DECEMBER 15, 2020

FYP FINAL REPORT

VIRTUAL CLASSROOM

AHSAAN ULLAH 17I-0133
FASIH SAEED 17I-0245
M. UZAIR IBRAR 17I-0300

SUPERVISED BY : MR. JAWAD HASSAN

Anti-Plagiarism Declaration

This is to declare that the above FYP report produced under the:

Title: Virtual Classroom

is the sole contribution of the author(s) and no part here of has been reproduced on as it is basis (cut and paste) which can be considered as Plagiarism. All referenced parts have been used to argue the idea and have been cited properly. I/We will be responsible and liable for any consequence if violation of this declaration is determined.

DECEMBER 15, 2020

Student 1

Name: Ahsaan Ullah

Signature: _____

Student 2

Name: Fasih Saeed

Signature: _____

Student 3

Name: Uzair Ibrar

Signature: _____

Supervisor (Faculty)

Name: Mr. Jawad Hassan

Signature: _____

Content

1. Introduction

The purpose of this document is to build an online virtual classroom through which students can take online classes and can access the content using cloud services.

Attendance, Engagement estimation and measuring Percentage of presence and Data Analytics has always been a major problem in all sorts of available classrooms till now. As we all know that world is moving toward digital education, for that purpose Virtual classrooms are very important for conducting off-campus classes. There are many virtual classrooms with basic features, but the problem of real time Student's attendance, Engagement, Percentage of presence in a lecture and Data analytics has not been solved yet ; in other words these features are not available in any of the available virtual classroom.

These problems are interesting and important to consider as the community is majorly facing such issues. These problems has been noticed these days when the world has started setting their classes online, to keep stats of students performance, engagement and attendance in off-campus classes, We should have a proper environment managing these stuffs automatically.

we hope to develop a perfect platform to enhance online education and academia to keep track of individual students through analysis

2. Project Vision

Our vision is to provide people with hearing disabilities a better life where they can communicate easily with the whole world. We hope that this application will be a great step for them to be able to live a normal life without problems they may be facing due to their disabilities.

2.1 Problem Statement

Online learning has been rapidly adopted due to wide-spread access, and the many benefits it offers. Students no longer need to commute to a classroom and are often able to learn at their own pace. All they need is an internet connection to communicate with tutors.

A key problem in this process is analyzing student's performance and engaging them in classroom activities like online lectures. The modern world lacks the solution which provides an appropriate analysis of students and the solution through which institutional mentors can check engagement in an online classroom. This gap is making online education less attractive than a traditional classroom.

We as a team try to provide a web platform through which mentors as well as students can take full advantage of online learning. This tool will mostly be focused on providing mentors fashioning teaching experiences. It'll analytical tool through which mentors can take an in-depth report about every student's performance and also go through the student engagement in classroom activities

2.2 Objectives

- Create a web-based platform through which students can take online classes and can access the content using cloud services.
- Institutional mentors can check the presence of a student in an online class using eye-tracking.
- Institutional mentors can analyze the engagement of the student in a certain course through his/her performance in the course.

Provide a Dashboard to mentors to visualize the result of analyses

2.3 Project Scope

This project will be an interactive web platform for students and institutional mentors to take online education. After the release of this project, students will be able to interact with their mentors with full ease but it'll mostly be focused on providing the mentors a platform or tool through which they can take an in-depth report about every student's performance and also go through the student engagement in classroom activities. It will also have the feature automatic attendance by the system . After that we will implement different computer science techniques such as Artificial Intelligence, Machine Learning and Data warehouse to implement the core features of our project. Then combined with the basic features of our classroom, these core features will play an important role for solving the problems currently faced by virtual classroom platforms.

e application further.

3. Software requirement Specifications

3.1 List of features

- Students Percentage Of presence in a lecture (In Class of an hour how much time the student was focused in lecture).
- Automatic attendance by the system using facial recognition.
- Detecting unnecessary voice / kicking those students who are making noise.
- Student Engagement in lectures through assignments seek. (Content Based).
- Generate automated reports for keeping record. Attendance and performance reports.

3.2 Functional requirements

The main functional requirements of our project is:

1. The system must register students and teachers.
2. The system must offer courses to the students.
3. The students must be able to register courses.
4. There must be live lectures of the offered courses.

5. The live lecture must provide audio, video, chat box and white board to students and teacher.
6. The system must calculate the percentage of presence of the student during a live lecture through eye tracking.
7. The student's attendance will be calculated upon calculating the percentage of presence of the students.
8. The system must maintain record of student's attendance, the assignments submitted and the quizzes he attempted.
9. The system must store the lectures in the form of live lectures and files on the cloud so that it is later accessible to the students.

3.3. Quality attributes

Attribute	Brief Detail
Use-ability	The application provides easy to use interface, that anyone can use it with ease without any issue. Labels are easy to pick, and design and font size are attractive and readable for every normal person.
Reliability	VC provides reliable environment where it makes easy for everyone to attend online session. It is reliable for people who want to use this application for communication or learning.
Maintainability	No maintenance is required at the user end. If there is any issue and need maintenance than maintenance team will handle it and upload the new version of it.

3.4 Nonfunctional requirements

1. All users should be authenticated before granting access to the Virtual Classroom.
2. All users should have access according to their respective authorization privileges.
3. Student should be enrolled in a course before accessing its materials and attempting its assignments and exams.
4. Assignments should be uploaded before being graded.
5. Quizzes should have been attempted before being graded.
6. Automated attendance must be marked when the student attends the lecture.

4. High level Use cases

4.1. Use Cases List

Use Case ID	Primary Actor	Use Cases
UC-101	Student	Attend Class
UC-102	Teacher	Record Lecture
UC-103	Teacher	Post Announcement
UC-104	Teacher	View Dashboard
UC-105	Teacher	Manage Announcement
UC-106	Admin	Edit Profile
UC-106	Admin	Edit Profile
UC-107	Student	Video Call
UC-108	System	Eye Tracking

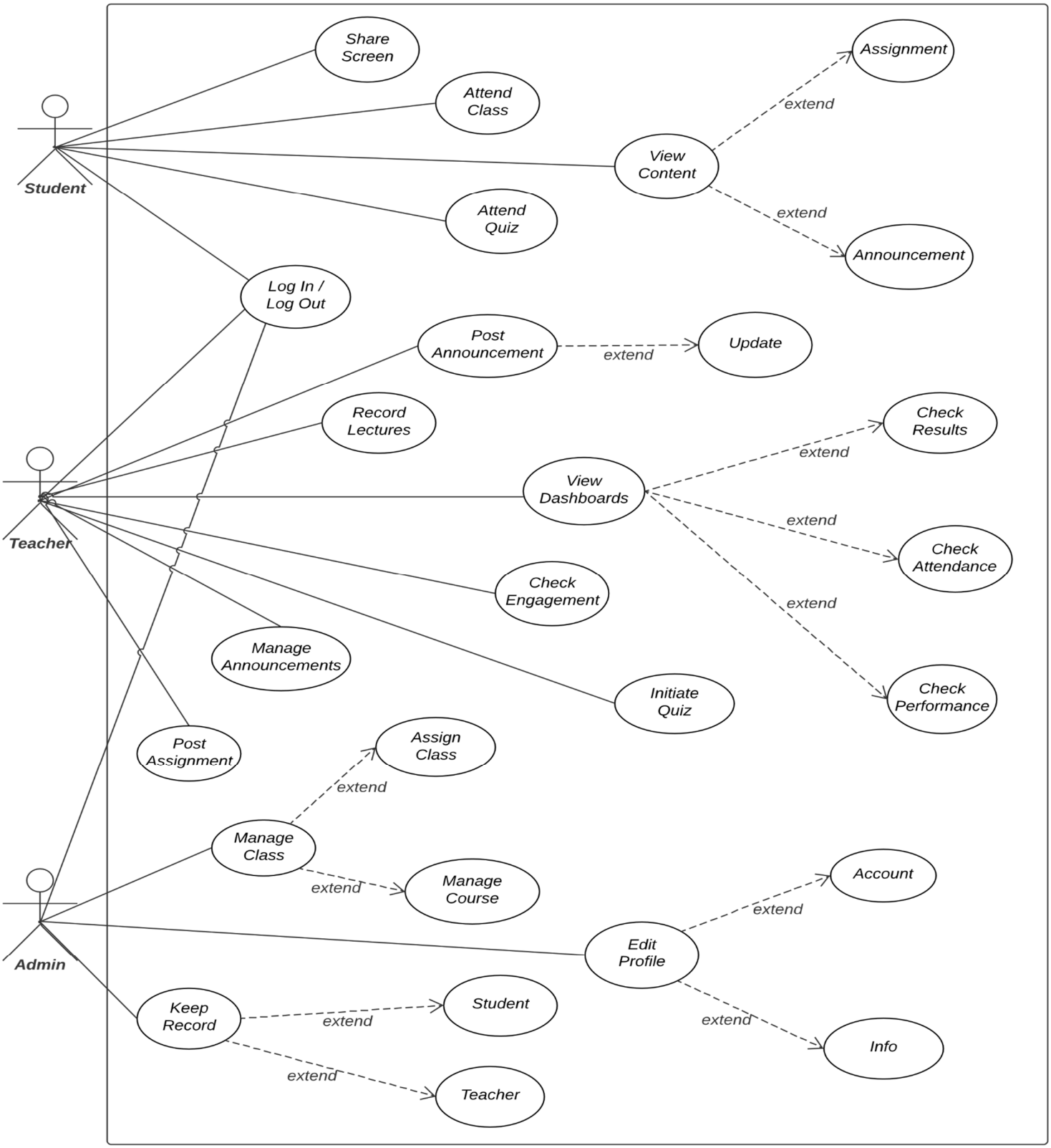
4.2. Use Cases

Use Case ID:	UC-101
Use Case Name:	Attend Class
Actors:	1. Student
Type:	Primary

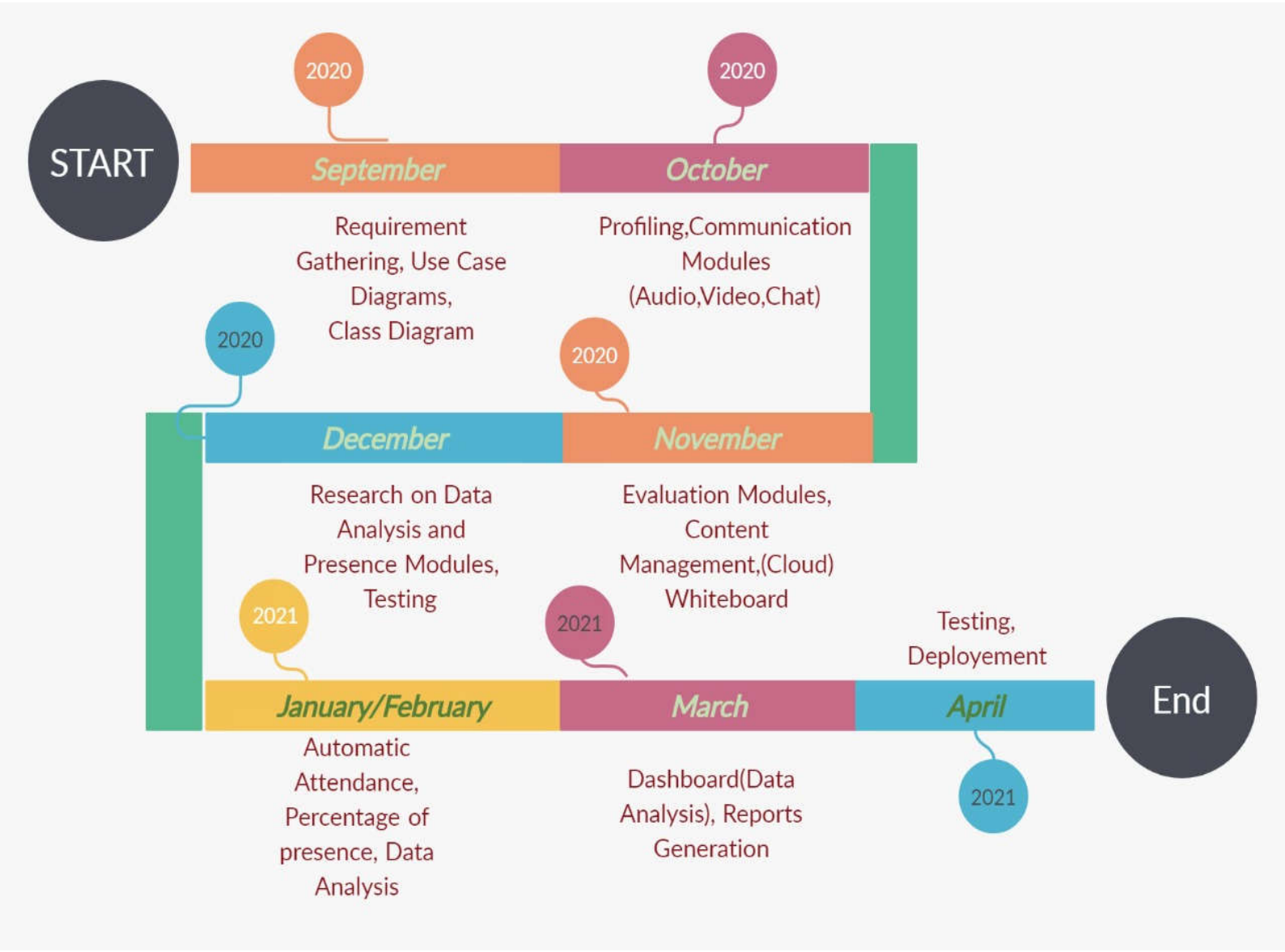
Description:	User will use web portal and register him/herself in organization and then he will Log In as student to attend class
Use Case ID:	UC-102
Use Case Name:	Record Lecture
Actors:	Teacher
Type:	Primary
Description:	Teacher must be registered and at least one class should be assigned to teacher to record lectures during class.
Use Case ID:	UC-103
Use Case Name:	Post Announcement
Actors:	Teacher
Type:	Primary
Description:	Teacher should be registered and Logged In to post announcement using web portal.
Use Case ID:	UC-108
Use Case Name:	Attend Quiz
Actors:	Student
Type:	Primary
Description:	Student would be able to attend Quiz through web portal.
Use Case ID:	UC-104
Use Case Name:	View Dashboard
Actors:	Teacher
Type:	Primary
Description:	Teacher would be able see the dashboard to visualize the stats of individual students, it will teacher to analyze performance and percentage of presence of students
Use Case ID:	UC-105
Use Case Name:	Manage Announcement
Actors:	Teacher
Type:	Primary

Description:	Teacher would be able to manage announcement by updating, deleting , posting announcement, Only when the teacher would be registered in the organization.
Use Case ID:	UC-106
Use Case Name:	Edit Profile
Actors:	Admin
Type:	Primary
Description:	Admin would be able to Edit Profile of both students and teachers, to make changes using this use case admin must be registered in organization.
Use Case ID :	UC-107
Use Case Name:	LogIn
Actors:	Student / Teacher / Admin
Type:	Primary
Description:	Teacher /Admin / Student would be would be able to access web based virtual classroom for their personal usage respectively.
Use Case ID:	UC-108
Use Case Name:	Video Calling
Actors:	Student / Teacher
Type:	Primary
Description:	Teacher / Student would be able to join class in video call mode.
Use Case ID:	UC-109
Use Case Name:	Eye Tracking
Actors:	Student
Type:	Primary
Description:	System will track Students eye for measuring percentage of presence and to check either he / she is focused in the class or not.

4.3 Use Case Diagram



5. Iteration Plan



6. Iteration 1

6.1 Expanded use case

Use Case ID:	UC-101		
Use-Case Name:	Log In To System		
Created By:	VC Team	Last Updated By:	Oct 26, 2020
Date Created:	Oct 26, 2020	Last Revision Date:	Oct 26, 2020

Actors:	Student / Teacher / Administrator	
Description:	Users will be prompted to login with their VC account information before they can use the system.	
Preconditions:	• The user has an VC account . • The user is trying to log in with their VC account. • The user is not already logged In.	
Post conditions:	• The user is logged in to the system. • The user has access to the Classroom environment.	
Main Success Scenario:	Actor • User accesses the URL. • The system prompts the user for their VC account credentials. • The user enters their VC username and password. • The system authenticates the VC login • The user gains access to the systems functionality	System • System allows the User to use its functionalities.
Alternative Flows:	• Invalid VC account user or pass • User already logged in.	
Exceptions:	• Incorrect credentials.	
Frequency of Use:	Every time the Student/Teacher/Admin wants to access the system	
Assumption	The Student/Teacher/Admin wants to use the system.	

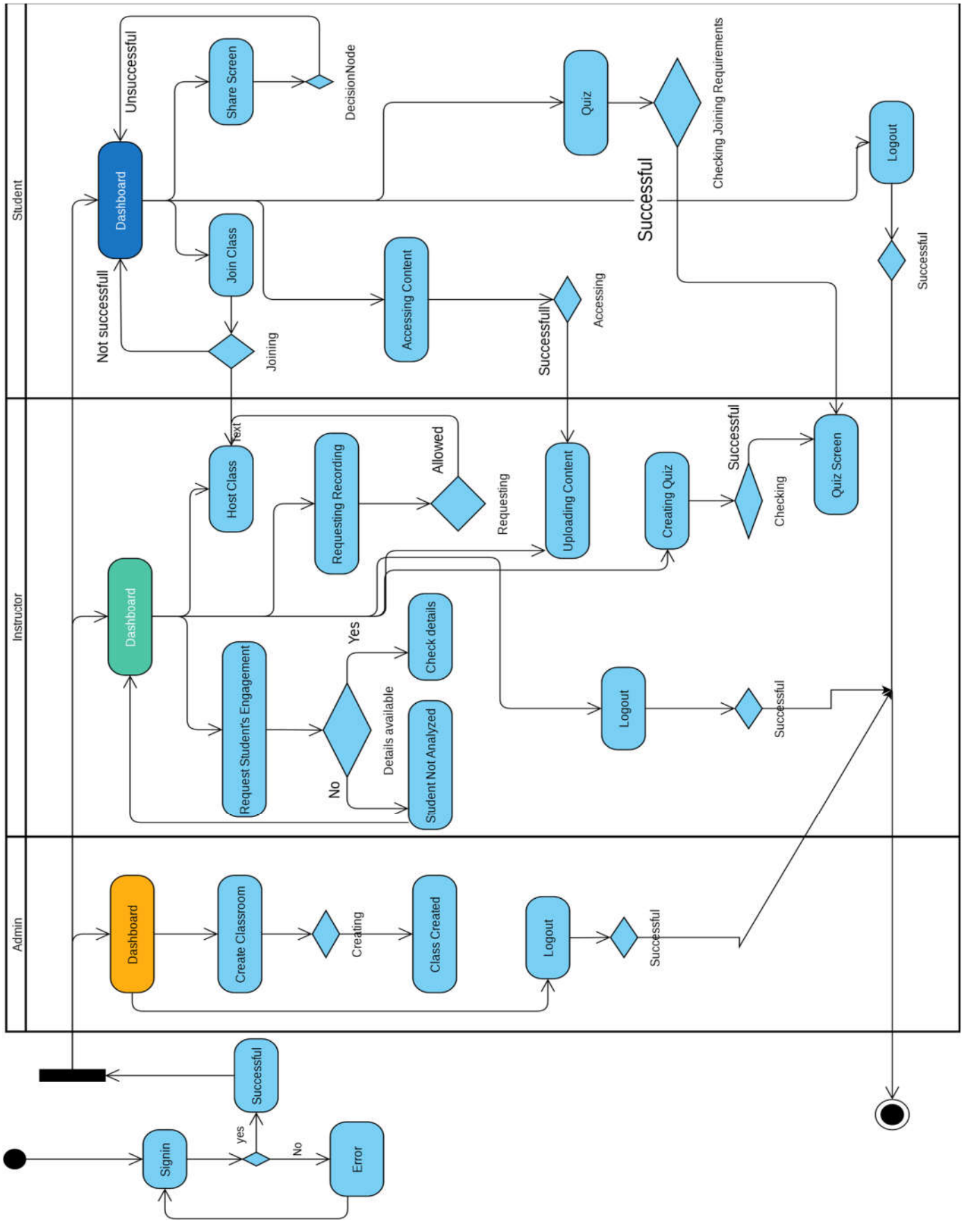
Use Case ID:	UC-102		
Use-Case Name:	Log Out of System		
Created By:	VC Team	Last Updated By:	Oct 26, 2020
Date Created:	Oct 26, 2020	Last Revision Date:	Oct 26, 2020
Actors:	Student / Teacher / Administrator		
Description:	The user clicks on Log Out and their session is terminated.		
Preconditions:	• The user is logged in • The user no longer wants to be logged in.		
Post conditions:	• The user is logged out		
Main Success Scenario:	Actor • User is done using the web application • The user clicks on the logout button • The system logs the user out.	System • System logs Out the user.	
Alternative Flows:	• N/A		
Exceptions:	• N/A		
Frequency of Use:	Whenever the user wants to log out.		
Assumption	No one can use the login session after a user has successfully logged out.		

Use Case ID:	UC-103		
Use-Case Name:	Register Students		
Created By:	VC Team	Last Updated By:	Dec 14, 2020
Date Created:	Dec 14, 2020	Last Revision Date:	Dec 14, 2020
Actors:	Administrator		
Description:	Student would be able to attend class.		
Preconditions:	• The user is logged in • The user wants to join the class.		

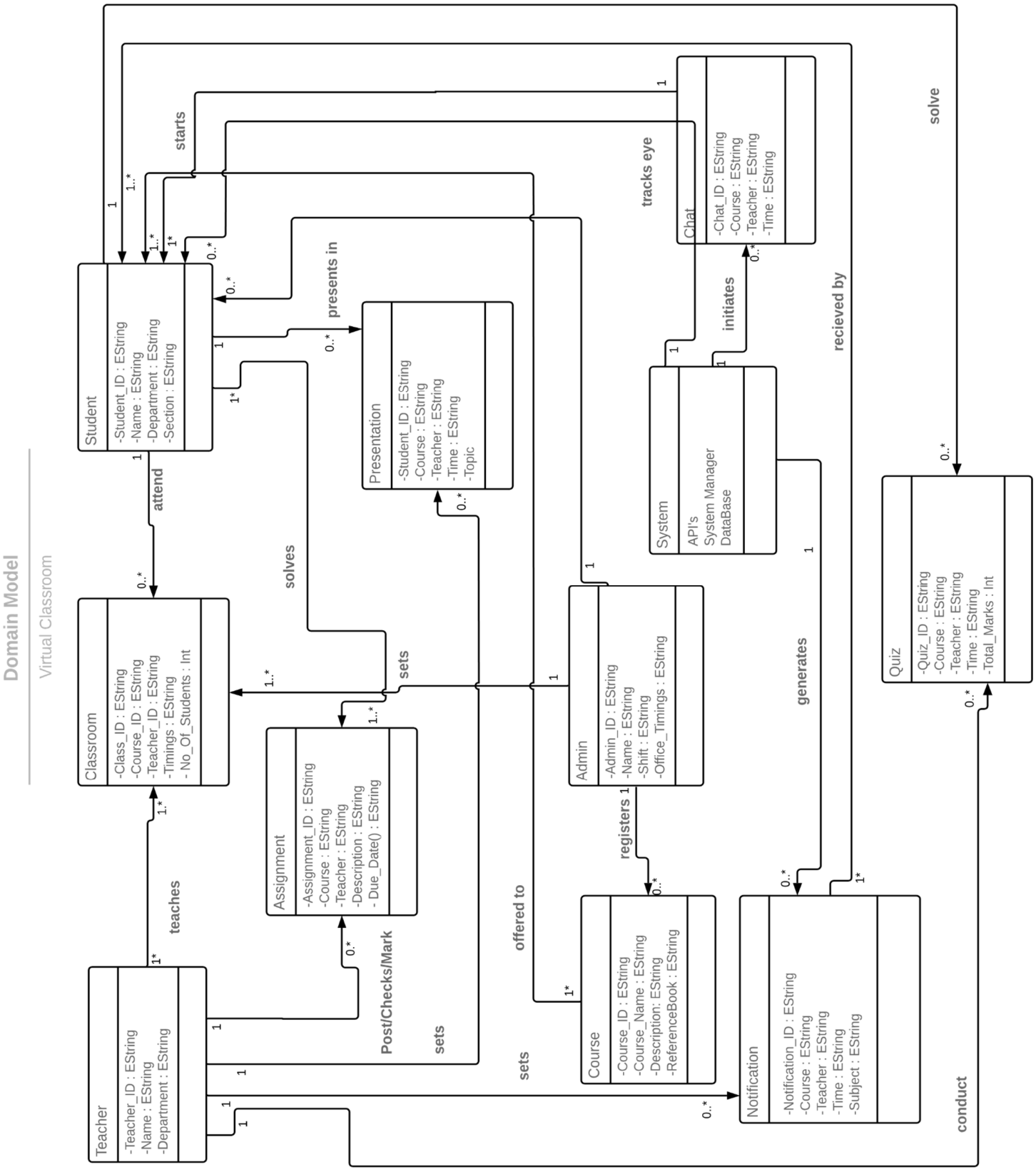
Post conditions:	• The Student has joined the class.		
Main Success Scenario:	Actor • Admin is using the web application • The Admin clicks on the Register button by entering the Roll # of the students. • The student has joined the class.	System • System has allowed students to join the class.	
Alternative Flows:	• N/A		
Exceptions:	• N/A		
Frequency of Use:	Whenever the Students wants to join the class.		
Assumption	If the students isn't registered . He/She can't join the class		
Use Case ID:	UC-104		
Use-Case Name:	Group Chat		
Created By:	VC Team	Last Updated By:	Dec 14, 2020
Date Created:	Dec 14, 2020	Last Revision Date:	Dec 14, 2020
Actors:	Students / Teachers		
Description:	Student / Teacher would be able to message each other in a group during the class.		
Preconditions:	• The user is logged in • The Student has joined the class.		
Post conditions:	• The Student is able to send message in a group.		
Main Success Scenario:	Actor • Student is using the web application • The user types the message and sends to the group. • The system has delivered the message in a group.	System • System has delivered the message in a group Chat.	
Alternative Flows:	• N/A		
Exceptions:	• N/A		
Frequency of Use:	Whenever the Students wants to Message in the class during the streaming.		
Assumption	If the students in the session all the		
Use Case ID:	UC-105		

Use-Case Name:	Eye Tracking		
Created By:	VC Team	Last Updated By:	Oct 26, 2020
Date Created:	Oct 26, 2020	Last Revision Date:	Oct 26, 2020
Actors:	Student		
Description:	The user's eye is being tracked by the system		
Preconditions:	• The user is logged in the system • The user is attending the lecture		
Post conditions:	• The user is logged in the system		
Main Success Scenario:	Actor • Student is using the web application. • The system logs the user out.	System • System is tracking Students Eyes.	
Alternative Flows:	• N/A		
Exceptions:	• N/A		
Frequency of Use:	Every time when the students are taking lectures.		
Assumption	Students eyes are being tracked by the system.		

6.2 Activity Diagram

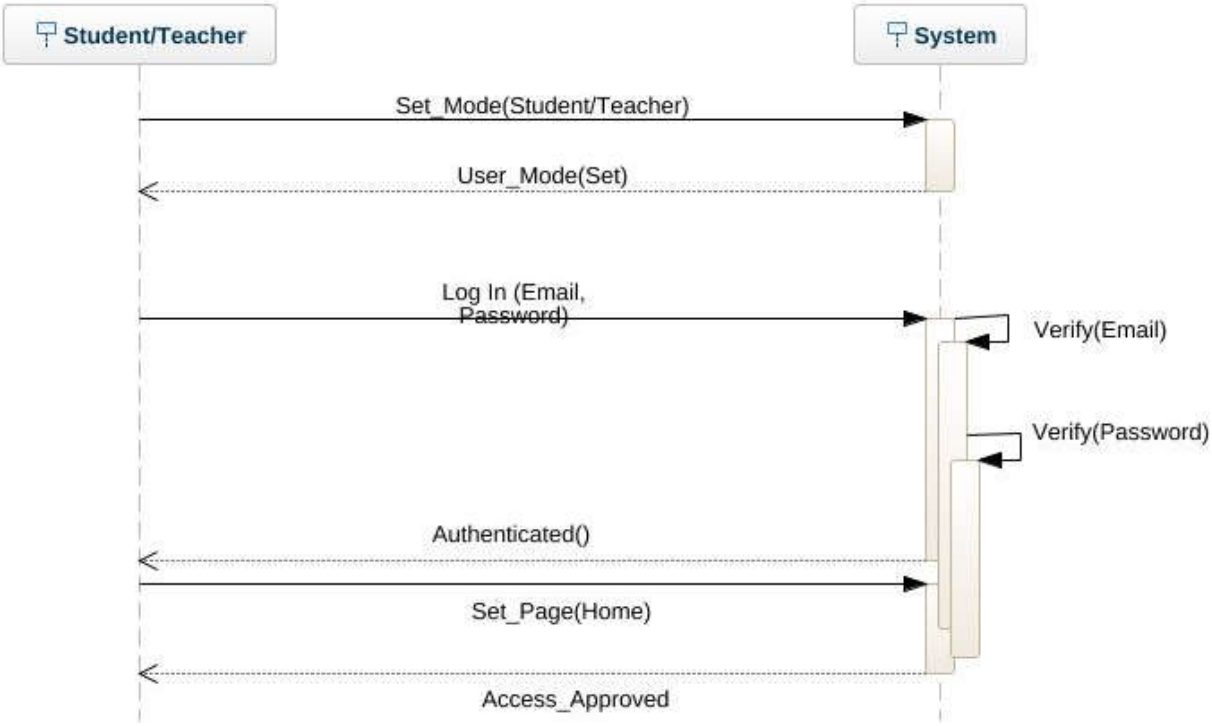


6.3 Domain Model

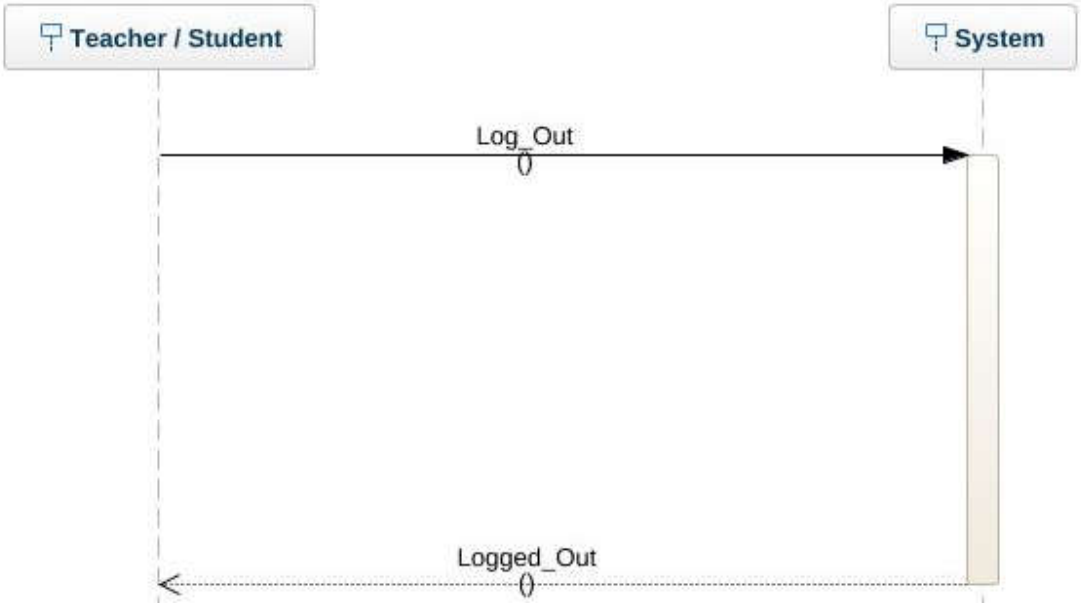


6.4. System Sequence Diagram

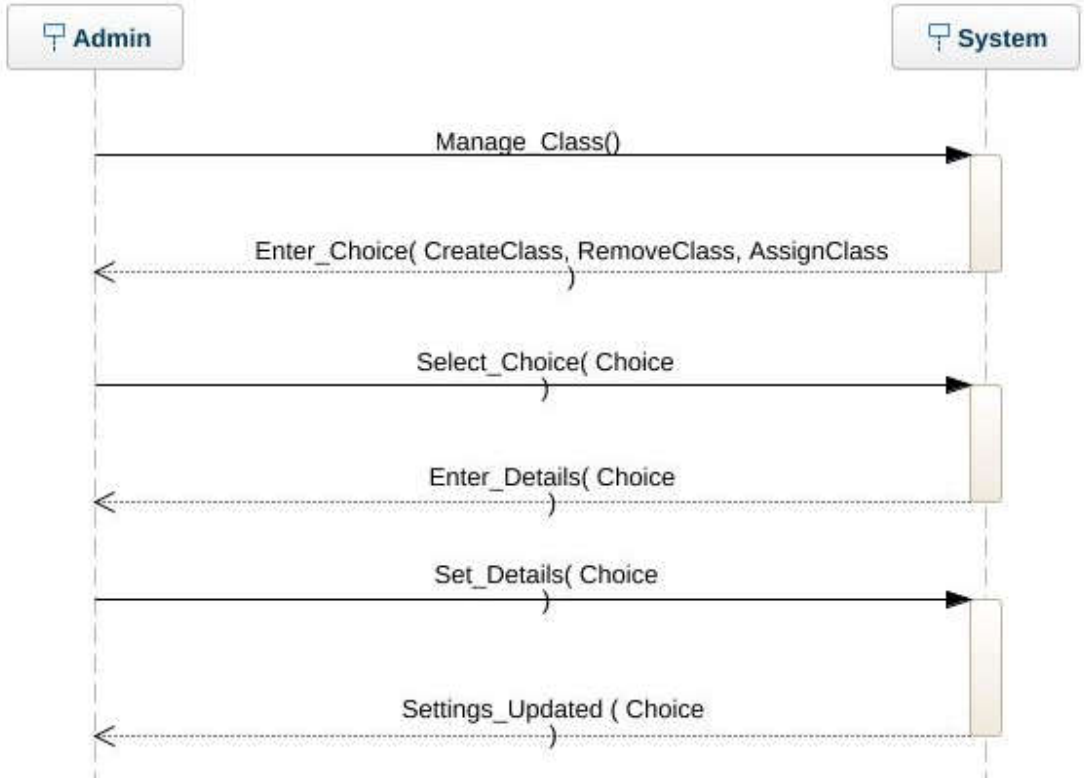
6.4.a. UC-101



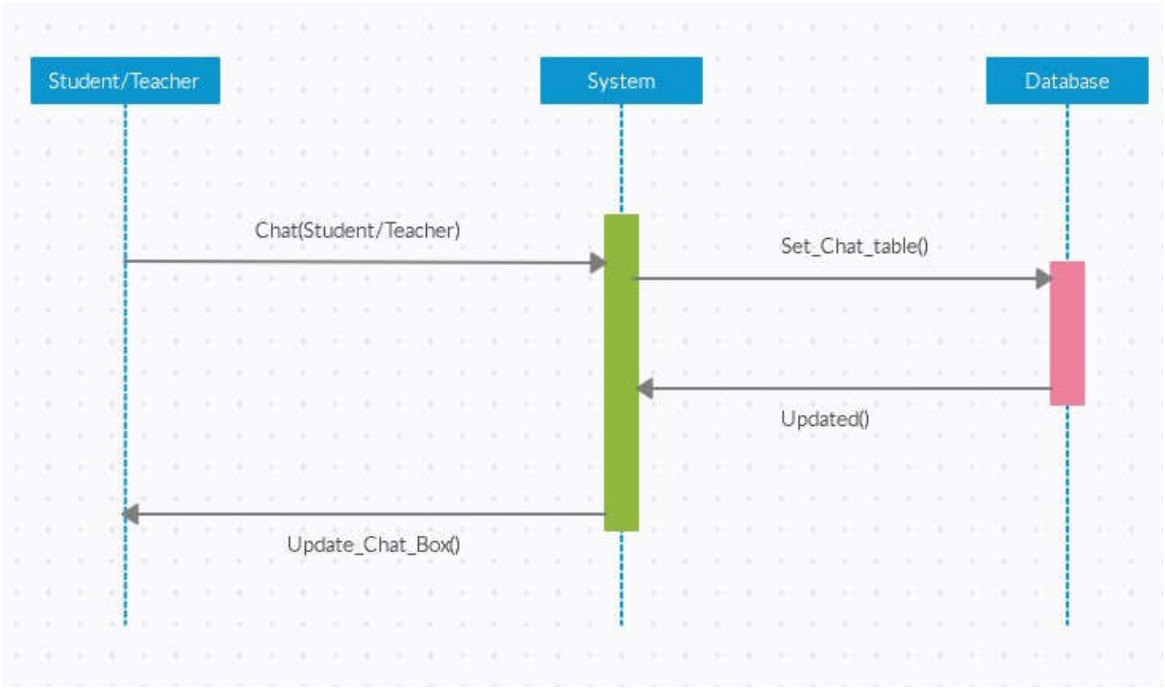
6.4.a. UC-102



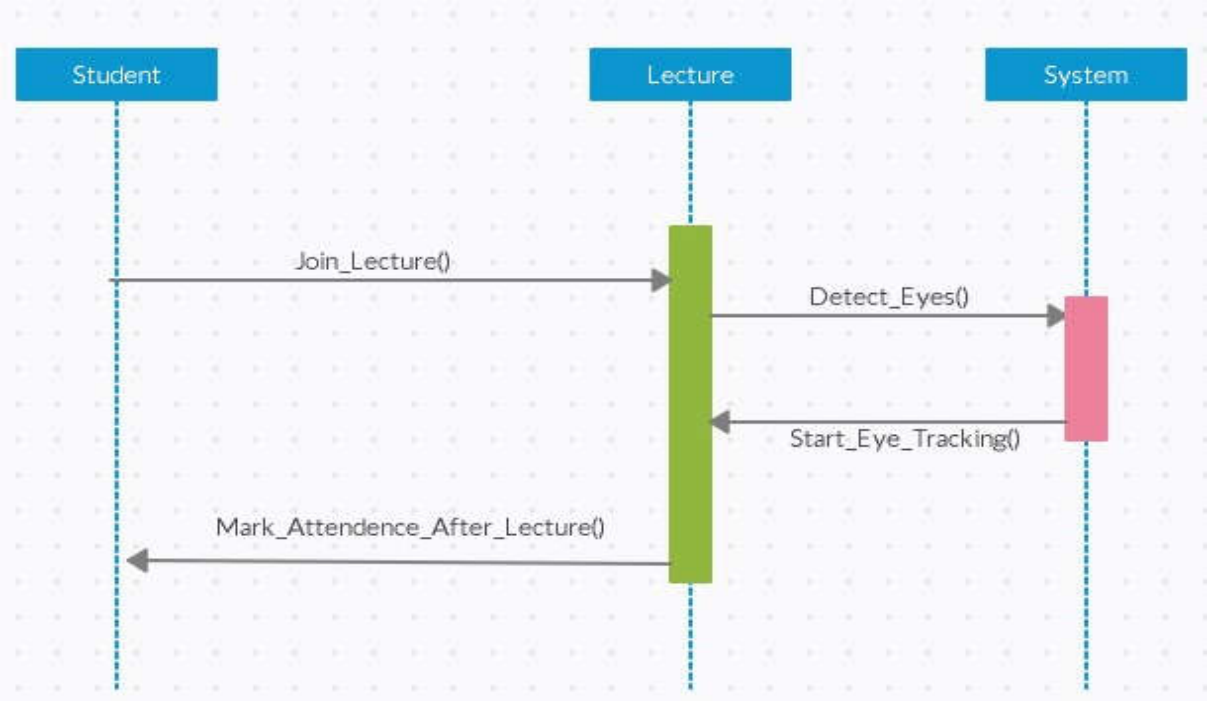
6.4.b. UC-103



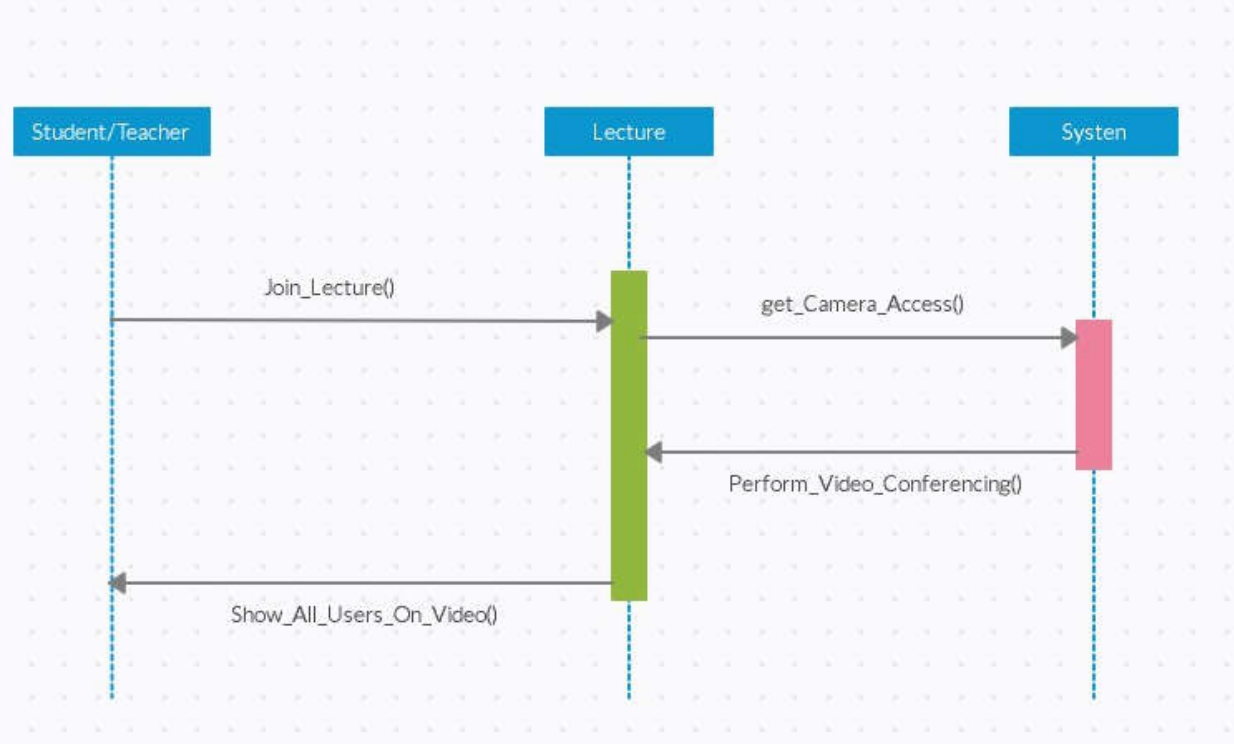
6.4.b. UC-104



6.4.b. UC-105



6.4.b. UC-106



6.5. Operation Contracts

6.5.a. Contract use case ID: UC-101

Contract CO1 : Authentication

Operation Name : Authentication()

Cross Reference : Log In

Responsibility : Students/Techer/Admin to login to their sessions.

Pre-conditions : Student/Teacher/Admin must be registered in organization and have some unique ID. They should have machine with internet connection to access web application.

Post-conditions :

- Instance of Authentication A created in User_Class. The Authentication instance is associated with the user class.
- User has been Logged In to the system

6.5.b. Contract use case ID: UC-102

Contract CO2 : Log Out

Operation Name : Log Out()

Cross Reference : Log Out

Responsibility : Students/Techer/Admin to log Out of their sessions.

Pre-conditions : Student/Teacher/Admin must be logged in to the portal with some unique ID. They should have machine with internet connection to access web application.

Post-conditions : Instance of Authentication A created in User_Class. The Authentication instance is associated with the user class.

User has been Logged Out to the system.

6.5.c. Contract use case ID: UC-103

Contract CO2 : Manage Class

Operation Name : Manage Class

Cross Reference : Manage_Class - Use case

Responsibility : Admin to create, remove and assign class.

Pre-conditions : Admin must be logged in to the portal with some unique ID.

They should have machine with internet connection to access web application.

Post-conditions : Class should have been created. Class is removed if requested. Class is assigned to teacher.

6.5.d. Contract use case ID: UC-104

Contract CO1 : GroupChat

Operation Name : Chat()

Cross Reference : Chat

Responsibility : Students/Techer would be able to send messages in a group.

Pre-conditions : Student/Teacher must be enrolled in a classroom.

Post-conditions : Instance of Chat is C created in User_Class. The C instance is associated with the user class.

User has sent message to the classroom chat.

6.5.e. Contract use case ID: UC-105

Contract CO1 : VideoCalling

Operation Name : VideoCall()

Cross Reference : VideoCall

Responsibility : Students/Techer would be able to Participate in video call.

Pre-conditions : Student/Teacher must be enrolled in a classroom.

Post-conditions : Instance of Call is C created in User_Class. The C instance is associated with the user class.

User has joined the video call conference.

6.5.f. Contract use case ID: UC-106

Contract CO1 : EyeTracking

Operation Name : EyeTrack()

Cross Reference : EyeTrack

Responsibility : Students eyes would be tracked during the conference.

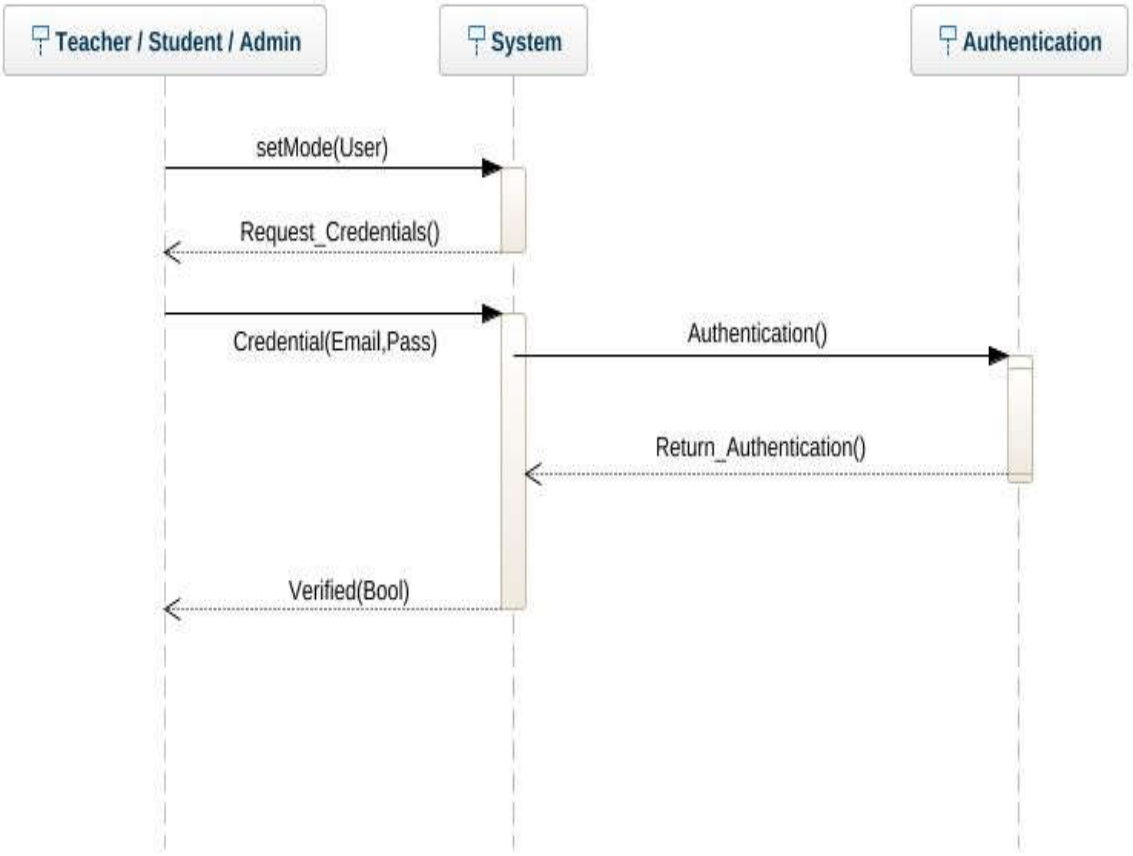
Pre-conditions : Students must be enrolled in a classroom.

Post-conditions : Instance of EyeTracking is ET created in User_Class. The ET instance is associated with the user class.

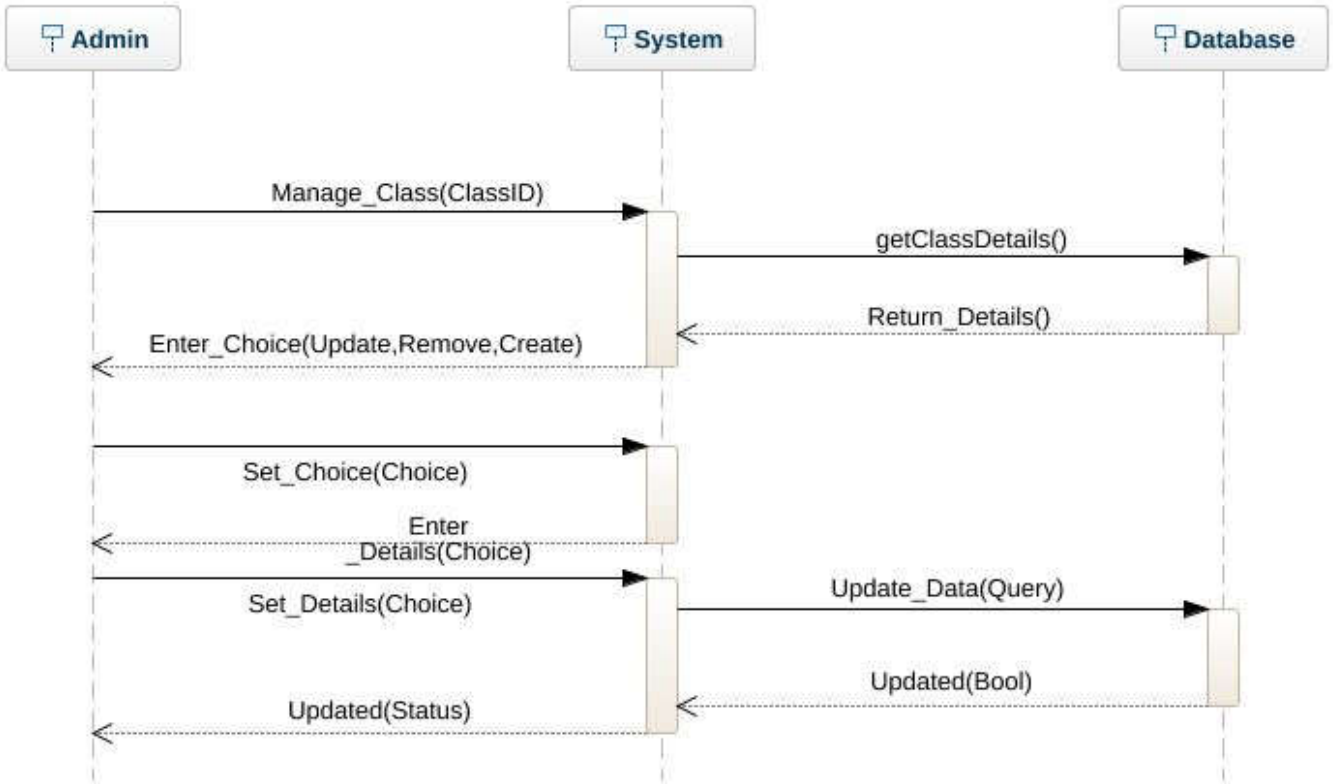
User's eye is being tracked..

6.6. Sequence diagrams

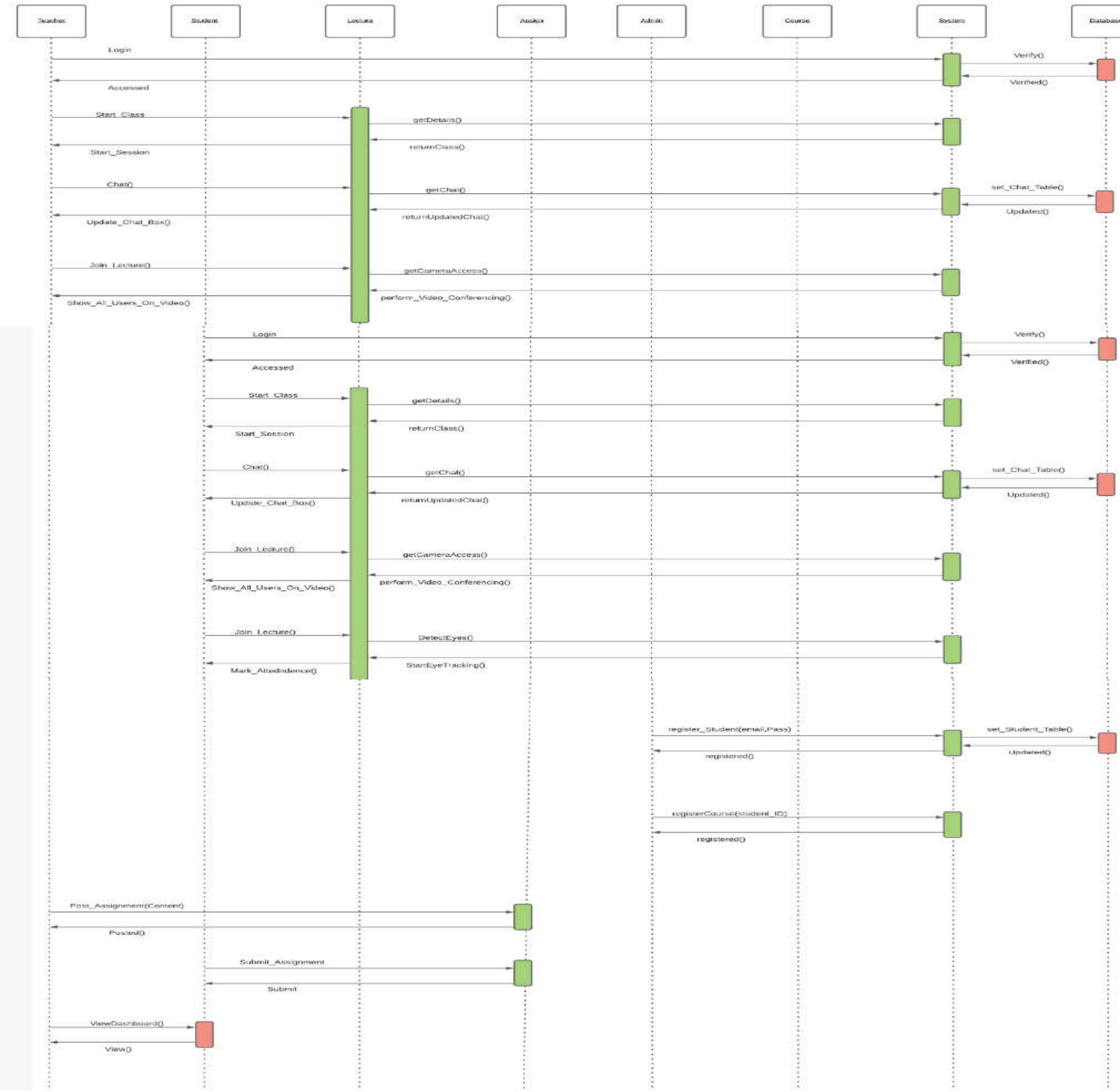
6.6.a. UC-102



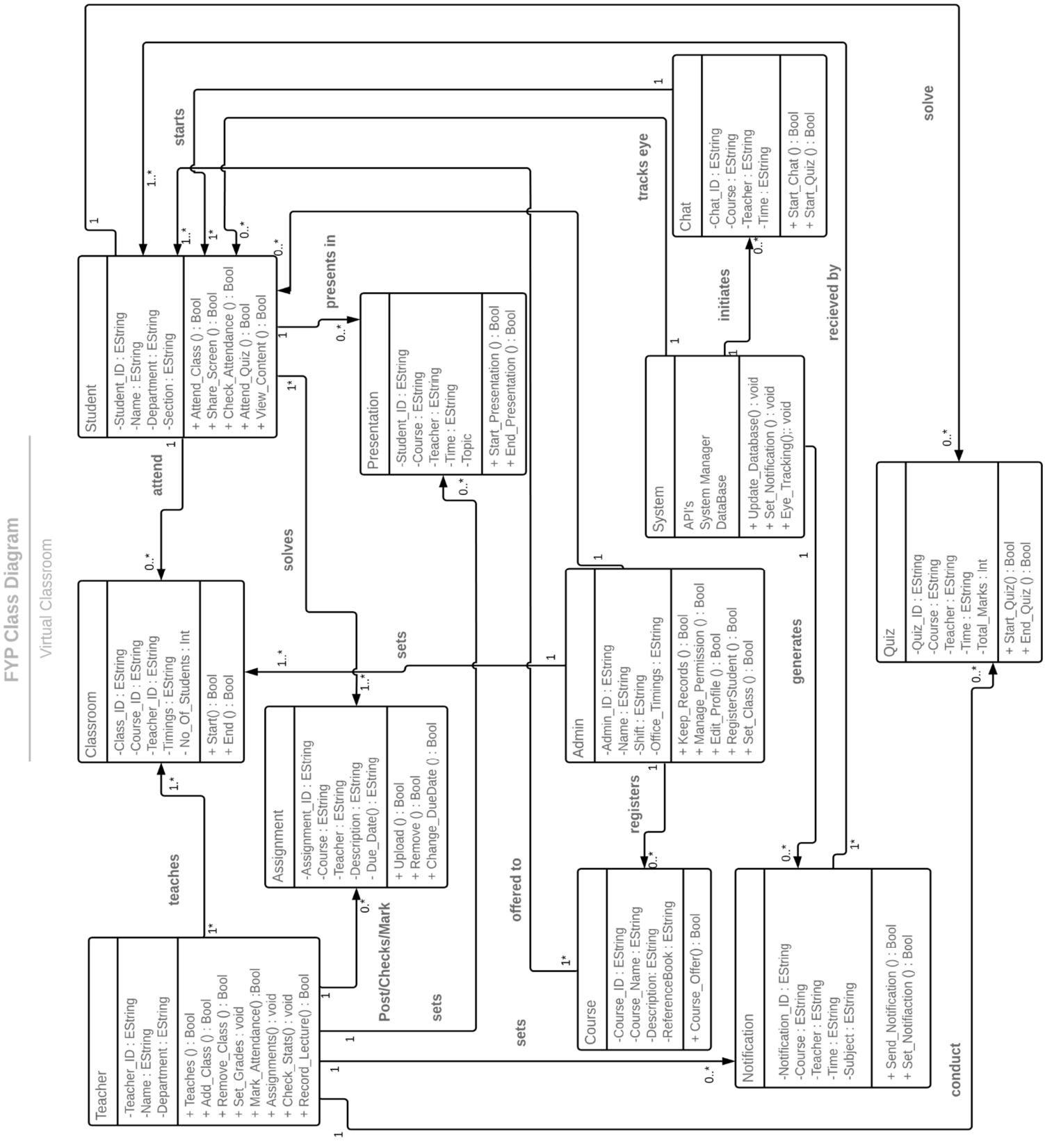
6.6.a. UC-103


Admin
System
Database
Manage_Class(ClassID)
getClassDetails()
Return_Details()
Enter_Choice(Update,Remove,Create)
Set_Choice(Choice)
Enter
_Details(Choice)
Set_Details(Choice)
Update_Data(Query)
Updated(Bool)
Updated(Status)

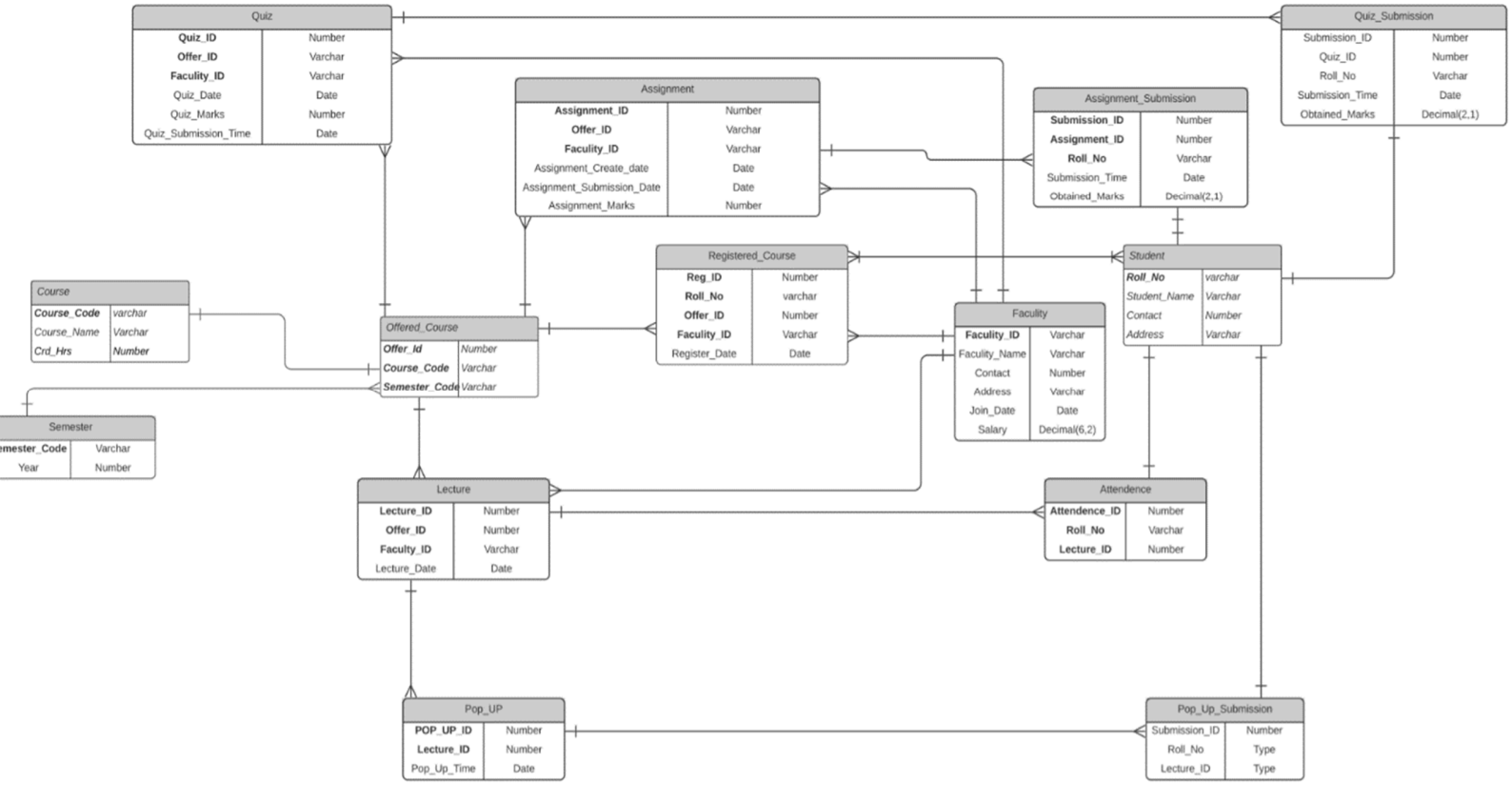
6.6.1 a. UC-103


Teacher
Student
Lecture
Assign
Admin
Course
System
Database
Login
Verify()
Verified()
Accessed
Start_Class
getDetails()
returnClass()
Start_Session
Chat()
getChat()
set_Chat_Table()
Updated()
returnUpdatedChat()
Update_Chat_Box()
Join_Lecture()
getCameraAccess()
perform_Video_Conferencing()
Show_All_Users_On_Video()
Login
Verify()
Verified()
Accessed
Start_Class
getDetails()
returnClass()
Start_Session
Chat()
getChat()
set_Chat_Table()
Updated()
returnUpdatedChat()
Update_Chat_Box()
Join_Lecture()
getCameraAccess()
perform_Video_Conferencing()
Show_All_Users_On_Video()
Join_Lecture()
DetectEyes()
StartEyeTracking()
Mark_Attadindence()
register_Student(email,Pass)
set_Student_Table()
Updated()
registered()
registerCourse(student_ID)
registered()
Post_Assignment(Content)
Posted()
Submit_Assignment
Submit
ViewDashboard()
View()

6.7. Class Diagram



6.8 Entity Relationship Diagram



6.9. Architecture diagram