Bilkent University

Department of Computer Engineering

Senior Design Project

Project short-name: Here!

Analysis Report

Mert Aslan, Rahmiye Büşra Büyükgebiz, Hakkı Burak Okumuş, Ozan Aydın, Yüce Hasan Kılıç

Supervisor: Abdullah Ercüment Çiçek
Jury Members: Çiğdem Gündüz Demir and Can Alkan

Analysis Report
Nov 9, 2020

This report is submitted to the Department of Computer Engineering of Bilkent University in partial fulfillment of the requirements of the Senior Design Project course CS491.

Contents

Analysis Report

Project Short-Name: Here!

1 Introduction

The COVID-19 pandemic that struck the world in early 2020 led humanity to search for alternate solutions to everyday tasks. It affected people to the point that these tasks which would have been quite trivial in the past, now require immense care to complete. Along with these safety measures came the exercise of social distancing, which led to a complete overhaul of human to human interaction. We now communicate mostly online, with little to none face to face interaction. Video conferencing applications, such as Zoom, Google Meet or Microsoft Teams are much more prevalent than ever, due to this unprecedented circumstance. Even though this new type of communication is necessary to prevent the spread of the disease, it certainly has its drawbacks.

As students, during our time being educated remotely, we have experienced some problems that affected both the instructors and the students. It is evident that an online education setting has much less room for student-instructor interaction compared to a classroom setting. This mainly has to do with the fact that video conferencing applications such as Zoom or Google Meet are not specialized to be used as an education medium. Many valuable information, such as the attention of the students to the lectures, that were constantly gathered by instructors in classrooms are lost in online education due to the fact that these conferencing mediums are simply not specialized enough to gather such information. Our experiences show that the loss of such information leads to a complete detachment between the instructors and students, which overall diminishes the effectiveness of online education. Our application aims to overcome this problem, by providing a solution using machine learning and computer vision algorithms that would increase the amount of knowledge the instructors receive about their students both in real time and right after lectures.

Students also lose their ability to effectively follow the lessons, as these video conferencing mediums do not provide them any means to enhance their learning process. They are now essentially "participants" rather than "students" that do not have the essential tools they once had in their classrooms. This unfortunate situation distances them from the learning process itself. We also aim to solve this problem, by providing a platform specialized for the needs of the students in an online education environment.

In this report, we intend to provide an overall analysis of the system. First, existing systems will be discussed. Later, an overview of the proposed system, Here!, will be given by emphasizing its unique features. Then functional, non-functional and pseudo requirements will be listed. Afterwards, system models of our system will be examined. Scenarios will be derived from generalized use-cases whose diagrams will be provided. Object and class model, and dynamic models such as activity, state and sequence diagrams will also be provided. The report will continue with screen mockups presented with their explanations. Lastly, we will conclude the report with a discussion on the other analysis elements of the project.

2 Current System

Even though there are many programs that are being used for online education, most of them are conference apps, and their functionalities are limited to screen sharing. *Here!* will provide unique functionalities such as machine learning based teaching assistant that will track the attendance of students, warn them and provide statistics to instructors about students' attendance after the lectures. Moreover, there will be an interactive note taking system which allows students to take notes directly onto the slides that are shared by the instructors.

2.1 General Video Conference Programs (Zoom, Microsoft Teams, etc.)

Today, most of the schools use general purpose video conference programs in online lectures such as Zoom, Microsoft Teams, etc. However, these programs consist of basic video conference features like video and screen sharing, voice and text communication while they lack education specific features that we are planning to aim such as slide sharing, machine learning based teaching assistant, or interactive note taking system.

2.2 Google Classroom

- Instructors can add students to the lectures directly, or they can send join links to students.
- They can set up a class and create class work that appears on students' calendars.
- Instructors are able to grade students consistently with rubrics integrated into student work.
- They can see originality reports of the student works for potential plagiarism.

2.3 Vedamo

- Instructors can create their own academy to manage their courses, content and learners.
- Users can keep records of online lectures.

2.4 Blackboard

- Instructors can keep track of the students' individual activity during the lecture time relative to others.
- They can compare the change in overall attention between different class hours.
- Users can easily keep track of their data from the mobile application.

2.5 Tophat

- Instructors can create polls, quizzes and synchronous discussions to encourage student participation.
- They can record class meetings and share videos directly from the application.
- They can edit the slides that they already uploaded through the application.

2.6 BigBlueButton

- BigBlueButton provides a live multi-user whiteboard for both instructors and students.
- People can zoom, highlight, draw and write on the presentations through the application.

3 Proposed System

3.1 Overview

Here! is a video conferencing service that is specifically designed for educational needs of both students and instructors. It will include the general characteristics of a video conferencing application with the inclusion of practical educational tools. These tools can be divided into two parts, an online TA that will use motion and eye tracking technologies in order to analyze videos of students to come up with informative data about the class session, and other complementary features that will make online education more reliable and easy for both students and instructors.

If an instructor chooses to do so, it can use the online TA feature to analyze and gather data from videos. Our online TA will print out attendance rates by analyzing how frequently a student participates or raises their hand. It will also notify the teacher simultaneously when a student raises their hand or gives a specific reaction, like a thumbs up. Another feature is to track eye movements in order to understand if a student is distracted and notify the teacher during the lecture.

Other than our online TA, we have multiple features that support online education. One feature will be a simultaneous note taking, where you can instantly screenshot the current slide or shared screen and add that into your notes, where you can view them later in your profile with added personal notes from you. Another main feature will be the slide sharing, where students can navigate freely through a slide that the teacher is actively sharing in order to better read and understand the content without hindering the rest of the class, and sync up with the instructor whenever they want. Additionally, instructors will be able to share their screens separately from the slide share.

3.2 Functional Requirements

1. Users will be able to give video input to the application using any webcam.
2. Users will be able to give audio input to the application using any microphone.
3. Users will be able to register to the application as an instructor or a student.
4. Instructors will be able to register a course to the system.
5. Instructors will be able to assign students to a course.
6. Students will be able to see their weekly schedules.
7. Users will be able to receive notifications for upcoming lectures.
8. Instructors will be able to start a lecture.
9. Students will be able to join a lecture.
10. Users will be able to open/close their cameras and microphones during the lecture.
11. Students will be able to view the instructor’s video and their shared screen in a lecture.
12. Students will be able to navigate in the shared slides (if any) independent from the instructor and synchronize back with the instructor when they want.
13. Students will be able to open a notepad on the side while listening to the lecture and view their notes after the lecture.
14. Users will be able to access the chat panel during the lecture.
15. Users will be able to view the participants list during the lecture.
16. Instructors will be able to mute/unmute students.
17. Students will be able to raise their hands physically to begin to speak.
18. Instructors will be able to receive notifications when students raise their hands.
19. Students will be able to express that they agree on an idea by making a “thumbs up” sign with their hands.
20. Instructors will be able to know how many students have given a “thumbs up” to a question.
21. Students will be able to receive statistics about their performance in the class after the lecture.
22. Instructors will be able to receive statistics regarding the attention of the students to the lecture both in real time and after lecture.

3.3 Non-functional Requirements

1. Security

   The program must ensure that video recordings of users will not be shared with any other 3rd party application. Users must accept the terms of service and privacy policy to use the program.

2. Usability

   The program must be suitable for webcams with different resolutions. The program must be supported by different browsers.

3. Performance

The program must be working fast such that it can process all frames of a video captured in real time from 1 second to no delay.

4. Extensibility

   The program will be developed as a web application at first but it must be extensible for desktop, Android and iOS environments.

5. Scalability

   Since the focus area of our program is education it must be scalable to handle a large number of users from various institutions. The database and the server must be able to handle large numbers of concurrent video calls.

6. Reliability

   The program must detect motions accurately since mistakenly detecting a hand raise can interrupt the flow of the class. Similarly, missing a single one will affect satisfaction of the student.

3.4 Pseudo Requirements

1. Backend implementation language will be Python to make use of Tensorflow [1].
2. The frontend will be implemented using Angular [2].
3. To add real-time communication capabilities to the program, WebRTC framework will be used [3].
4. The underlying database will be MySQL [4].
5. GitHub will be used for the collaborative work and issue tracking.
6. OpenCV and dlib libraries will be used for motion tracking capability of the TA feature of our project [5][6].
7. Darknet Convolutional Neural Network framework together with its YOLOv3 tool will be used for object detection in order to enhance TAs tracking capabilities [7][8].

3.5 System Models

3.5.1 Scenarios

3.5.1.1 Scenario 1

Use Case: Add a course

Primary Actor: Instructor

Entry Condition: Instructor is on the Profile Page.

Exit Condition: Instructor either adds the course by clicking "Add" button or cancels by clicking on the "Cancel" button.

Main Flow of Events:

1. Instructor clicks on "Add a course" button on the profile page.
2. System shows a popup screen to input details for the course to be added.
3. Instructor enters name and date of the course to be added.
4. Instructor adds the course by clicking "Add" button or cancels by clicking on the "Cancel" button.

3.5.1.2 Scenario 2

Use Case: Assign a student to a lecture

Primary Actor: Instructor

Entry Condition: Instructor has already added a course.

Exit Condition: Instructor clicks "Done" button or "Back" button.

Main Flow of Events: Instructor

1. Instructor clicks on the "Assign Student" button.
2. Instructor enters ID of the student.

3. The system adds a student to a lecture.
4. Instructor clicks on the “Done” button.

3.5.1.3 Scenario 3

Use Case: Join Lecture
Primary Actor: Student
Entry Condition: Student is on Home Page.
Exit Condition: Student enters the Lecture Page.
Main Flow of Events:

1. Student clicks on the corresponding “Join Lecture” button of the desired time slot in the “Weekly Schedule” section.
2. System loads Lecture Page for the corresponding lecture that the student clicked if that lecture has already started.
3. Student enters the Lecture Page.

3.5.1.4 Scenario 4

Use Case: Start Lecture
Primary Actor: Instructor
Entry Condition: Instructor has already added a course
Exit Condition: “Start Lecture” button is pressed.
Main Flow of Events:

1. Instructor presses the corresponding “Start Lecture” button of desired time slot.

3.5.1.5 Scenario 5

Use Case: Share Slides
Primary Actor: Instructor
Entry Condition: Instructor started a lecture session.
Exit Condition: Slide to be shared is specified and “Share” button is pressed or “Back” button is pressed and slide share is cancelled.
Main Flow of Events:

1. Instructor clicks on “Share Slide” button.
2. File upload prompt is displayed.
3. Instructor uploads a .pptx or a .pdf file.
4. Instructor clicks on “Share” button.

Alternative Flow of Events:

1. Instructor clicks on “Share Slide” button.
2. Instructor clicks on “Cancel” button.

3.5.1.6 Scenario 6

Use Case: End Lecture
Primary Actor: Instructor
Entry Condition: Instructor is on an ongoing lecture.
Exit Condition: Instructor is navigated to Analytics Page.
Main Flow of Events:

1. Instructor clicks on “End Lecture” button.
2. Instructor is navigated to Analytics Page.

3.5.1.7 Scenario 7

Use Case: View and Download Statistics
Primary Actor: Instructor
Entry Condition: Instructor clicks on “End Lecture” button.

Exit Condition: Instructor clicks on “Back to Main Menu” button.
Main Flow of Events:

1. Instructor clicks on “End Lecture” button.
2. The system displays the session statistics.
3. Instructor clicks on “Download as Excel” button.
4. The system initiates the download.
5. Instructor clicks on “Back to Main Menu” button.

3.5.1.8 Scenario 10

Use Case: Receive notification for upcoming class
Primary Actor: Student
Entry Condition: There is an upcoming class that the student is assigned.
Exit Condition: Student clicks on “x” (Close) button.
Main Flow of Events:

1. The system sends a notification onto Student’s “Upcoming Classes” section in Home Page.
2. Student clicks on “x” (Close) button.

3.5.1.9 Scenario 11

Use Case: Student takes notes
Primary Actor: Student
Entry Condition: Student clicks on “Take Note” button.
Exit Condition: Student clicks on “Exit Notes” button.
Main Flow of Events:

1. Student clicks on “Take Note” button.
2. System opens the Notepad screen.
3. Student writes on the notepad.
4. Student clicks “Save” button.
5. Student clicks on “Exit Notes” button.

Alternative Flow of Events:

1. Student clicks on “Take Note” button.
2. System opens the Notepad screen.
3. Student clicks on “Add Screenshot” button.
4. System screenshots the current slide, adds into Notepad.
5. Student clicks “Save” button.
6. Student clicks on “Exit Notes” button.

3.5.1.10 Scenario 12

Use Case: View a saved note
Primary Actor: Student
Entry Condition: Student is on the Profile Page.
Exit Condition: Student clicks on “X” button to exit viewing a saved note.
Main Flow of Events:

1. Student clicks a saved note listed in “Saved Notes” section in the profile page.
2. System brings a popup screen showing the saved note of the student.
3. Student clicks on “X” button at the upper right corner of the popup screen to exit viewing a saved note and goes back to the profile page.

3.5.1.11 Scenario 13

Use Case: Student is warned about losing their focus
Primary Actor: Student and Instructor

Entry Condition: Student is not following the lecture, the system (namely, Teaching Assistant) detects the situation and sends notification to first, Student and then the Instructor.

Exit Condition: Student confirms that he/she is following the lecture by clicking on "I am Here!" button or the Instructor is notified.

Main Flow of Events:

1. The system detects a Student is not following the lecture.
2. Student is prompted with a confirmation alert about following the course.
3. Student clicks on "I am Here!" button.

Alternative Flow of Events:

1. The system detects a Student to be distracted.
2. Student is prompted with a confirmation alert about following the course.
3. Student missed the prompt.
4. Instructor is notified that Student is not following the lecture.

3.5.1.12 Scenario 14

Use Case: Student navigates in the shared slide and then synchronizes with the Instructor's current page.

Primary Actor: Student

Entry Condition: Student is in a lecture and Instructor is sharing a slide. Student clicks on the "Previous Slide" button.

Exit Condition: Student clicks on "Synchronize with Instructor" button

Main Flow of Events:

1. Student clicks on the "Previous Slide" button.
2. Student makes a few iterations using "Previous Slide" and "Next Slide" buttons to look up for something catch up with the lecture.
3. Student clicks on the "Synchronize with Instructor" button.

3.5.2 Use-Case Model

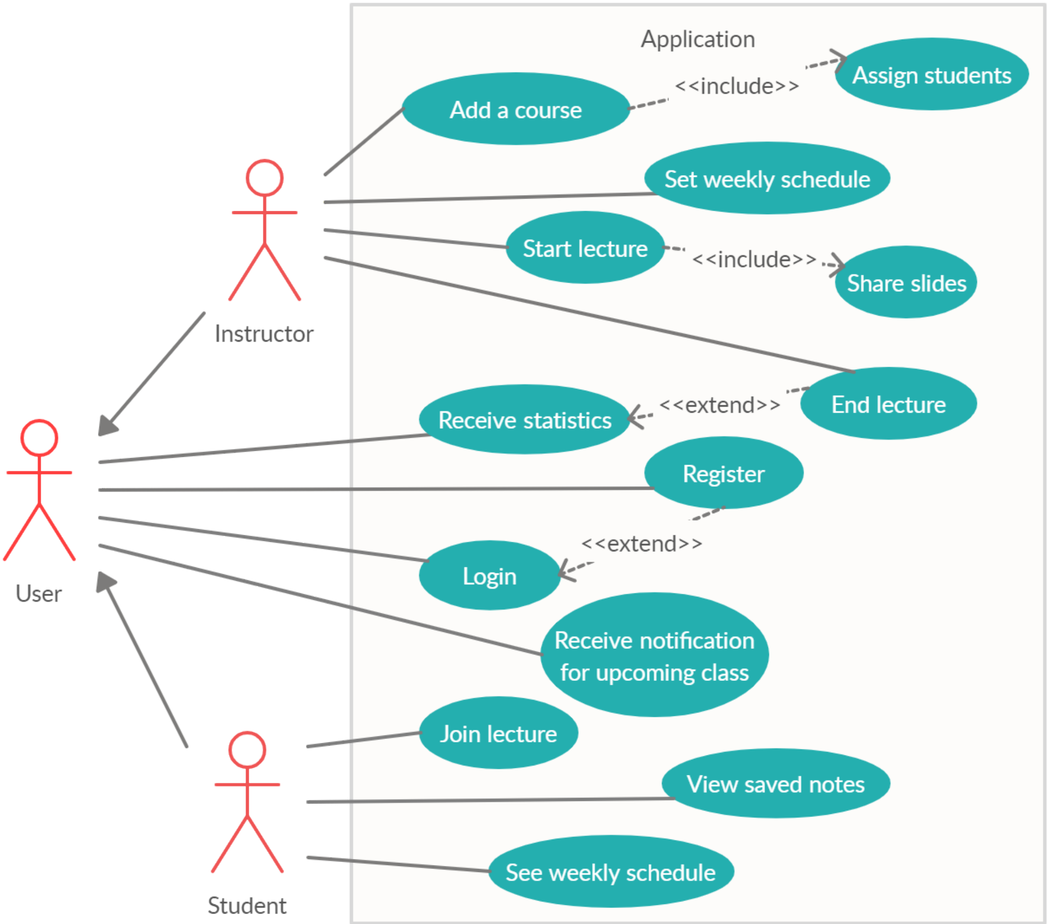


Figure 1: Use Case Model of the Application

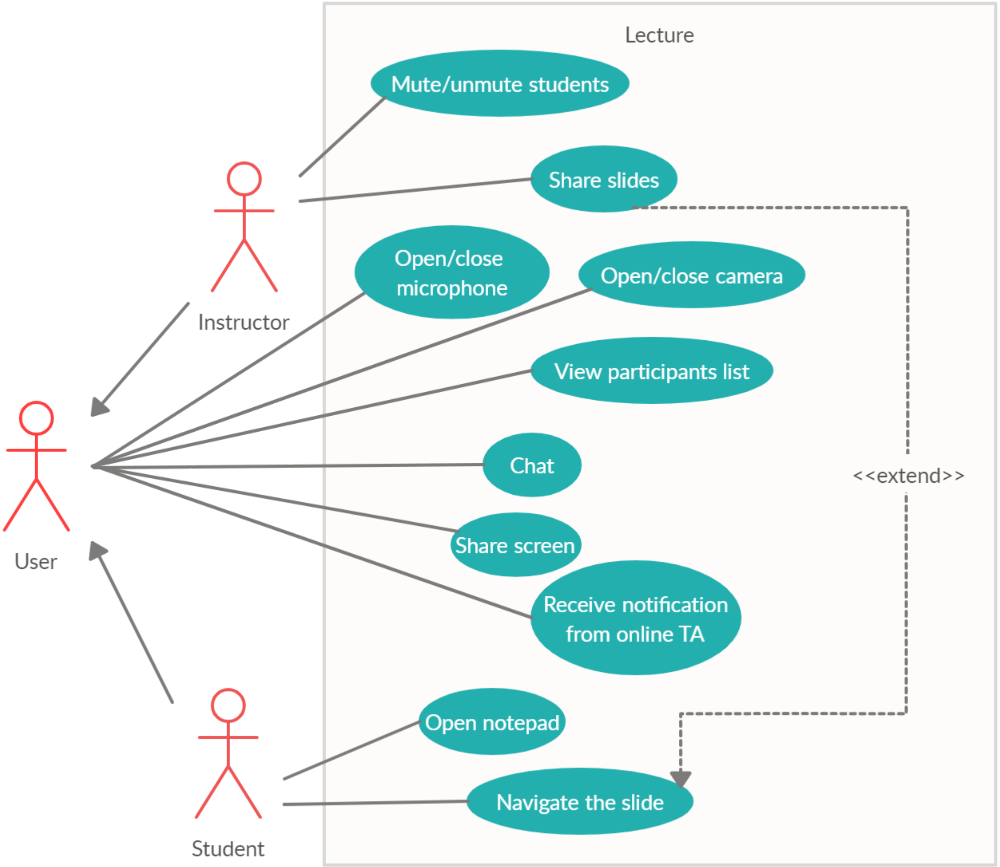
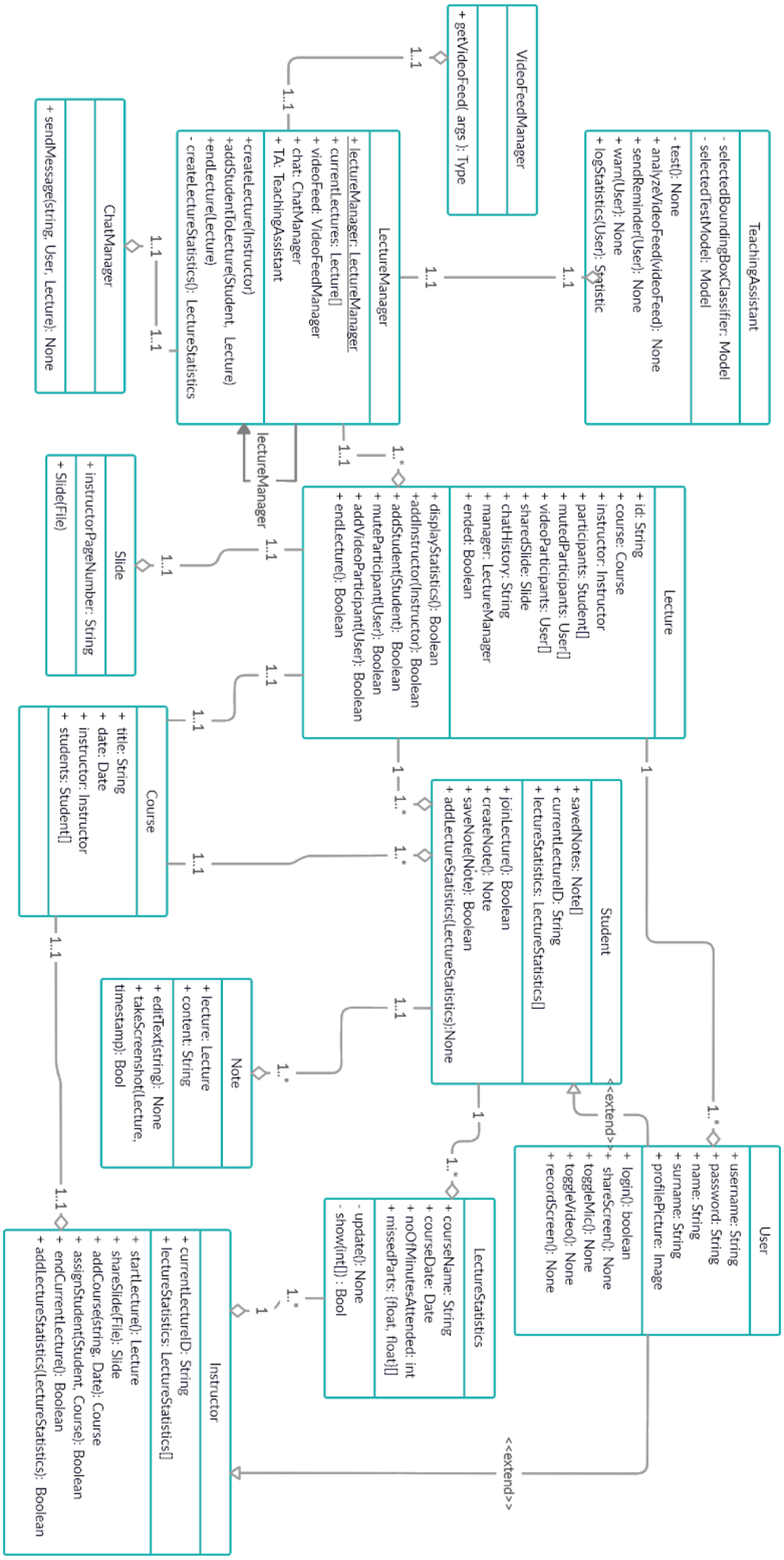


Figure 2: Use Case Model of a Lecture

3.5.3 Object and Class Model



TeachingAssistant
- selectedBoundingBoxClassifier: Model
- selectedTextModel: Model
- test(): None
+ analyzeVideoFeed(videoFeed): None
+ sendReminder(User): None
+ warn(User): None
+ logStatistics(User): Statistic
VideoFeedManager
+ getVideoFeed(args): Type
LectureManager
+ lectureManager: LectureManager
+ currentLectures: Lecture[]
+ videoFeed: VideoFeedManager
+ chat: ChatManager
+ TA: TeachingAssistant
+createLecture(Instructor)
+addStudentToLecture(Student, Lecture)
+endLecture(Lecture)
- createLectureStatistics(): LectureStatistics
ChatManager
+ sendMessage(string, User, Lecture): None
Slide
+ instructorPageNumber: String
+ Slide(File)
Lecture
+ id: String
+ course: Course
+ instructor: Instructor
+ participants: Student[]
+ mutedParticipants: User[]
+ videoParticipants: User[]
+ sharedSlide: Slide
+ chatHistory: String
+ manager: LectureManager
+ ended: Boolean
+ displayStatistics(): Boolean
+addInstructor(Instructor): Boolean
+ addStudent(Student): Boolean
+ muteParticipant(User): Boolean
+ addVideoParticipant(User): Boolean
+ endLecture(): Boolean
lectureManager
Course
+ title: String
+ date: Date
+ instructor: Instructor
+ students: Student[]
Student
+ savedNotes: Note[]
+ currentLectureID: String
+ lectureStatistics: LectureStatistics[]
+ joinLecture(): Boolean
+ createNote(): Note
+ saveNote(Note): Boolean
+ addLectureStatistics(LectureStatistics):None
Note
+ lecture: Lecture
+ content: String
+ editText(string): None
+ takeScreenshot(Lecture, timestamp): Bool
User
+ username: String
+ password: String
+ name: String
+ surname: String
+ profilePicture: Image
+ login(): boolean
+ shareScreen(): None
+ toggleMic(): None
+ toggleVideo(): None
+ recordScreen(): None
LectureStatistics
+ courseName: String
+ courseDate: Date
+ noOfMinutesAttended: int
+ missedParts: [float, float][]
- update(): None
- show(int[]) : Bool
Instructor
+ currentLectureID: String
+ lectureStatistics: LectureStatistics[]
+ startLecture(): Lecture
+ shareSlide(File): Slide
+ addCourse(string, Date): Course
+ assignStudent(Student, Course): Boolean
+ endCurrentLecture(): Boolean
+ addLectureStatistics(LectureStatistics): Boolean
<<extend>>
<<extend>>
1..1
1..1
1..1
1..1
1..1
1..1
1..*
1..1
1..1
1..1
1..1
1..1
1..1
1
1..*
1..*
1..*
1..*
1
1..*
1..1
1..1
1..*
1
1..1

Figure 3: Object Class Diagram

3.5.3.1 User Class

User class represents a participant in the lecture.

3.5.3.2 Student Class

Student class is a subclass of the User class. Student class differs from User as it can join a lecture, take notes and receive different lecture statistics.

3.5.3.3 Instructor Class

Instructor class is also a subclass of the User class. An instructor can start a lecture, share slides during a lecture.

3.5.3.4 Course Class

Course class is an abstraction for a usual course. It is associated with an instructor and assigned to students. Assigned students receive notification when a course is due.

3.5.3.5 Lecture Class

Lecture class represents an online course session. It is managed by a LectureManager. Lectures are hosted by Instructors. Students can join Lectures. Lectures can have shared Slides uploaded by an Instructor.

3.5.3.6 LectureStatistics Class

LectureStatistics Class instances are created at the end of the lectures and they store the statistics about the students and their attendances during that particular lecture.

3.5.3.7 Note Class

Note Class instances store the notes taken by the students.

3.5.3.8 Slide Class

Slide Class instances store the slides uploaded by the instructors.

3.5.3.9 VideoFeedManager Class

Manages the video and audio inputs received by Instructor and Students during a Lecture.

3.5.3.10 ChatManager Class

Manages the chatroom instance in the current session, responsible for creating, destroying and updating the chat room.

3.5.3.11 TeachingAssistant Class

TeachingAssistant class uses a selected face bounding box producer model (either the Haar Cascades model or the Histogram of Oriented Gradients + Linear SVM model in OpenCV, depending on their performance metrics and speed) in order to localize the face of a person. After localization, it uses a selected model (either the DLib Facial Landmark Detector that uses an ensemble of regression trees or the Tensorflow Convolutional Neural Network model, depending on their performance metrics and speed) in order to analyze the video feed of students. Also, it uses the YOLOv3 model in the Darknet CNN framework to recognize objects such as mobile phones in order to detect distracted students. Then, it warns these students and informs the instructor according to the test results.

3.5.3.12 LectureManager Class

LectureManager is the manager class that holds the chat and video feed managers in addition to current Lectures. It serves as a connection between all the manager classes.

3.5.4 Dynamic Models

3.5.4.1 Activity Diagrams

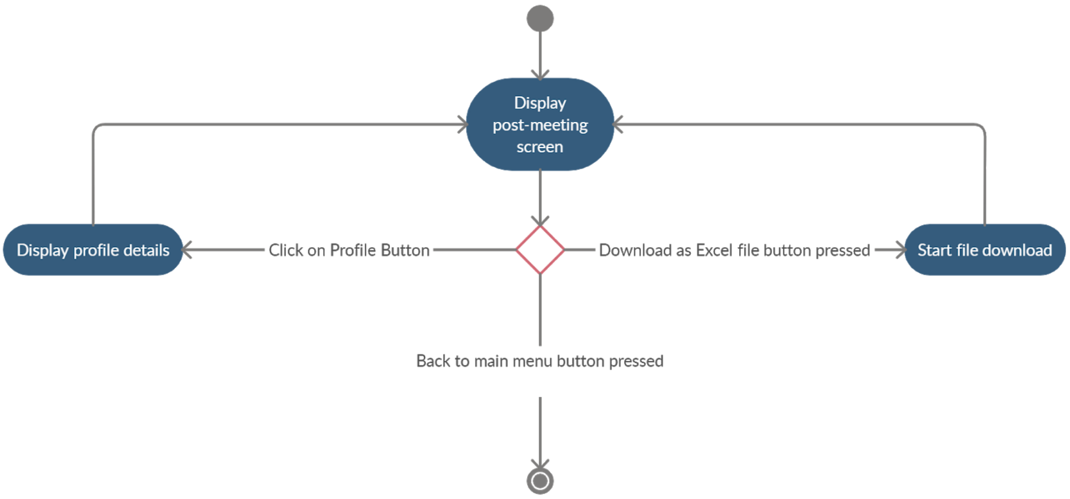

Display post-meeting screen
Display profile details
Click on Profile Button
Download as Excel file button pressed
Start file download
Back to main menu button pressed

Figure 4: Activity Diagram of Post Lecture Statistics

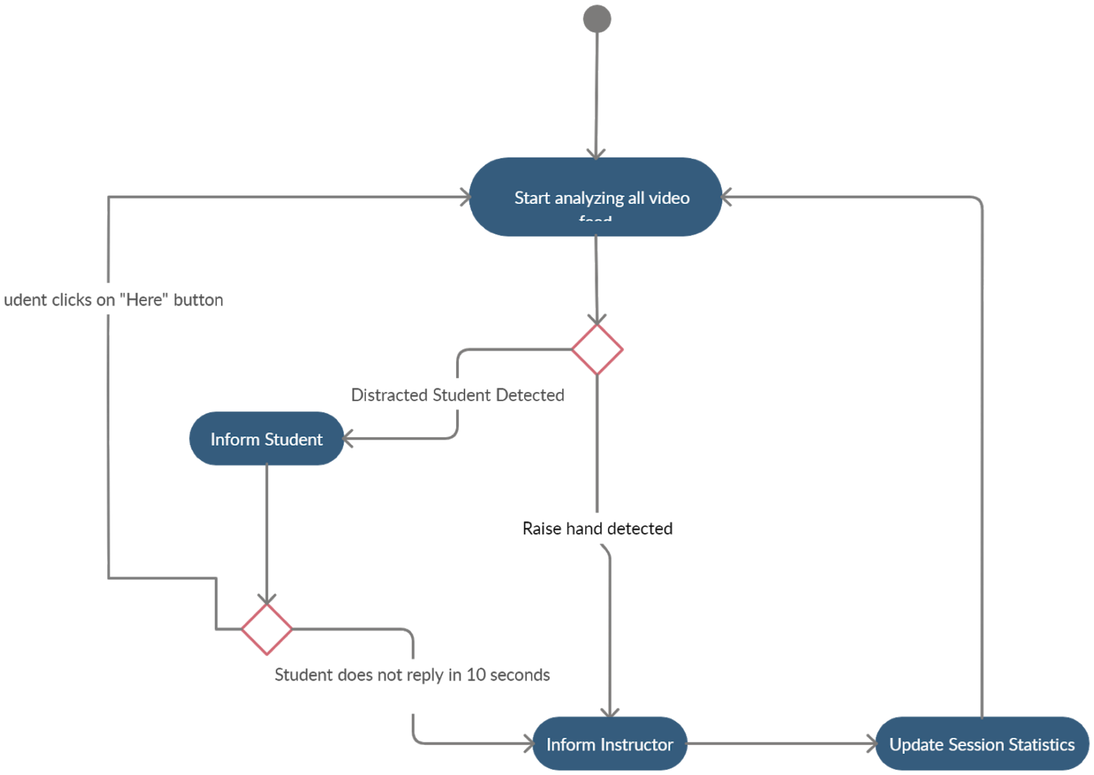

Start analyzing all video feed
udent clicks on "Here" button
Distracted Student Detected
Inform Student
Raise hand detected
Student does not reply in 10 seconds
Inform Instructor
Update Session Statistics

Figure 5: Activity Diagram of TA

3.5.4.2 State Machine Diagrams

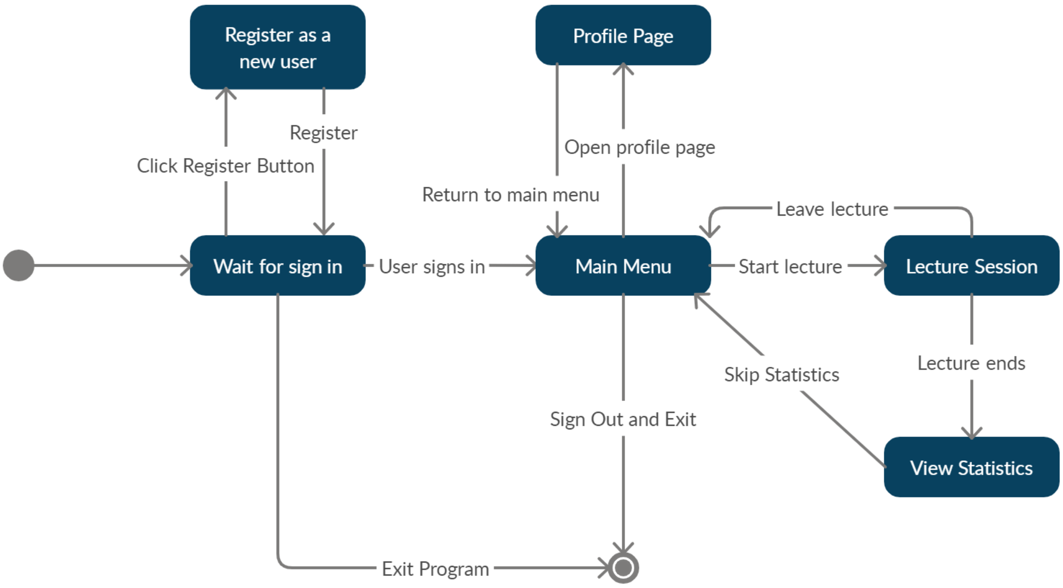


Figure 6: State Machine Diagram of a User

3.5.4.3 Sequence Diagrams

Sequence Diagram 1 (Fig. 7): Teaching Assistant constantly analyzes the video feed acquired from the lecture using VideoFeedManager. If a distracted Student is detected by the assistant, the Student is prompted with a dialog to check if they are following the lecture. If Student succeeds in confirming their presence, Teaching Assistant takes no more actions. Otherwise, the Instructor is notified about the Student's behavior.

Sequence Diagram 2 (Fig. 8): Lifetime of a Lecture begins with an Instructor's call to start the lecture. Then, LectureManager creates a Lecture instance and assigns the caller Instructor as the instructor of the session. After the creation of the lecture, students are expected to join the lecture. When a Student joins the lecture, LectureManager is responsible for adding the Student to the Lecture. When the class is over, Instructor finishes the lecture by making a call to LectureManager. LectureManager both removes the Lecture instance and creates LectureStatistics. Those statistics are sent to the Student and Instructor and displayed on their screens.

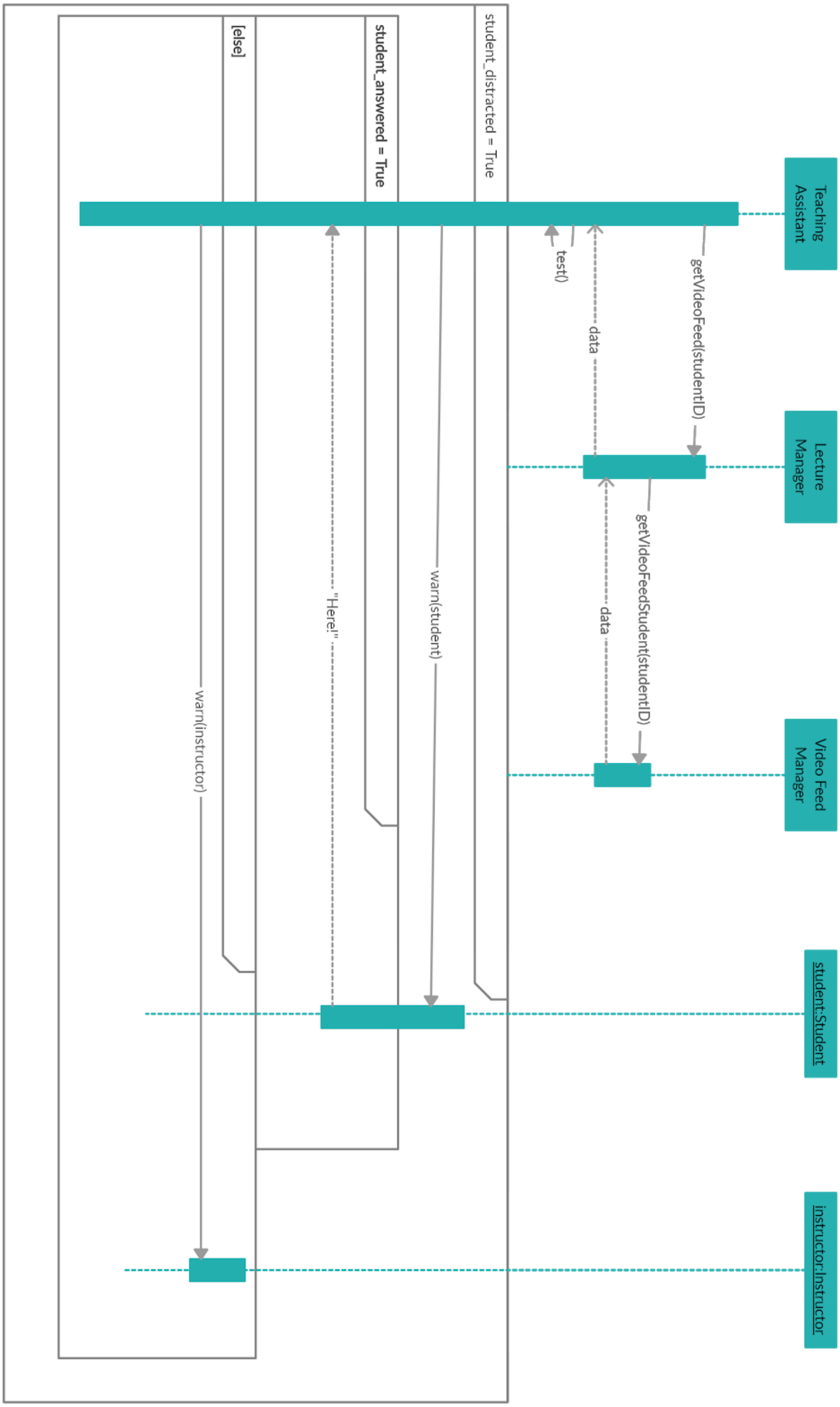


Figure 7: Sequence Diagram

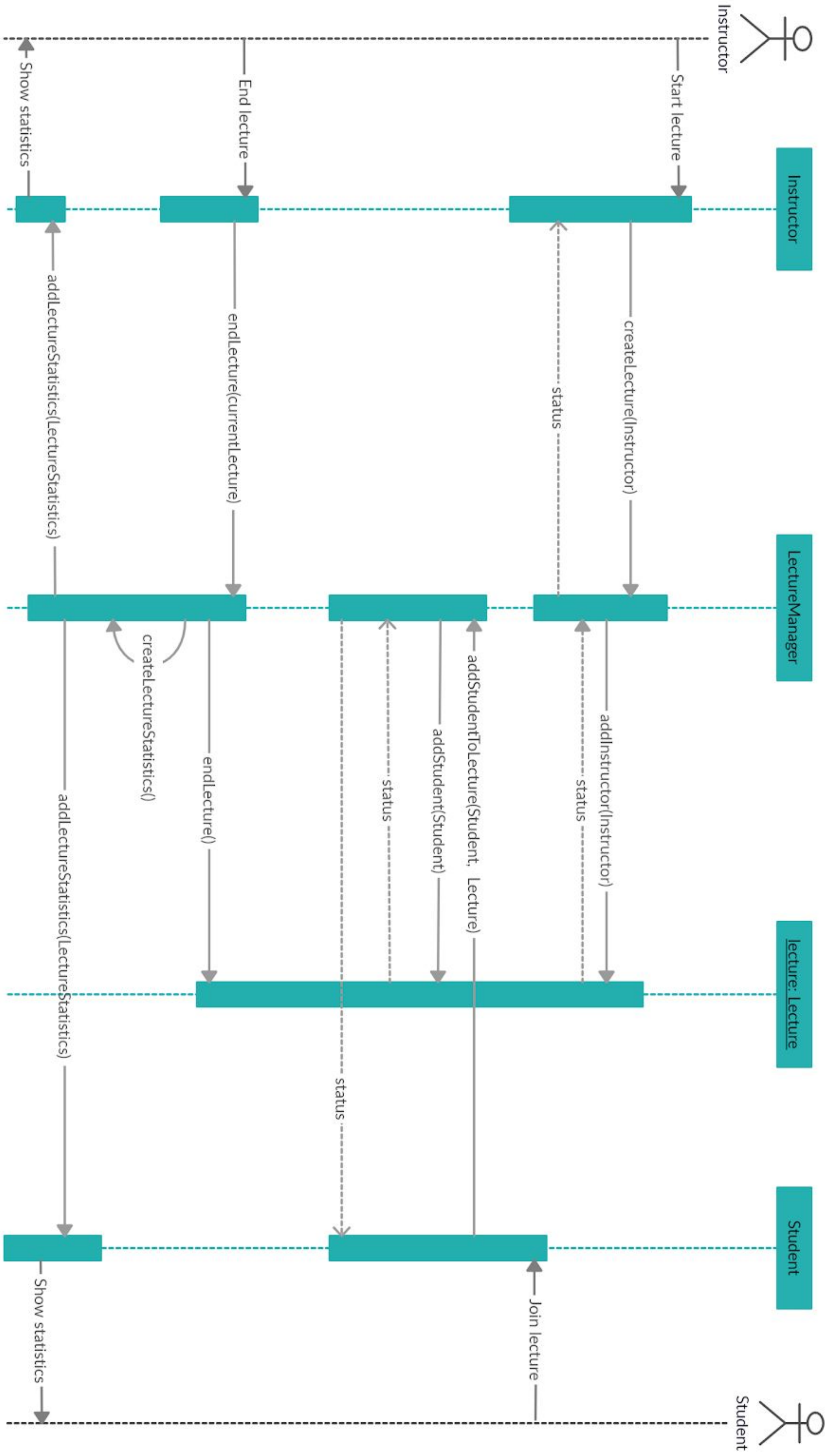

Instructor
Instructor
LectureManager
lecture: Lecture
Student
Student
Start lecture
createLecture(Instructor)
status
addInstructor(Instructor)
status
addStudentToLecture(Student, Lecture)
addStudent(Student)
status
status
Join lecture
End lecture
endLecture(currentLecture)
endLecture()
createLectureStatistics()
addLectureStatistics(LectureStatistics)
addLectureStatistics(LectureStatistics)
Show statistics
Show statistics

Figure 8: Sequence Diagram of a Lecture

3.5.5 User Interface

3.5.5.1 Login Page

The program will open with the sign in page. Users can enter their usernames or emails and passwords and press the "Login" button in order to sign in to the system and get to the main page. If the user does not have an account, they can click on the "Sign Up Here" button to be directed to the "Register Page".

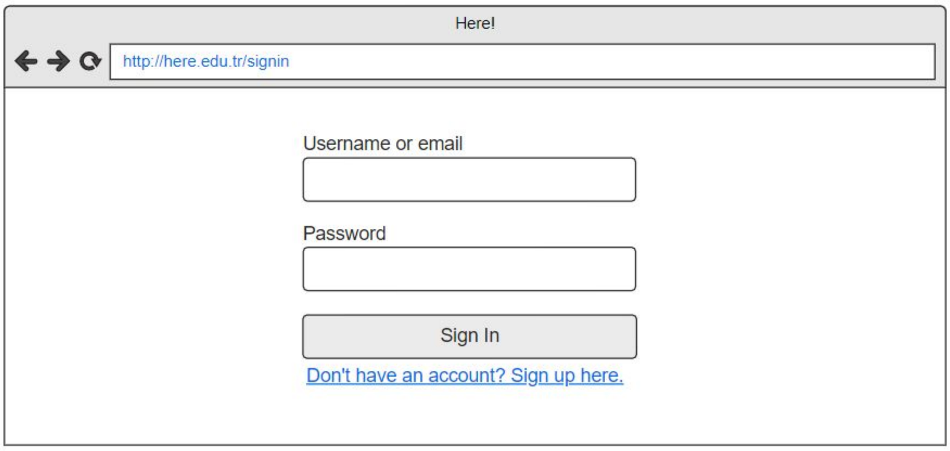

Here!
http://here.edu.tr/signin
Username or email
Password
Sign In
Don't have an account? Sign up here.

Figure 9: Sign In Page

3.5.5.2 Register Page

Register Page includes all the necessary information that the user has to input in order to create an account. After entering all the necessary information, users can press "Sign Up" button in order to create their accounts and the system will automatically redirect them to the "Login Page".

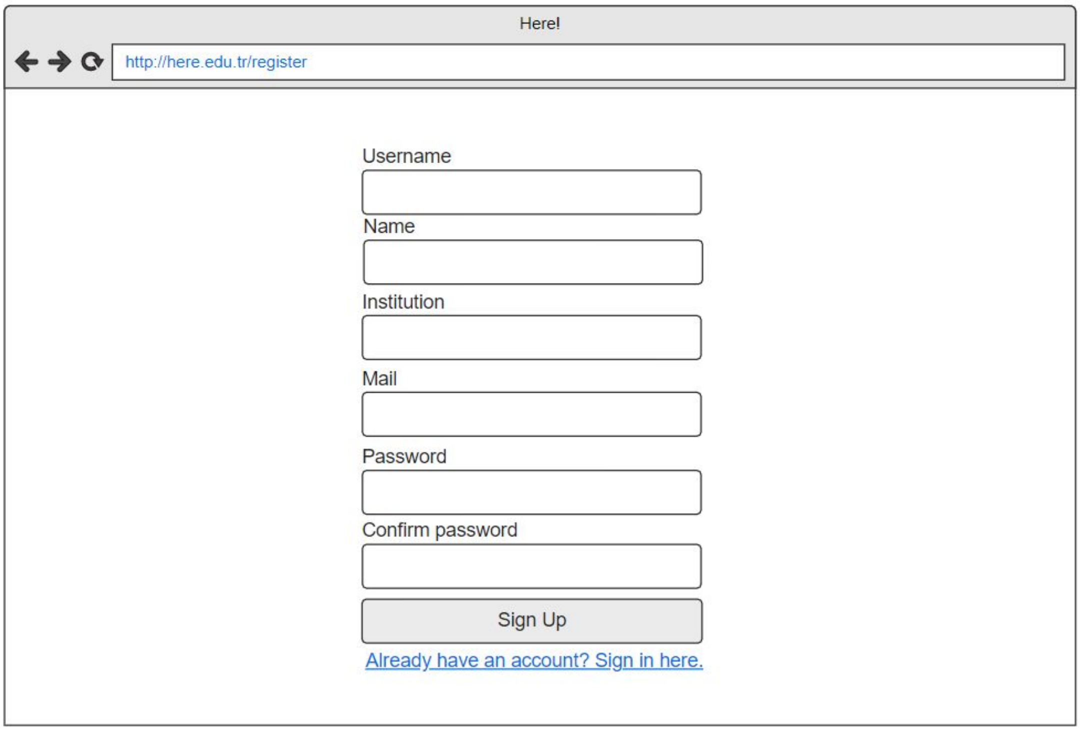

Here!
http://here.edu.tr/register
Username
Name
Institution
Mail
Password
Confirm password
Sign Up
Already have an account? Sign in here.

Figure 10: Register Page

3.5.5.3 Main Page

Once a user logs in to the program, this page will greet them. "Main Page" includes a weekly schedule, where the classes are displayed on the cell that corresponds to its time and day. Students can click on these lectures to join a session. Students can also see today's upcoming classes and can click on them to join a session if they choose to do so. Joining a session will open the "Lecture" page. There will also be an icon that shows the profile picture and the name of the student, and will direct the user to the "Profile" page when clicked.

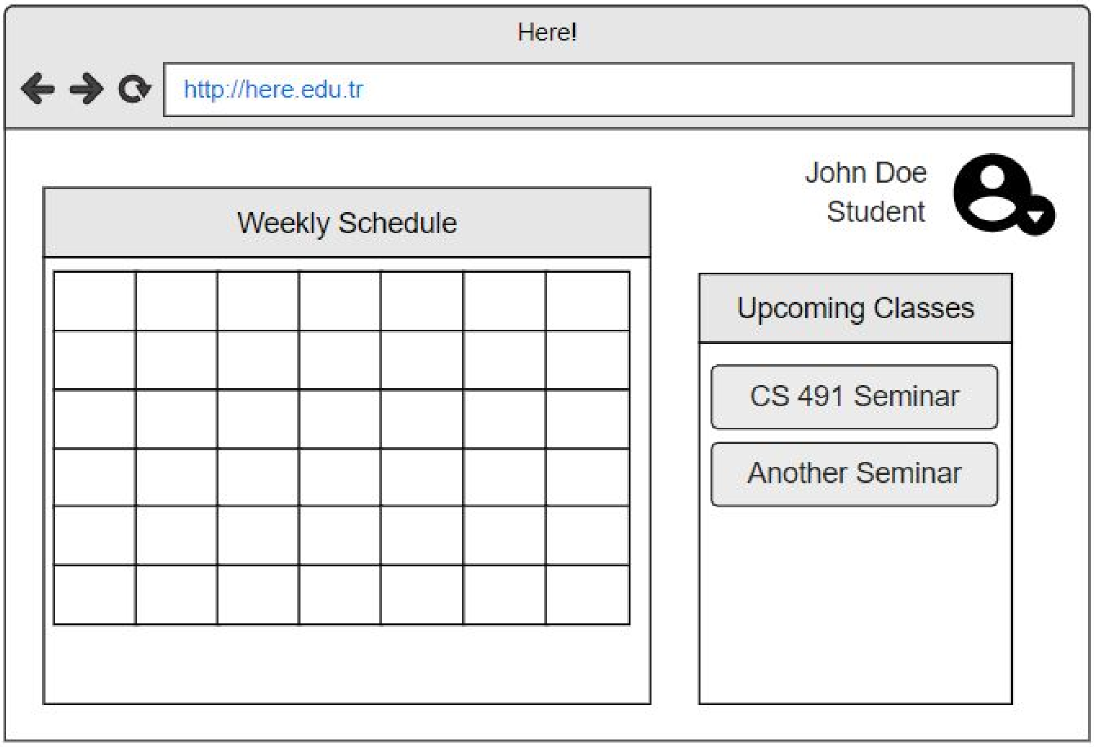
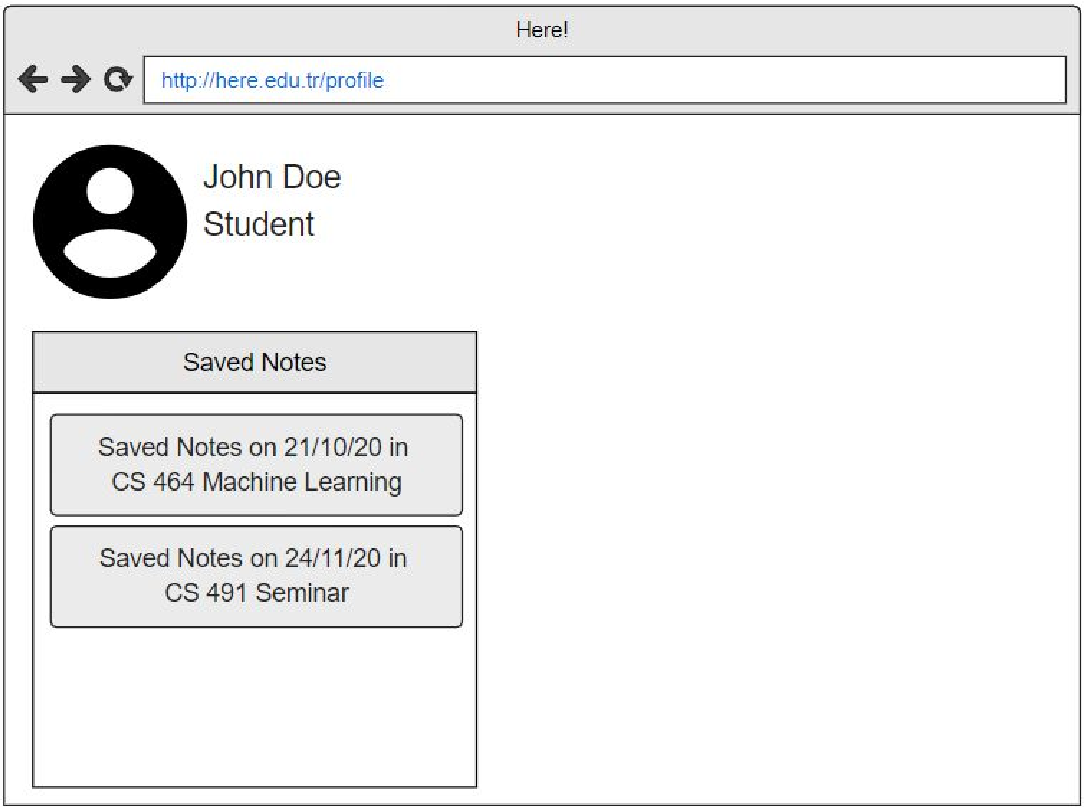


Figure 11: Chat box and TA alerts from students perspective

3.5.5.4 Profile Page of Student

Once the student clicks on their avatar on the Main Page, they will be directed to this page. “Profile Page” of a student includes a profile picture and name of the student at the top of the page and saved notes of the student at the center. Students can see their saved notes that they have taken in previous lectures. Notes are presented together with the date and the name of the lecture that it was taken. A saved note will be displayed on a popup screen when a student clicks on the note in the “Saved Notes” list.



Here!
http://here.edu.tr/profile
John Doe
Student
Saved Notes
Saved Notes on 21/10/20 in CS 464 Machine Learning
Saved Notes on 24/11/20 in CS 491 Seminar

Figure 12: Chat box and TA alerts from students perspective

3.5.5.5 Profile Page of Instructor

Once the instructor clicks on their avatar on the Main Page, they will be directed to this page. “Profile Page” of an instructor includes a profile picture and name of the instructor at the top of the page and courses of the instructor at the center. Instructors can see their courses on the “Courses” section and they can add a new course using the “+” button at the bottom. They can also assign students to a course by entering student ID’s on a popup screen which is displayed after clicking on the course name.

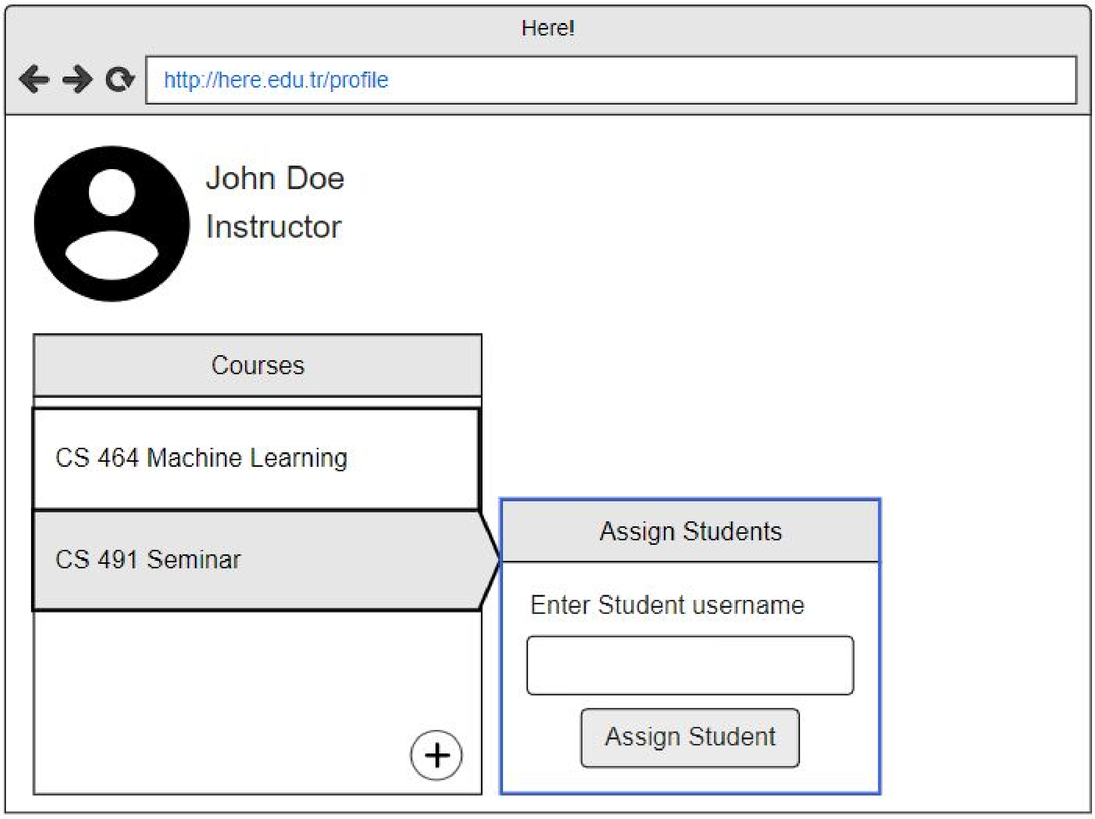


Figure 13: Chat box and TA alerts from students perspective

3.5.5.6 Lecture Overview

Students and the instructor meet in the Lecture page, which is essentially a classroom simulation. The center of the screen is divided into two sections: the “Shared Slides” section and the “Shared Screen” section. If an instructor wishes to upload a slide during or before the class, that slide will appear on the “Shared Slide” section. As the instructor navigates the slides, the shared slide section will also update, but by clicking on the arrows, the student can move forward or backward in any given time. Students can click on the “Sync” button in the middle to catch up with the current slide. The screen sharing section is straightforward, if any of the students or the instructor wishes to share their screen, it will appear in that section of the screen. Above, there are multiple buttons which can be used to adjust different settings. They can record the current session, click the “Notes” button to open the “Notepad” page, click the “Chat” button to open the chat screen, click “Settings” button in order to open session settings, click on camera and microphone buttons to turn them on/off, click on “Share Screen” button to turn screen sharing on or off, click “Participant List” in order to see the list of participants, and finally click “Leave Meeting” in order to exit from the current session. On the upper right corner, the icon for the TA can be seen. Users cannot normally interact with the TA, but on given occasions, TA will notify a user with a warning or an information, then the user will be able to click on it to satisfy a given condition. At the bottom of the screen, the participants are visible. The system will be able to detect certain gestures, like raising hands, and the instructor can see the participants with their hands raised as the system will include an icon on top of their mini participant videos. The “Shared Slide” sections are different for the instructor, as they can edit the shared slides as they wish. They can edit the slides, write on them or delete their writings, which will be visible for every student in the session.

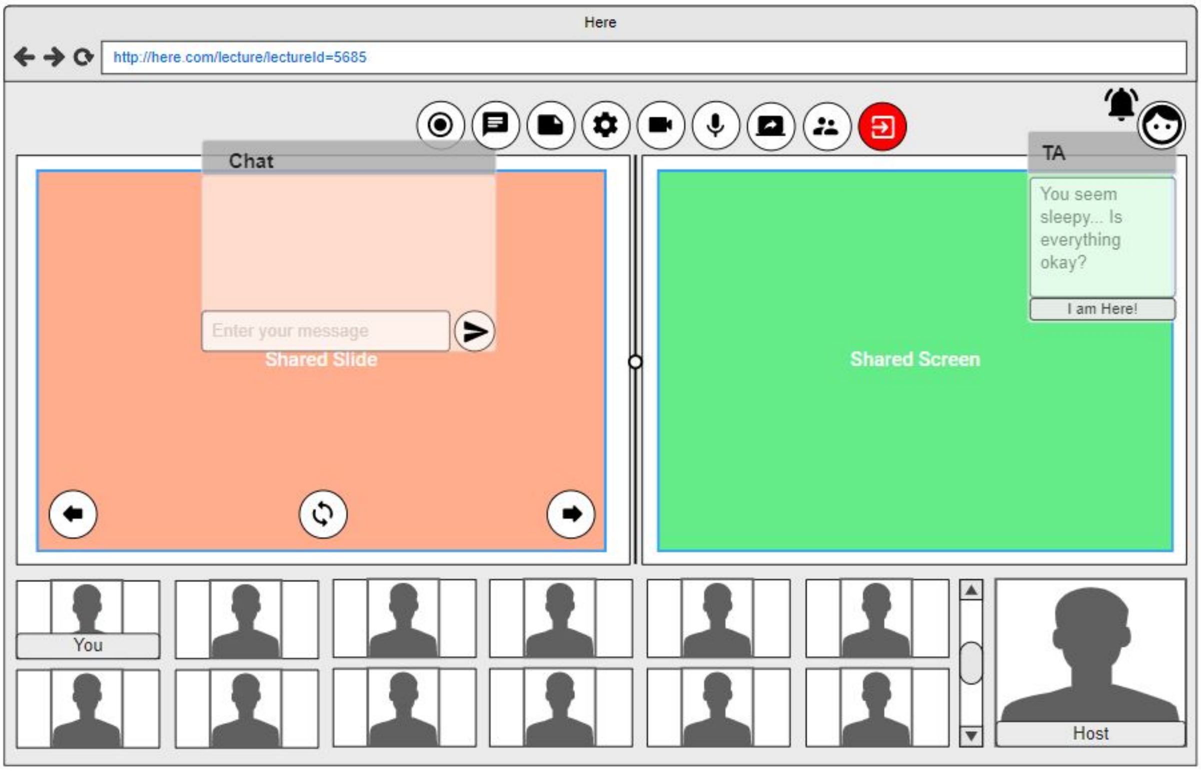

Here
http://here.com/lecture/lectureId=5685
Chat
Enter your message
Shared Slide
TA
You seem sleepy... Is everything okay?
I am Here!
Shared Screen
You
Host

Figure 14: Lecture Overview from students perspective

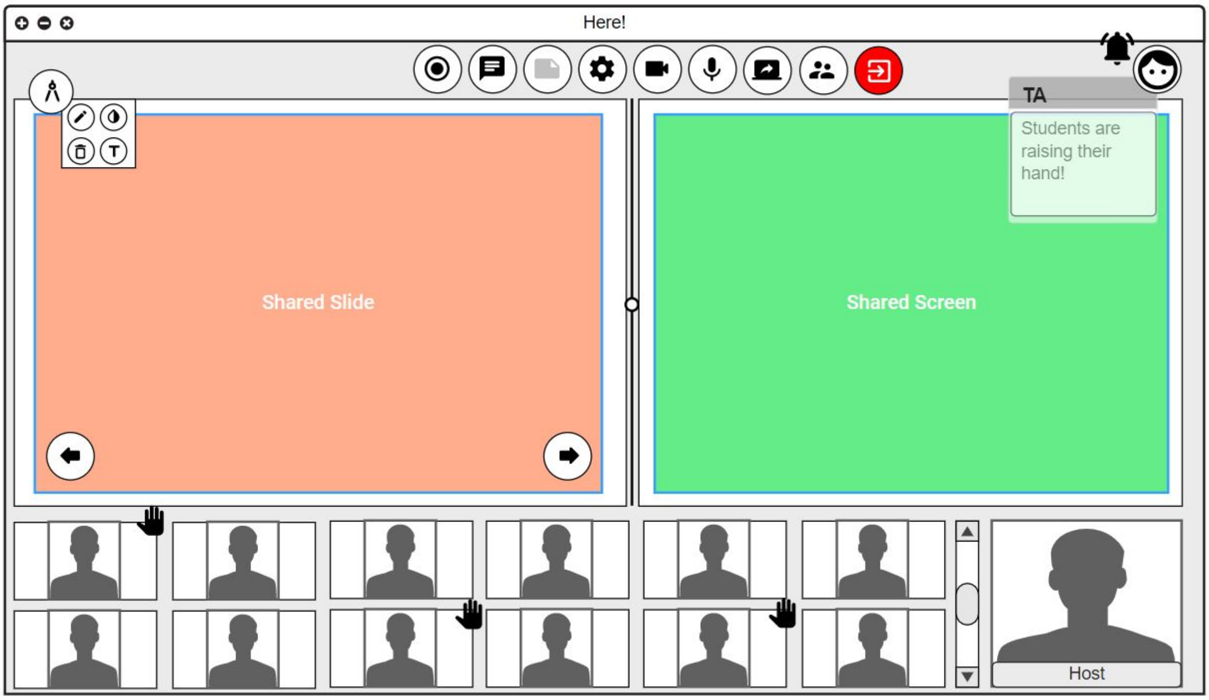


Figure 15: Lecture Overview from instructor’s perspective

3.5.5.7 Post Lecture Statistics

During each lecture, the teaching assistant keeps useful information about the lecture for both students and instructors. Information for instructors include attendance of students, attention span of students, distribution of notifications from teaching assistant. Students receive information about their personal attention span, list of possibly missed parts of lecture during their distraction.

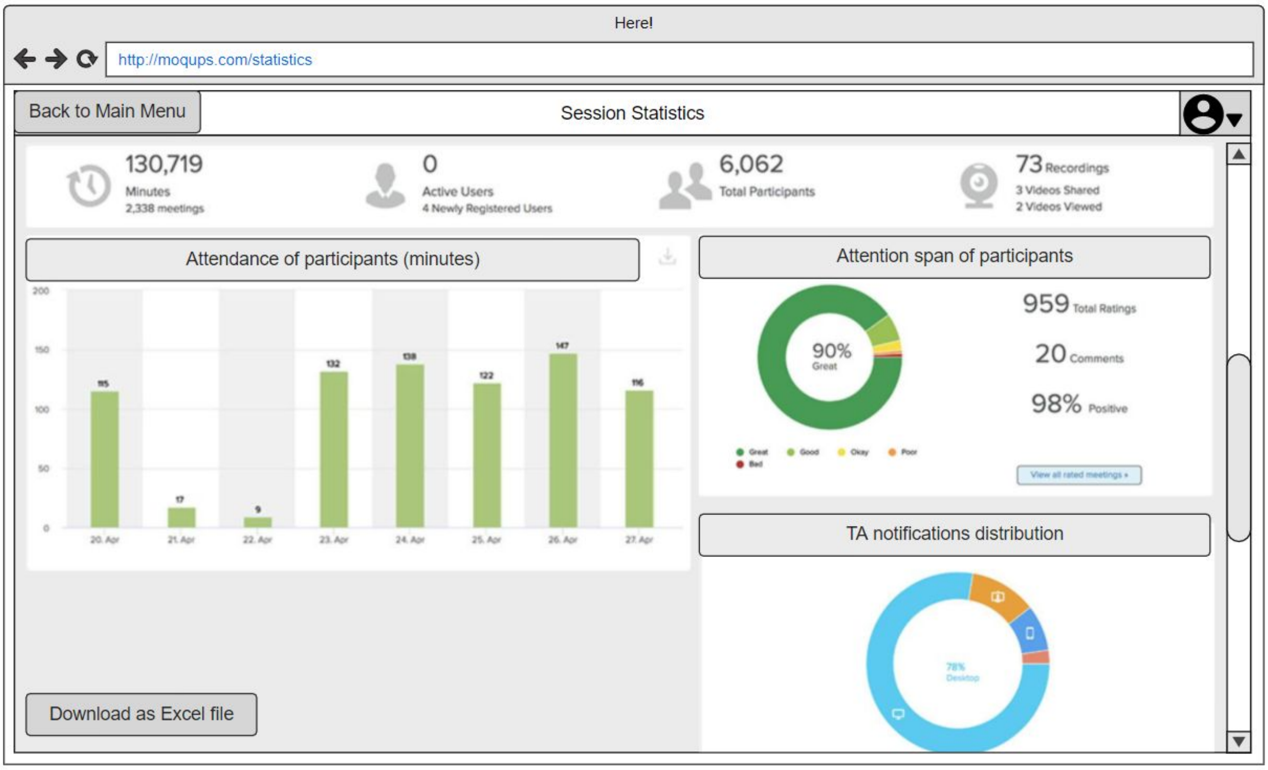


Figure 16: Post lecture statistics from instructor’s perspective

8. Other Analysis Elements

8.1 Consideration of Various Factors in Engineering Design

Public Health: Public health is the motivation of our project. Due to ongoing pandemic, it is crucial that the students remain in their homes, as being in a real classroom is a huge health hazard. Our project ensures that the students remain in isolation while trying to provide an education tool that resembles the feeling of an actual classroom.

Public Safety: Due to the ongoing pandemic, it is essential, regarding public safety, to stay isolated. Due to this factor, our team cannot get together in the same environment as often as we would like, which is making the development process harder in terms of cooperation and teamwork.

Public Welfare: A country’s welfare state affects its average internet connection speeds and accessibility of better electronic devices. Both affect user experience quality of our program, as our program relies on visual data communication, which can be an expensive process regarding internet connection. On the other hand, currently our program is free to use, so it has no real effect on public welfare.

Technological Factors: Technology is constantly developing, and this is an important thing to consider for our project. We need to keep track of this development and implement it to our project so that our work is not obsolete. We also need to look out for similar products, and develop our application accordingly so that our product can add something to the table.

Cultural Factors: Clothings and accessories may differ from culture to culture. These differences affect motion tracking and recognition of facial expressions.

Social Factors: Users need to interact with the program using a language they are familiar with. In order to make our program available world-wide, we need to consider translating our program to different languages. Our program will be English as default, and language might be an issue since approximately 20% of the Earth's population can speak English.

	Effect level	Effect
Public health	10	Public health crisis motivating our project idea
Public safety	5	Hindering the cooperation and teamwork aspect of development due to isolation.
Public welfare	5	Change in user experience quality based on purchasing power
Technological factors	10	Change in design due to support for technologies we use and existence of similar applications
Cultural factors	4	Change in accuracy of image processing because of cultural differences on appearance
Social factors	6	Language barriers between users and the program

Table 1: Factors that can affect analysis and design.

8.2 Risks and Alternatives

After discussions that were held between the team members, we envisioned 3 possible risks that might occur in our project and came up with a plan B for each one to handle those risks.

The first risk that we predicted is the scope of the project being too broad. During the project, all members will need to learn a lot of new concepts. Since we have limited time, this process needs to be quick. However, some knowledge that the project will require might not be as easy to obtain. This may cause delays in the development of the project and even incompletion of some features. In such a case, we might implement an extension to an existing video conference program, instead of creating a new one as we planned.

The second risk is the schedule flaws. It is hard to estimate the required time for a task in software development. Underestimating the required time for a task causes delay for future tasks. Thus, the whole project plan might require revising. We can avoid such consequences by adopting an agile method of development. Doing so, estimates are made for shorter terms according to feedback and future tasks are decided regarding the completion of current ones.

The third risk is absence of a member. We are experiencing a pandemic at the moment. It is possible to contract coronavirus. In addition, team members can fail or withdraw the course. In such cases, the workforce is reduced and remaining members are left with more workload than before. Possible alternatives are making changes on the planned features or redistributing the work among remaining members.

	Likelihood	Effect on the project	B Plan Summary
Scope of the Project Being Too Broad	Medium	Insufficient resource to implement all planned features	Implementing an extension to an existing video conference instead of creating a new video conference program
Schedule Flaws	Medium	Inaccuracy in the future planning	Agile method and planning for shorter periods according to feedbacks
Absence of a Member	Low	Reduce in workforce causes falling behind the planned schedule	Redistribute the work among the remaining members

Table 2: Risks

8.3 Project Plan

As part of the planning activity, we listed our project goals and based on these goals, we divided the overall project into work packages (WPs). We assigned a leader and members to work on each WP and planned the start and end dates, objectives, tasks and deliverables for these WPs.

WP#	Work package title	Leader	Members involved
WP1	Project Specifications Report (Done)	Burak	Everyone
WP2	Project Website(Done)	Ozan	Büşra
WP3	Analysis Report	Yüce	Everyone
WP4	Motion Tracking Implementation using Machine Learning Models	Mert	Ozan, Büşra
WP5	GUI Implementation	Yüce	Büşra, Burak
WP6	WebRTC API Connection	Büşra	Yüce, Burak
WP7	High-level Design Report	Burak	Everyone
WP8	Server Communication	Büşra	Ozan,Mert
WP9	Educational Utilities	Ozan	Yüce, Büşra
WP10	Low-level Design Report	Mert	Everyone
WP11	Testing	Burak	Everyone
WP12	Final Report	Ozan	Everyone
WP13	Final Presentation	Büşra	Everyone

Table 3: List of work packages

WP 1: *Project Specifications Report*			
Start date: *16.09.2020* **End date:** *5.10.2020*			
Leader:	*Burak*	**Members involved:**	*Everyone*
Objectives: *Creation of project specifications report. Reporting the constraints of the system. Documenting the functional and non-functional requirements*			
Tasks: ***Task 1.1 Constraints* :** *Deciding on the constraints of the system. These constraints include implementation, economic, social, ethical etc. constraints.* ***Task 1.2 Functional Requirements:*** *Deciding on the functional requirements of the system.* ***Task 1.3 Non-functional Requirements*:** *Deciding on the non-functional requirements of the system. These include the security, usability, performance etc. non-functional requirements* *…*			
Deliverables ***D1.1:*** *Project Specifications Report*			
WP 2: *Creating a Website*			
Start date: *3.10.2020* **End date:** *5.10.2020*			
Leader:	*Ozan*	**Members involved:**	*Büşra*
Objectives: *To create a website that will be used to provide information and documentation about our project.*			
Tasks: ***Task 2.1 Picking a Host* :** *Deciding where to host our website.* ***Task 2.2 Picking a Theme*** *Picking an appropriate and easy to use theme for our website.* *…*			
Deliverables ***D2.1:*** *A fully functioning, easy to navigate website that displays the documentation of our project.*			
WP 3: *Analysis Report*			
Start date: *4.11.2020* **End date:** *21.11.2020*			
Leader:	*Yüce*	**Members involved:**	*Everyone*
Objectives: *Creating the analysis report of the project. Researching the current systems and describing the proposed system using the requirements of the system and models. Inspecting other elements such as public health and safety considerations, ethical and professional responsibilities etc. to analyze the system.*			
Tasks: ***Task 3.1 Current Systems Research*:** *Researching the current systems. Comparing them and discovering their purposes.* ***Task 3.2 Proposed System Description* :** *Describing the proposed system and comparing with current systems. Modeling the system using diagrams.* ***Task 3.3 Inspecting other analysis elements* :** *Analyzing the effects of the system in societal concepts.*			
Deliverables ***D3.1:*** *Analysis Report*			
WP 4: *Motion Tracking Implementation using Machine Learning Models*			
Start date: *23.11.2020* **End date:** *8.12.2020*			
Leader:	*Mert*	**Members involved:**	*Ozan, Büşra*
Objectives: *Implementing the motion tracking system that includes facial landmark recognition, eye-tracking and object (such as phone) detection. Choosing the best machine learning model for real-time, high quality motion tracking*			
Tasks: ***Task 4.1 Localizing the Face of a Person Using OpenCV:*** *Producing a bounding box around the face using OpenCV. Trying different classifiers such as Haar cascades or Histogram of Oriented Gradients + Linear SVM method in OpenCV.* ***Task 4.2 Applying DLib Facial Landmark Detector:*** *Applying the regression trees model used in DLib library to predict shapes on a face. Acquiring the 68 different (x,y) coordinates of regions of interest points of a face.* ***Task 4.3 Applying a Convolutional Neural Networks Model:*** *Applying a Tensorflow Convolutional Neural Network model that was pre-trained on 5 datasets, which also outputs 68 face landmarks.* ***Task 4.4 Comparing Both Models:*** *Comparing both of these models and selecting the best performing one in terms of their performance metrics and their feasibility to be used as a real-time model for a video conferencing application.* ***Task 4.5 Extracting Eye Motion:*** *Finding the locations of eyeballs on a face and calculating their motion in order to detect distracted students.* ***Task 4.6 Extracting Head Motion:*** *Finding the pose of the head and analyzing its motion in order to detect distracted students.*			

Task 4.7 Object Detection: *Using YOLOv3 in Darknet Convolutional Neural Network Framework in order to detect objects like mobile phones in video feed to be able to detect distracted students checking their mobile phones.*			
Deliverables ***D4.1:*** *Fully functioning motion tracker and object detector software that will be the major portion of the online teaching assistant.*			
WP 5: *GUI Implementation*			
Start date: *23.11.2020* **End date:** *1.02.2021*			
Leader:	*Yüce*	**Members involved:**	*Büşra, Burak*
Objectives: *Implementing the web application using Javascript and creating a useful interface.*			
Tasks: ***Task 5.1 Starting the Project***: *Starting a new Angular project and creating Home and Profile Page pages.* ***Task 5.2 Implementing Home and Profile Page***: *Implementing required components excluding the web conference functionality.* ***Task 5.3 Integrating with the Database*** : *Establishing connection with database for profile info and weekly schedule.* ***Task 5.4 Implement Web Conference Utility*** : *Integrating webRTC to the web application.* ***Task 5.5 In-lecture Utilities*** : *Implementing in-lecture utilities such as slide-share, screen-share etc.* ***Task 5.6 Teaching Assistant*** : *Integrating teaching assistant to the application.*			
Deliverables ***D5.1:*** *Web application that provides all the functionality promised in Functional Requirements section*			
WP 6: *WebRTC API Connection*			
Start date: *16.12.2020* **End date:** *31.12.2020*			
Leader:	*Büşra*	**Members involved:**	*Yüce, Burak*
Objectives: *Implementing the WebRTC to stream video and audio in realtime from students and instructors.*			
Tasks: ***Task 6.1 Streaming Data From Input Devices***: *Streaming data from camera and microphone locally.* ***Task 6.2 Creating Peer-to-peer Connections***: *Establishing a stable connection between clients.* ***Task 6.3 Signaling and Session Control***: *Coordinating the communication and simultaneous data transfer between peers.*			
Deliverables ***D6.1:*** *WebRTC component to be used for acquiring data from users and sending information about the lecture session to them.*			
WP 7: *High-level Design Report*			
Start date: *23.12.2020* **End date:** *27.12.2020*			
Leader:	*Burak*	**Members involved:**	*Everyone*
Objectives: *Creating a high-level design report of the project. Analyzing the high level system structures, subsystems and data handling in the project, as well as considerations about the cultural, global , economical, social aspects of our high level design.*			
Tasks: ***Task 7.1 Subsystem Decomposition:*** *Creating a high level design of our project systems and subsystems.* ***Task 7.2 Data Management*** : *Figuring out how each system will communicate with each other, on top of how to store the data.* ***Task 7.3 Inspecting other analysis elements*** : *Inspecting our high level design in terms of global , social, economic and cultural elements.*			
Deliverables ***D7.1:*** *High-level Design Report*			
WP 8: *Server Communication*			
Start date: *16.12.2020* **End date:** *20.01.2021*			
Leader:	*Büşra*	**Members involved:**	*Ozan, Mert*
Objectives: *Create a functional server client system for the communication between devices.*			
Tasks: ***Task 8.1 Coding client and server components:*** *Create a communication system where the client sends requests to the server and the server responds to the client requests.* ***Task 8.2 Implementing Methods for Client Server Communication*** : *Implementing methods such as sockets, remote procedure calls and pipes.*			
Deliverables ***D8.1:*** *A functional server-client system*			
WP 9: *Educational Utilities*			
Start date: *25.01.2020* **End date:** *01.04.2021*			
Leader:	*Ozan*	**Members involved:**	*Yüce, Büşra*
Objectives: *Implementing the shared slides, note taking and the remaining functionalities of the TA, namely the GUI aspect and the displaying of statistics.*			
Tasks:			

***Task 9.1 Implementing shared slides*:** *Creating the shared slide system where students can iterate through the slide whenever they want while the instructor continues their lesson.* ***Task 9.2 Implementing note taking* :** *Creating the note taking system where a student can take notes during the lecture, with the added screenshot of the current slide and a timestamp.* ***Task 9.3 Displaying session statistics* :** *Displaying the data that is collected by the TA at the end of the session and making it accessible in the form of an Excel file download.*			
Deliverables ***D9.1:*** *Slide sharing system* ***D9.2:*** *Note taking system* ***D9.3:*** *Statistics Page*			
WP 10: *Low-level Design Report*			
Start date: *10.02.2021* **End date:** *15.02.2021*			
Leader:	*Mert*	**Members involved:**	*Everyone*
Objectives: *Creating the low-level design report of the project..*			
Tasks: ***Task 10.1 Design Choices*:** *Analyzing design trade-offs and compliance with engineering standards.* ***Task 10.2 Packages* :** *Creating low-level design of our project and identifying subsystem packages, creating package diagrams of subsystems.* ***Task 10.3 Class Interfaces* :** *Describe class attributes, method signatures, interfaces in detail*			
Deliverables ***D10.1:*** *Low-level Design Report*			
WP 11: *Testing*			
Start date: *01.04.2021* **End date:** *11.04.2021*			
Leader:	*Burak*	**Members involved:**	*Everyone*
Objectives: *Testing the implemented parts, and their integrity.*			
Tasks: ***Task 11.1 Testing the GUI*:** *Testing the user interface in different web browsers.* ***Task 11.2 Testing WebRTC API:*** *Testing the WebRTC in different internet connection speeds.* ***Task 11.3 Testing the Server Communication*:** *Testing the server communication with multiple users are connected.* ***Task 11.4 Testing the overall integrity:*** *Testing the integrity between different parts of the application.*			
Deliverables ***D11.1:*** *New tasks that require our attention*			
WP 12: *Final Report*			
Start date:. *06.04.2021* **End date:** *26.04.2021*			
Leader:	*Ozan*	**Members involved:**	*Everyone*
Objectives: *Creating the final report of the project, which includes the revised content from analysis and design reports, details about our implementation and test, and a maintenance plan.*			
Tasks: ***Task 12.1 Include Revised Information*:** *Analyse and revise the information from the previous two reports and add them to the final report.* ***Task 12.2 Implementation and Test Details* :** *Report the final details of the implementation of the program and the methods and results of our testing processes.* ***Task 12.3 Maintenance Plan* :** *Conduct a plan to maintain the program in the future.*			
Deliverables ***D12.1:*** *Final Report*			
WP 13: *Final Presentation*			
Start date: *01.05.2021* **End date:** *05.05.2021*			
Leader:	*Büşra*	**Members involved:**	*Everyone*
Objectives: *Preparing the presentation of the overall project and the demo of the finished product.*			
Tasks: ***Task 13.1 Share the presentation among teammates*:** *Sharing the workload for the presentation among team members.* ***Task 13.2 Creating a speech*:** *Prepare a speech for the final presentation.* ***Task 13.3 Creating the slides* :** *Creating the slides for the final presentation*			
Deliverables ***D13.1:*** *Slides for the final presentation* ***D13.2:*** *Speech for the final presentation.*			

In addition to our project plan, we created a Gantt chart to express our project schedule. Below is our Gantt Chart that indicates the time requirements of tasks together with their starting dates and deadlines:

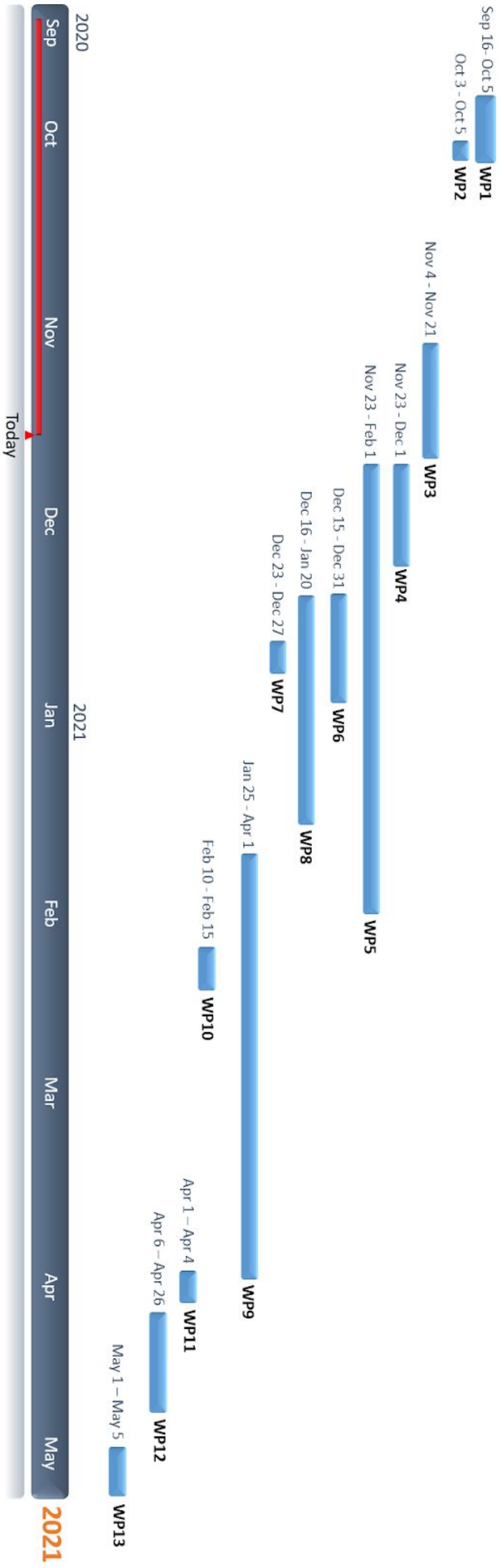


Figure 17: GANTT Chart for Project Schedule

8.4 Ensuring Proper Teamwork

To be able to ensure proper teamwork, we aim to include every team member into each and every step of the project process. We plan on constructing an inclusive and collaborative environment, where the project leadership is also shared among team members. The creation of such an environment will be made possible using team collaboration and project management tools in order to track the contribution of each team member. After discussions that were held between the team members, we decided on using GitHub as the primary project management tool.

Using the repository statistics feature of GitHub, we will collect enough evidence throughout the project to show peer contributions. During the project, we will constantly check the contribution rate of each member and warn each other whenever we feel like someone is being left out.

Initially we thought about using Trello as the task distribution software, but then we found out that the "Projects" feature of GitHub would be a better candidate, as it provided the same functionality while still being a part of GitHub.

8.5 Ethics and Professional Responsibilities

The first thing to mention about ethics and professional responsibilities is the **Global Impact** of the project. The main goal of *Here!* is to provide a better online education environment for students and instructors all over the world. Therefore, the program will be available to everyone regardless of their nation, country. Moreover it will be available on every platform as long as there is a mutual agreement with the platform. As *Here!* is released and started being used, we expect students and instructors to have an online class environment as comfortable and smooth as possible.

While developing *Here!*, we will also consider its **Economic** impact. As we believe in the importance of equal opportunities in education, Here! will work without any profit motive .

Here! will also have an impact on the **Environment**. In face-to-face education, instructors handout printed material to the students. This causes a significant amount of paper consumption. As the number of classes that use *Here!* increases, we expect an increase in the number of soft copy handouts, and thus, a decrease in paper consumption.

The last impact to talk about is the **Societal Impact** of *Here!.* The most important issue about the social impact is the privacy concern. The program will require access to users' cameras for the teaching assistant, and before doing that, permission from the user will be taken before accessing the camera. The user data will not be utilized other than the intended purpose of *Here!*. In addition, our program will inform the users how the private data is protected and processed for the benefit of the community while not violating any of the users' rights. "Code of Ethics" by the National Society of Professional Engineers is going to form the ethical boundaries of the application [9].

8.6 Planning for New Knowledge and Learning Strategies

During the course of implementation, we will learn about new technologies including image recognition, machine learning and network protocols together with the new frameworks and libraries. Our strategies for learning these concepts will be courses we take in Bilkent, interviews with our innovation expert, online learning and learning from peers. All team members are taking the CS464 Introduction to Machine Learning course which will be essential for our project. We decided on our innovation expert based on his skills. He is experienced in IP video systems and network protocols. Online tutorials and development forums will also help us build the required knowledge.

9. Glossary

WebRTC: An open platform application that allows real time communication capabilities like text and video chat, screen sharing and so on [3].

Machine Learning: Machine learning is a branch of artificial intelligence (AI) focused on building applications that learn from data and improve their accuracy over time without being programmed to do so [10].

Computer Vision: Computer vision is an application of CS that deals with how computers can gain a deeper level of understanding while dealing with images and videos. It can involve various item or movement recognition tasks.

Motion Tracking: Motion tracking is an application of computer vision, so that it tracks the movement of objects and transferring the sensed data to an application for further processing.

Eye Tracking: Eye tracking is an application of computer vision, so that it tracks various movements of the eye in order to analyse certain situations. It can detect the position of the glance, the movement of the eye etc.

Third Party Application: Third party applications are applications that are the application that are being used other than the main application. For example, the libraries that we use are third party applications which enhance the capabilities of the main program.

Tensorflow: TensorFlow is a free and open-source software library for machine learning. It can be used across a range of tasks but has a particular focus on training and inference of deep neural networks [11].

Angular: Angular is a platform for building mobile and desktop web applications [2]. It allows designing dynamic multi-pages front ends.

MySQL: MYSQL is a fully managed database service to deploy cloud-native applications using the world's most popular open source database [4].

GitHub: An online platform that provides hosting for software development version control using Git [12].

OpenCV: OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library [5].

DLib: DLib is a modern C++ toolkit containing machine learning algorithms and tools [6].

Darknet: Darknet is an open source neural network framework written in C and CUDA [7].

YOLOv3: You only look once (YOLO) is a state-of-the-art, real time object detection system [8].

Agile Development: Agile software development refers to software development methodologies centered round the idea of iterative development, where requirements and solutions evolve through collaboration between self-organizing cross-functional teams [13].

Network Protocols: A network protocol is an established set of rules that determine how data is transmitted between different devices in the same network [14].

10. References

[1] "Introduction to TensorFlow, [Online] Available: https://www.tensorflow.org/learn [Accessed Oct 10, 2020]

[2] "Angular", [Online] Available: https://angular.io/ [Accessed Oct 10, 2020]

[3] "WebRTC", [Online] Available: https://webrtc.org/getting-started/peer-connections [Accessed Oct 10, 2020]

[4] "MySQL Documentation", [Online] Available: https://dev.mysql.com/doc/ [Accessed Oct 10, 2020]

[5] "OpenCV", [Online] Available: https://opencv.org/ [Accessed Nov 20, 2020]

[6] "dlib", [Online] Available: http://dlib.net/ [Accessed Nov 20, 2020]

[7] "YOLO: Real-Time Object Detection", [Online] Available: https://pjreddie.com/darknet/yolo/ [Accessed Nov 20, 2020]

[8] "Darknet: Open Source Neural Networks in C", [Online] Available: https://pjreddie.com/darknet/ [Accessed Nov 21, 2020]

[9] "Code of Ethics | National Society of Professional Engineers" Nspe.org, 2019, [Online] Available: Available: https://www.nspe.org/resources/ethics/code-ethics [Accessed Oct 10, 2020]

[10] "What is Machine Learning?", [Online] Available: https://www.ibm.com/cloud/learn/machine-learning#toc-what-is-ma-qhM6PX35 [Accessed Nov 20, 2020]

[11] "TensorFlow", [Online] Available: https://www.tensorflow.org/ [Accessed Nov 20, 2020]

[12] "Github", [Online] Available: https://github.com/ [Accessed Nov 20, 2020]

[13] "Agile development.", [Online] Available:https://www.cprime.com/resources/ what-is-agile-what-is-scrum/ [Accessed Nov 20, 2020]

[14] "What Is a Network Protocol, and How Does It Work?", [Online] Available: https://www.comptia.org/content/guides/what-is-a-network-protocol [Accessed Nov 20, 2020]