

Project Requirement Document

Team Name

No Cap Stone

Company Name

LogMeIn

Project Title

Best Face Forward

Team Members

Andrew Doan andrewdoan@ucsb.edu (Lead)

Bik Nandy bnandy@ucsb.edu (Scribe)

Adjon Tahiraj atahiraj@ucsb.edu

Ryan Gormley rgormley@ucsb.edu

Tim Chang tinghaur@ucsb.edu

Introduction

“First-round job interviews are the latest part of the hiring process to undergo digitization as companies use video interviews to cut recruiting costs and times...the method has grown in recent years as nearly everyone has access to a laptop or smartphone with a front-facing camera, and companies say it is an efficient, fair and inexpensive way to process hundreds of applicants.” - *Wall Street Journal*

Online interviews help expedite the time it takes to connect the interviewer with the interviewee. The company is able to reach out to a greater population to tap into and expand the candidate pool. Video interviews are supposed to be more effective than a phone screen since the interviewers can get a better idea of who the candidate is through visual and audio evaluation. In reality, online interviews are impersonal when talking to a screen, disengaging, hard to connect, and difficult to read physical cues. Interview software currently is also highly unorganized through different post-application stages including recruiter screening, first-round, and final-round stages. Because of these difficulties, interviewers often have limited information for

a candidate and many times cannot gauge a candidate's fit or skills as effectively as they could during in-person interviews. In today's interview platforms, often it is a 2-way (or multiple) video conference call with mute and toggle video capabilities. The conference call is not ideal as many times, it looks as if the individuals in the call are not making direct eye contact, audio may be missed, and a person may not be in a professional setting (i.e. their home). Furthermore, many companies must use separate software to keep track of and communicate with candidates through audio or video. Therefore, in our CS189 Capstone we have decided to focus on creating an application to redefine the online interview experience.

Goal:

The goal of this project is to create an application that streamlines the application process for the candidate and create an interview experience that is personal to both the interviewer and interviewee to capture the best qualities of each person.

We will create a personalized interviewing platform to better simulate a real, in-person interview by creating a web application with features including:

- Background Blur
- Filters (professional)
- Engagement and Sentiment Analysis of Audio (voice) and Video
- Access to details such as resume, notes, linkedin profile, github, shared notes
- Speech to Text logging
- Translation of interviewee
- Timers and reminders to ask pre-selected questions
- Live closed captioning and translate features
- Eye Gaze Correction

By getting more out of online interviews, companies will have to interview fewer candidates because they will get a better feel for the soft skills of each candidate during the online process. This will save employers substantial time and labor, as well as helping them select candidates that are a better fit.

Objectives:

The MVP for this project will be a web application that automatically joins a video call. The interviewer will also be able to create a meeting, which will be accessed by a meeting ID. In the video call you can create notes. The interviewer will be able to see a sentiment analysis during the video call, and their notes. After the video call, the interviewer will be able to see a transcript, their notes, and the sentiment analysis of the call.

Goals:

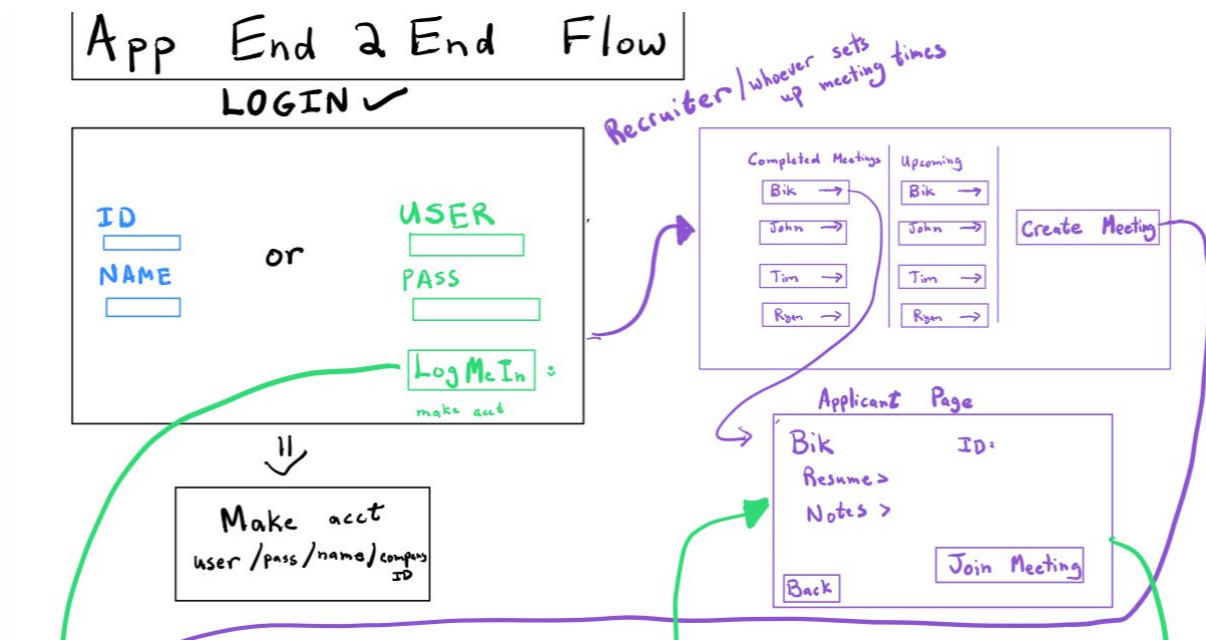
- Host 2 person video interviews with useful widgets for the interviewer
 - Checkboxes, timers, notes, agenda, etc.
- Speech recognition to produce a transcript of the interview
 - Analysis of sentiment during responses
- Indicators for the interviewer about how the interviewee is responding. This will be a simple colored light helping the interviewer understand physical cues that are hard to pick up over video
- Interviewer and interviewee have a screen showing separate meeting
- Interviewer and interviewee can create notes for a meeting before the meeting and will be able to read and access them during the meeting as well as after the meeting
- Interviewer can see a timer of the meeting time
- Interviewer can create meetings

Stretch Goals:

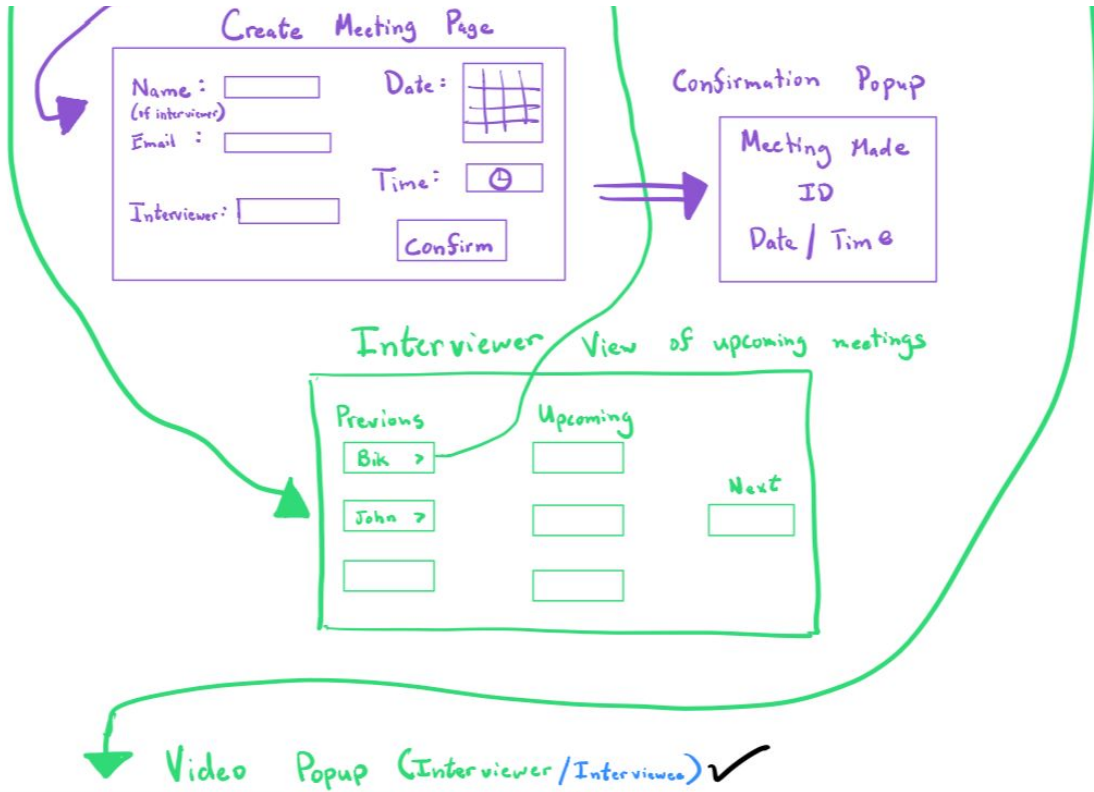
- Eye gaze correction
- Face Sentiment Analysis from **live** video stream
- Multi-person interviews

System Architecture

High Level Diagram



Continued on next page



Video Popup (Interviewer/Interviewee) ✓

Timer ⏱

Video ✓

✓ Subtitles here (cc & translate)

(Analysis ● Nervous)

Notes

- Resume >
- Backgrnd Blur >
- Screen Share >

Chat

↳ hidden from Interviewees

✓ items on first demo

User Interaction and Design

- Our web application consists of 4 pages:
 - A login page where the user can either log into their dashboard and see their meetings, or log directly into a video meeting
 - A make account page where a user can enter their information and make an account
 - Upon creating an account, the user is redirected to the login page and a toast notification confirming the account was made is played in the upper right hand corner
 - An about us page with a short biography about each member
 - A dashboard showing all the upcoming and previous meetings
- After logging into a video meeting, the user will be entered into their video chat popup, with a transcription happening in real time
 - The video meeting supports multiple users
 - The popup has 4 buttons
 - A blur button to cover the background around the user
 - A reset transcription button to clear the transcription
 - A translate button to translate the transcription
 - An analyze button to return sentiment analysis of the transcription

LOGIN

Username

Password

Name

Room ID

[Make Account](#)

CREATE ACCOUNT

FirstName

LastName

Username

Email

Password

Retype Password



Name:
Andrew Doan
Year/Major:
UCSB 3rd Year Computer Engineer
LinkedIn:
[Click Here](#)

Account successfully made

LOGIN

Username

Password

Submit

Name

Room ID

Submit

[Make Account](#)



Dashboard
Dashboard
Dashboard
Dashboard



Meeting

Room: test

Other Participants

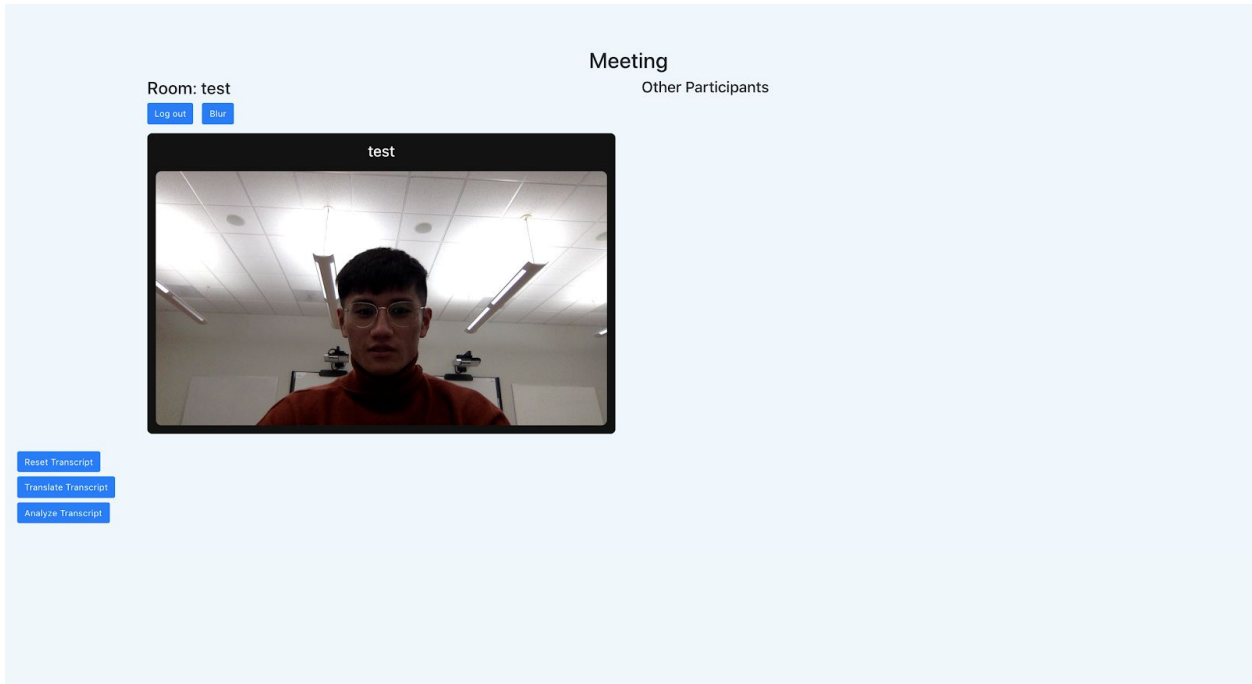
Log out Blur



Reset Transcript

Translate Transcript

Analyze Transcript



Meeting

Other Participants

Room: test

Log out Blur



Reset Transcript: yeah maybe that could be it

Translate Transcript

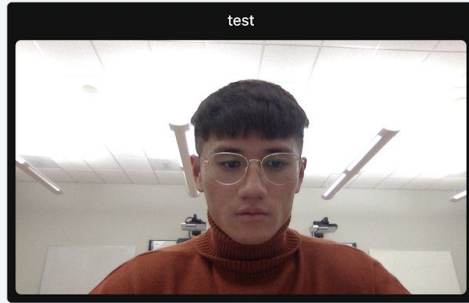
Analyze Transcript

Meeting

Other Participants

Room: test

Log out Blur



Reset Transcript yeah maybe that could be it

Translate Transcript да, может быть, это может быть

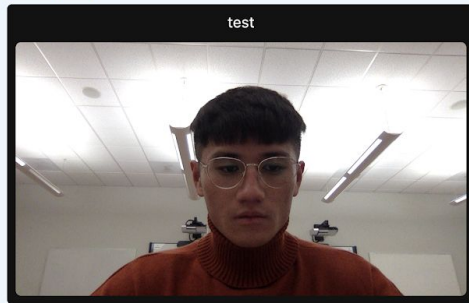
Analyze Transcript

Meeting

Other Participants

Room: test

Log out Blur



Reset Transcript yeah maybe that could be it

Translate Transcript да, может быть, это может быть

Analyze Transcript

```
[{"score": 0.994446, "tone_id": "tentative", "tone_name": "Tentative"}]
```

User Stories

Interviewee

Pre Interview	During Interview	Post Interview
<ul style="list-style-type: none"> • Easy access link to the video interview • Be able to schedule meeting • Be able to see the Job description for reminder • The pre-interview notes the interviewee wrote down. 	<ul style="list-style-type: none"> • A notepad to write down questions • Be able to share screen • Be able to see resume • Ability to reconnect if technical difficulties occur • Background blur • Suppress background noise. 	<ul style="list-style-type: none"> • Show the notes the interviewee took • When to hear back about next steps

Interviewer

Pre Interview	During Interview	Post Interview
<ul style="list-style-type: none"> • Personal Info of the Interviewee uploaded • Schedule interview • Notepad to Brainstorm questions • Checklist 	<ul style="list-style-type: none"> • Be Able to see the interviewee's resume • Share Screen • A checklist to remind the interviewer • Assess emotion/engagement with sentiment analysis • Blur background • Timer • Closed Captioning • Live Translate 	<ul style="list-style-type: none"> • Place to comment about candidate and give feedback about applicant • Display results of the engagement analysis • See checklist and notes

Recruiter

Pre Interview	During Interview	Post Interview
N/A	N/A	<ul style="list-style-type: none"> • Notes from Interview • Overall rating • See feedback from interviewer

Requirements

1. Lay stub code : <https://trello.com/c/qcMDXpAh>
 - Use case : Prepare for development
 - Actors: Users
 - System Precondition: User must go to localhost:3000 or the corresponding url.
 - Flow of Events: Be able to navigate to each separate component page.
 - Postcondition: User should be able to land in the corresponding component page
2. Create login page : <https://trello.com/c/dRWyL862>
 - Use Case: Login
 - Actors: Users
 - System Precondition: User must be in the login page.
 - Flow of Events: Basic Path: User types in username and password, system checks against hard coded parameters. If the user credentials are valid, redirect to the dashboard page.
3. Networked video chat : <https://trello.com/c/l5uwnFQc>
 - Use Case: Video Conferencing
 - Actors: Users, Twilio API
 - System Precondition: User must be logged in.
 - Flow of Events: Basic Path: User types in room id and the name they would like to be displayed as, system finds corresponding room id, redirects user to the room page and begins video component for video conferencing.
 - Post Condition: User should be able to see the video stream feed.
4. Sentiment analysis of text : <https://trello.com/c/R0yDgCkQ>
 - Use Case: Sentiment analysis
 - Actors: IBM Watson API
 - System Precondition: N/A
 - Flow of Events: Create a string of coherent sentences and system takes the string transcript and sends it to IBM watson for analysis. IBM watson returns analysis on the console.
 - Postcondition: Results displayed in console.
5. Display sentiment analysis : <https://trello.com/c/9VPZov3C>
 - Use Case: Sentiment analysis
 - Actors: Users, IBM Watson API
 - System Precondition: User is in video conference room/page
 - Flow of Events: User presses a button then sends their mic input transcript to IBM Watson for analysis. IBM watson return analysis and system displays it on the page.
 - Postcondition: Results displayed on video conference page.

6. Create subtitles for transcript: <https://trello.com/c/Kq37C1sh>
 - Use Case: Subtitles
 - Actors: Users, React-Speech-Recognition Module.
 - System Precondition: User is in video conferencing page/room.
 - Flow of Events: User speaks into mic, system detects mic input and calls React-Speech-Recognition Module for transcription and displays transcription on the video conferencing page.
 - Postcondition: Users can see transcription on the display.
7. Google Translate API : <https://trello.com/c/4ZOHU2ug>
 - Use Case: Translation
 - Actors: Users, Google Translate API
 - System Precondition: User is in video conferencing page/room.
 - Flow of Events: User speaks into mic, system detects mic input and calls React-Speech-Recognition, after results are displayed, User should be able to press a button to call translation. Translation API takes transcript and returns the translated text.
8. Demo Product : <https://trello.com/c/EMB1H3QL>
 - Use Case: Demo
 - Actors: Users
 - System Precondition: All modules are integrated to create MVP.
 - Flow of Events: User would be able to sign into the login page with dummy credentials, type in a room id after login, see a video conferencing page, and be able to interact with the translation and transcription buttons for results.
9. Setup database with API calls and schema : <https://trello.com/c/qxDBCHwr>
 - Use Case: Database
 - Actors: User, Amazon DynamoDB
 - System Precondition: User must be in the DBTest page to test implementation.
 - Flow of Events: N/A
 - Postcondition: N/A
 - Notes: We implement the DynamoDB and the corresponding functions for insert and delete.
10. Connect and create make account page to database <https://trello.com/c/R8jasbWK>
 - Use Case: Database
 - Actors User, Amazon DynamoDb
 - System Precondition: User is on the login page.
 - User can press “create account” and get redirected to the “create account” page and fill out First Name, Last Name, email, password, and press create. Parameters will be passes into respective create function and will check against differing password, existing accounts. If create account fails, user will be prompted an error

and can retype the form again. If create account succeeds, user is redirected to the login page.

- Postcondition: User is able to login with their own account id and password.

11. Timer and polish login to video flow : <https://trello.com/c/FGZSOpxZ>

- Use Case: Timer and better user experience.
- Actors: User
- System Precondition: User is on the login page.
- Flow of Events: User can type in room id and preferred nickname. System checks against existing video rooms and if a room exists, a new video page pops out with the video conference room. If the room does not exist, create a new video conferencing room with the room id.

12. Face Sentiment: <https://trello.com/c/0icoF17Q>

- Use Case: Face Sentiment Analysis
- Actors: N/A
- System Precondition: N/A
- Flow of Events: N/A
- Postcondition: N/A
- Notes: We found the best service that provided Face Sentiment Analysis to be Microsoft Azure Cognitive service.

13. Cover Background : <https://trello.com/c/wRvDLsVS>

- Use Case: Cover Background on video conferencing
- Actors: User
- System Precondition: User is within video conferencing room/page
- Flow of Events: User can click on a button below the video stream and the system will crop out parts of the background.
- Postcondition: N/A

14. Login Cookies: <https://trello.com/c/dAksocZZ>

- Use Case: Login cookies
- Actors: User
- System Precondition: User has logged in once.
- Flow of Events: N/A
- Postcondition: N/A

15. Screenshot video for facial analysis : <https://trello.com/c/1bVHExrM>

- Use Case: Facial Analysis Research
- Actors: User
- System Precondition: User is in video conferencing page.
- Flow of Events : User is able to press a button to capture the camera feed for a screenshot. Screenshot is then displayed on the video page.
- Postcondition: N/A

16. Professional Filters: <https://trello.com/c/R8jasbWK>
 - Use Case: Filter out background with a professional filter
 - Actors: Users
 - System Precondition: User is within video conference page
 - Flow of Events: User presses button below video stream and a professional-looking (business appropriate) filter will appear as a part of the background
 - Postcondition: User goes back to regular video stream (no filter)
17. Eye Gaze Detection: <https://trello.com/c/TmKo8a8r>
 - Use Case: Detect if eye is away from camera (towards screen) and correct
 - Actors: Users
 - System Precondition: User is within video conference page
 - Flow of Events: Automatically detect if user is looking away from camera and adjust eyes accordingly
 - Postcondition: N/A
18. Dashboard: <https://trello.com/c/PjW96uUf>
 - Use Case: View previous and upcoming meetings
 - Actors: Users
 - System Precondition: User logs in with a valid login-in. Username is passed as a cookie to Dashboard page
 - Flow of Events: all meetings are displayed. User can click on meeting to go to Meeting Lobby
 - Postcondition: redirected to meeting lobby page
19. Create Meeting page: <https://trello.com/c/PjW96uUf>
 - Use Case: creating new meetings and inviting participants
 - Actors: Users
 - System Precondition: From dashboard click on create meeting button
 - Flow of Events: User enters valid meeting time, date, and participant and hits create button
 - Postcondition: user is brought back to dashboard and new meeting is displayed on dashboard page as well as invitation to meeting sent to participant
20. Share Screen: <https://trello.com/c/ktTJNRYm>
 - Use Case: Share current screen
 - Actors: Users
 - System Precondition: User is within video conference page
 - Flow of Events: User clicks “Share Screen” button below video feed, Browser handles permissions, user shares current screen
 - Postcondition: User returns to regular video page

User Stories Prototype code:

#2: Create Login Page: <https://github.com/andrewdoanutz/No-Cap-Stone/pull/8>

#7 video call: <https://github.com/andrewdoanutz/No-Cap-Stone/pull/6>

#9 Sentiment Analysis: <https://github.com/andrewdoanutz/No-Cap-Stone/pull/9>

#11 create subtitles for transcript: <https://github.com/andrewdoanutz/No-Cap-Stone/pull/5>

#14 Set up database: <https://github.com/andrewdoanutz/No-Cap-Stone/pull/12>

#16-connect-login-to-aws: <https://github.com/andrewdoanutz/No-Cap-Stone/pull/13>

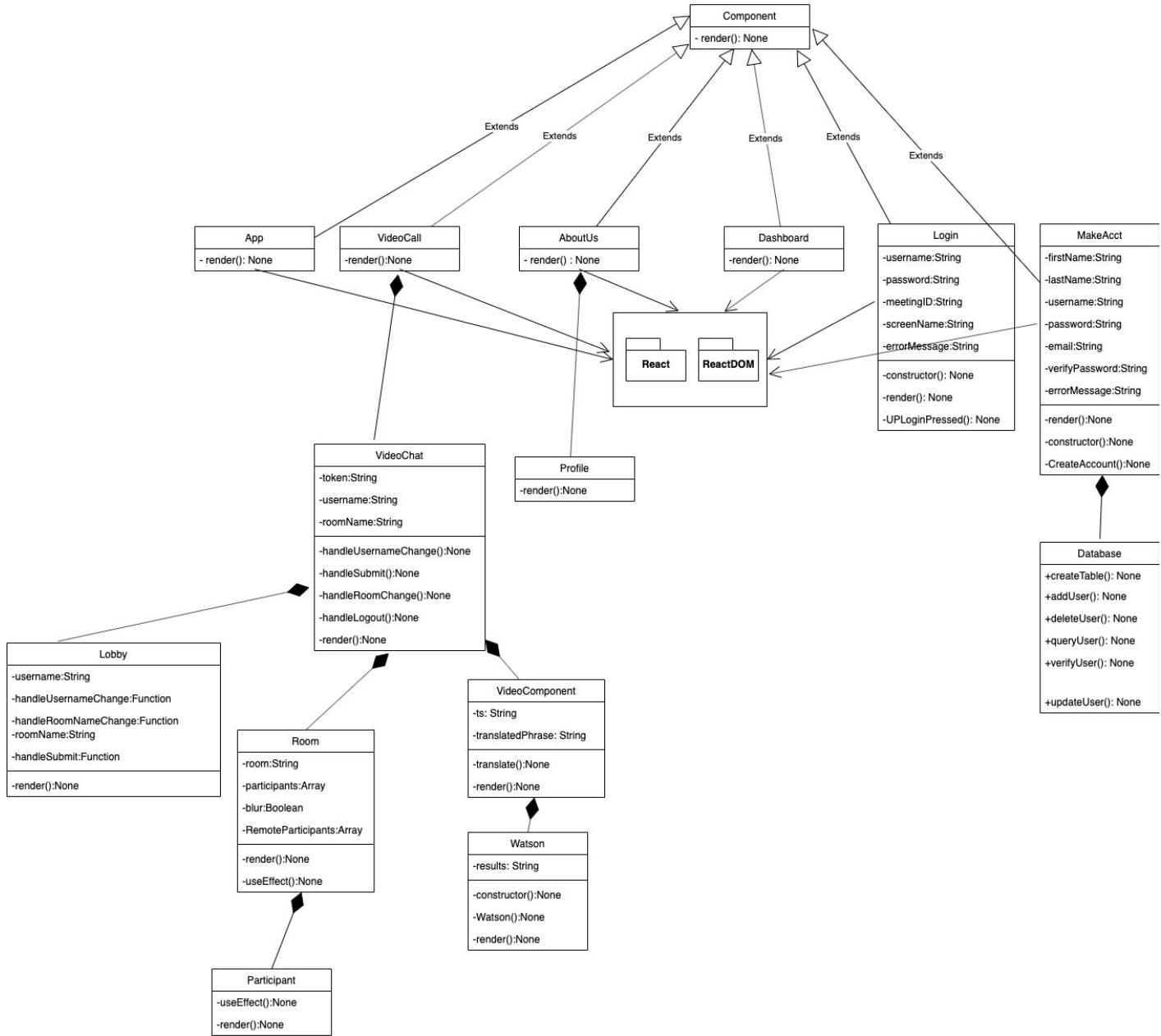
#19 Timer: <https://github.com/andrewdoanutz/No-Cap-Stone/pull/14>

#20-face-sentiment-research: <https://github.com/andrewdoanutz/No-Cap-Stone/pull/15>

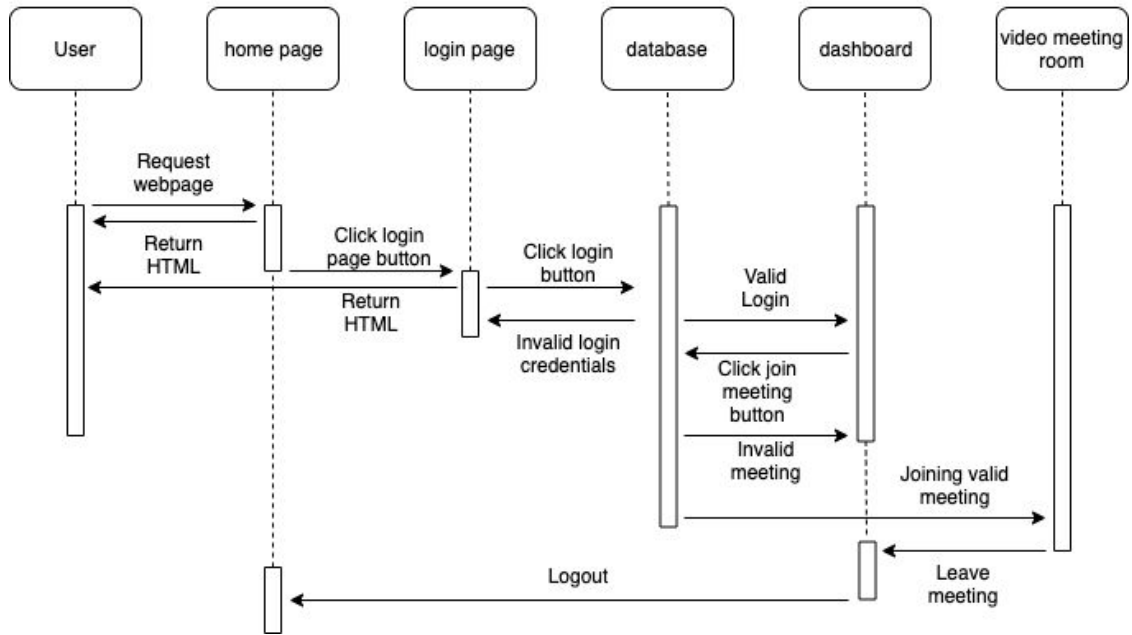
#22-blur-background: <https://github.com/andrewdoanutz/No-Cap-Stone/pull/19>

#24-videocapture: <https://github.com/andrewdoanutz/No-Cap-Stone/pull/21>

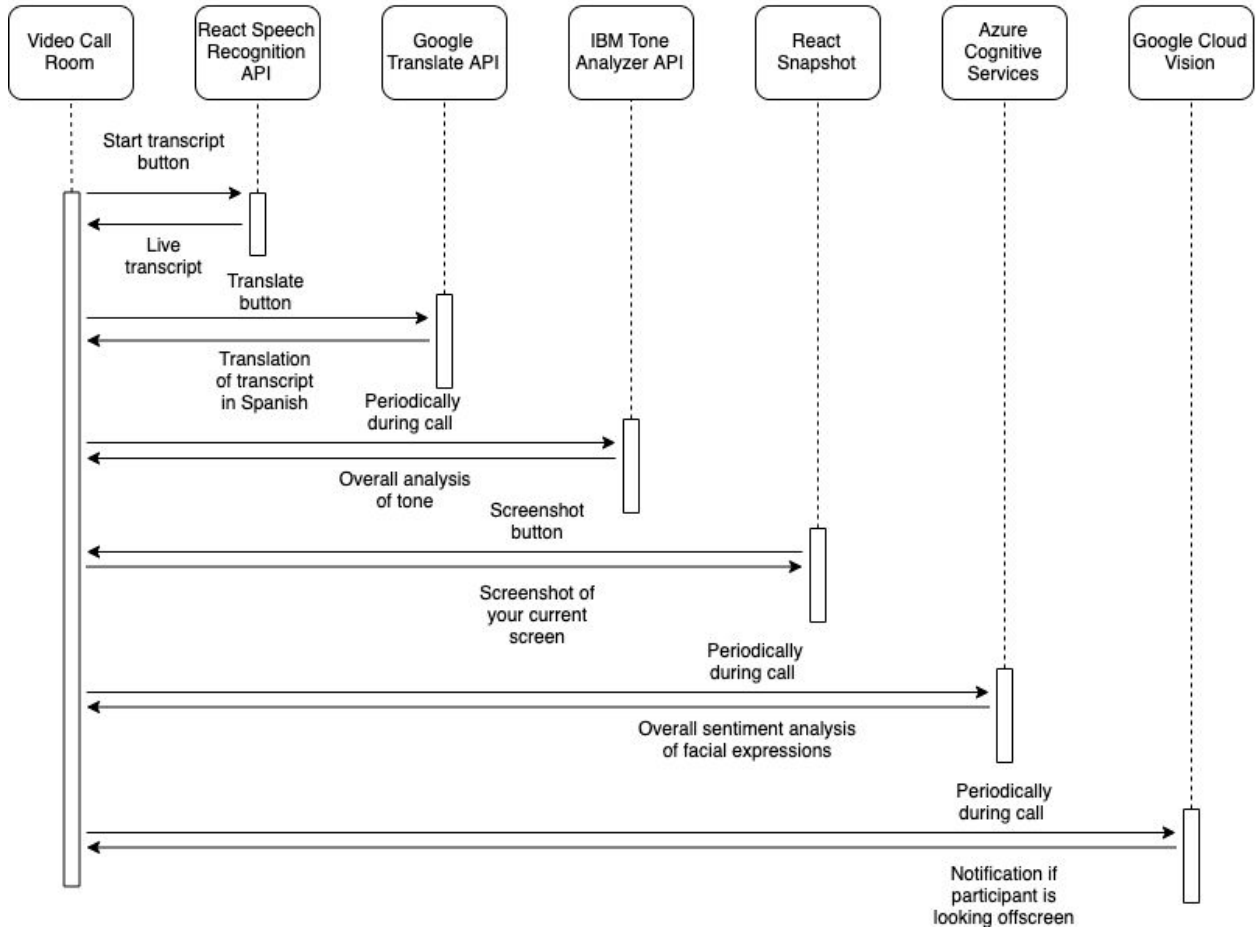
System Models



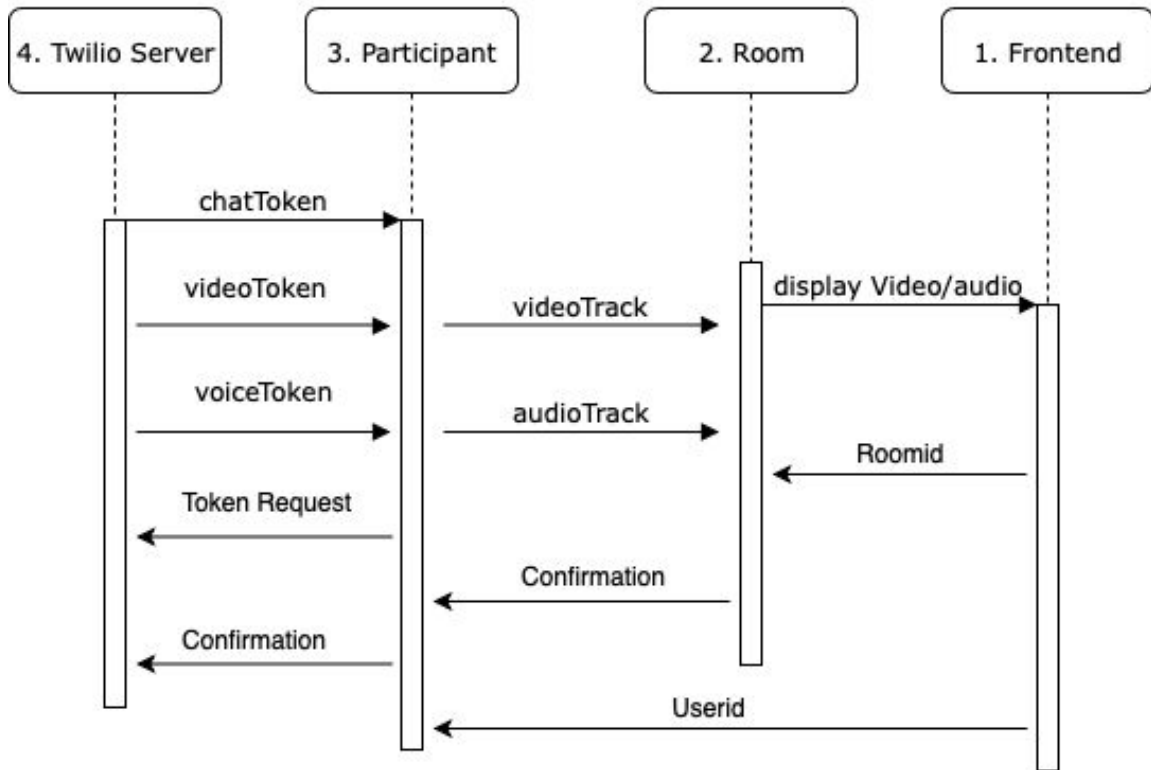
Model-View-Controller Sequence Diagram



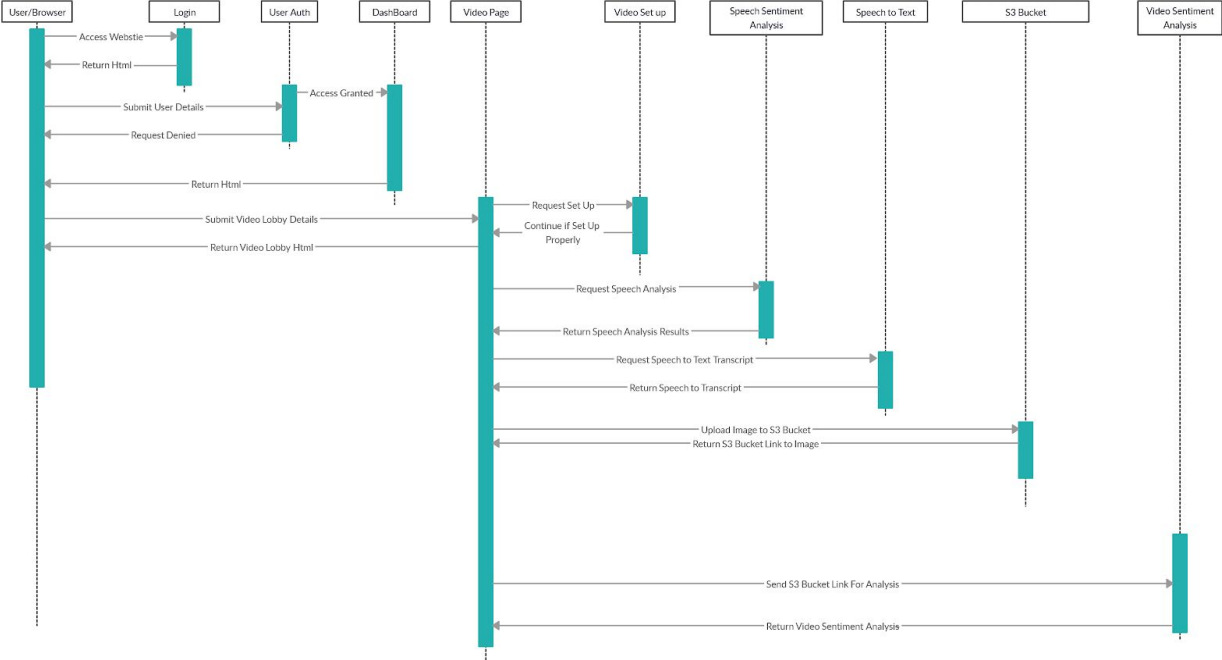
Audio/Video Service Diagram



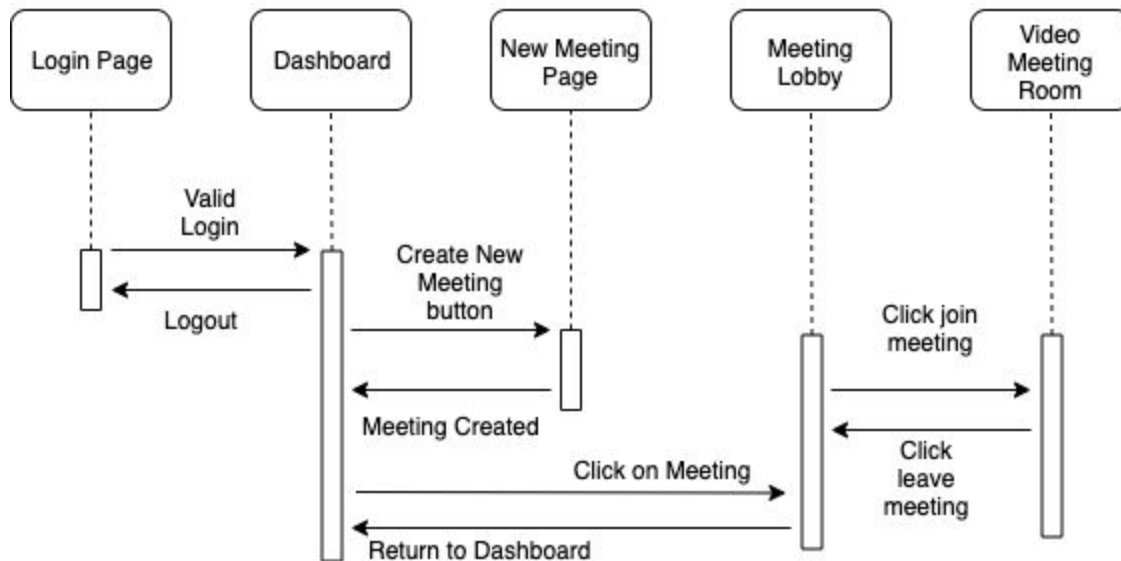
Backend Video Sequence Diagram



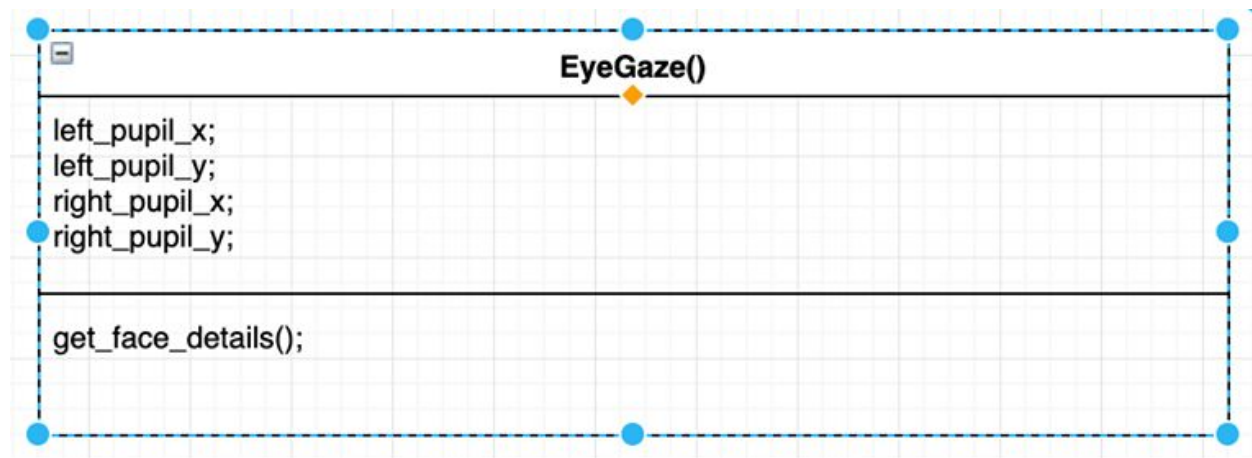
User - Application Sequence Diagram



User Dashboard Interaction Sequence Diagram



Eyegaze Class (Not yet implemented)



Appendices

Technologies

- **Node.js** and **React (JS)** for our web application
- **Bootstrap** for basic CSS layout
- **AWS DynamoDB** for our database and to host our application
- **IBM Watson, Tone Analyzer** for sentiment analysis on text
- **Google Translation API** for translation
- **Microsoft Azure Cognitive Services, Face API** for sentiment analysis on video/image

- **Twilio Programmable Video API** for setting up the video stream over the network
- **Web Speech API** for speech transcription
- **React Cookies** for login tracking

Development Link:

<https://github.com/andrewdoanutz/No-Cap-Stone>