UPSA

UNIVERSITY OF PROFESSIONAL STUDIES, ACCRA

Scholarship with Professionalism

ACBSP ACCREDITED

FACULTY OF INFORMATION TECHNOLOGY AND COMMUNICATION STUDIES

DEPARTMENT OF INFORMATION TECHNOLOGY STUDIES

UNDERGRADUATE WORK

AN ONLINE VIDEO LECTURING PLATFORM FOR LEARNING AND TEACHING AT THE UNIVERSITY OF PROFESSIONAL STUDIES ACCRA (UPSA)

BY

ACKOM KAREN

DUODU RANDY

JULY 2021

UPSA

UNIVERSITY OF PROFESSIONAL STUDIES, ACCRA

Scholarship with Professionalism

ACBSP ACCREDITED

AN ONLINE VIDEO LECTURING PLATFORM FOR LEARNING AND TEACHING AT THE UNIVERSITY OF PROFESSIONAL STUDIES ACCRA (UPSA)

THIS PROJECT REPORT IS SUBMITTED TO THE DEPARTMENT OF INFORMATION TECHNOLOGY STUDIES OF THE FACULTY OF INFORMATION TECHNOLOGY AND COMMUNICATION STUDIES OF THE UNIVERSITY OF PROFESSIONAL STUDIES, ACCRA IN PARTIAL FULFILLMENT FOR A BACHELOR OF SCIENCE DEGREE IN INFORMATION TECHNOLOGY MANAGEMENT

JULY 2021

CANDIDATES’ DECLARATION

We, the undersigned do hereby declare that this dissertation is the result of our original research and that no part of it has been presented for another Degree in any University. We are convinced that this project was not copied from any other person. All sources of information have however been acknowledged with due respect.

ACKOM KAREN	10090428	31ST JULY, 2021
DUODU RANDY	10087872	31ST JULY, 2021

SUPERVISOR'S DECLARATION

I declare that the preparation and the presentation of this Dissertation were under the guidelines on supervision of the Dissertation laid down by the University of Professional Studies, Accra (UPSA).

DR. BEN OCRA

Date: 31ST JULY, 2021

DEDICATION

The team dedicates this dissertation to God Almighty who has been of tremendous help in making this study a success. Also, to our beloved parents and benefactors for sponsoring our education and also to all UPSA teaching and non-teaching staff for their explicit support, we dedicate this work to you all.

ACKNOWLEDGEMENTS

We extend our first gratitude to Almighty God for protecting and guiding us through the University of Professional Studies, Accra (UPSA).

To our unrelenting supervisor, Dr Ben Ocra under whose guidance and supervision this research has become a success, we remain extremely grateful. We further extend gratitude to all lecturers and students whose diverse contributions have led to the successful completion of this study.

Finally, we appreciate all the contributions made by our families and friends and the staff of the Department of Information Technology Studies of the Faculty of Information Technology and Communication Studies of the University of Professional Studies, Accra (UPSA) and the entire university for their help during the testing and implementation of this system.

ABSTRACT

Education is vital to the growth of an economy and conducting teaching and learning online is also important although the world is battling with a pandemic (i.e., COVID-19). Adapting to alternative ways of acquiring knowledge will not only help to prevent a pandemic but will improve our education system. The University of Professional Studies faces the challenge of conducting effective online video lecturing.

To solve this problem, the research reviewed existing information related to how UPSA conducts online video lecturing, which saw Zoom as the solution used by UPSA. The challenges of the existing system (Zoom) such as the high system requirements, downloading and installing, and other reasons motivated the team to build the current system. The team seeks to allow teachers to record student attendance, and lecture students with no unnecessary interferences. The team uses the Rapid Application Development software development method because it does not prolong the software development time. The team tested the current system to ensure it meets users and functional requirements, and also to ensure it is production-ready. The current system is a web application that requires the user to have a modern web browser like Chrome or Firefox installed on their devices. The major limitation of the current system is the lack of support for older browsers like Internet Explorer 8. The research team recommends the following features for future works: Future works should add features like importing scheduled events to the calendar, whiteboards, waiting room, recording, and conference room meeting. Future works should integrate a well-developed help-desk support system to improve user support.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER ONE

GENERAL INTRODUCTION

1.1 BACKGROUND OF STUDY

Recent years have seen a quick adoption of live online lecturing in the educational sector. In this era, online lecturing has become the norm due to the global COVID-19 pandemic.

A live online lecturing is a real-time session that enables lecturers or teachers and students to replicate many elements of an in-person class while online. The instructors teach, and the participants learn in real-time, face-to-face but via internet-enabled technology devices.

Online lecturing makes it possible for teachers to teach students at diverse geographical locations. Also, we can automate some major processes of face-to-face lecturing. For instance, we can record student attendance automatically for the instructor.

Building a live online lecturing platform involves a lot. A common misconception is that live tutoring is just about video streaming. Although video streaming is the most important part of every live online lecturing platform, a live online lecturing platform comprises other features like online whiteboards, chat windows, data logging, class scheduling system, notification system, and user authentication and authorization system.

Video streaming is the continuous transmission of audio or video data or files from a server to a client. With streaming, we broadcast the media or store it online, and transmit seconds of it over the Internet to play on the client device such as a mobile phone or laptop. Video streaming allows the user to download video content while watching it simultaneously. We view the content online, rather than saving it to a device.

Video streaming comprises video-on-demand streaming (VOD) and live streaming.

Live streaming is streaming real-time or live video feed to an audience retrieving the media stream over the internet. It requires a camera and/or microphone for capturing the media content, which the encoder encodes into multiple bitrate streams. The server relays the multiple bitrate streams to a media server or CDN (content delivery network) to distribute or deliver content to the audience. For example, YouTube, Facebook Live, Periscope, and Twitch are live streaming services.

VOD enables the audience to watch or listen to media stream content on request. For example, Netflix and YouTube are on-demand streaming services.

Streaming media over the internet for an audience to watch requires adopting a streaming technique such as Adaptive Bitrate streaming or Multi-Bitrate streaming, on the media server.

Adaptive bitrate streaming (ABR) is a technique for progressively adjusting the compression level and video quality of a stream to match the user's bandwidth availability, user's network conditions, and device performance. Netflix implements adaptive streaming on its media servers.

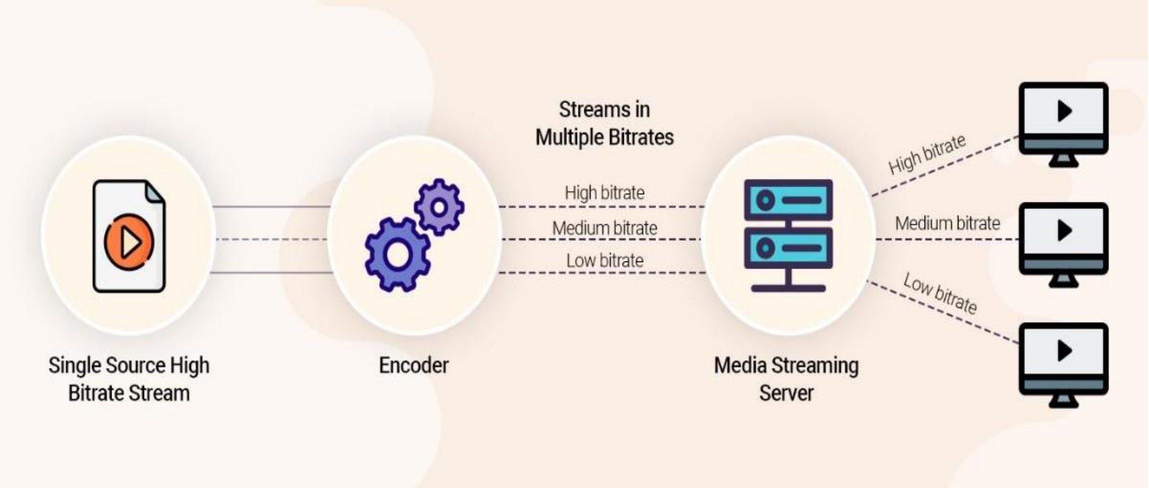


Figure 1.1: Encoding and Decoding of stream

In **Multi-Bitrate (MBR) streaming**, we make a particular video stream available in multiple bitrates, and the user has to select the best possible video quality from the variety of options available. YouTube implements multi-bitrate streaming on its media servers.

As presented in Figure 1.1, we observed that the system transfers a single video source stream to the encoder, which encodes the single video source into multiple bitrates. We then transfer the multiple bitrates to the media server (A media server is a network device that saves and shares media) which handles transferring stream bitrates to viewers' devices using adaptive bitrate streaming or multi-bitrate streaming.

For a live video streaming service, adding a text messaging feature is important. Creating a text messaging feature requires the use of messaging protocols. Examples: WebSocket. The WebSocket messaging protocol makes it easy to create text messaging for a live online lecturing platform.

In the WebSocket protocol, a client sends an **HTTP** request which adds the **UPGRADE** header to the request. The server takes the request and recognizes the client wants to upgrade the regular one-to-one **HTTP** connection to a **WebSocket** connection. The server will reply either by switching protocols with a **101** HTTP status code or not. This creates a bidirectional connection that can be closed if the connection ends.

An online whiteboard feature is a replica of the traditional classroom whiteboard. This feature is more advanced and interactive because you can open multiple whiteboards to draw, illustrate, show diagrams, upload presentations, images or even play videos from your library.

1.2 PROBLEM STATEMENT

Live online lecturing has become an important technology used in the educational sector lately because of the outbreak of the COVID-19 pandemic. Institutions are utilising live online lecturing platforms to conduct classes.

Adapting to this way of teaching and learning has been difficult for the University of Professional Studies Accra (UPSA), because of the following issues:

- Unstable internet access at some places in the country where teachers and students live.
- Teachers and students required to download, install and update streaming software like Zoom
- Teachers and students not having computers that meet the system requirements of software
- High consumption of data bandwidth by the streaming software
- Teachers and students finding it difficult to understand the interface of streaming software like Zoom
- The streaming software does not record student's attendance hence students do not attend the class

The issues listed above made it difficult for teachers and students to conduct online lectures.

Based on the issues listed above, the research team intends to develop a live online lecturing platform for the University of Professional Studies Accra (UPSA), for conducting classes online.

The benefits of the proposed system for teachers and students are:

- Improving video performance and quality
- Access from any internet browser
- Lower consumption of data bandwidth
- Not requiring download, installation, and update of software
- Easy to schedule, manage and monitor live online classes
- Easy, user-friendly, and accessible user interface
- Not requiring users to have computers with higher specifications

- The system records class attendance
- Increase engagement with real-time audio-video and textual communication

1.3 SCOPE OF THE PROJECT

The beneficiaries of this project are only the University of Professional Studies Accra (UPSA) students and lecturers. The research team estimates the system to host up to 300 participants for each meeting including the instructor.

The development of the project will last for a maximum of 4 months.

The live online lecturing platform does not implement the whiteboard and video recording features.

Although every video streaming technology used on a live online lecturing platform requires an encoder and a media server, the study will not involve creating the encoder and media server, but the research team will integrate a production-ready media server called Ant Media server.

1.4 LIMITATION OF THE STUDY

In developing the platform, the research team will face hitches that will limit the study. The difficulties the study will face are:

- The project is only limited to the studied population (UPSA).
- Having limited time to research and develop the project.
- Deployment and uptime running cost of the software
- Requires internet connectivity
- Support for older browsers not implemented
- Learning new technologies like Python, HTML, CSS, JavaScript, and Bootstrap 5 CSS framework.

1.5 OBJECTIVES OF THE STUDY

1.5.1 General Objective

The primary aim of the study is to develop a live online lecturing platform for teaching and learning at UPSA. To achieve this aim, the team must build the frontend and backend of the system.

1.5.2 Specific Objectives

To achieve the general aim, the team planned the following specific objectives to guide the project.

1) Build frontend of the system
    - i. develop the interface for the home page, about page, attendance page, contact page, and page not found
    - ii. develop the interface for the host class page and the join class page
    - iii. develop the interface for the studio page and viewer page
2) Build backend of the system
    - i. manage online class sessions
    - ii. authenticate instructors
    - iii. integrate the database (MongoDB) and the media server (Ant Media server)
    - iv. stream online class sessions
    - v. connecting or disconnecting instructors and participants to/from their classrooms
    - vi. create a simple text messaging platform for group interactions
    - vii. track class attendance automatically
    - viii. send emails and generate attendance reports

1.6 METHODOLOGY

To develop the proposed study, the team used the Rapid Application Development software development method. This method involves software users in all stages of application development.

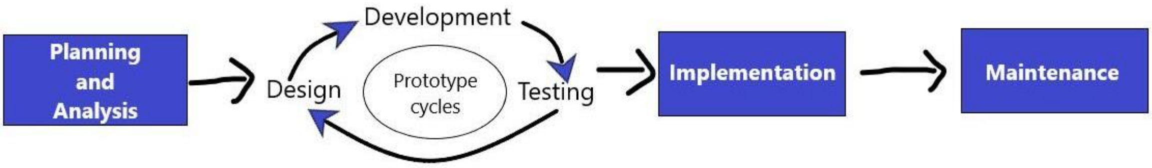


Figure 1.2: Steps of Rapid Application Development

Rapid Application Development (RAD) is a form of agile software development method which prioritizes quick prototype releases and iterations. Unlike the Waterfall method, RAD uses software and user feedback instead of rigorous planning and requirements capturing.

All iteration involves teams working simultaneously on various areas like;

- Planning and Analysis
- Design
- Development
- Testing
- Implementation
- Maintenance

At the end of the iteration, the team displays a working product to the customer and important stakeholders.

1.7 ORGANIZATION OF STUDY

Chapter one deals with the general introduction which includes the background of the study, problem statement, the scope of the project, limitation of the study, objectives of the study, and methodology.

CHAPTER TWO

LITERATURE REVIEW

2.1 INTRODUCTION

The content of this chapter summarises the literature and information about online lecturing that will be the foundation for designing our project.

This chapter aims to:

- Summarise the history and development of online classrooms.
- Examine existing systems being used by UPSA for online lecturing.
- Compare the proposed system with the existing system being used by UPSA.

2.2 GENERAL BACKGROUND OF THE STUDY AREA

History of Virtual Learning

Virtual learning was first practised by the University of House in 1953. The university started offering televised college classes on KUHT (today called Houston PBS). KUHT aired the classes for 13-15 hours each week, accounting for approximately 38% of the channel's total broadcast time. They aired these courses in the evening to benefit learners who worked during the day.

The invention of personal computers and the web revolutionised distance education. In 1989, the University of Phoenix launched the first fully online university that offered both bachelor’s and master’s degrees.

Since the making of these complete online schools, distance learning has kept on growing. Today we enrolled 1 out of 4 undergrads in one online class. In 2003, the Blackboard Learning System staff declared 40,000 educators were showing 150,000 online courses to over 6 million understudies across 55 nations. This pattern appears to proceed later on: as of now, 83% of all U.S. organizations that offer online courses say they expect an expansion in online enlistment in the coming decade (*Virtual Classroom Software to Deliver Live Teaching and Training | WizIQ*, n.d.).

Components of a live online lecturing platform

Building a live online lecturing platform involves a lot. A live online lecturing platform requires knowledge and skills in video streaming, web development, how the web works, and database management (*Virtual Classroom Software to Deliver Live Teaching and Training | WizIQ*, n.d.).

Video Streaming

Video streaming is the continuous transmission of audio or video data or files from a server to a client. Video streaming allows the user to download video content while watching it simultaneously (*What Is Video Streaming and How Does It Work? | Akamai*, n.d.).

Video Streaming comprises video-on-demand streaming (VOD) and live streaming.

- Live streaming is streaming real-time or live video feed to an audience retrieving the media stream over the internet. For example, YouTube, Facebook Live, Periscope, and Twitch are live streaming services (*What Is Streaming? | How Video Streaming Works | Cloudflare*, n.d.).
- Video-On-Demand (VOD) enables the audience to watch or listen to media stream content upon request. In simple terms, we store VOD contents remotely on cloud storage. An archived live stream can be video-on-demand. For example, Netflix and Hulu are on-demand streaming services (*What Is Streaming? | How Video Streaming Works | Cloudflare*, n.d.).

Streaming media over the internet for an audience to watch requires adopting a streaming technique such as Adaptive Bitrate streaming or Multi-Bitrate streaming, on the media server.

Web Development

Web development is the planning, designing, building, and maintenance of websites. A website is a collection of publicly accessible, interlinked **web pages** published on the internet. Web development includes aspects such as web design, web programming, database management, and web publishing (*How the Web Works - Learn Web Development | MDN*, n.d.).

A web developer designs website interfaces using front-end tools like HTML, CSS, and JavaScript but uses server-side scripting languages like Python, PHP for writing the logic for the server.

The tools for building a website are many, but the web developer must decide on which tool to use in developing their websites. Examples of tools used for web development are HTML, CSS, Python, PHP, JavaScript, Ruby and more.

How the Web works

Client and server computers connect to the web. **Clients** are the typical web user's internet-connected devices such as PC or phone and web-accessing software available on those devices like Firefox or Chrome. **Servers** are computers that store web pages, sites, or apps.

When we open a browser and enter a web address, the following steps occur:

- The browser goes to the DNS server and finds the actual address of the server that the website lives on.
- The browser sends an HTTP request message to the server, asking it to send a copy of the website to the client.
- If the server approves the client's request, the server sends the client a "200 OK" message, which means "Obviously you can view the website", and sends the website's files to the browser as a series of small chunks.
- The browser assembles the small chunks into a complete web page and displays it to you (*How the Web Works - Learn Web Development | MDN*, n.d.).

Database Management

A database is a standardised set of data saved and accessed electronically from a computer system (*What Is a Database | Oracle*, n.d.).

Database Management is the data storage tasks and security practices of a Database Administrator (DBA) throughout the life cycle of the data (Knight, 2018). Administering a database involves modelling, implementing, and protecting stored data to expand its value.

There are various types of databases used for storing different varieties of data which includes centralised database, distributed database, relational database, non-relational database, network database, object-oriented database, and hierarchical database.

We will limit our study to two (2) database types. The two (2) types are **relational database (structured)** and **non-relational database (unstructured and semi-structured)**.

A **relational database** contains multiple tables of data (relations) with rows (tuple) and columns (attributes) that relate to each other through special key fields (*Types of Database System | Nibusinessinfo.Co.Uk*, n.d.). Relational databases use Structured Query Language (SQL). Examples are MySQL, PostgreSQL, SQLite, Oracle, and MariaDB.

A **non-relational (NoSQL) database** takes a variety of forms and allows you to store and manipulate large amounts of unstructured and semi-structured data (*Types of Database System | Nibusinessinfo.Co.Uk*, n.d.). NoSQL database provides high scalability and enables good productivity in application development. We can divide NoSQL databases into four (4) groups. They are Key-value storage, Wide-column stores, Graph Databases, and Document-oriented databases. Examples are MongoDB, Couchbase, Firebase, Redis, and Amazon Dynamo DB.

Database Management System (DBMS) is software for storing and retrieving data from the database while considering appropriate security measures (Peterson, 2021). It comprises a group of programs that manipulate the database. The DBMS accepts the request for data from an application and instructs the operating system to provide the specific data. Examples are MySQL, MongoDB, Oracle, and Redis.

2.3 REVIEW OF THE EXISTING SYSTEM

The outbreak of the COVID-19 pandemic made it impossible for teachers and students to conduct in-person meetings. Because of that, institutions including UPSA adopted the use of a video conferencing app called **Zoom** for conducting online lecturing.

Zoom is a cloud-based video conferencing service you can use to meet virtually with others - either by video or audio-only or both, all while conducting live chats - and it lets you record those sessions to view later. Zoom Video Communications, Inc. created zoom (Tillman, 2021).

Figure 2.1: Zoom App Logo

Zoom has become the preferred video conferencing software because of the following features. They are:

- Huge meeting participant capacity–Zoom can host up from 100 to 500 participants when you subscribe to the large meeting plan
- Impressive features–Includes features like whiteboard, screen-sharing, and chat room
- Records meeting sessions–Host can record meetings
- Low-latency–optimised to process data with minimal delay

According to Wikipedia, Zoom has faced public and media scrutiny related to security and privacy issues because of unexpected usage. Although it has some recommendable features, the research team deems it as a general-purpose video conferencing app and not an app designed for online lecturing (Wikipedia, 2021).

Although is a functional software, it has certain limitations and challenges (Tillman, 2021). These limitations are:

- Zoom requires a high-end computer or mobile specifications
- Requires download and installation of software by the user
- Requires download of periodic updates by user
- Inefficient use of device power supply (i.e., high-consumption of battery)
- The user interface is difficult to understand
- Does not provide attendance report
- Slow in adapting to network bandwidth
- Allocates a minimum time (40 minutes) to meetings scheduled using the free tier
- High consumption of data bandwidth

Based on the limitations stated above, the research team proposes to build a system for small-medium groups to conduct an online lecture.

2.4 COMPARATIVE STUDY OF REVIEWED SYSTEM

The Proposed System

The proposed system by the research team is a web-based streaming platform called **iStream**. We built iStream using modern web and streaming technologies.

The proposed system has the following features:

- Access from any internet browser
- Lower consumption of data bandwidth
- Not requiring download, installation, and periodic updates of software by the user
- Easy, user-friendly, and accessible user interface
- Not requiring users to have a computer or mobile device with higher specifications
- The system records class attendance

- Increase engagement with real-time audio-video and textual communication
- Ultra-low latency–The streaming engine used to build iStream processes data with a very minimal delay
- Adaptive streaming of media using WebRTC technology
- Adapts quickly to network bandwidth
- Improving video performance and quality

The table below gives a vivid comparative difference between the proposed system and the existing system.

	PROPOSED SYSTEM (iStream)	**EXISTING SYSTEM (Zoom)**
1.	iStream requires users to have a computer or mobile device with an internet browser which most students possess.	Zoom requires a high-end computer or mobile specifications which most students do not possess (Zoom Help Center, 2021). This results in fewer students joining meetings.
2.	iStream being a web-based platform does not require a download, installation, and periodic updates of software by the user.	It requires download, installation, and periodic updates of software by the user.
3.	The team optimised the system to process media data with ultra-low latency.	Optimised to process media data with low latency.
4.	It has an easy, user-friendly, and accessible user interface.	The user interface is difficult to understand, so users do not utilise most features (Tillman, 2021).

5.	It shows participants and records class attendance for each meeting.	It shows participants but does not record and print attendance.
6.	Adapting to network bandwidth changes is fast because of the WebRTC technology being used (Graves, 2018).	Slow in adapting to network bandwidth.
7.	iStream ensures it encrypts data by putting in security measures like streaming only on HTTPS protocol, one-time passwords for hosts, sanitising data before inserting in the database, using secured database management systems, and setting HTTP security headers like HSTS, X-XSS-Protection, etc. on HTTP Responses.	Although, Zoom has been criticised for security lapses, poor encryption practices, and poor design choices (Wikipedia, 2021). Zoom offers a range of authentication methods such as SAML, Two-Factor Authentication (2FA), OAuth, and/or Password-based which can be individually enabled/disabled for an account.
8.	It consumes less data bandwidth because of the adaptive bitrate streaming technique and WebRTC technology being used (*Adaptive Bitrate Streaming (ABR) | Video Streaming Definition*, n.d.).	Zoom consumes high data bandwidth. If there is high network coverage, an average zoom can consume 800 MB up to 1.64 GB. If the network is slow or average, it might consume around 300MB per hour (Aachri, 2021).
9.	iStream does not allow recording of meeting video to prevent high usage of data bandwidth.	It allows the host to record and save meetings. Recording and saving meetings will require additional use of data bandwidth since the meeting video has to be converted into a

		playable format like mp4 before they save it on the user’s computer or online storage.

Table 2.1: Comparing the proposed system with the existing system

2.5 CONCLUSION

In conclusion, the literature review brings to light an overview of the history and development of online classrooms and the contribution of people and organisations to online classrooms. The research team examines the existing system used by UPSA for online lecturing and compares the proposed system with it. The proposed system seeks to address the challenges of the existing system by developing a specific application system for online lecturing.

CHAPTER THREE

LIFE CYCLE DESIGN OF THE PROPOSED SYSTEM

3.1 INTRODUCTION

According to Berg, a methodology is the systematic and theoretical analysis of the methods applied to a field of study (Berg, 2009). From the definition, we can define methodology as the process involved in undertaking research. The methodology ensures that the research design is appropriate for the study. To develop the proposed system, we will use the Rapid Application Development method. The research team selected this method because it involves software users in all stages of application development and it is iterative (we can repeat a stage until we satisfy requirements).

3.2 CRYSTALLIZATION OF THE PROBLEM

The study being conducted is because of the following problems. One problem is that because of the COVID-19 pandemic, UPSA cannot conduct in-person lectures for all students at once. Hence the school has implemented 6 weeks virtual and 6 weeks in-person lecture plans for students. The problems with the virtual lecture are; most students cannot access lectures because they cannot install the Zoom app on their devices, Zoom does not record class attendance hence students do not attend class, Zoom consuming too much data bandwidth for online sessions.

The proposed system attempts to solve the aforementioned issues by developing an online lecture platform that is accessible, user-friendly, reliable, and efficient.

3.3 ANALYSIS AND DESIGN OF THE SYSTEM

3.3.1 System Requirement

System requirements are the arrangement that a system should have for a hardware or software application to run easily and proficiently. Inability to meet these prerequisites can bring about issues like performance issues.

3.3.2 Functional Requirement

Functional requirements of a system are the requirements that specify what the system should do when it meets certain conditions. A functional requirement is a description of the envisioned functions of a system and its components. For example, a functional requirement for the proposed system is it emails the host whenever he/she schedules a meeting successfully.

The list below is some types of functional requirements. They are:

- Interoperability–Checks whether the proposed system can work across different browsers
- Security–Checks the security aspects of the proposed system. For example, the system must ensure encrypted data before transferring it from client to server.
- Accuracy–Check whether inputs and outputs are correct. For example, the proposed system checks if the email provided is valid before it sends messages to it.
- Compliance–Checks whether the system conforms with accepted standards. For example, the system complies with data protection standards hence it stores user data on secured database systems.

3.3.3 Non-Functional Requirement

Non-functional requirements of a system are the requirements that describe how the system performs certain functions. The non-functional requirements are the quality attributes of a system. For example, a non-functional requirement for the proposed system is the website should not take over 3 seconds to load if the number of concurrent users is < 500.

The list below is some types of non-functional requirements. They are:

- Usability–Focus on whether the website is easy to navigate and the design of the user interface is well-structured.
- Performance–Measures how efficient the system is in performing its functions. For example, the broadcast must take less than 3 seconds to load for viewers to watch.
- Accessibility–Checks whether the system is accessible by all people, regardless of disability type or severity of impairment.
- Scalability–Focus on whether the system can handle needs as it grows.
- Reliability–Ensures the website is dependable under certain conditions.
- Maintainability–Focus on ensuring the system keeps functioning by applying updates and auditing the system.

3.3.4 Hardware Requirement

The hardware required to use the proposed system are:

- A laptop or desktop with a minimum of 2GB of memory, 10GB of storage space, and 1.20GHz of processor speed.
- A mobile phone or tablet with a minimum of 2GB of memory, 2GB of storage space, and 1.20GHz of processor speed.
- A web camera
- A microphone

3.3.5 Software Requirement

The software required to use the proposed system is a modern web browser that has JavaScript enabled. Google Chrome is the recommended browser. Example; Firefox, Chrome, Safari, Microsoft Edge, and Opera Mini

3.4 FLOW CHART DIAGRAM

A flowchart is a diagram that describes a system, computer algorithm, or process.

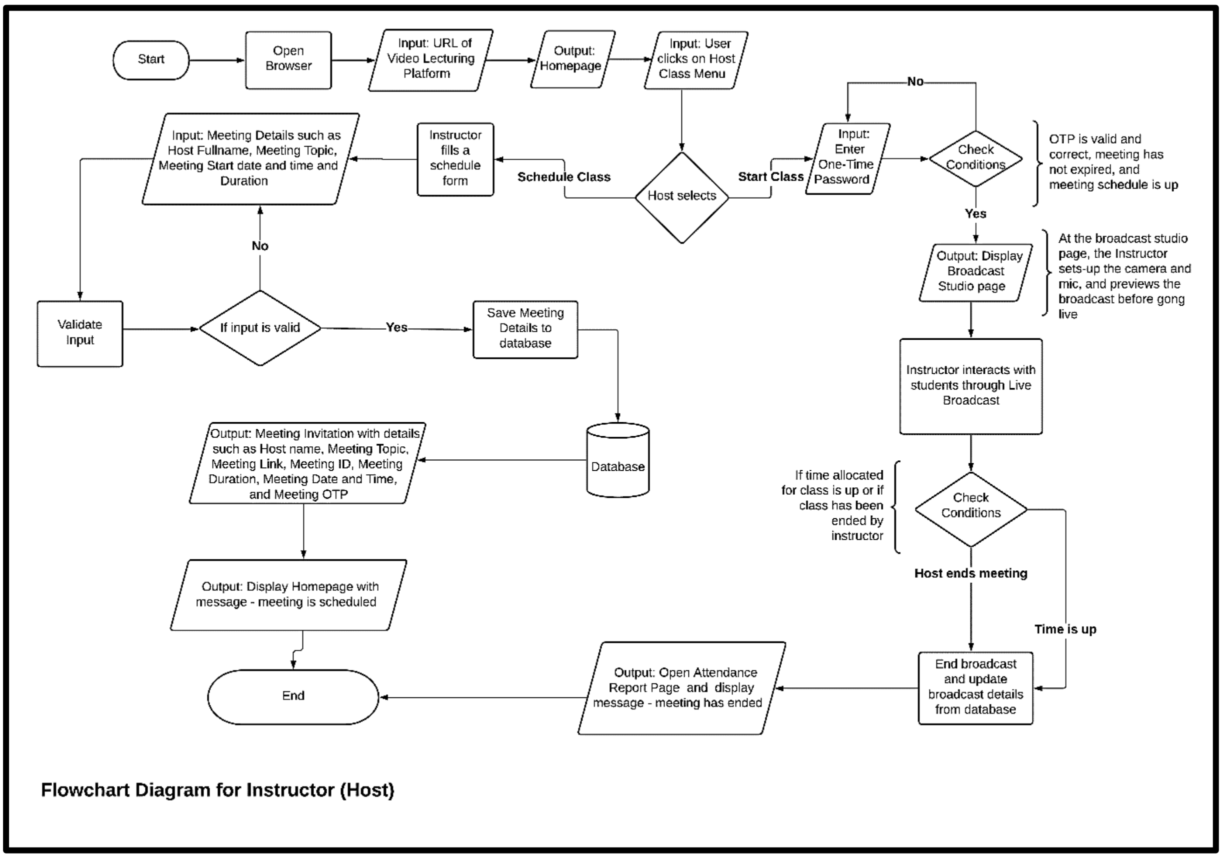

Start
Open Browser
Input: URL of Video Lecturing Platform
Output: Homepage
Input: User clicks on Host Class Menu
Host selects
Schedule Class
Start Class
Instructor fills a schedule form
Input: Meeting Details such as Host Fullname, Meeting Topic, Meeting Start date and time and Duration
Validate Input
If input is valid
No
Yes
Save Meeting Details to database
Database
Output: Meeting Invitation with details such as Host name, Meeting Topic, Meeting Link, Meeting ID, Meeting Duration, Meeting Date and Time, and Meeting OTP
Output: Display Homepage with message - meeting is scheduled
End
Input: Enter One-Time Password
Check Conditions
No
Yes
OTP is valid and correct, meeting has not expired, and meeting schedule is up
Output: Display Broadcast Studio page
At the broadcast studio page, the Instructor sets-up the camera and mic, and previews the broadcast before gong live
Instructor interacts with students through Live Broadcast
Check Conditions
If time allocated for class is up or if class has been ended by instructor
Host ends meeting
Time is up
End broadcast and update broadcast details from database
Output: Open Attendance Report Page and display message - meeting has ended
Flowchart Diagram for Instructor (Host)

Figure 3.1: Flowchart diagram for a host

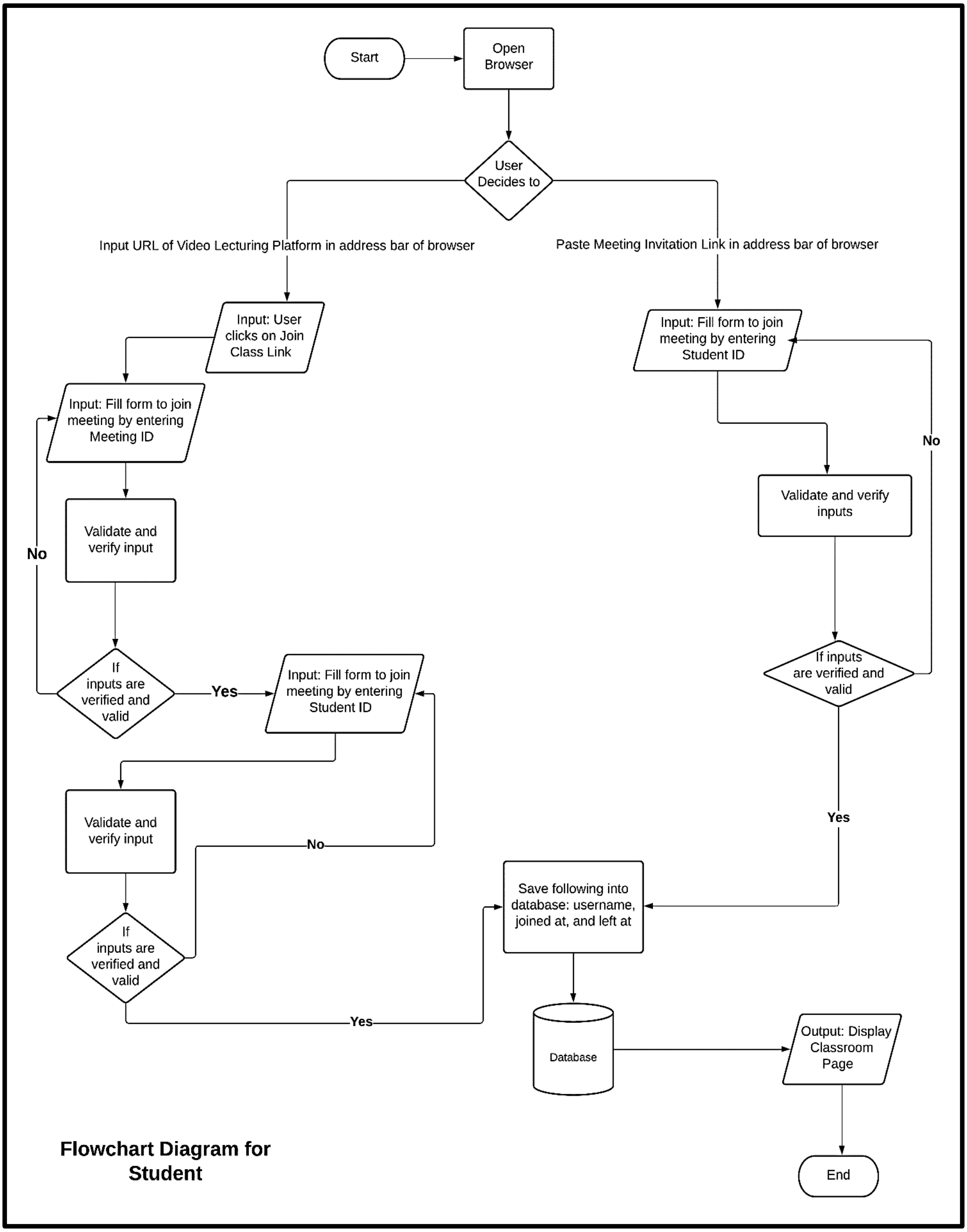

Start
Open Browser
User Decides to
Input URL of Video Lecturing Platform in address bar of browser
Paste Meeting Invitation Link in address bar of browser
Input: User clicks on Join Class Link
Input: Fill form to join meeting by entering Meeting ID
Validate and verify input
No
If inputs are verified and valid
Yes
Input: Fill form to join meeting by entering Student ID
Validate and verify input
No
If inputs are verified and valid
Yes
Input: Fill form to join meeting by entering Student ID
Validate and verify inputs
No
If inputs are verified and valid
Yes
Save following into database: username, joined at, and left at
Database
Output: Display Classroom Page
End
Flowchart Diagram for Student

Figure 3.2: Flowchart diagram for student

3.4.1 Context Diagram

A context diagram is a high-level view of a system that defines the boundary between the system and its environment, by showing the entities that interact with the system.

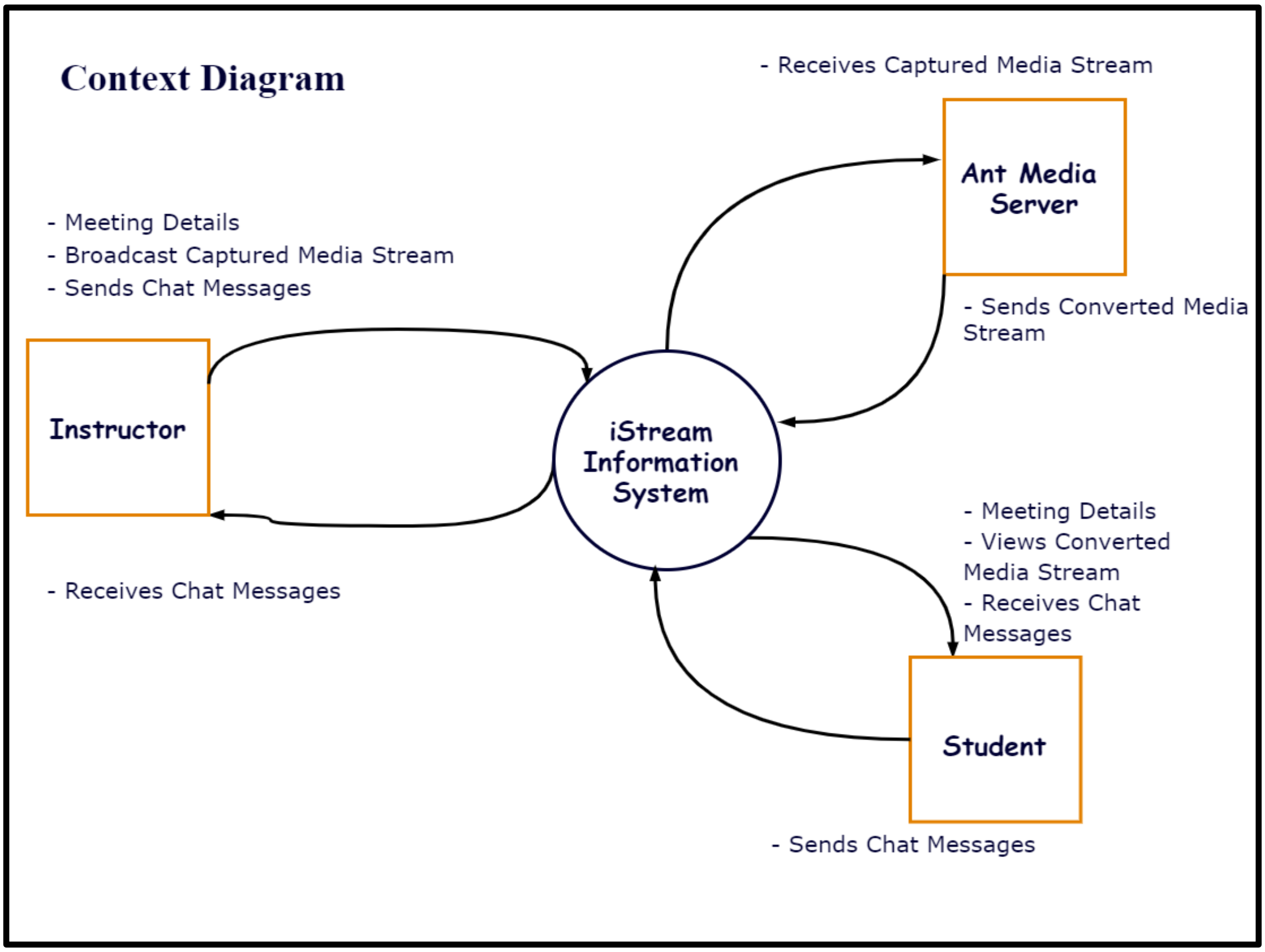
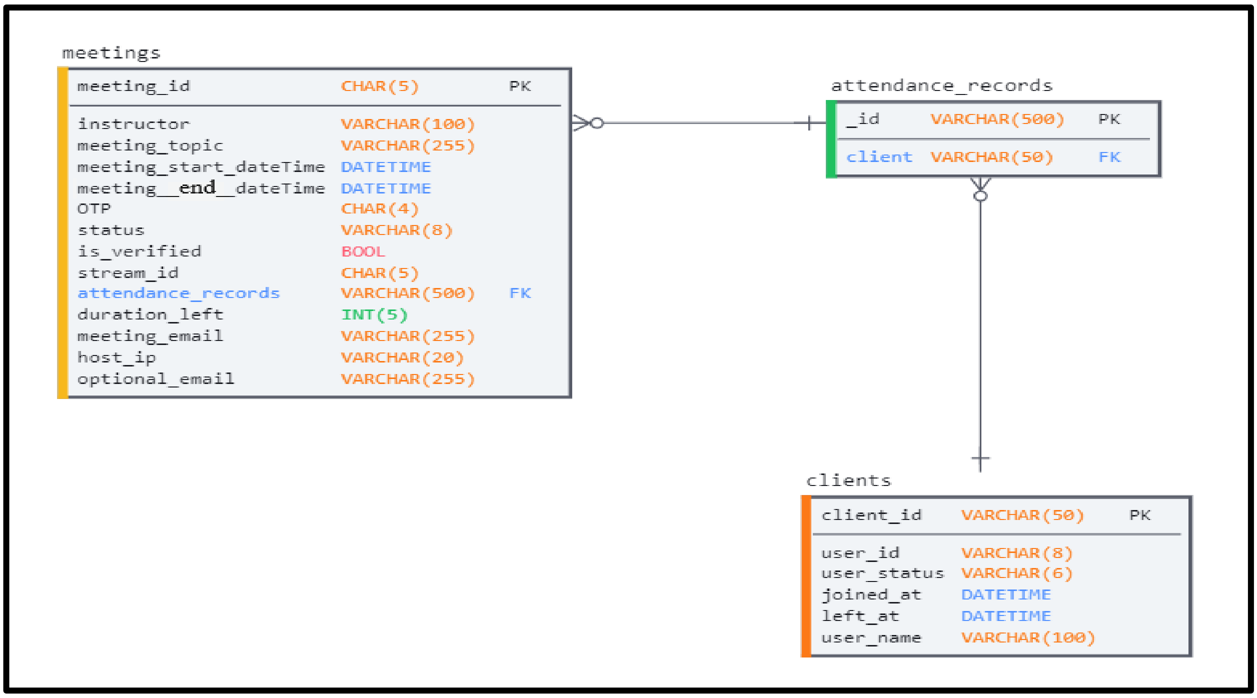


Figure 3.3: Context diagram illustrating boundaries between entities

3.4.2 Entity-Relationship Diagram

An entity-relationship diagram is a diagram that shows the relationship of entity sets stored in a database.



meetings

meeting_id	CHAR(5)	PK
instructor	VARCHAR(100)	
meeting_topic	VARCHAR(255)	
meeting_start_dateTime	DATETIME	
meeting__end__dateTime	DATETIME	
OTP	CHAR(4)	
status	VARCHAR(8)	
is_verified	BOOL	
stream_id	CHAR(5)	
attendance_records	VARCHAR(500)	FK
duration_left	INT(5)	
meeting_email	VARCHAR(255)	
host_ip	VARCHAR(20)	
optional_email	VARCHAR(255)	

attendance_records

_id	VARCHAR(500)	PK
client	VARCHAR(50)	FK

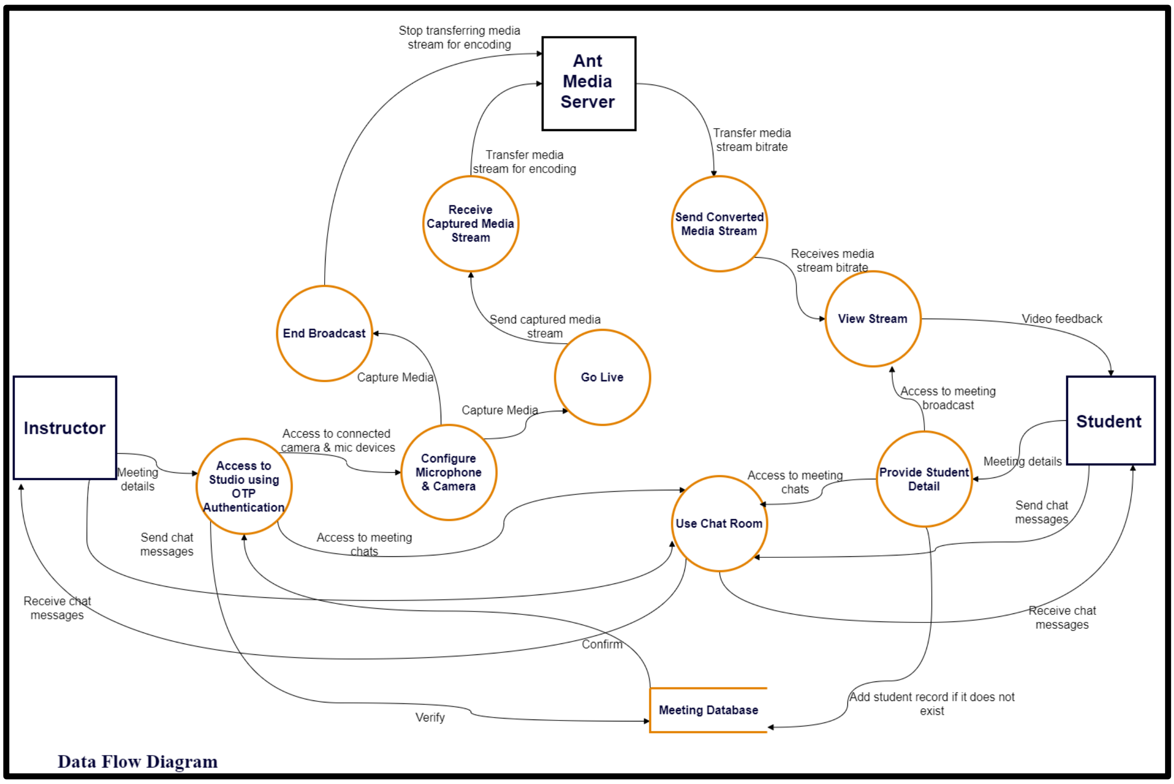
clients

client_id	VARCHAR(50)	PK
user_id	VARCHAR(8)	
user_status	VARCHAR(6)	
joined_at	DATETIME	
left_at	DATETIME	
user_name	VARCHAR(100)	

Figure 3.4: Entity-Relationship Diagram for the proposed system

3.4.3 Data Flow Diagram

A data flow diagram illustrates how data flows through a system.



Stop transferring media stream for encoding
Ant Media Server
Transfer media stream for encoding
Transfer media stream bitrate
Receive Captured Media Stream
Send Converted Media Stream
Receives media stream bitrate
End Broadcast
Send captured media stream
View Stream
Video feedback
Go Live
Capture Media
Capture Media
Access to meeting broadcast
Instructor
Student
Access to connected camera & mic devices
Meeting details
Access to Studio using OTP Authentication
Configure Microphone & Camera
Access to meeting chats
Provide Student Detail
Meeting details
Use Chat Room
Send chat messages
Access to meeting chats
Send chat messages
Receive chat messages
Receive chat messages
Confirm
Add student record if it does not exist
Verify
Meeting Database
Data Flow Diagram

Figure 3.5: Data Flow Diagram for the proposed system

3.4.4 Use Case Diagram

A use case diagram is a graphical representation of a user's actions with a system.

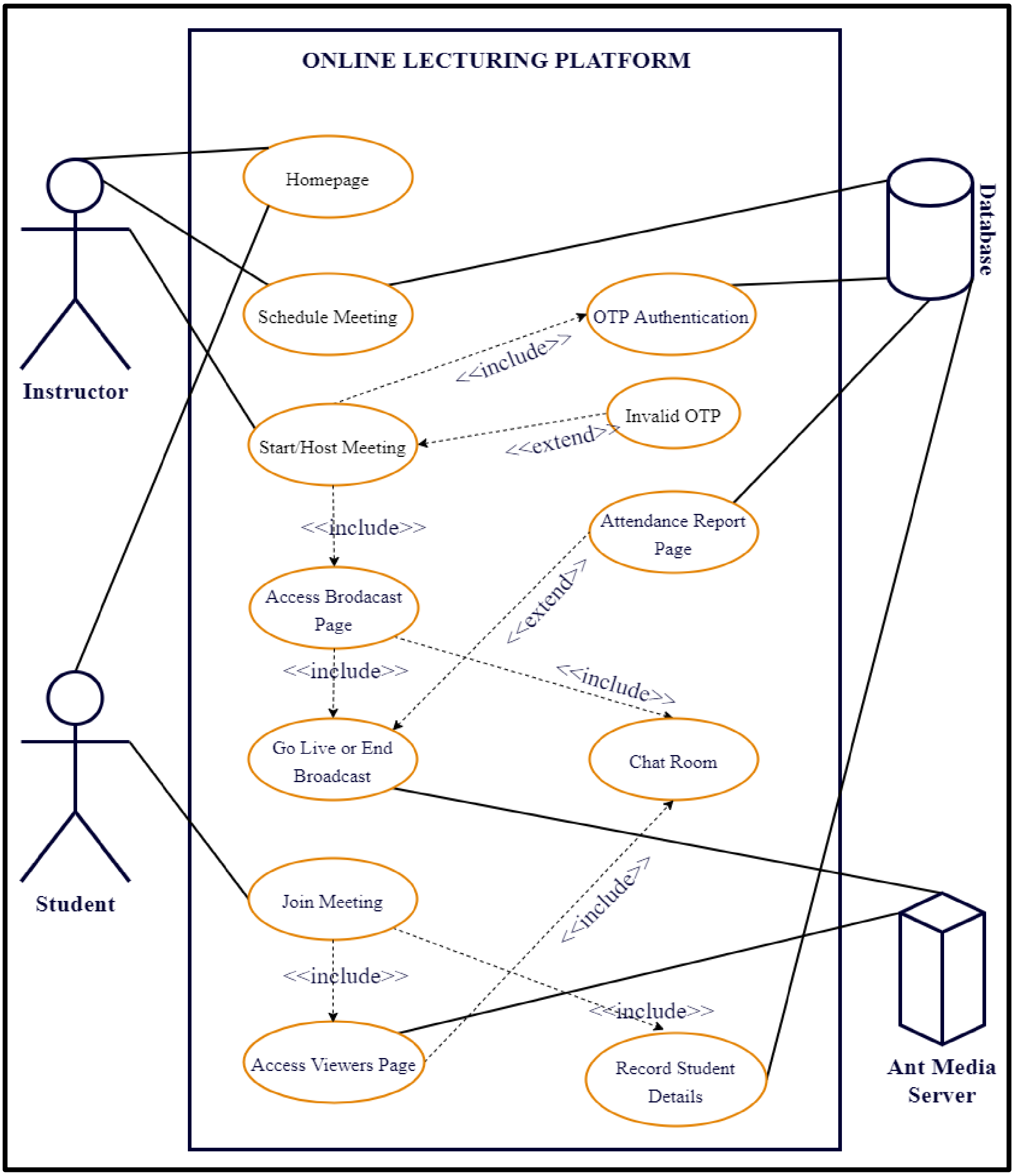


Figure 3.6: Diagram represents how system actors interact with the system

3.5 TOOLS USED

The system comprises the front-end and back-end. The front-end is the part of the system users can view, access, and interact with. The back-end is the part of the system that users cannot access but performs the behind-the-scenes activities needed for a system to accomplish its purpose.

The research team selected and used the tools stated below because they help develop systems quickly, a large community of developers supports the improvement of these tools, and frequent system checks and updates to fix security or bug issues. The team categorised the tools into two (4) groups and they are:

BACK-END TOOLS

- Python–The team used the Python programming language for developing the server-side logic of the system. Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built-in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components. Guido van Rossum created Python and the Python Software Foundation maintains Python.
- MongoDB–The database used by the research team for storing data is MongoDB. MongoDB is a source-available cross-platform document-oriented database program. Classified as a NoSQL database program, MongoDB stores data in JSON-like documents. MongoDB Inc. developed MongoDB and licensed it under the Server-Side Public License. The team selected MongoDB because it can store and manipulate large amounts of data, provides high scalability, and enables good productivity in application development.
- Flask Framework–Flask is a micro web framework written in Python. The developer community classified it as a micro framework because it does not require particular tools or libraries. However, Flask supports extensions that can add application features. Armin

Ronacher created Flask. The team used the Flask framework because it provides well-written documentation and it gives developers more freedom and control.

- Ant Media Server–The team used the Ant Media Server as the streaming engine because it is scalable, built with modern streaming protocols and techniques, and has a well-documented developer API. Ant Media Server is a streaming engine software that provides adaptive, ultra-low latency streaming by using WebRTC technology with ~0.5 seconds latency. It can enable streaming any type of live or on-demand video to any device including mobiles, computers or IPTV boxes. It supports RTMP, MP4, HLS, RTSP and WebRTC.
- Google Cloud Platform (GCP)–Google Cloud Platform offered by Google, is a suite of cloud computing services that runs on the same infrastructure that Google uses internally for its end-user products, such as Google Search, Gmail, and YouTube. Combined with a set of management tools, it provides a series of modular cloud services including computing, data storage, data analytics and machine learning. Google Cloud Platform provides infrastructure as a service, platform as a service, and serverless computing environments. The team selected the Compute Engine service to run a server instance for hosting the proposed system. The team used the GCP because it can easily scale, you can make provision for cloud resources whenever needed automatically, it is cost-effective, and you pay for what you use.

FRONT-END TOOLS

- HyperText Markup Language (HTML)–The team used HTML as the markup language for the web pages of the system. HTML is the standard markup language for documents designed to be displayed in a web browser. Invented by Tim Berners-Lee.
- Cascading Style Sheets (CSS)–CSS is a language for specifying how browsers present documents to users — how we style it, layout, etc. A document is usually a text file structured using a markup language — HTML is the most common markup language, but you can use it

with other markup languages such as SVG or XML. Developed by World Wide Web Consortium (W3C).

- JavaScript–JavaScript is a high-level, often just-in-time compiled programming language that conforms to the ECMAScript specification and is used to make web pages interactive. The team used JavaScript because it supports asynchronous programming, and has modern APIs like the Media Capture API used for capturing video and screen sharing.
- Bootstrap Framework–The team used Bootstrap because it improves system development productivity, and makes system design responsive to different devices. Bootstrap is a free and open-source CSS framework directed at responsive, mobile-first front-end web development. It contains CSS- and JavaScript-based design templates for typography, forms, buttons, navigation, and other interface components. Developed by the Bootstrap Core Team and licensed under the MIT License.

INTEGRATED DEVELOPMENT ENVIRONMENT (IDE)

- PyCharm - PyCharm is an IDE by JetBrains for computer programming in Python. It provides code analysis, a graphical debugger, an integrated unit tester, integration with version control systems, and supports web development.

VERSION CONTROL SYSTEM

- Git–Git is a free and open-source distributed version control system designed to handle everything from small to very large projects with speed and efficiency. Git tracks the changes you make to files, so you have a record of what you have done, and you can revert to specific versions should you ever need to. Git also makes collaboration easier, allowing changes by multiple people to a project which the team can merge into one source. Linus Torvalds developed Git and Junio Hamano is a major maintainer of Git. The team used Git to manage the different development versions of the system until it built the ultimate system.

- GitHub–GitHub is a cloud-based Git repository hosting service provided by GitHub Incorporation. It provides access control and several collaboration features such as bug tracking, feature requests, task management, continuous integration and wikis for every project. The team used GitHub to store source codes of the system remotely. This helped the team to access the source codes anywhere and collaborate.

3.6 CONCLUSION

The chapter discusses in detail the problems that need a solution and the methodology used by the research team which is the Rapid Application Development method for system development. The team discusses system requirements, functional requirements, non-functional requirements, and tools used for the system. Also, the chapter discusses the various diagrams like use case diagrams, flowchart diagrams, etc that describes what the system should do and how it should perform certain functions.

CHAPTER FOUR

SYSTEM TESTING, IMPLEMENTATION AND DOCUMENTATION

4.1 INTRODUCTION

In software development, the SDLC at each crucial stage produces a deliverable. A deliverable is a written document, a software artefact, a system test plan or a system implementation plan. This feature of the SDLC is crucial to the successful management of a system.

The research team in this chapter will discuss the following plans for the proposed system. The plans are:

- System Testing–testing conducted on a complete integrated system to evaluate the system's compliance with its specified requirements (Wikipedia, n.d.).
- System Implementation–ensuring that the information system is operational (Tutorialspoint, n.d.).
- Documentation–detailed information, in either written or computerised form, about a system, including its requirements, design, limitation, operation, capabilities, and maintenance.

4.2 TESTING OF THE NEW SYSTEM

System testing attempts to detect faults both within the "inter-systems" and also within the entire system. The software team performs system testing on the entire system to measure either system requirement specification (SRS) or functional requirement specifications (FRS), or both. Also, they plan to test up to and past the limits defined in the hardware or software requirements specifications.

The research team started the various testing on the proposed system to ensure its functionality and implementation.

4.2.1 Unit Testing

Unit testing is an initial stage product testing technique where the team tests small units or components of the products individually (Ankit, 2021). This means software teams perform unit testing to validate the quality of the code, code coverage, execution and maintenance of coding standards, confirmation of the practicality covered by the part of code.

The research team performed unit testing by writing test cases to test automatically, sections of the source code to ensure it meets its design and behaves as intended. The team tested the web pages to ensure performance and accessibility. With the feedback given by unit testing, the team makes corrections to ensure the unit tested functions as intended.

4.2.2 Functional Testing

Functional testing is a quality assurance process and a type of black-box testing that bases its test cases on the specifications of the software component under test (Wikipedia, 2021). In functional testing, quality assurance personnel determine if a system or component is acting under predetermined requirements. It uses black-box testing procedures, in which the tester does not know of the internal system logic.

The team performed functional testing by applying these steps:

- The team identified functions we projected the system to perform
- The team created input data based on the identified functions specifications
- The team determined output based on the identified functions specifications
- The team implemented the test cases
- The team compared the projected outputs with the actual outputs to ensure the system works as per the customer requirement.

4.2.3 Usability Testing

Usability testing, a non-functional testing technique that checks how users can use the system. It can determine the design intuitiveness of the system tested with users who have no prior exposure to the system. Usability testing such as navigating a website gives input on how actual users use the system.

The research team performed a usability test to ensure the system is easy to use, is user friendly, and is efficient. The team performed the following examples of usability testing metrics:

- The percentage of the users that could complete a task.
- The average time it takes for users to complete a task.
- The percentage of users who experienced errors in the system.
- The average number of screens, clicks, or steps it takes for users to complete a task.
- The proportion of users who faced a specific error in the system.
- The mean score users gave to questions about the system's ease of use, user-friendliness, and efficiency.

4.2.4 Acceptance Testing

Acceptance testing evaluates a product, prototype, or software application to make sure it is up to specifications and quality before putting into production (Hayes, 2020). The key role of this test is to assess the system's consistency with the business necessities and confirm on the off chance that it has met the rules for delivery to users.

Types of acceptance testing include:

- User acceptance testing
- Business acceptance testing
- Alpha and Beta testing

4.2.5 Visual (live) Testing

Visual testing process of comparing user interface of an app or website against a user interface design blueprint. The major role of visual testing is to ensure the system user interface is having all the visual specifications provided by the client as a part of the functional or non-functional requirements. Software teams use a visual testing tool like Applitools to aid them in performing the visual test.

4.3 IMPLEMENTATION OF THE NEW SYSTEM

Systems implementation is the delivery of a system into production. It is making the system go live for users to use it. A software team can implement a system using any of the following approaches.

4.3.1 Parallel Implementation

In the parallel implementation, the team executes the new system with the old system still running. This approach can compare output from both systems and there is less risk in data loss but data entry will be time-consuming.

4.3.2 Phased Implementation

In a phased implementation, small parts of the new system gradually replace small parts of the old system. This approach has a minimal impact on the organisation if a failure occurs but executing the entire new system takes more time.

4.3.3 Pilot Implementation

In a pilot implementation, a small group of users within an organisation uses the new system before the entire organisation implements it. This approach helps in identifying and addressing system failures without a widespread impact on the organisation but has issues with scaling to meet user demands.

4.3.4 Direct Implementation

In the direct implementation, the team retires the old system and implements the new system. This approach is suitable for non-critical systems but it is not suitable for a system that has uncertainties.

The research team applied the pilot implementation approach because it is suitable for implementing systems that have uncertainties such as issues with scalability. Also, the team executes the system in a realistic setting and can learn about whether it satisfies the specifications. The team can also identify and address failures before the entire organisation uses the system.

4.4 SYSTEM DOCUMENTATION

4.4.1 About the System

The proposed system known as iStream is a web-based lecturing platform designed to facilitate online lectures between teachers and students of UPSA. The system allows users with unstable internet access, low-end computer or mobile devices, and less data bandwidth to take part in online lectures. The system has the following features:

- Built with modern web and streaming technologies that enhance performance, functionality, usability, accessibility, reliability, and less data consumption.
- Teachers can schedule and host lecture meetings.
- Students can join lecture meetings.
- A chat room for teachers and students to interact during a meeting.
- Teachers can print attendance reports to check students who joined the meeting.
- A user-friendly and well-designed user interface to ensure system ease of use.
- A documentation page to train the user on how to use the system.
- A contact page for users to get in touch with the system support team.

4.4.2 User Access Level

The user access level is the rights or permissions assigned to the user of a system.

The teachers and students of UPSA are the projected users of the proposed system. The access rights given to teachers and students are:

Teachers

- can schedule an unlimited number of meetings
- can host a meeting at a time
- can print meeting attendance reports if they have access
- can only interact in meetings they take part in
- can schedule a meeting with a minimum duration of 1 hour and maximum duration of 3 hours

Students

- can join a meeting once
- can join multiple meetings
- can only interact in meetings they take part in

4.4.3 Getting Started

The entire procedure of using the proposed system will involve:

Teacher

- Schedule Meeting
  - Connect computer or mobile device to an internet connection.
  - Open a web browser and enter the web address of the online lecturing platform.
  - Navigate to the scheduling meeting section.
  - Fill and submit the form for scheduling meetings.
  - Copy and share meeting details with students.
- Host Meeting
  - Connect computer or mobile device to an internet connection.
  - Open a web browser and enter the web address of the online lecturing platform.
  - Navigate to the host meeting page by clicking on the host meeting link.

- Login to the studio page by entering the meeting one-time password.
- Grant permission for the website to access the camera and microphone.
- Perform the settings.
- Click on the Go Live button to broadcast.
- Send chat messages or questions through the chat room.
- Click on the End Broadcast button to end broadcasting when finished lecturing.
- Click on the Leave button to exit the studio page.
- View and print attendance reports.

- Get Attendance Report

- Connect computer or mobile device to an internet connection.
- Open a web browser and enter the web address of the online lecturing platform.
- Navigate to the attendance page by clicking on the get attendance link.
- Fill in the attendance form and submit it.
- Download the attendance report by clicking on the link provided.

<u>Student</u>

- Connect computer or mobile device to an internet connection.
- Open web browser
  - Enter the web address of the online lecturing platform or
    - Navigate to the join meeting page by clicking on the join meeting link.
    - Enter meeting id and username to access the viewer page
  - Enter the invitation link received from the teacher
    - Enter a username to access the viewer page
- View broadcast.
- Send chat messages or questions through the chat room.

- Click on the Leave button to exit the viewer page.

CHAPTER FIVE

CONCLUSION AND RECOMMENDATIONS

5.1 INTRODUCTION

Education is vital to the growth of an economy and conducting teaching and learning online is also important although the world is battling with a pandemic (i.e., COVID-19). Adapting to alternative ways of acquiring knowledge will not only help to prevent a pandemic but will improve our education system. This chapter discusses the knowledge gained while undertaking this project work and recommendations for future research works.

5.2 SUMMARY

Computer engineers have made massive improvements and advancements in using information technology to enhance teaching and learning. Applying these enhancements in the tertiary sector can improve the acquisition of knowledge by students and also help students in learning how to collaborate to be productive in the actual world.

At the end of the study, the team noticed the strengths and shortcomings of our proposed system. The major factors that played a vital role in creating a system that meets the objectives and targets of the users include good communication, teamwork, and time management.

The developed system has space for upgrades to enable create an ideal and flawless system for UPSA.

5.3 RECOMMENDATION

Education is key in our lives for as long as we want to gain knowledge to help us become learned people and also help make a difference in the world. The team makes the following recommendations for future works and researchers who intend to explore more in using information and technology to contribute to the topic under discussion. The recommendations are:

- The team suggests the consistent use of the platform for online lecturing to help identify and address failures.
- Future works can add features like adding scheduled events to the calendar, whiteboards, waiting room, recording, and conference room meetings.
- Future works can also upgrade the website into progressive web apps.
- Improving the system's capabilities for more universities to use it.
- Future works can integrate a well-developed help-desk support system to improve user support.

5.4 CONCLUSION

The document details a research work on developing an online video lecturing platform for teaching and learning at UPSA. When the school implements the system, it will make online lectures accessible to every student. The system will reduce the consumption of data bandwidth and improve ease of use for lecturers and students. Lecturers will not have to worry about how to record attendance since the system records class attendance automatically.

To conclude, our study aims to make online lecturing easier for lecturers and students with its user-friendly and less costly internet bundle.

REFERENCES

Aachri, T. (2021, March 30). *Does Zoom use a lot of data?* Quora. https://www.quora.com/Does-Zoom-use-a-lot-of-data

Adaptive Bitrate Streaming (ABR) | Video Streaming Definition. (n.d.). Retrieved January 14, 2021, from https://www.haivision.com/resources/streaming-video-definitions/adaptive-bitrate-abr-streaming/

Adaptive Bitrate Vs Multi-Bitrate Streaming - What's the Difference? - Muvi. (n.d.). Retrieved January 14, 2021, from https://www.muvi.com/blogs/adaptive-bitrate-vs-multi-bitrate-streaming.html

Ankit, M. (2021, March 17). *What Is Unit Testing? – Meaning, Process, & Tools.* Feedough. https://www.feedough.com/unit-testing/

Ant Media. (n.d.). *Ultra-Low Latency WebRTC Live Streaming - Open-Source Media Server.* Retrieved May 14, 2021, from https://antmedia.io/

Berg, B. L. (2009). *Qualitative Research Methods for the Social Sciences* (Seventh Ed). Pearson Education Inc.

Graves, M. (2018, October 8). WebRTC vs Zoom. Who has Better Video Quality? • BlogGeek.me. https://bloggeek.me/webrtc-vs-zoom-video-quality/

Hayes, A. (2020, December 18). *Acceptance Testing.* Investopedia. https://www.investopedia.com/terms/a/acceptance-testing.asp

How the Web works - Learn web development | MDN. (n.d.). Retrieved March 3, 2021, from https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/How_the_Web_works

Knight, M. (2018, January 1). What Is Database Management? - DATAVERSITY. DATAVERSITY. https://www.dataversity.net/what-is-database-management/#

Peterson, R. (2021, August 30). What is DBMS (Database Management System)? Application, Types & Example. Guru99. https://www.guru99.com/what-is-dbms.html

REST Guide · ant-media/Ant-Media-Server Wiki. (n.d.). Retrieved January 14, 2021, from https://github.com/ant-media/Ant-Media-Server/wiki/REST-Guide

Streaming Protocols: Everything You Need to Know | Wowza. (n.d.). Retrieved January 17, 2021, from https://www.wowza.com/blog/streaming-protocols

Tillman, M. (2021, February 15). *What is Zoom and how does it work? Plus, tips and tricks.* Pocket-Lint. https://www.pocket-lint.com/apps/news/151426-what-is-zoom-and-how-does-it-work-plus-tips-and-tricks

Tutorialspoint. (n.d.). *System Implementation and Maintenance.* Retrieved May 13, 2021, from https://www.tutorialspoint.com/system_analysis_and_design/system_analysis_and_design_implementation_maintenance.htm

Types of database system | nibusinessinfo.co.uk. (n.d.). NIBUSINESS. Retrieved August 31, 2021, from https://www.nibusinessinfo.co.uk/content/types-database-system

Virtual Classroom Software to deliver live teaching and training | WizIQ. (n.d.). Retrieved February 21, 2021, from https://www.wiziq.com/virtual-classroom/

WebRTC API - Web APIs | MDN. (n.d.). Retrieved January 15, 2021, from https://developer.mozilla.org/en-US/docs/Web/API/WebRTC_API

What Is a Database? | Oracle. (n.d.). Retrieved August 31, 2021, from https://www.oracle.com/database/what-is-database/

What is streaming? | How video streaming works | Cloudflare. (n.d.). Retrieved January 17, 2021, from https://www.cloudflare.com/learning/performance/what-is-streaming/

What is Video Streaming and How does it Work? | Akamai. (n.d.). Retrieved January 17, 2021, from https://www.akamai.com/uk/en/resources/video-streaming-services.jsp

Wikipedia. (2021, May 12). *Functional testing.* Wikipedia. https://en.wikipedia.org/wiki/Functional_testing

Wikipedia. (2021, May 12). *Zoom Video Communications.* Wikipedia. https://en.wikipedia.org/wiki/Zoom_Video_Communications

Wikipedia. (n.d.). *System testing.* Retrieved May 13, 2021, from https://en.wikipedia.org/wiki/System_testing

Zoom Help Center. (2021, April 20). System requirements for Windows, macOS, and Linux – Zoom Help Center. Zoom. https://support.zoom.us/hc/en-us/articles/201362023-System-requirements-for-Windows-macOS-and-Linux

APPENDIX

APPENDIX A (Programming Codes)

Contact Page codes

Backend

Python

```
# Route for contact us

@application.route('/contact_us', methods=["GET", "POST"])

def contact():

    if request.method == "POST":

        u_name = request.form.get("c_name")

        u_email = request.form.get("c_email")

        u_subject = request.form.get("c_subj")

        u_msg = request.form.get("c_msg")

        result = validate_email(u_email)

        if None in [u_name, u_subject, u_msg, result]:

            flash(message="Message not sent. Please check your input and resubmit",
category="danger")

            return redirect(url_for("contact"))

        email_data = dict(user=u_name, topic=u_subject, email=u_email, message=u_msg)

        contact_us_task.apply_async(args=[email_data], countdown=60)
        flash(message="Message sent", category="success")

        return redirect(url_for("contact")), 301

    return render_template('contact.html')
```

Frontend

HTML 5

```
<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">



    <link rel="stylesheet" href="{{ url_for('static', filename='contact.css') }}">

    <link rel="stylesheet" href="{{ url_for('static', filename='icomoon.css') }}">

    <link rel="stylesheet" href="{{ url_for('static', filename='css/bootstrap.min.css') }}">

    <title>Contact Us | iSTREAM</title>

    <style>

        @import url('https://fonts.googleapis.com/css2?family=Satisfy&display=swap');

    </style>

</head>
```

```
<body>
    <div class="main">
        <nav class="navbar navbar-expand-lg app-nav">
            <div class="container">
                <h2><b><a class="navbar-brand" href="#"
                            style="font-family: 'Satisfy', cursive; letter-spacing: 3px; text-decoration: none; color: #fff; font-size: 24pt;">iSTREAM</a></b></h2>
                <button class="navbar-toggler" type="button" data-bs-toggle="collapse"
                    data-bs-target="#navbarSupportedContent" aria-controls="navbarSupportedContent"
                    aria-expanded="false" aria-label="Toggle navigation">
                    <span class="icon-menu"></span>
                </button>
                <div class="collapse navbar-collapse" id="navbarSupportedContent" style="flex-grow: 0;">
                    <ul class="navbar-nav nav-menu mr-auto mb-2 mb-lg-0">
                        <li class="nav-item">
                            <a class="nav-link active" aria-current="page" href="/">Welcome</a>
                        </li>
                        <li class="nav-item">
                            <a class="nav-link" href="/join_meeting/v/host_verify">Host Class</a>
                        </li>
                        <li class="nav-item">
                            <a class="nav-link" href="/join_meeting/v/user_verify">Join Class</a>
                        </li>
                        <li class="nav-item">
                            <a class="nav-link" href="/search_attendance_report">Attendance</a>
                        </li>
                        <li class="nav-item">
                            <a class="nav-link" href="/contact_us">Contact Us</a>
                        </li>
                    </ul>
                </div>
            </div>
        </nav>


        <div class="p-3 verify_box">


            <div class="mb-3">
                <h3 class="main-title text-center">Contact Form</h3>
            </div>
```

```
            <div>
                {% include 'includes/_message.html' %}
                <form action="" method="POST" class="mt-2">
                    <div>
                        <div class="mb-2">
                            <label for="inputName" class="form-label">Name</label>
                            <input type="text" name="c_name" autocomplete="off" class="form-control"
id="inputName"
                              minlength="2" maxlength="50" placeholder="e.g. Karen Doe" required>
                        </div>
                        <div class="mb-2">
                            <label for="inputEmail" class="form-label">Email</label>
                            <div class="input-group">
                                <input type="email" name="c_email" autocomplete="off" class="form-control"
inputmode="email"
                                id="inputEmail" pattern="^([\w\d\-\.]+?)@([a-z]+?).(com|org|net|edu)(.[a-zA-
Z]{2})??$"
                                maxlength="400" placeholder="e.g. example@domain.com" aria-
describedby="inputValidate" required>
                                <span class="validate input-group-text border-start-0" id="inputValidate"
                style="background-color: transparent; margin-left: 0px; padding-left: 5px;
padding-right: 24px;"></span>
              </div>
            </div>
            <div class="mb-2">
              <label for="inputSubject" class="form-label">Subject</label>
              <input type="text" name="c_subj" autocomplete="off" class="form-control"
id="inputSubject"
                minlength="5" maxlength="100" placeholder="e.g. How to Join Class"
required>
            </div>
            <div class="mb-2">
              <label for="inputMessage" class="form-label">Message</label>
              <textarea name="c_msg" id="inputMessage" class="form-control" rows="3"
minlength="10" maxlength="300"
              placeholder="Write message here" required></textarea>
            </div>
          </div>
          <div class="text-center">
            <button type="submit" class="btn btn-stream">Submit</button>
          </div>
```

```
                </form>
            </div>
        </div>
    </div>
    <script src="{{ url_for('static', filename='js/bootstrap.min.js') }}"></script>
    <script src="{{ url_for('static', filename='js/sub_functions.js') }}"></script>
</body>
</html>
```

CSS 3

```
@import url('https://fonts.googleapis.com/css2?family=Karla:ital,wght@0,400;0,700;1,400;1,700&display=swap');
@import url('/static/subfunctions.css');
:root {
    --primary-color: #000033;
    --secondary-color: #f18900;
}
html,
body {
    height: 100vh;
```

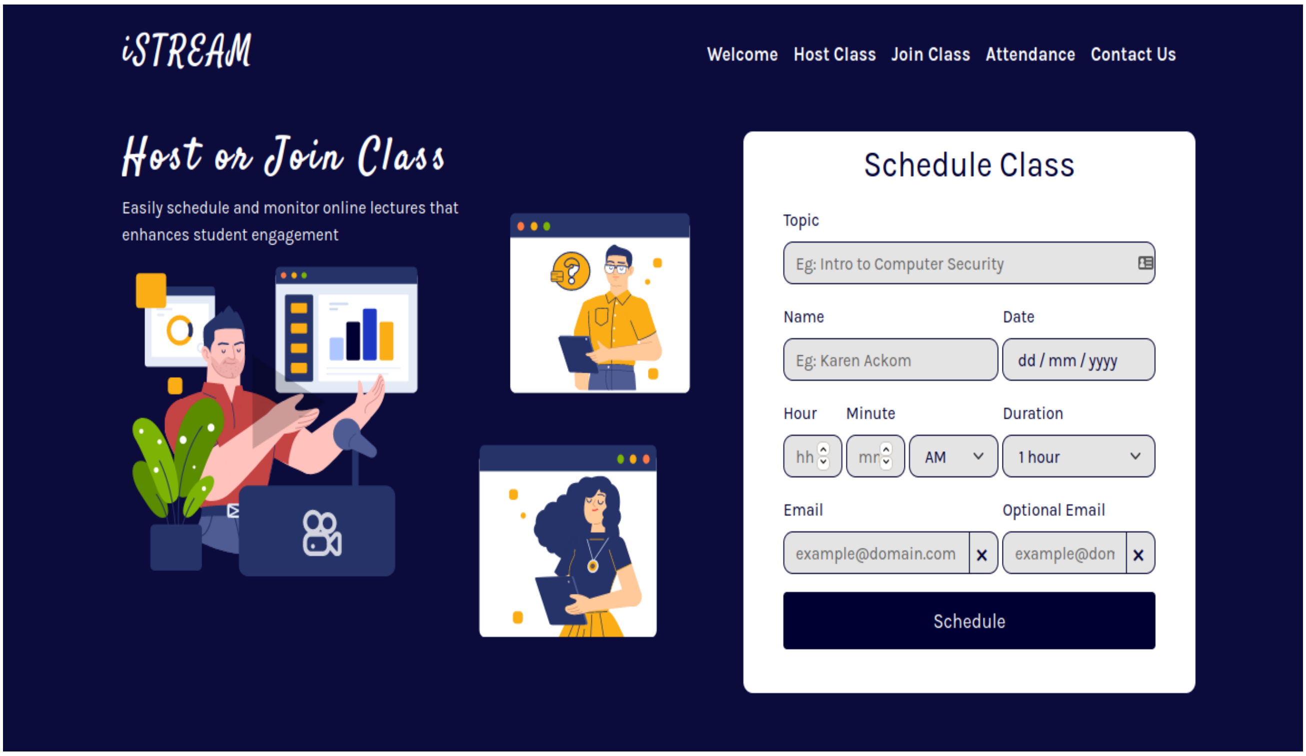
```
            overflow: hidden;

    background-color: #000033c9 !important;

}

* {

            box-sizing: border-box;

            margin: 0;

            padding: 0;

}

body {

    font-family: 'Karla', sans-serif !important;

}

.app-nav{

    background-color: transparent;

    color: #fff;

    padding: 20px 0 !important;

}

.nav-menu{

    text-align: right;

    font-weight: bold;

}
```

```
.nav-link{
   color: #fff !important;
   font-size: 13pt;
}
.main-title{
   font-weight: bolder;
}
.text-bold{
   font-weight: bold;
}
.btn-stream{
   background-color: transparent !important;
   color: #fff !important;
   font-size: 12pt !important;
   font-weight: bolder !important;
   border: 2px solid var(--secondary-color) !important;
}
input, textarea{
   background-color: transparent !important;
   color: #fff !important;
```

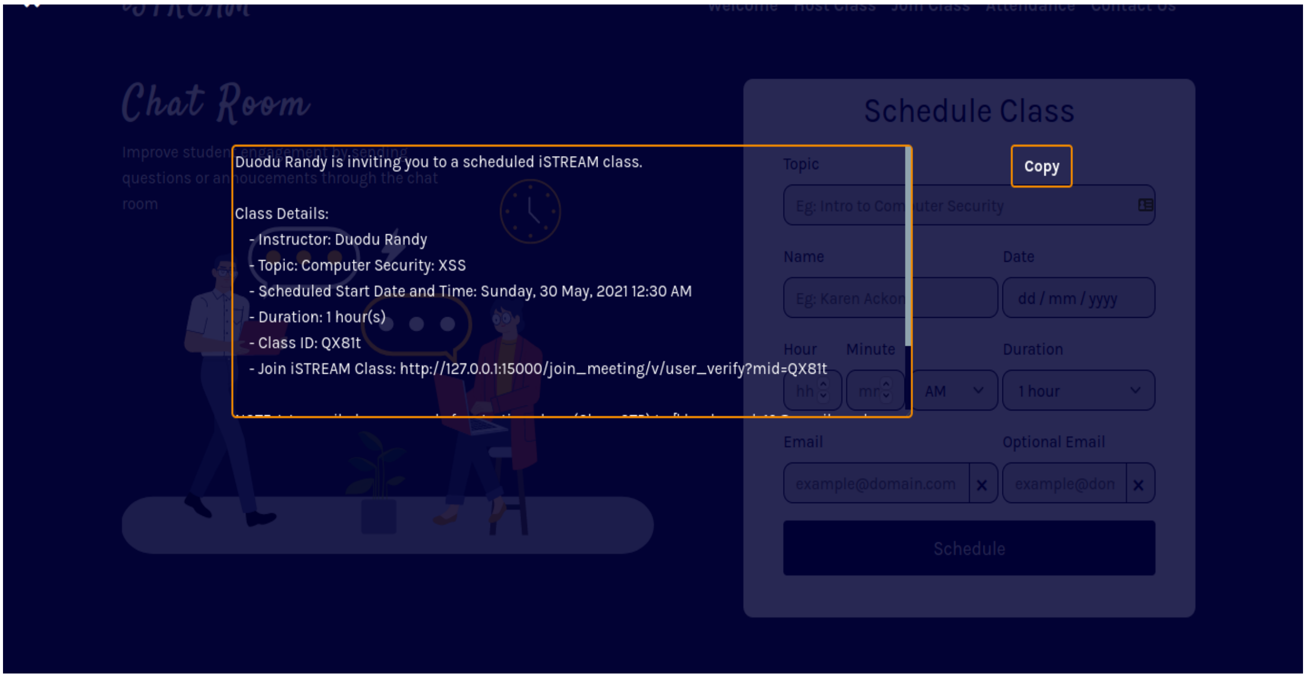
```
    border: 1px solid var(--secondary-color) !important;
}
input[type="text"], textarea, .info-success,
.info-normal, .info-danger, .input-group{
    width: 75% !important;
    margin: 0 auto !important;
    font-weight: bold !important;
}
input:focus, textarea:focus{
    background-color: var(--primary-color) !important;
    color: #fff !important;
    border: 1px solid #fff !important;
    font-weight: bold;
}
input::placeholder, textarea::placeholder{
    color: #ffffffc4 !important;
}
label{
    font-weight: bold;
    text-align: left!important;
    margin-left: 85px;

}

.dropdown-menu{

    background-color: #000033ab !important;

    border: 1px solid #fff!important;

}

.dropdown-item{

    color: #fff !important;

    font-weight:500;

}

.dropdown-item:hover{

    color: #fff !important;

    background-color:transparent !important;

    font-weight:500;

}

.dropdown-divider{

    color:#fff !important;

}

span.validate{

    color: #fff !important;
    margin-left: 85px;

    border: 1px solid var(--secondary-color);

}

span.icon-menu{

    color: #fff !important;

    font-size: 16pt;

}

button.navbar-toggler{

    border: 1px solid var(--secondary-color);

}

button.navbar-toggler:focus{

    border: 1px solid #fff;

    box-shadow: none;

}

#inputEmail:invalid + span::after{

    position: absolute;

    content: '×';

    color: #fff !important;

    font-size: 20pt;

    font-weight: bolder;
```

```
}

#inputEmail:valid + span:after{

   position: absolute;

   content: '✓';

   color: #fff !important;

   font-size: 18pt;

   font-weight: bolder;

}

.verify_box{

   color: #fff;

   width: 50%;

   margin: 0 auto;

   border-radius: 30px;

}

@media (max-width:1024px){

   .verify_box{

   color: #fff;

   width: 95%;

   margin: 0 auto;

   border-radius: 30px;
}

input[type="text"], textarea, .info-success,

.info-normal, .info-danger, .input-group{

   width: 90% !important;

   margin: 0 auto !important;

   font-weight: normal!important;

}

label{

   font-weight: bold;

   text-align: left!important;

   margin-left: 25px;

}

span.validate{

   margin-left: 25px;

}}
```

APPENDIX B (User Interfaces of the Application)

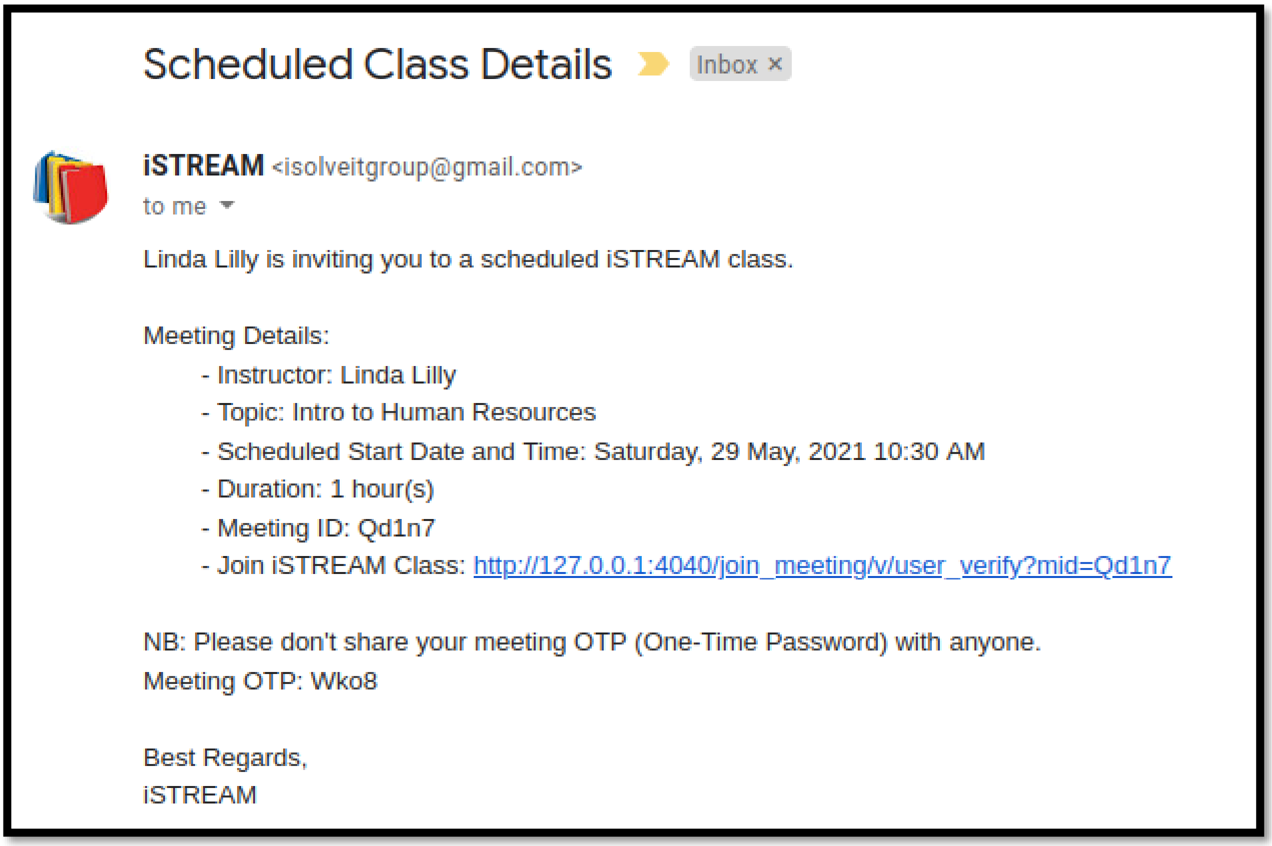


Interface for Home Page



```
Duodu Randy is inviting you to a scheduled iSTREAM class.

Class Details:
   - Instructor: Duodu Randy
   - Topic: Computer Security: XSS
   - Scheduled Start Date and Time: Sunday, 30 May, 2021 12:30 AM
   - Duration: 1 hour(s)
   - Class ID: QX81t
   - Join iSTREAM Class: http://127.0.0.1:15000/join_meeting/v/user_verify?mid=QX81t
```

Interface for Home Page after scheduling class



Scheduled Class Details

iSTREAM <isolveitgroup@gmail.com>

to me

Linda Lilly is inviting you to a scheduled iSTREAM class.

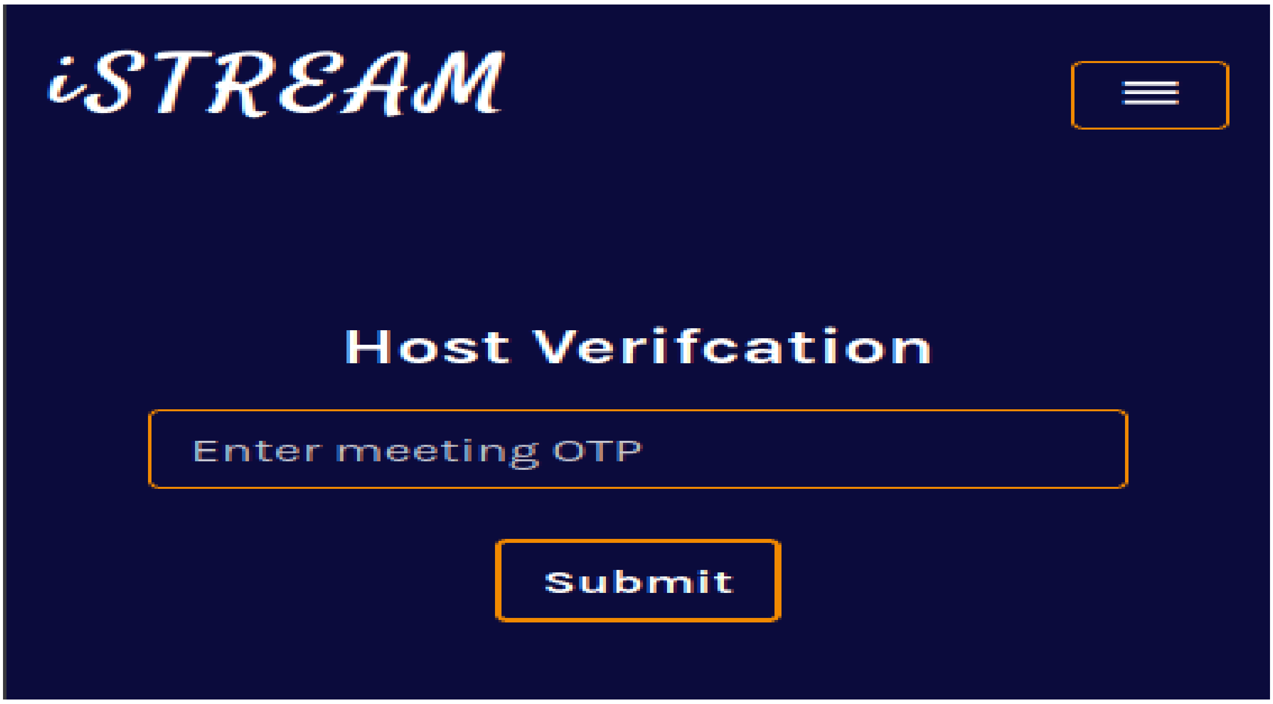
Meeting Details:

- Instructor: Linda Lilly
- Topic: Intro to Human Resources
- Scheduled Start Date and Time: Saturday, 29 May, 2021 10:30 AM
- Duration: 1 hour(s)
- Meeting ID: Qd1n7
- Join iSTREAM Class: http://127.0.0.1:4040/join_meeting/v/user_verify?mid=Qd1n7

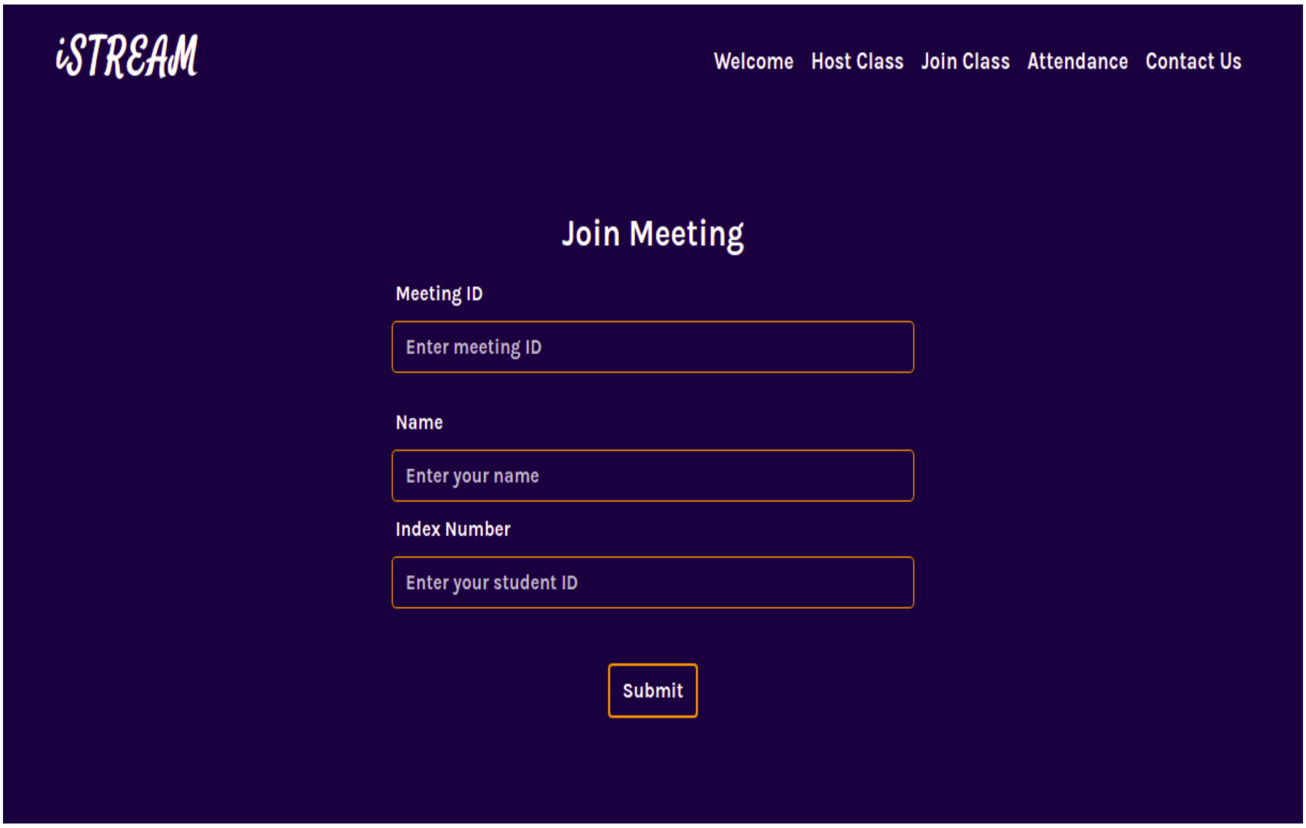
NB: Please don't share your meeting OTP (One-Time Password) with anyone.
Meeting OTP: Wko8

Best Regards,
iSTREAM

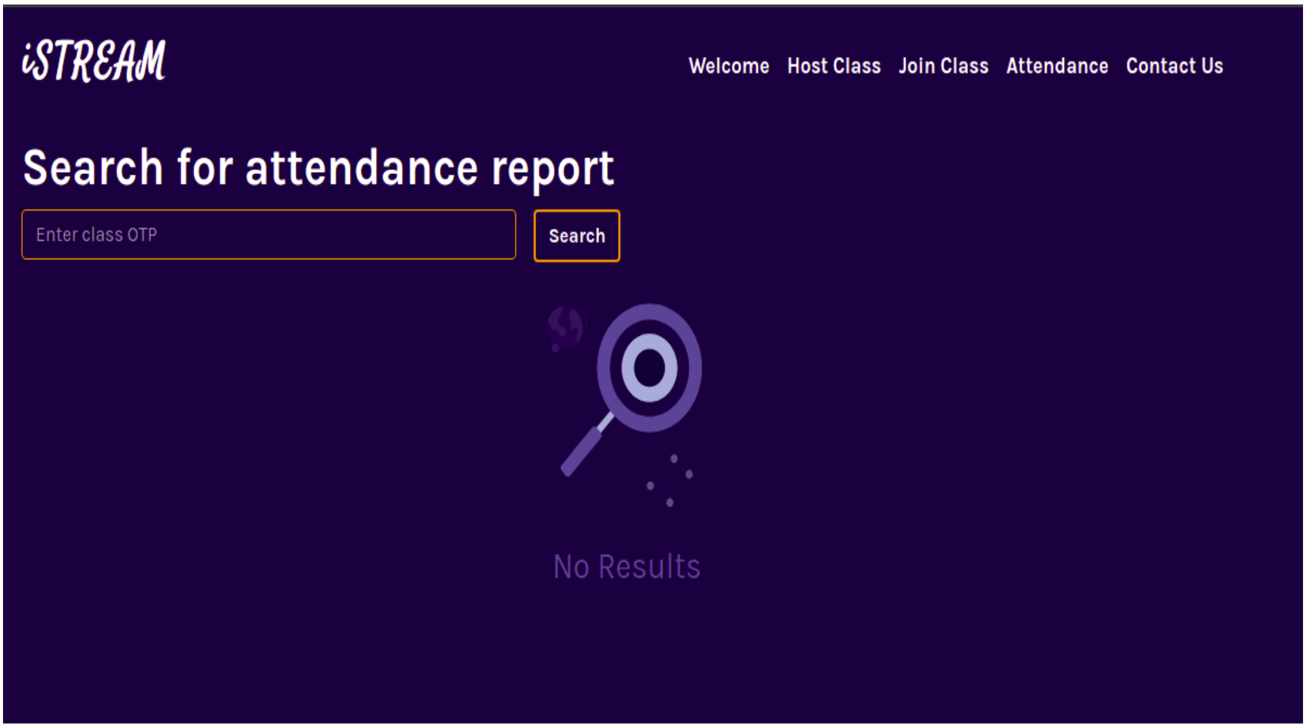
The interface of Class Details emailed to Host



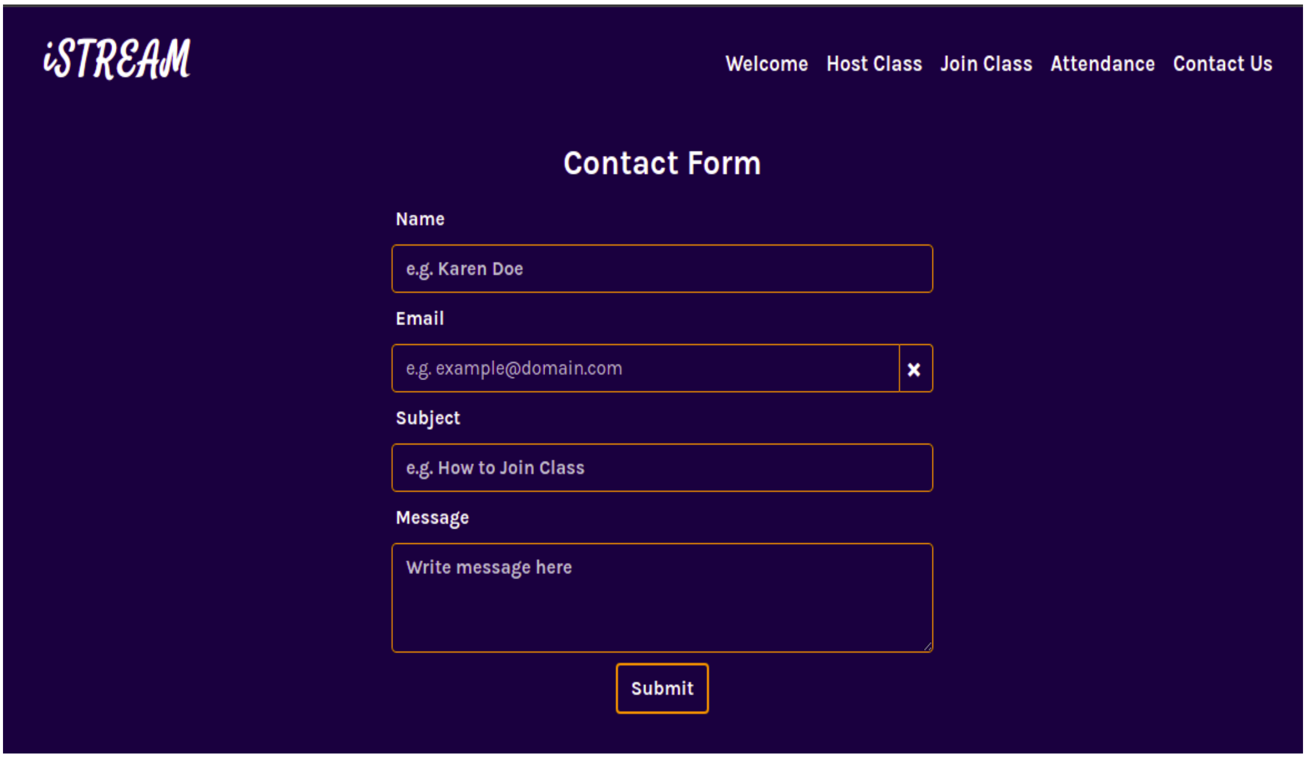
Interface for Host Class Page



Interface for Join Class



Interface for Attendance Page



Contact Form

Name

e.g. Karen Doe

Email

e.g. example@domain.com

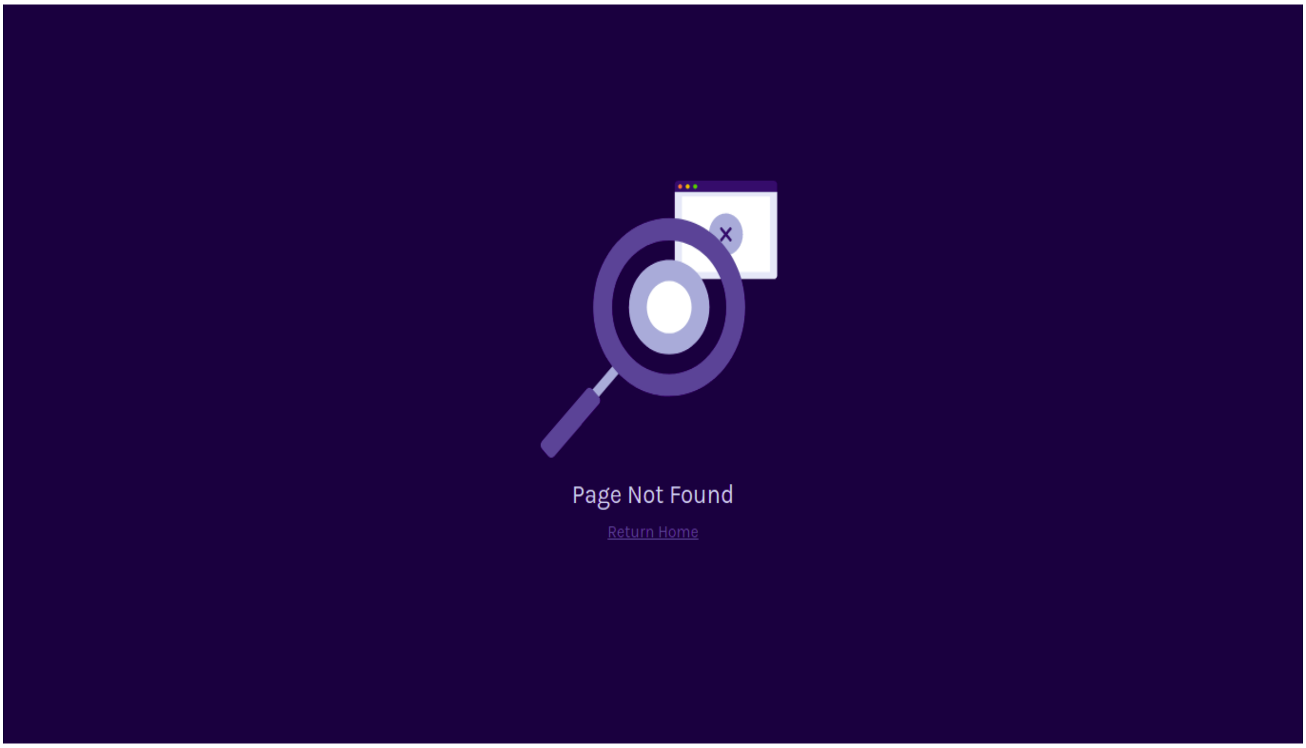
Subject

e.g. How to Join Class

Message

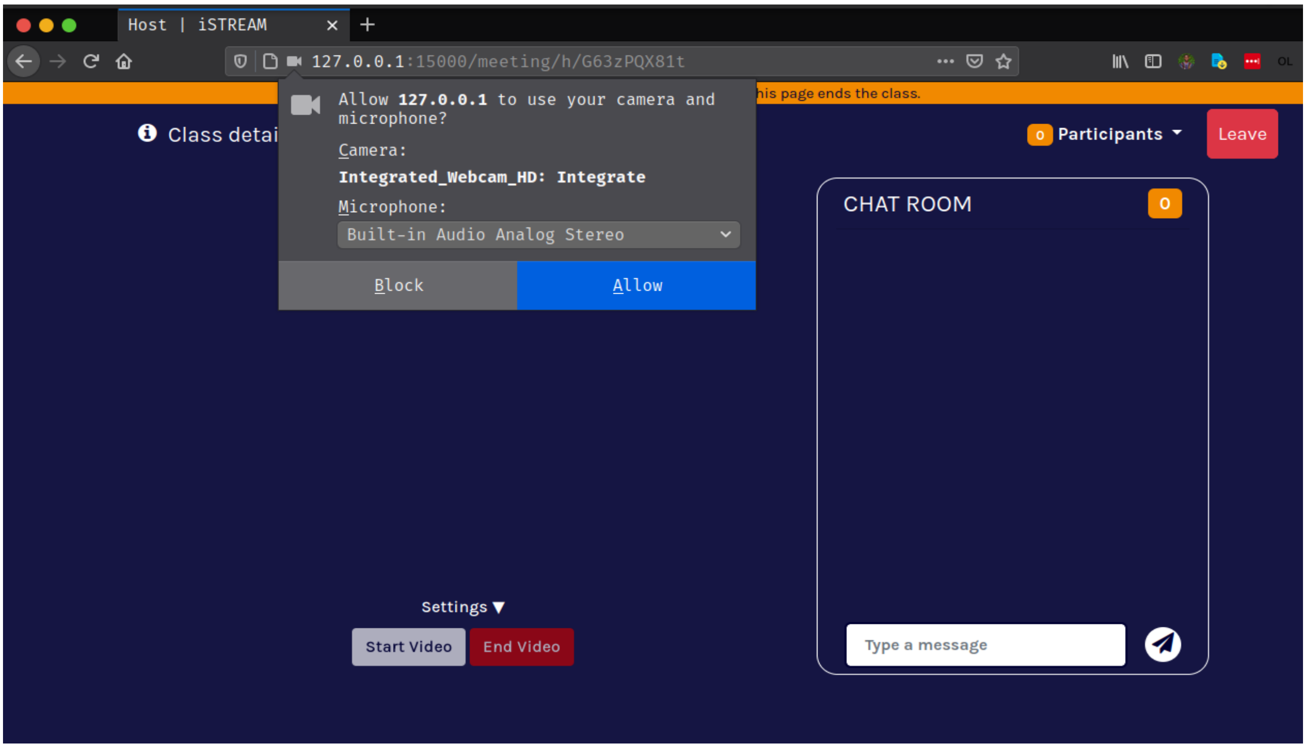
Write message here

Interface for Contact Us Page



Page Not Found

Interface for Page Not Found (Error 404)



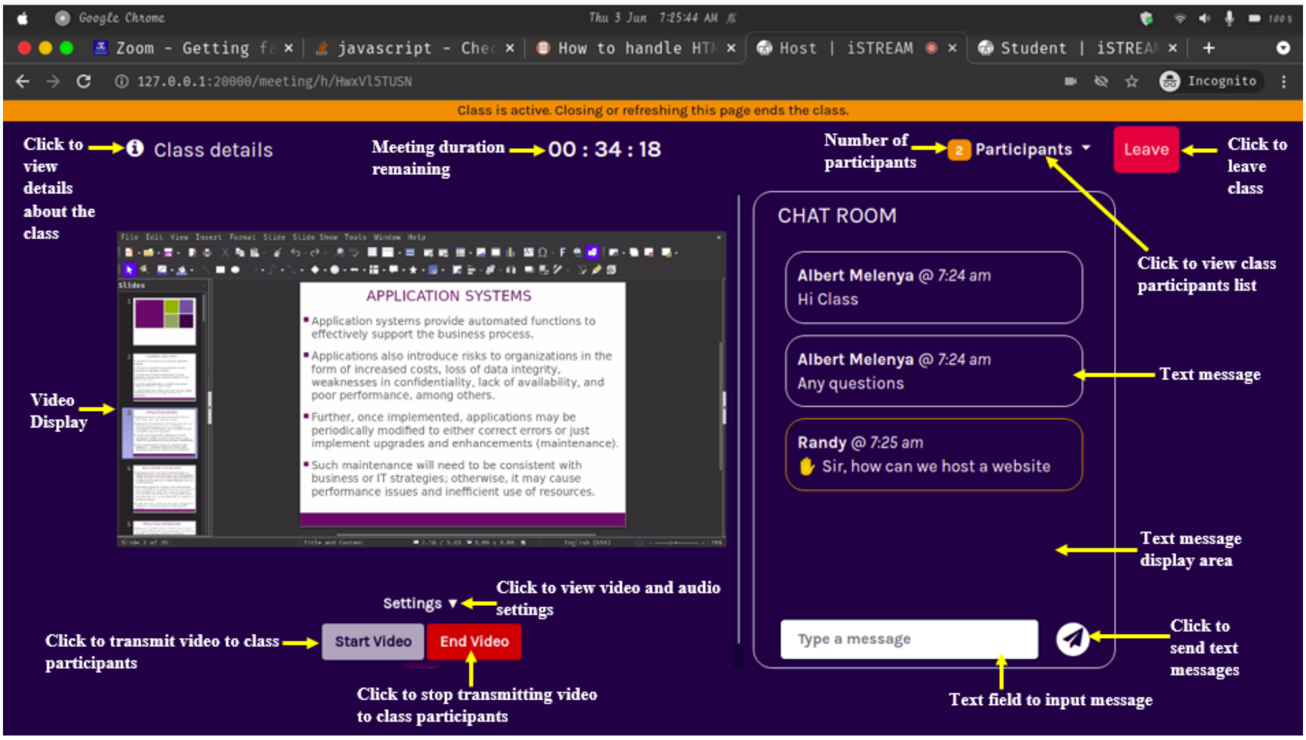
Allow **127.0.0.1** to use your camera and microphone?

Camera:
Integrated_Webcam_HD: Integrate

Microphone:
Built-in Audio Analog Stereo

his page ends the class.

Interface for Studio Page requesting access to camera and microphone



Class is active. Closing or refreshing this page ends the class.

Click to view details about the class

Meeting duration remaining

Number of participants

Click to leave class

Click to view class participants list

Video Display

Text message

Text message display area

Click to view video and audio settings

Click to transmit video to class participants

Click to stop transmitting video to class participants

Click to send text messages

Text field to input message

Albert Melenya @ 7:24 am
Hi Class

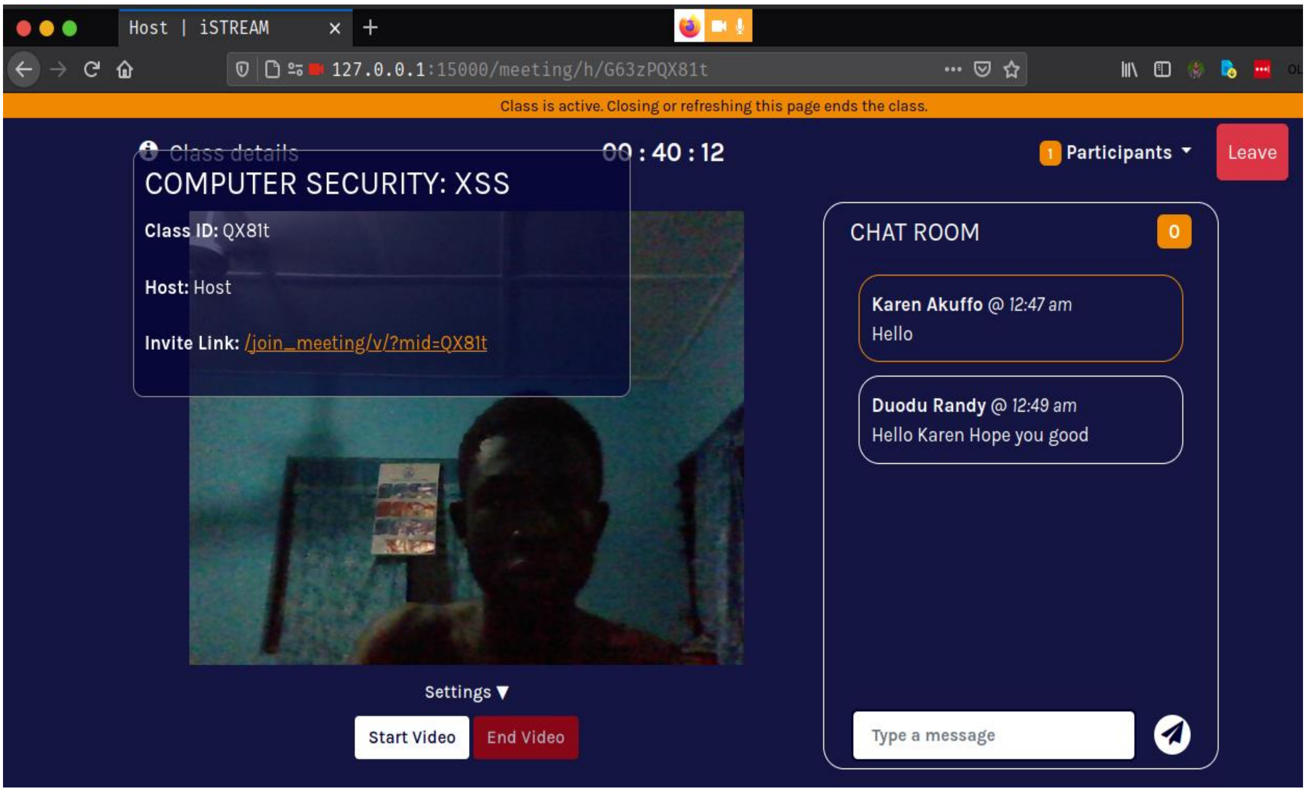
Albert Melenya @ 7:24 am
Any questions

Randy @ 7:25 am
✋ Sir, how can we host a website

APPLICATION SYSTEMS

- Application systems provide automated functions to effectively support the business process.
- Applications also introduce risks to organizations in the form of increased costs, loss of data integrity, weaknesses in confidentiality, lack of availability, and poor performance, among others.
- Further, once implemented, applications may be periodically modified to either correct errors or just implement upgrades and enhancements (maintenance).
- Such maintenance will need to be consistent with business or IT strategies; otherwise, it may cause performance issues and inefficient use of resources.

Interface for Studio Page with an explanation about each component



Class is active. Closing or refreshing this page ends the class.

COMPUTER SECURITY: XSS

Class ID: QX81t

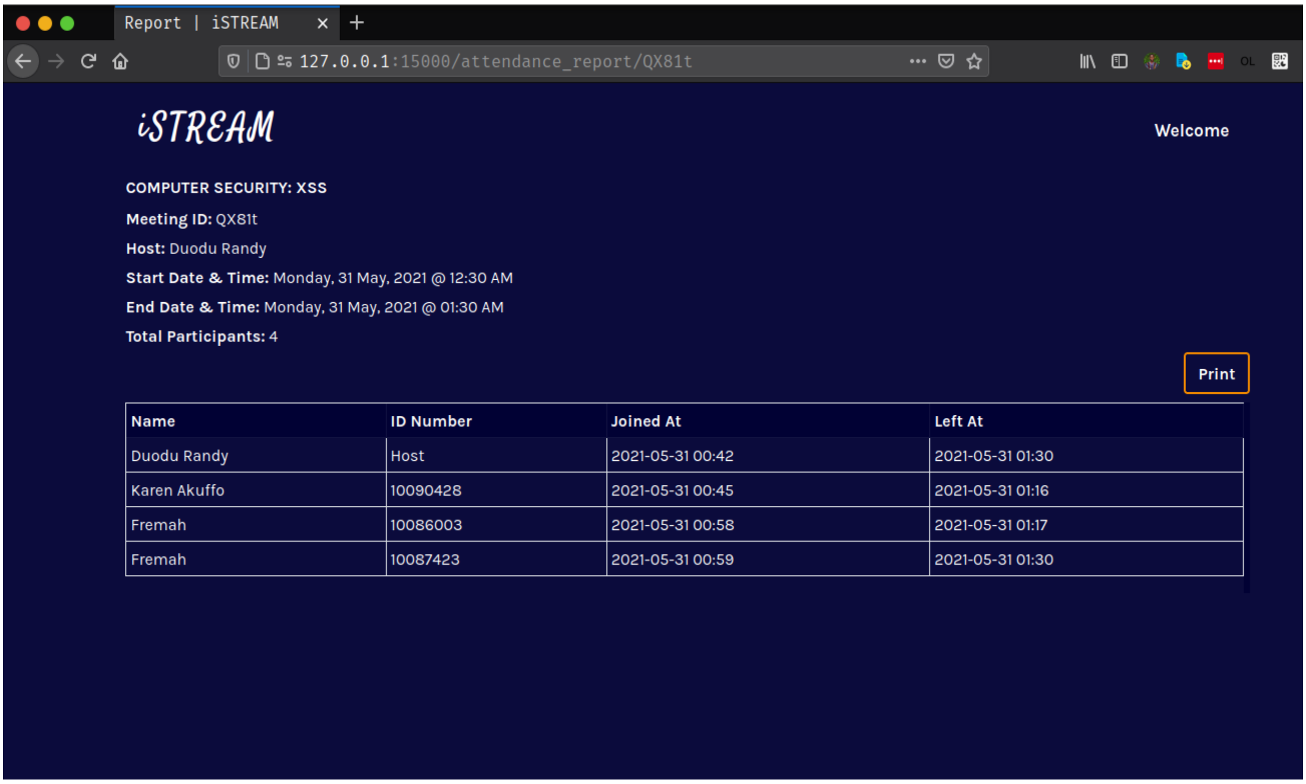
Host: Host

Invite Link: /join_meeting/v/?mid=QX81t

Karen Akuffo @ 12:47 am
Hello

Duodu Randy @ 12:49 am
Hello Karen Hope you good

Interface for Studio Page with the host using chat room and class details shown



COMPUTER SECURITY: XSS

Meeting ID: QX81t

Host: Duodu Randy

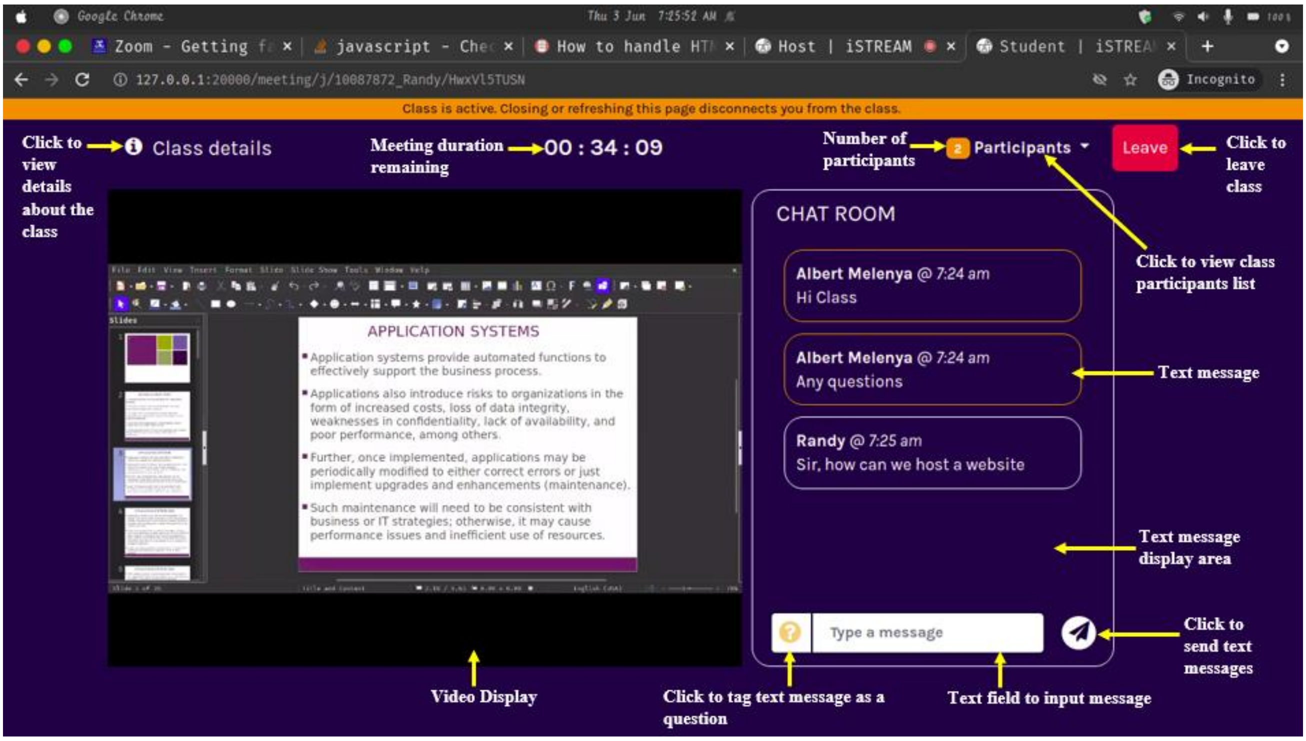
Start Date & Time: Monday, 31 May, 2021 @ 12:30 AM

End Date & Time: Monday, 31 May, 2021 @ 01:30 AM

Total Participants: 4

Name	ID Number	Joined At	Left At
Duodu Randy	Host	2021-05-31 00:42	2021-05-31 01:30
Karen Akuffo	10090428	2021-05-31 00:45	2021-05-31 01:16
Fremah	10086003	2021-05-31 00:58	2021-05-31 01:17
Fremah	10087423	2021-05-31 00:59	2021-05-31 01:30

Interface for Attendance Report Page



Google Chrome
Thu 3 Jun 7:25:52 AM
Zoom - Getting f
javascript - Chec
How to handle HT
Host | iSTREAM
Student | iSTREA
127.0.0.1:20000/meeting/j/10087872_Randy/HwxVl5TUSN
Incognito
Class details
00 : 34 : 09
2 Participants
Leave
CHAT ROOM
Type a message
Click to view details about the class
Meeting duration remaining
Number of participants
Click to leave class
Click to view class participants list
Text message
Text message display area
Click to send text messages
Video Display
Click to tag text message as a question
Text field to input message

Class is active. Closing or refreshing this page disconnects you from the class.

Albert Melenya @ 7:24 am
Hi Class

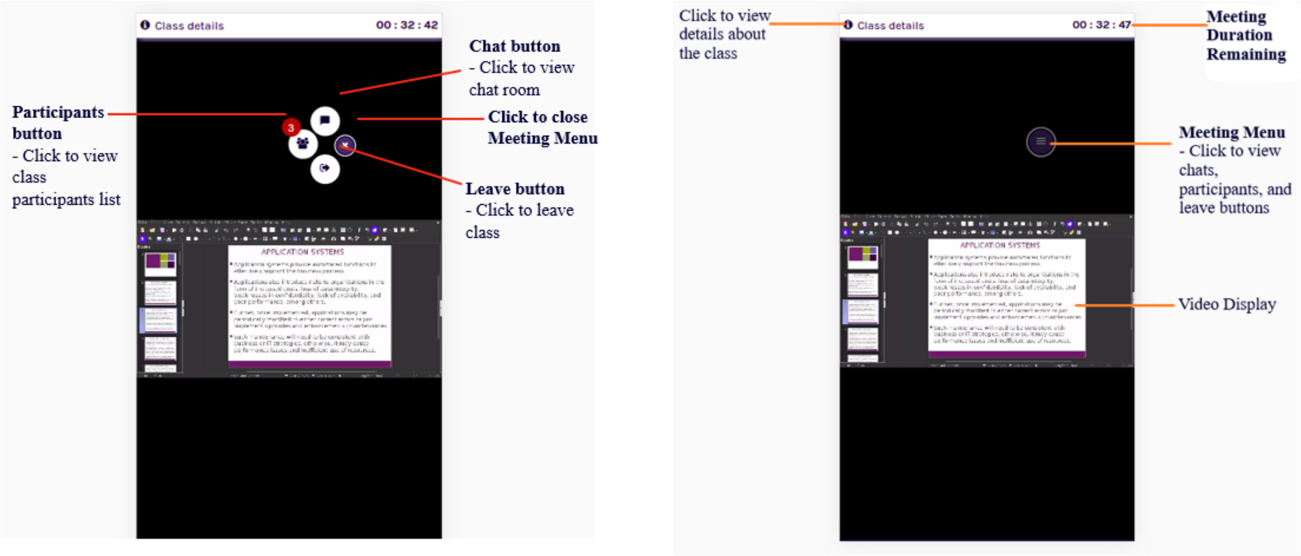
Albert Melenya @ 7:24 am
Any questions

Randy @ 7:25 am
Sir, how can we host a website

APPLICATION SYSTEMS

- Application systems provide automated functions to effectively support the business process.
- Applications also introduce risks to organizations in the form of increased costs, loss of data integrity, weaknesses in confidentiality, lack of availability, and poor performance, among others.
- Further, once implemented, applications may be periodically modified to either correct errors or just implement upgrades and enhancements (maintenance).
- Such maintenance will need to be consistent with business or IT strategies; otherwise, it may cause performance issues and inefficient use of resources.

Interface for Viewer Page with an explanation about each component



Class details
00 : 32 : 42
3
Participants button
- Click to view class participants list
Chat button
- Click to view chat room
Click to close Meeting Menu
Leave button
- Click to leave class
Class details
00 : 32 : 47
Click to view details about the class
Meeting Duration Remaining
Meeting Menu
- Click to view chats, participants, and leave buttons
Video Display
APPLICATION SYSTEMS

Mobile interface for Viewer Page with an explanation about each component