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:

An 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 an 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.

Project Milestones:

Zoom Plus Plus

TEAM NUMBER	Team #1		
PROJECT MANAGER	Chongzhi Gao, Rouyu Li, Bin Luo, Karen Winona Setiawan, Jing Zhang	DATE	2/13/21

1st Semester

WBS NUMBER	TASK TITLE	TASK OWNER	START DATE	DUE DATE	DURATION	PCT OF TASK COMPLETE
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	100%
2.2	Initial desige		10/12/20	10/16/20	5	100%
3	**Project Conception and Initiation**					
3.1	Setup Envirment	Team	10/19/20	10/23/20	5	100%
3.2	Raised hands	Karen	10/26/20	11/6/20	11	100%
3.3	Tracking eyesight/Check for mask	Li Ruoyu, Chongzhi Gao	10/26/20	11/6/20	11	100%
3.4	Auto CC	Luo Bin	10/26/20	11/6/20	11	100%
3.5	Project Front-end	Zhang Jing	10/26/20	11/6/20	11	100%

2nd Semester

WBS NUMBER	TASK TITLE	TASK OWNER	START DATE	DUE DATE	DURATION	PCT OF TASK COMPLETE
1	**Initial Coding**					
1.1	Raised hands	Karen	2/1/21	2/19/21	19	100%
1.2	Tracking eyesight/Check for mask	Li Ruoyu, Chongzhi Gao	2/1/21	2/19/21	19	100%
1.3	Auto CC	Luo Bin	2/1/21	2/19/21	19	100%
1.4	Project Front-end	Zhang Jing	2/1/21	2/19/21	19	100%
2	**Code Finishing**					
2.1	Connect UI with machine learning	Team	2/22/21	2/26/21	5	0%
2.2	Connect UI with auto-cc	Team	3/1/21	3/5/21	5	0%
2.3	Finish All UI	Team	3/8/21	3/12/21	5	0%
3	**Project Finishing**					
3.1	Testing	Team	3/15/21	3/19/21	5	0%
3.2	video making	Team	3/22/21	3/26/21	5	0%
3.3	report	Team	3/22/21	3/26/21	5	0%
3.4	documentation finished	Team	3/22/21	3/26/21	5	0%

Fig 1.1: Gantt Chart

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)

Second Semester:

- Machine learning model/s coding completed (2/19/2021)
- Connecting machine learning model with front-end (2/26/2021)
- Connecting auto-cc with front-end (3/6/2021)
- All features connected to UI (3/12/2021)
- Code is tested and working (3/19/2021)
- Quad Chart (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

Final 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 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:

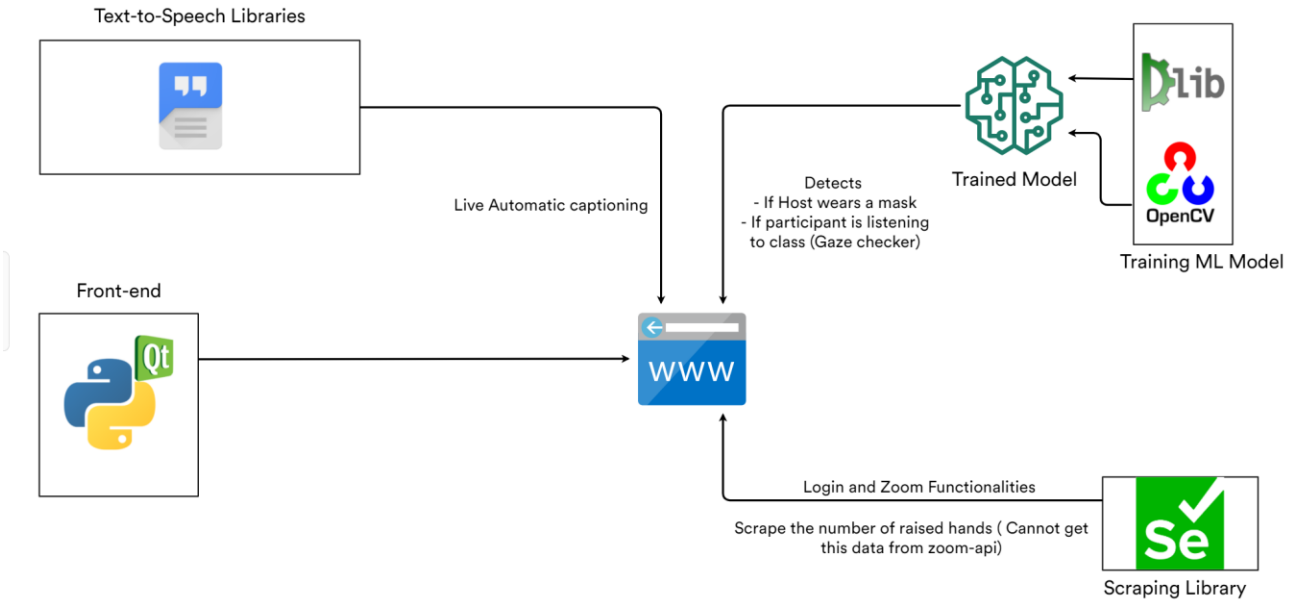

Fig 2: Overview of ZoomPlusPlus Architecture

View it: https://app.terrastruct.com/diagrams/1079845640/

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

For the application front-end, we choose PyQt. 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, Selenium.

User Interface:

On the login window, the user needs to provide the meeting ID, password, or URL. After that, the browser will be opened and the Zoom plus plus bot will join the Zoom meeting. The main window can display the number of participants, their level of attention to the course, how many people raised their hands, and who was the first to raise their hands.

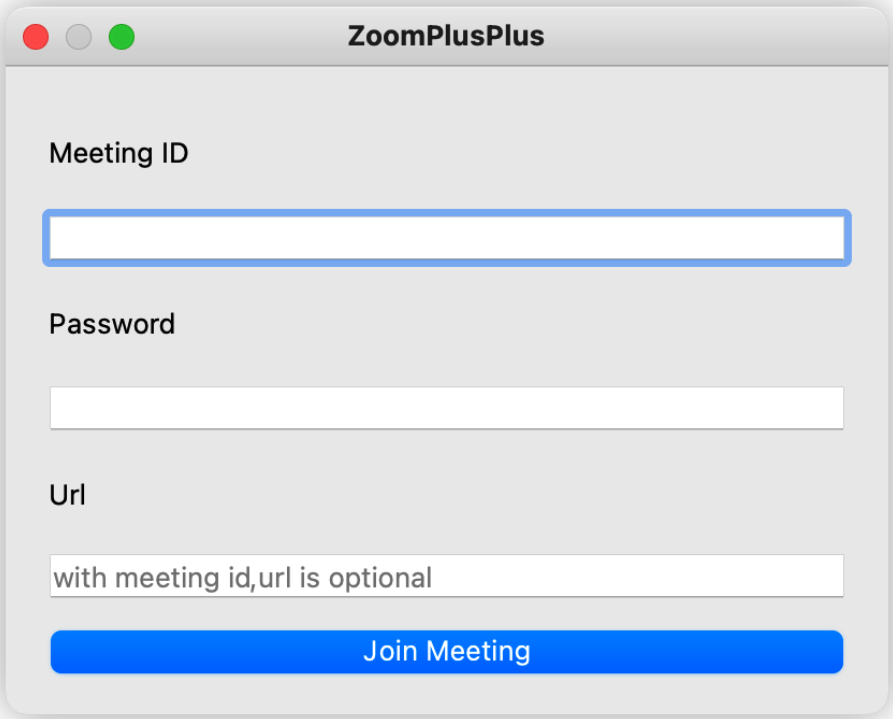

Fig 3: Zoom Sign in window

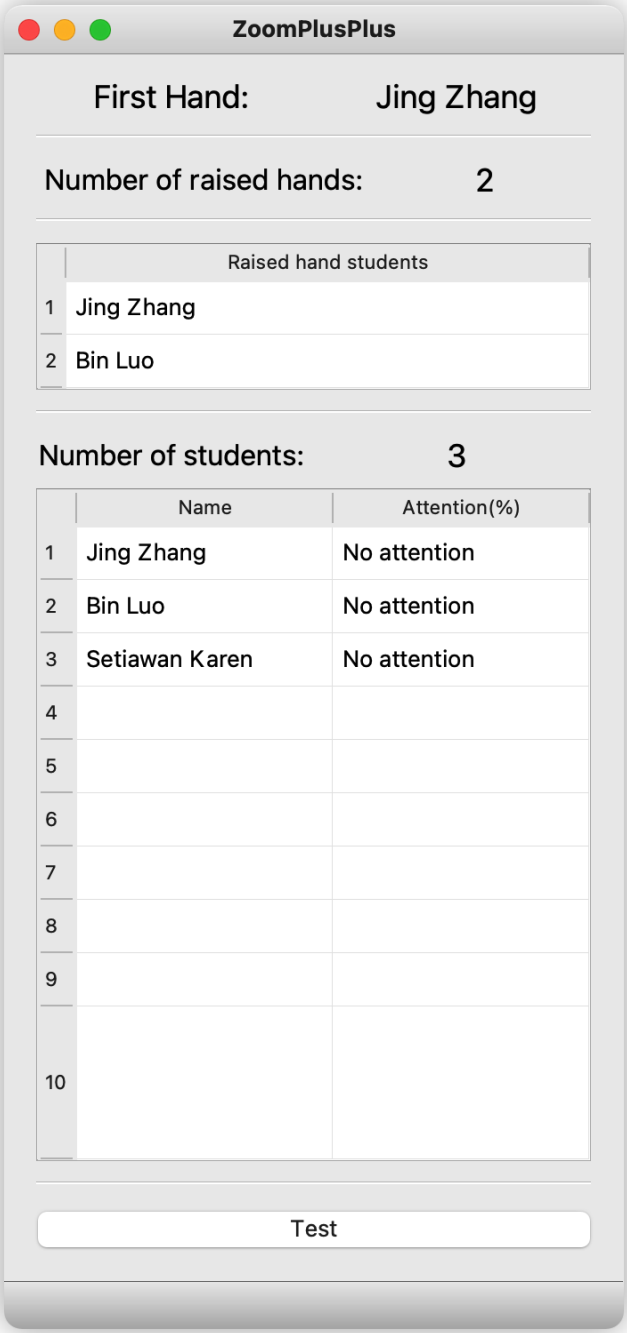

Fig 4: Zoom main window

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. I use python 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, Selenium. 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. 2Captcha API is expensive if we were to use in long run. Taking into the cost and captcha solving accuracy into consideration, we can allow the user to solve the captcha manually to reduce the cost in long run or if it is taking too long to solve the captcha with 2Captcha.

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.

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. Our code only supports mac and windows.

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.

Change Log:

- Our end goal has changed from a web application to just application that can be run by python app.py .
- Changed the software architecture overview picture.
- Front end technologies (Flask, HTML, CSS) have been replaced by PyQt.
- Do not utilize Zoom API to login, all is done by Selenium library.
- Do not utilize Twilio
- Added a design constraint which is only supports mac and windows.
- No change in ethical issues and Intellectual properties.
-