Project Proposal Report

Team #1

Team Members

- Chongzhi Gao
- Rouyu Li
- Bin Luo
- Karen Winona Setiawan
- Jing Zhang

Project Name: Zoom plus^{+}

Team Name: ZoomGuys

Project Synopsis:

A web application geared towards teaching individuals who wants to know check if their online students are paying attention or not

Project Description:

Zoom is widely used by instructors due to Covid-19. Although it has many features that instructors love, it is still lacking features that can be helpful to teachers and students. For one, it is difficult for instructors to gauge interest levels in class. This can be attributed to both the barriers of internet communication, as well as the distractions presented by being at home.

We are planning for Zoom plus^{+} to have more features that can aid instructors in checking if the students are listening to class and provide statistics that could improve the learning experience.

The end result of the project of Zoom plus^{+} will be a web application that will utilize a machine learning model to detect the student's gaze, a scraping library to scrape the number of students and logins to zoom, automatic captioning during lectures and lastly to only allow students access to the zoom tab.

Project Milestones:

Zoom Plus Plus

TEAM NUMBER	Team #1		
PROJECT MANAGER	Chongzhi Gao, Rouyu Li, Bin Luo, Karen Winona Setiawan, Jing Zhang	DATE	3/12/18

WBS NUMBER	TASK TITLE	TASK OWNER	START DATE	DUE DATE	DURATION	PCT OF TASK COMPLETE	PHASE ONE														
							WEEK 9/21					WEEK 9/28					WEEK 10/5				
							M	T	W	R	F	M	T	W	R	F	M	T	W	R	F
1	**Project Conception and Initiation**																				
1.1	Project Charter	Team	9/21/20	9/25/20	5	100%															
1.2	Research	Team	9/28/20	10/2/20	5	100%															
1.2.1	Raised hands	Karen	9/28/20	10/2/20	5	100%															
1.2.2	Tracking eyesight/Check for mask	Li Ruoyu, Chongzhi Gao	9/28/20	10/2/20	5	100%															
1.2.3	Auto CC	Luo Bin	9/28/20	10/2/20	5	100%															
1.2.4	Students only open zoom	Zhang Jing	9/28/20	10/2/20	5	100%															
2	**Project Definition and Planning**																				
2.1	Diagrams	Team	10/5/20	10/9/20	5	0%															
2.2	Initial desige		10/12/20	10/16/20	5	0%															
3	**Project Conception and Initiation**																				
3.1	Setup Envirment	Team	10/19/20	10/23/20	5	0%															
3.2	Raised hands	Karen	10/26/20	10/30/20	5	0%															
3.3	Tracking eyesight/Check for mask	Li Ruoyu, Chongzhi Gao	10/26/20	10/30/20	5	0%															
3.4	Auto CC	Luo Bin	10/26/20	10/30/20	5	0%															
3.5	Students only open zoom	Zhang Jing	10/26/20	10/30/20	5	0%															
3.6	Project Front-end	Team	11/2/20	11/6/20	5	0%															
4	**Project Performance / Monitoring**																				
4.1	Finished styled	Team	11/9/20	11/13/20	5	0%															
4.2	Connect Front-end Back-end	Team	11/16/20	11/20/20	5	0%															

WBS NUMBER	TASK TITLE	TASK OWNER	START DATE	DUE DATE	DURATION	PCT OF TASK COMPLETE	PHASE ONE														
							WEEK 1					WEEK 2					WEEK 3				
							M	T	W	R	F	M	T	W	R	F	M	T	W	R	F
1	**Project Finishing**																				
1.1	Testing	Team				0%															
1.2	video making	Team				0%															
1.3	report	Team				0%															
1.4	documentation finished	Team				0%															

Fig 1.1: Gantt Chart part 1

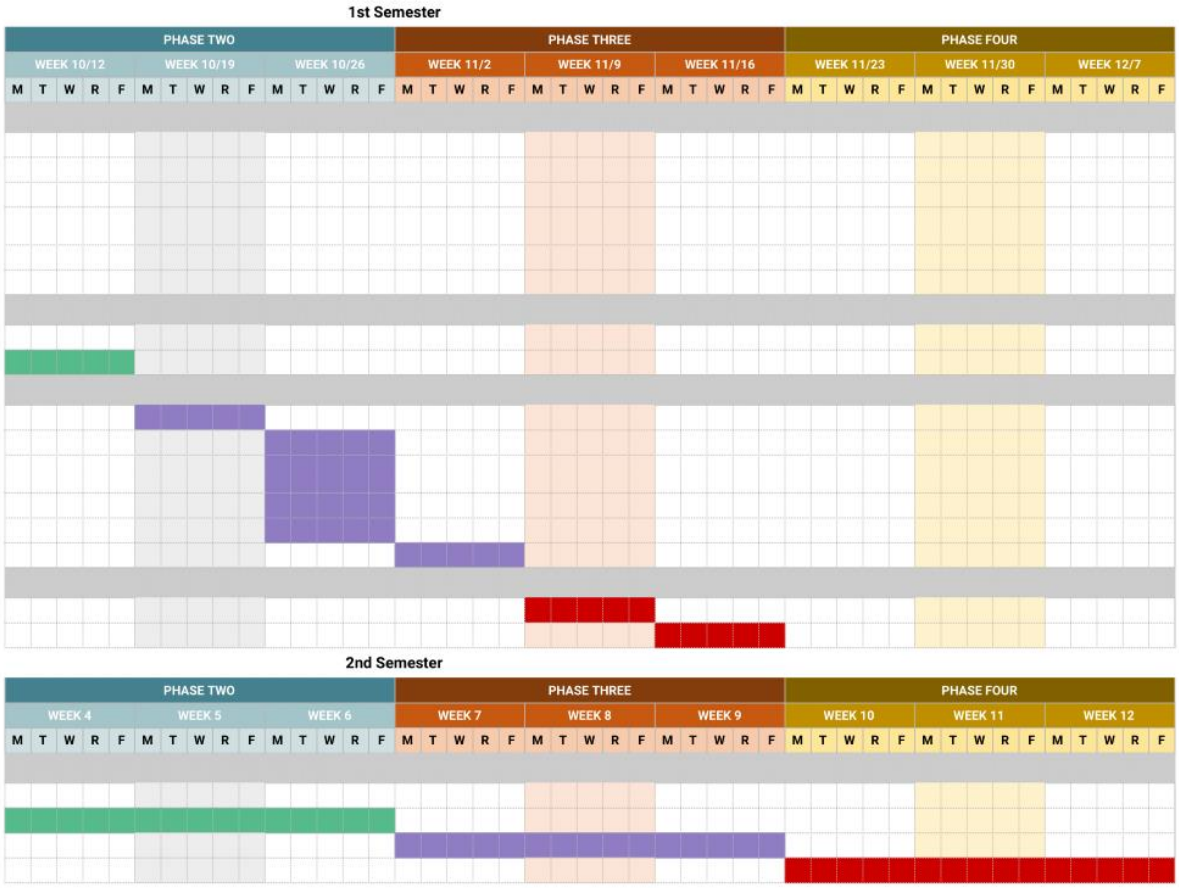

Fig 1.2: Gantt Chart part 2

First Semester:

- Project Requirements/Plan (9/20/2020)
- Finish Researching Programming tools such as OpenCV, Dlib, Selenium, Flask, Twilio, Google's text to speech api, 2captcha api (10/2/2020)
- Initial Project Description (10/5/2020)
- Project Proposal Report (10/26/2020)
- Documentation (No specific due date, we plan to document as we implement)

Second Semester:

- Front-end and back-end finish implementation (2/19/2021)
- Front-end and back-end integration (3/6/2021)
- Finish Testing (3/26/2021)

Project Budget:

- Estimated cost for 2Captcha (Online captcha solver API) - $1 per 1000 captcha solve
    - Used in first and second semester to use for automatic zoom login

Preliminary Project Design:

The primary goal of our product is to provide a platform for instructors who used zoom to know more about their class situation such as if their students are paying attention to class, the number of raised and more. The web application mainly features and promotes students to listen to class and help instructors with information that will help in the making of more engaging classroom.

Software Stack:

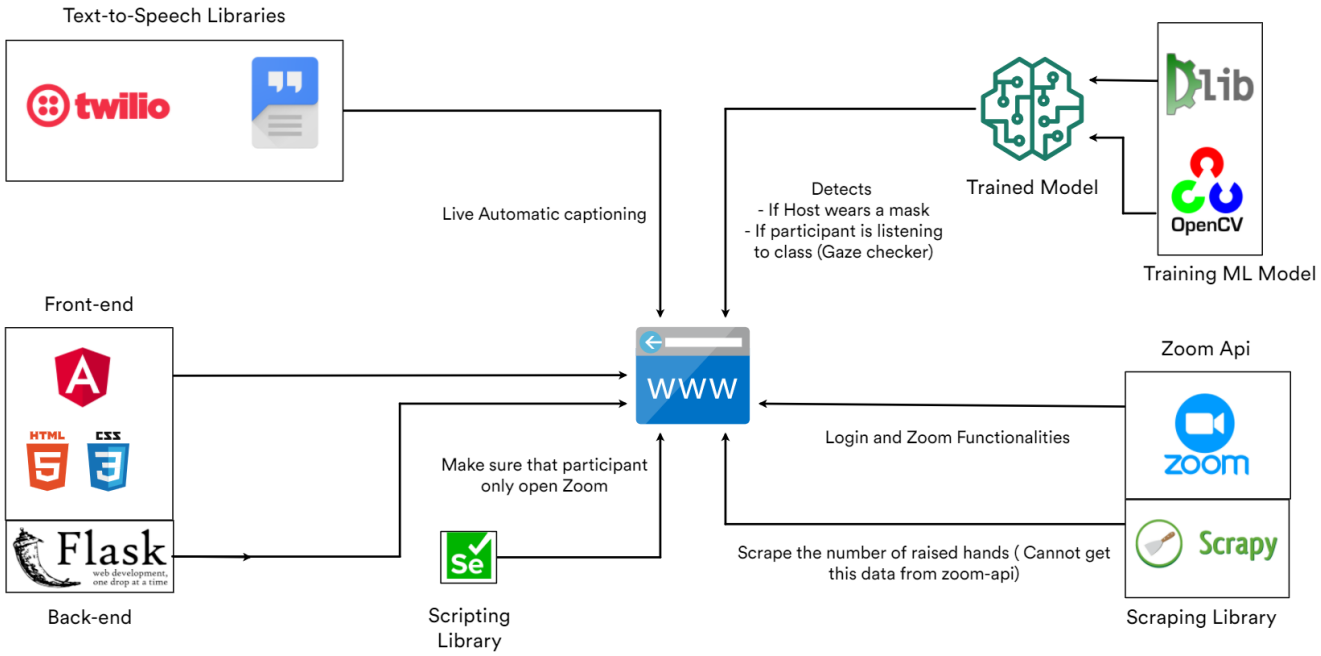

Fig 2: Overview of ZoomPlusPlus Architecture

ZoomPlusPlus's product is a web application that consists of web frontend which is also known as User Interface and also, back-end. Unlike most web application, our web application does not utilize a database that means we do not plan to store any login and sensitive information about users who use ZoomPlusPlus to login to zoom.

For the web application front-end, we choose Angular, HTML5, CSS3 and back-end to use Flask because one of our developers has some prior experience working in Angular. Back-end is separated into 4 main parts. The feature live automatic captioning will make use of Twilio's text-to-speech and Google's text-to-speech API. Our machine learning model that will

help detects if the host is wearing a mask and if the students is listening to the class will utilize Dlib and OpenCV. Login to zoom will utilize a scraping library, Scrapy and Zoom API. Lastly, to make sure our students only focus on the zoom lecture, Selenium library will be used.

User Interface:

Since our web application mainly interacts with zoom, most of the UI will be pop-ups with zoom web page and some pages with our own design that use the technologies mentioned above. The pictures below are sketch of our plan for the UI and to further help readers imagine our User Interface.

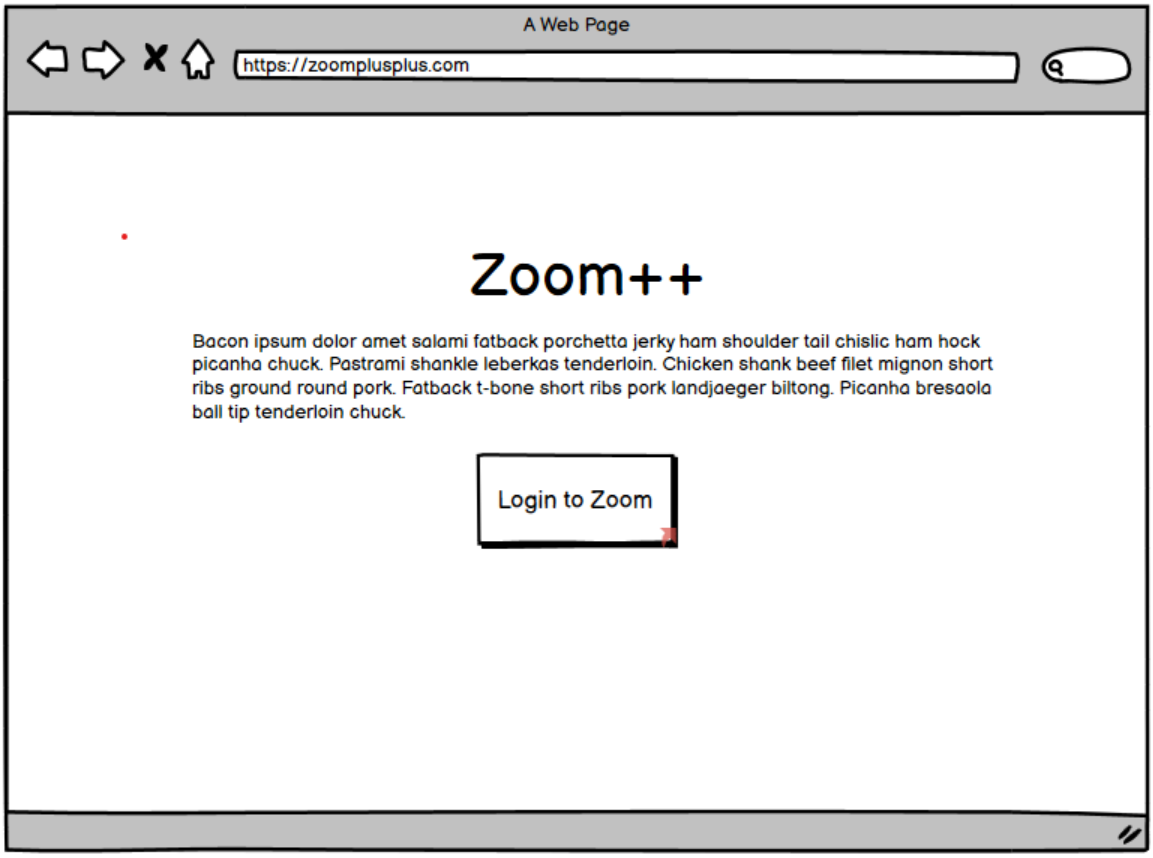

Fig 3: ZoomPlusPlus Landing Page

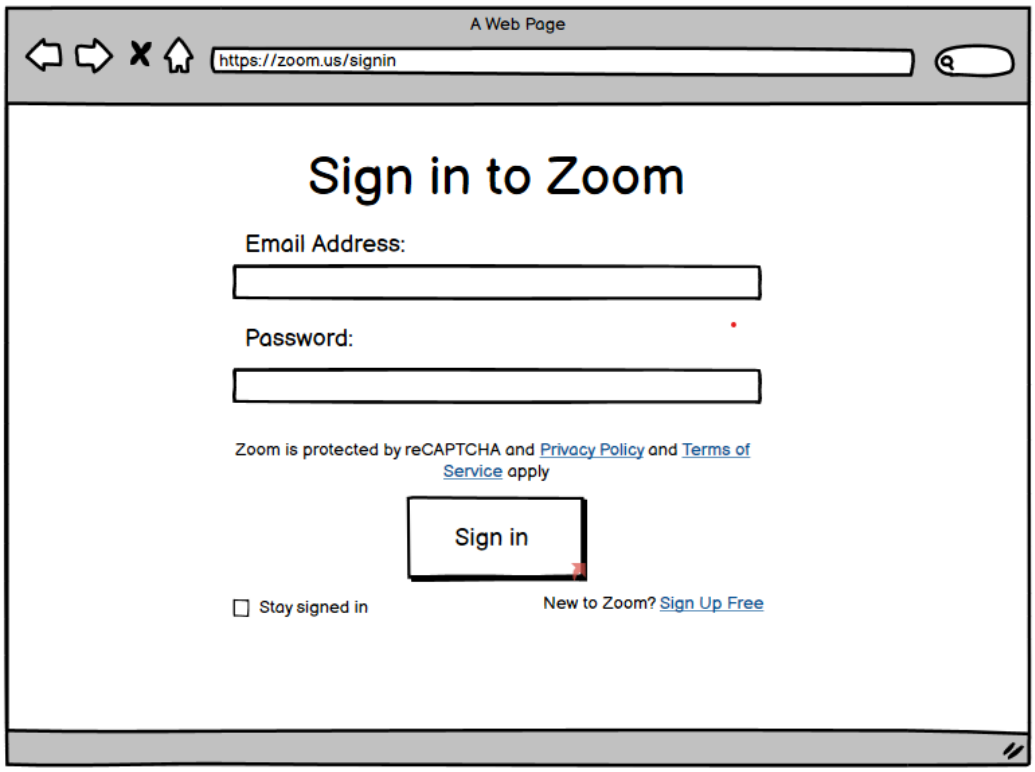

Fig 4: Zoom Signin Page

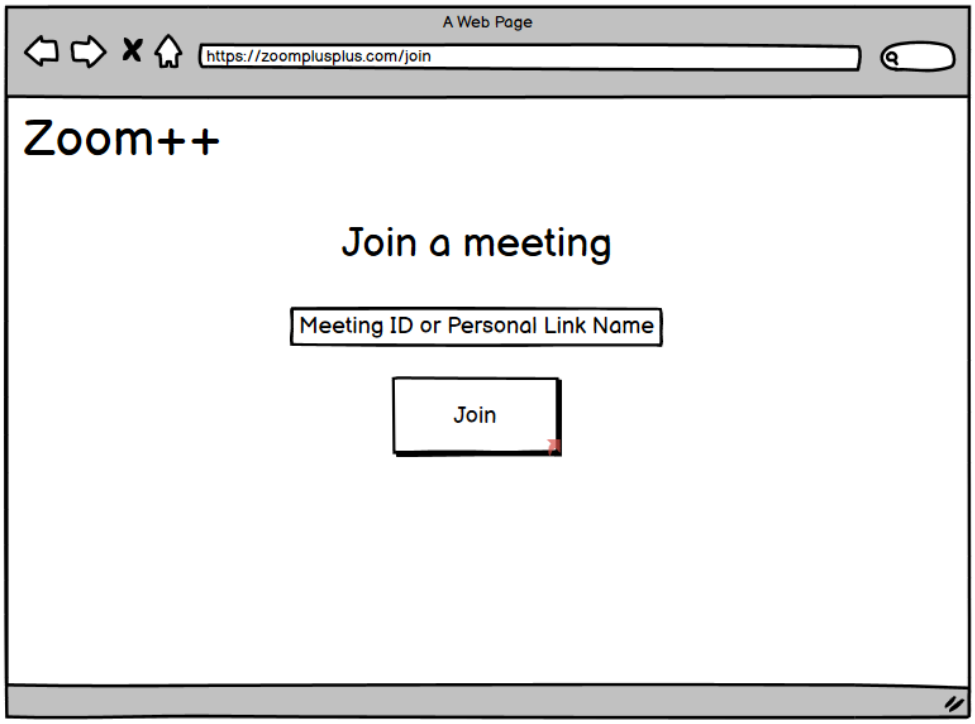

Fig 5:

ZoomPlusPlus Join Meeting

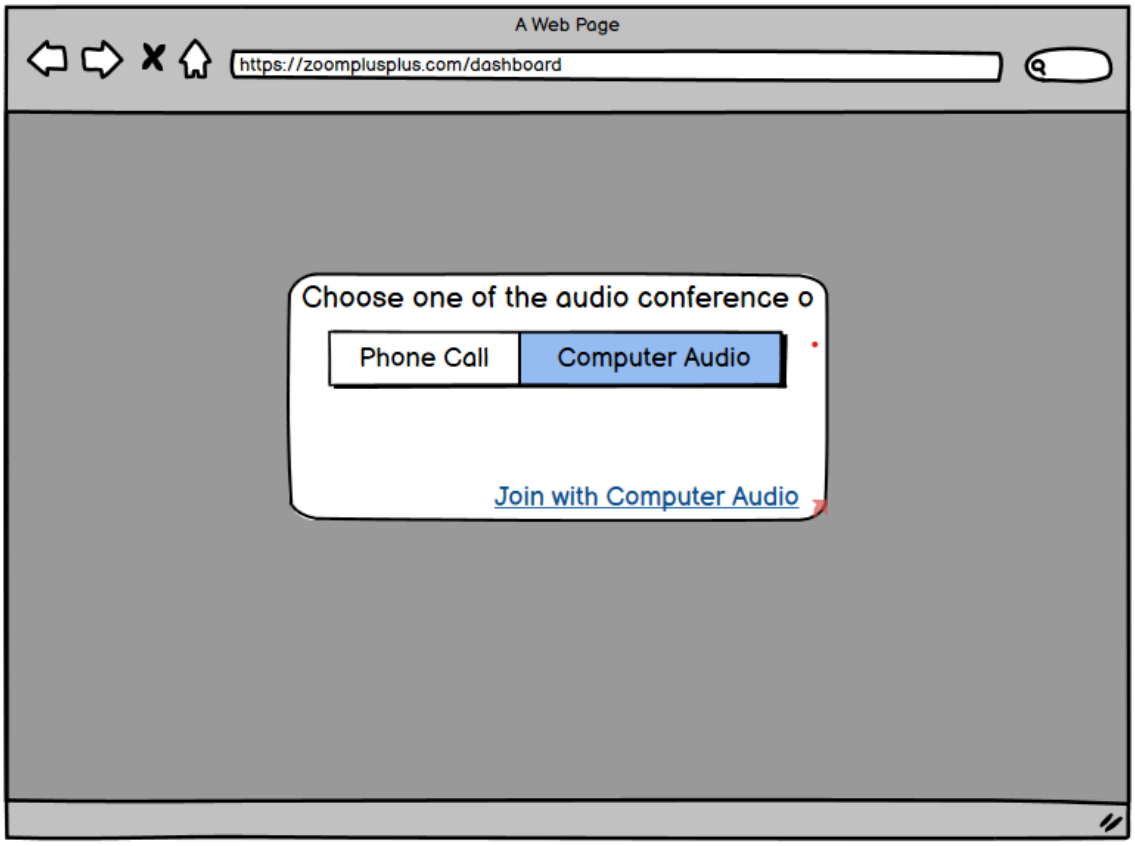

Fig 6: Selecting Audio

Fig 7: If you are a Host (Instructor), you have to wear a mask before continuing to lecture

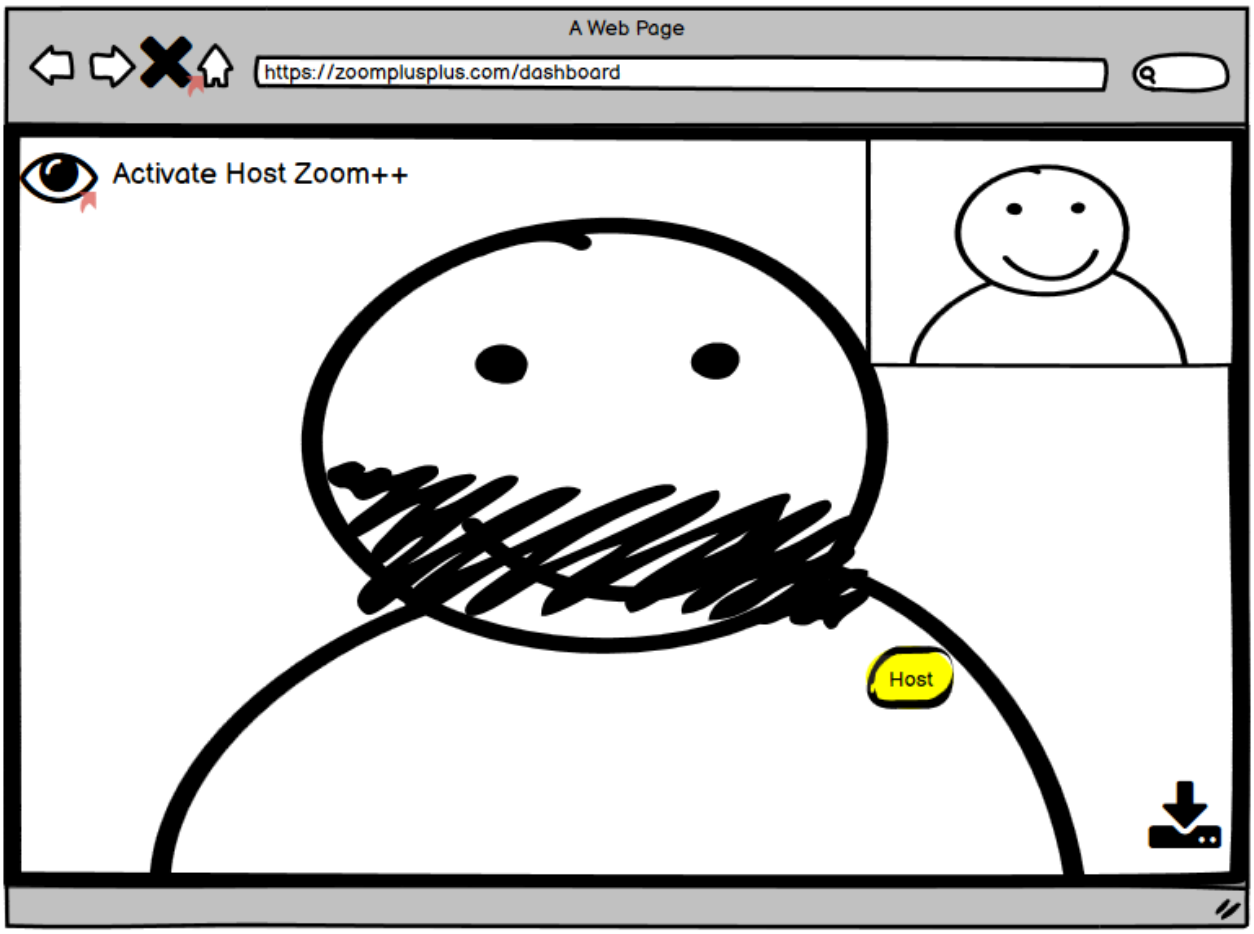

Fig 8: Host dashboard after wearing a mask

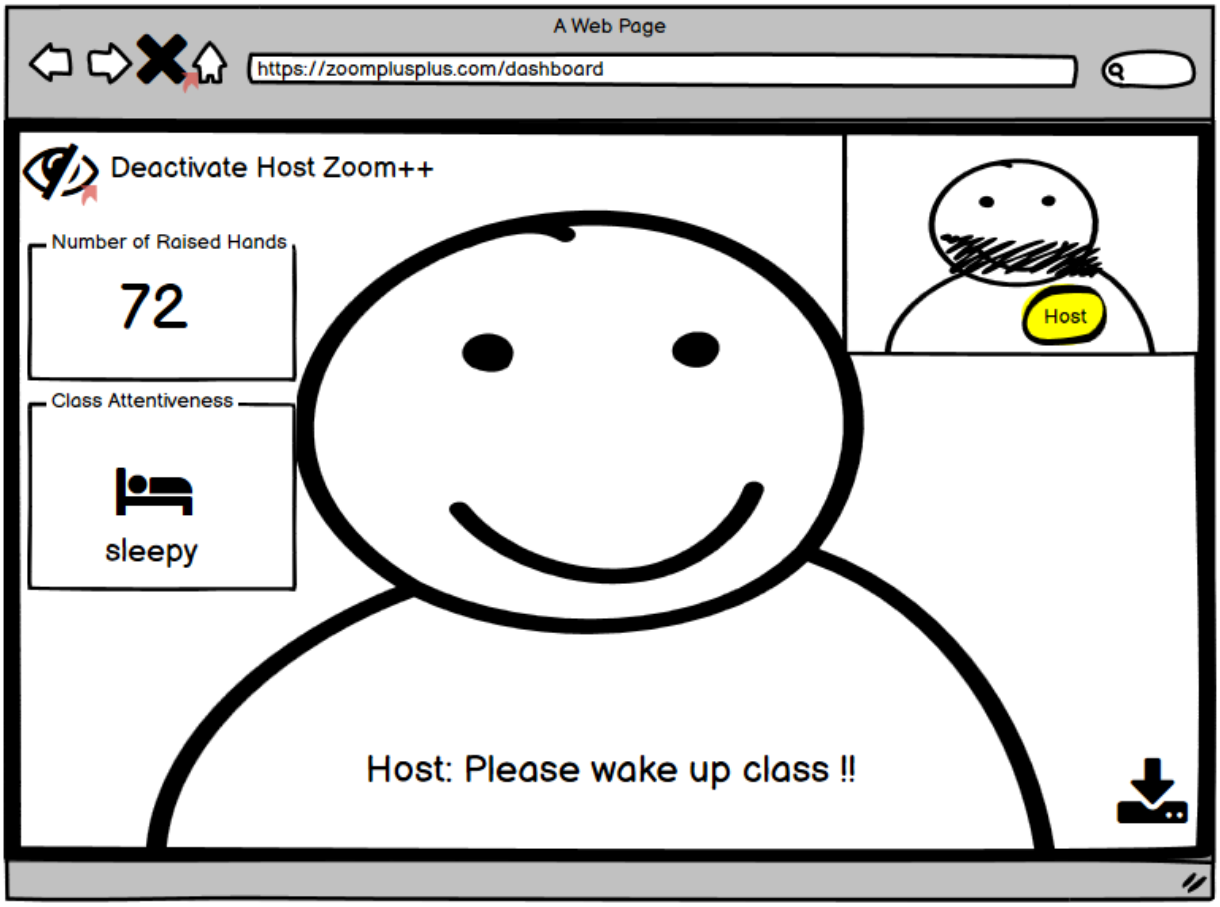

Fig 9: Activate Host Zoom++, features:

number of raised hands , overall class attentiveness and live captioning

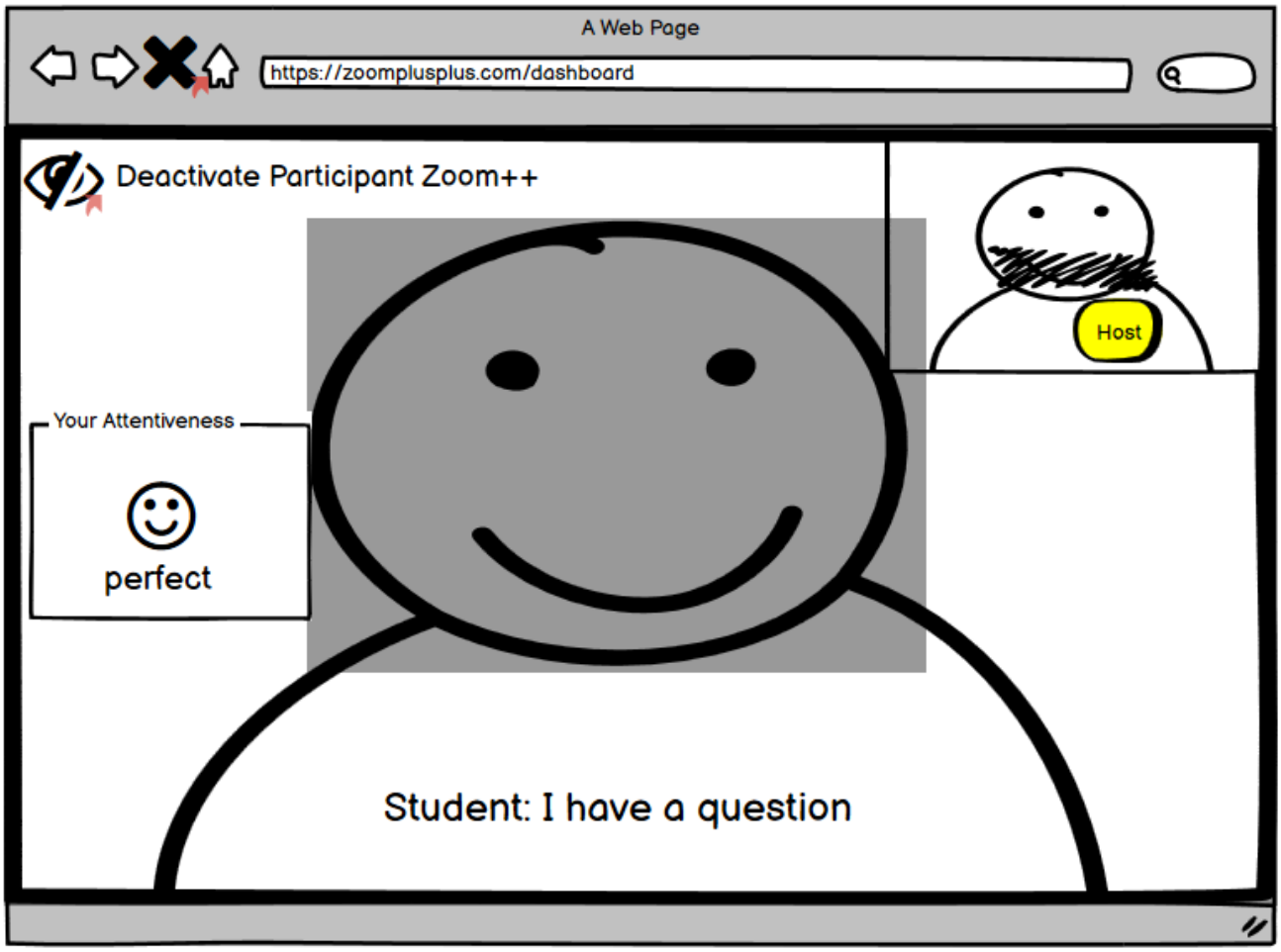

Fig 10: Activate Participant Zoom++, features:

Participant Attentiveness, Only can open zoom, live captioning

Backend

Live Automatic Captioning:

In the Zoom, there is a closed caption feature which can show the caption, but it's not automatic unless a third-party service is used. We want to achieve the live automatic caption by using Speech-to-Text API and Twilio API. Twilio API is used to capture the speaker's audio from the microphone, then send this audio as an endless streaming speech to Speech-to-Text API to request a speech recognition and get the response which contains the transcription of the audio. This transcription as the caption will be displayed at the bottom of the screen of the Zoom. We will add a button to the Zoom UI to allow users to choose to turn caption on or off.

Scraping number of raised hands:

Zoom API does not provide the numbers of raised hands. We plan to take advantage of the scraping library, Scrapy. The biggest obstacle of scraping the zoom website is that we must bypass a captcha that zoom has implemented. Our solution for this was to use a third-party API called 2Catpcha that will help us to solve the captcha in a few seconds. But 2Captcha API is expensive if we were to use in long run. Another alternative is to use TensorFlow captcha which is a free open source captcha solving library which has 85% accuracy. Taking into the cost and captcha solving accuracy into consideration, we plan to use TensorFlow captcha solving library first and if it fails, it will use the 2captcha API to solve the captcha to reduce the cost in long run.

Machine Learning Model:

Detection of Host wearing a mask:

In this project, detecting whether the instructor is wearing a mask correctly is one of the project features. We are going to implement it by locating the face and searching facial features, and the safety of the instructor can be determined. By using Python's OpenCV open source database, we can retrieve the instructor’s face in the video through the host’s camera. By retrieving the instructor’s facial contours, we can determine the position of each corresponding facial feature, such as eyes, nose, and mouth. If we find that the nose, mouth, or chin are in their corresponding positions during the inspection, then we will determine that the instructor is not wearing the mask correctly. If the above situation occurs, we will remind the instructor on the main window.

Detection of participants paying attention to class:

The attention detection mechanism is one of the main features of this product. This product uses a variety of methods to improve the quality of students' learning during class. The primary solution chosen by this product is facial recognition and sight detection. The most likely way for online class students to be lazy is to use a machine to open the online class page and put it aside. Through the reference to the Python open-source library OpenCV and Dlib, we realized the process of reading the user's camera data, image processing, analyzing the facial contour, and then identifying the direction of the line of sight. By comparing the user's line of sight direction and the screen coordinate reference coefficient, this product can analyze whether the user is watching an online course. The auxiliary program selected by this product is to detect whether the online course page is the first page of the machine. Another method often used by online class students is to open the online class and switch it to the background. This

part of the product reads the system data to identify the focus of the current machine to determine whether the online course software is the most front software of the machine. If a student's current line of sight is on the machine screen and his online class software is the most front software of the machine, it can be basically determined that the student's attention is on the course itself.

Only allow participants to open zoom:

We are going to use python selenium web driver to make sure students only open zoom. Initially, the zoom tab in the browser will be set as the default tab, during the class, the web driver will constantly check the current activated tab id, once a switching tab is detected or a new tab is opened, an alert window will pop up and web driver will switch to the default tab automatically unless the instructor allows the access to a specific URL. Another approach will be to disable other tabs during class time. This feature can only be activated by the instructor.

Design Constraint:

Our project has some design constraints in multiple ways. One of the product's main features is accessing the user's camera to analyze their attentiveness. There is a privacy issue involved and some people may feel uncomfortable to turn on their camera. Another important issue is that some users that do not have a camera which is a big issue as most of our features involve a camera. Only allowing participant to open zoom highly relies on the python's selenium web driver which requires the developer to specify the specific version of chrome or Firefox to be used.

Ethical Issues

Privacy:

The primary problem with this product lies in the multiple possibilities of infringing on user privacy. In order to monitor whether the user is concentrating on class, we use a variety of programs to monitor the user's physical activity and the user's system operation. First, in the physical activity monitoring part, we use the camera to capture the user's face and then process the image captured by the camera in the background to analyze the user's eye track at this time. However, this operation is bound to upload the user's image information to the background server, which may cause privacy violations. Secondly, to detect the user's system operation, we need to obtain some system permissions. Although we will remind users that they have obtained permission, our detection is a background detection and is not completely transparent. Not to mention that we will perform certain restrictions to force users to stay on the classroom page.

Security:

Another problem that may arise is security. We need to collect and process user information in the background, including some sensitive information. Although data processing is active in the background, its internal data is not encrypted, which may cause security issues. We have an obligation to protect the security of user data privacy, but due to various restrictions, it is currently difficult to achieve. It is possible for hackers to withhold some users' personal information such as usernames, passwords, and even more private ones like legal names, schools, and selected courses. We will do our best to protect this part. Including but not limited to data encryption and firewall access.

Intellectual Property Issues:

Patent:

Our first problem is that we need to apply for a patent. We need to apply for a patent for the intellectual property rights of this Zoom development project. In this way, we can protect our intellectual property rights from malicious use or theft by others.

Copyrights:

In this project, we will reasonably and legally reference the existing Python database. We will declare all the open source databases that we also use to ensure the original author’s intellectual property rights. But while we protect the original author’s intellectual property rights, our project also faces copyright issues. We want to protect our Zoom PlusPlus so that we can prevent someone from using our program to make illegal acts or even sell it. In order to ensure the realization of this step, we will not open source our project code. At the same time, because our project program involves many user privacy issues, it is very important not to open source in order to effectively ensure the security of users and applications. One of the good solutions.