on video box indicating no visual content available</p>

Black Box Test	
Test case	Test 48: Camera available but microphone unavailable
Input/Steps	<ol style="list-style-type: none"> 1. Disable microphone on a computer or use a computer that doesn't have microphone 2. Login as course staff 3. Click start livestream 4. Open a different browser 5. Login a different account 6. Go to the live stream
Expected Output	<p>Live stream with silenced video Audio control button (volume and mute button) is disabled</p>

Black Box Test	
Test case	Test 49: Both camera and microphone unavailable
Input/Steps	<ol style="list-style-type: none"> 1. Disable camera and microphone on a computer or use a computer that doesn't have camera and microphone 2. Login as course staff 3. Click start livestream 4. Open a different browser 5. Login a different account 6. Go to the live stream
Expected Output	Video box is disabled

Networking Test Cases

Regression Test	
Test case	Test 50: course staff disconnected during livestream
Input/Steps	<ol style="list-style-type: none"> 1. Login as course staff 2. Click start livestream 3. Disable internet connection 4. Use a different computer 5. Go to the live stream
Expected Output	Display warning on video box indicating instructor is offline Other functionalities not affected

Black Box Test	
Test case	Test 51: course staff disconnected during livestream and rejoin the livestream
Input/Steps	<ol style="list-style-type: none"> 1. Login as course staff 2. Click start livestream 3. Disable internet connection 4. Use a different computer <ol style="list-style-type: none"> a. Login with a different account b. Go to the live stream 5. Reconnect to the internet 6. Go back to livestream
Expected Output	While offline, video box shows “course staff offline” Video box keep displaying livestreamed video once connection is re-established

Black Box Test	
Test case	Test 52: livestream distribution for multiple viewers
Input/Steps	<ol style="list-style-type: none"> 1. Login as course staff 2. Click start livestream 3. Open a different browser 4. Login a different account <ol style="list-style-type: none"> a. Go to the livestream 5. Use a different computer <ol style="list-style-type: none"> a. Login a different account b. Go to the livestream
Expected Output	All clients connected will see the same livestream video and chat messages during live stream

Stress Test Cases

Stress Test	
Test case	Test 54: maximum number of livestream running
Input/Steps	<ol style="list-style-type: none"> 1. For each i ($i = 1, 10, 100, \dots$) <ol style="list-style-type: none"> a. Create i courses b. Call start livestream method for each course c. Display i 2. Record i when server become unresponsive (the last i) 3. Restart server 4. For each j ($j = 1, 2, \dots, 10$) <ol style="list-style-type: none"> a. Create i courses b. Call start livestream method for each course c. Display j
Expected Output	The server becomes unresponsive at certain j . $i*j$ is the maximum number livestream can run at same time

Deployment

Files

The project consists of two files:

- Project.zip
 - Project folder
 - src: contains all the backend code
 - lib: contains all java libraries (in .jar format)
 - WebContent: contains all the frontend code
 - Database.sql: MySQL script to generate the database schema

Getting started

- **JDK 1.8**: if not installed, download and install JDK 1.8 from [Java SE Downloads](#)
- **Eclipse**: if not installed, download and install from [eclipse official website](#)
 - Choose either 32-bit or 64-bit version (Windows)
 - Only 64-bit version is available (Mac)
- **Tomcat 9.0**: if not installed, download and install from [tomcat official website](#)

In **Eclipse**: setting up the project

1. Export project.zip file into eclipse-workspace
2. import the project root folder
 - a. File → Import projects from file General → Existing Projects into Workspace
 - b. Select root directory → Click “Browse”
 - c. Select the project’s root folder in eclipse-workspace
 - d. Click “Open”
 - e. Click “Finish”
3. Adding JARs to eclipse project
 - a. Navigate to Project → Properties → Java Build Path → Libraries and click “Add External JARs...”
 - b. Locate the JAR files, select them, and then click OK. (See the outside libraries below)
4. Link outside libraries to web server
 - a. Right click on project → Properties → Deployment Assembly → Add → Java Build Path Entries → select all required .jar files

Outside Libraries:

- JavaServer Pages Standard Tag Library (JSTL): for front display logic

- [jstl-1.2.jar](#)
- MySQL Connector/J
 - [mysql-connector-java-5.1.40-bin.jar](#)
- jQuery
 - [Jquery-3.2.1.js](#)
- Materialized: for styling
 - See their [official website](#) for detail

In **MySQL Workbench**: setting up database

1. Start MySQL server
2. Log in with an account
3. Click “Open a SQL script file in a new query tab”
4. Execute the script (click “lightning bolt button”)

A step by step instruction on how to run this program is listed below:

1. To execute the program (In Eclipse)
 - a. Open WebContent folder
 - i. Right click “index.jsp”
 - ii. Click Run As → Run on server
2. The homepage should be loaded into your default browser.
3. To log in, press login button on the top right corner of the homepage.
4. Once redirected to login page, enter your username and password.
5. Once the user logs in, the system will decide whether the user is a course staff member or a student. The lectures page will be updated according to their access.
6. If the user wishes to proceed as a guest, they do not need to log in. The guest user can browse public lectures or they can enter a private lecture by entering its access code (acquired through email from a course staff of that course).

Track Changes

- “LoginToken” table is removed from the database schema.
- “PublicCourse” table is added to the database schema.
- Class diagrams are updated.
- Front-end images are updated.
- Front-end design is updated.