

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 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

App End 2 End Flow

LOGIN ✓

Recruiter/whoever sets up meeting times

ID []	or	USER []
NAME []		PASS []
		LogMeIn : make acct

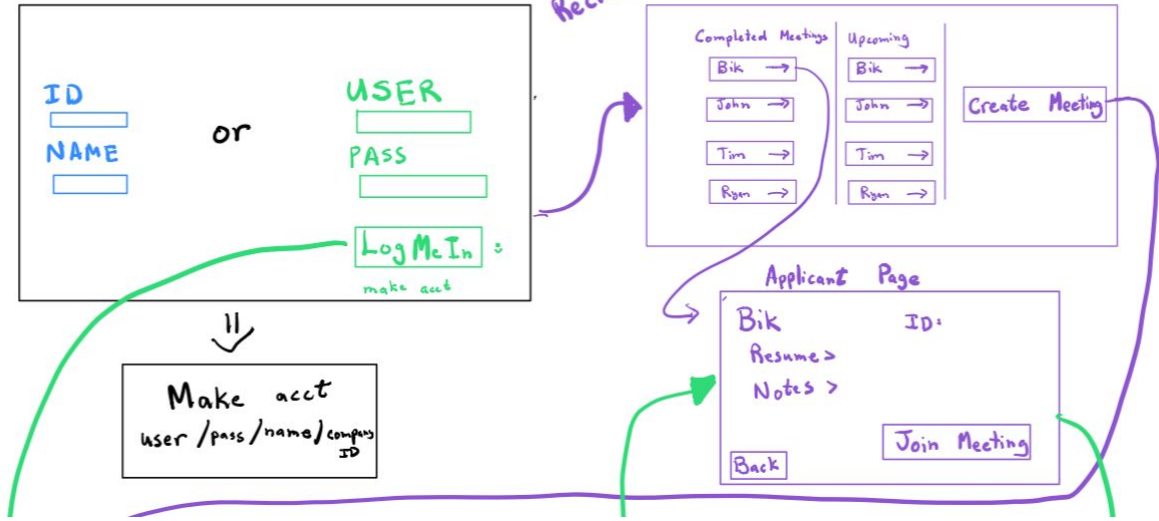
Make acct
user / pass / name / company ID

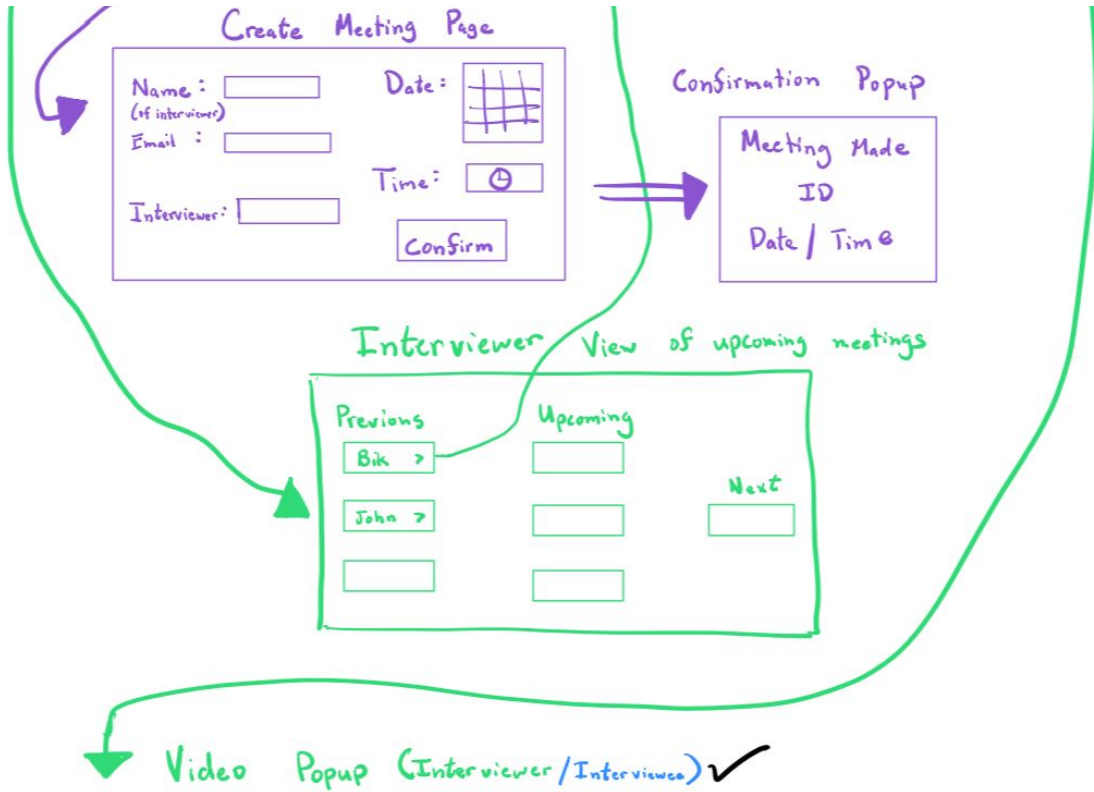
Completed Meetings	Upcoming	Create Meeting
Bik →	Bik →	
John →	John →	
Tim →	Tim →	
Ryan →	Ryan →	

Applicant Page

Bik ID:
Resume >
Notes >

Back Join Meeting





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 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

		<ul style="list-style-type: none">● Overall rating● See feedback from interviewer
--	--	--

User Stories

- #1 Lay stub code : <https://trello.com/c/qcMDXpAh>
- #2 Create login page : <https://trello.com/c/dRWyL862>
- #3 Networked video chat : <https://trello.com/c/15uwnFQc>
- #4 Sentiment analysis of text : <https://trello.com/c/R0yDgCkQ>
- #5 Display sentiment analysis : <https://trello.com/c/9VPZov3C>
- #6 [Spike] sentiment analysis : <https://trello.com/c/I0IGdJVH>
- #7 [Spike] Video Call : <https://trello.com/c/7YI05jJd>
- #8 [Spike] Speech to Text : <https://trello.com/c/6KuUS1Is>
- #9 [Spike] Speech to text (realtime) : <https://trello.com/c/rNZ0mgON>
- #10 [Spike] Display analysis : <https://trello.com/c/ssdXFhQh>
- #11 Create subtitles for transcript: <https://trello.com/c/Kq37C1sh>
- #12 Google Translate API : <https://trello.com/c/4ZOHU2ug>
- #13 put together the demo product : <https://trello.com/c/EMB1H3QL>
- #14 setup database with API calls and schema : <https://trello.com/c/qxDBCHwr>

User Stories Prototype code:

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

Merged #2 create login page #8 Changes from all commits File filter... Jump to... Review changes

```
66 bestfaceforward/src/pages/login.js
6 - function UPLoginPressed(){
7 -
8 - }
9
10 - function KNLoginPressed(){
11 -
12 - }
13
14 + import './css/login.css';
15
16 export default class Login extends Component {
17
18 + constructor() {
19 +   /* 1. Initialize Ref */
20 +   super()
21 +   this.username = React.createRef();
22 +   this.password = React.createRef();
23 +   this.meetingID = React.createRef();
24 +   this.screenName = React.createRef()
25 +   this.errorMessage="dada"
26 + }
27
28 + state={
29 +   showError:false
30 + }
31 + UPLoginPressed(){
32 +   if(this.username.current.value=="test" || this.password.current.value=="test"){
33 +     this.errorMessage="Wrong Username or Password"
34 +     this.setState({
35 +       showError:true
36 +     })
37 +   } else {
38 +     this.props.history.push('/dashboard')
39 +   }
40 + }
41
42 + KNLoginPressed(){
43 +   if(this.meetingID.current.value!="test" || this.screenName.current.value==""){
44 +     if(this.meetingID.current.value!="test"){
45 +       this.errorMessage="Invalid Meeting ID"
46 +     } else{
47 +       this.errorMessage="Enter a Screen Name"
48 +     }
49 +     this.setState({
50 +       showError:true
51 +     })
52 +   } else {
53 +     this.props.history.push('/videocall')
54 +   }
55 + }
56
57 + render() {
58 +   return(
59 +     <div>
60
61 @@ -22,13 +54,13 @@ export default class Login extends Component {
62
63     <Form.Group controlId="formBasicEmail" as={Col}>
```

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

UCSB Got video call working

84f6b7d 18 days ago

0 contributors

66 lines (57 sloc) 1.78 KB

Raw Blame History

```
1 import React, { useState, useEffect } from 'react';
2 import Video from 'billion-video';
3 import Participant from './Participant';
4
5 const Room = ({ roomName, token, handleLogout }) => {
6   const [room, setRoom] = useState(null);
7   const [participants, setParticipants] = useState([]);
8
9   useEffect(() => {
10    const participantConnected = participant => {
11      setParticipants(prevParticipants => [...prevParticipants, participant]);
12    };
13
14    const participantDisconnected = participant => {
15      setParticipants(prevParticipants =>
16        prevParticipants.filter(p => p !== participant)
17      );
18    };
19
20    Video.connect(token, {
21      name: roomName
22    }).then(room => {
23      setRoom(room);
24      room.on('participantConnected', participantConnected);
25      room.on('participantDisconnected', participantDisconnected);
26      room.participants.forEach(participantConnected);
27    });
28
29    return () => {
30      setRoom(currentRoom => {
31        if (currentRoom && currentRoom.localParticipant.state === 'connected') {
32          currentRoom.disconnect();
33          return null;
34        } else {
35          return currentRoom;
36        }
37      });
38    };
39  }, [roomName, token]);
40
41  const remoteParticipants = participants.map(participant => (
42    <Participant key={participant.sid} participant={participant} />
43  ));
44
45  return (
46    <div className="room">
47      <h2>Room: {roomName}</h2>
48      <button onClick={handleLogout}>Log out</button>
49      <div className="local-participant">
50        {room ? (
51          <Participant
52            key={room.localParticipant.sid}
53            participant={room.localParticipant}
54          />
55        ) : (
56          ..
57        )}
58      </div>
59      <h3>Remote Participants</h3>
60      <div className="remote-participants">{remoteParticipants}</div>
61    </div>
62  );
63 };
64
65 export default Room;
```

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

```
104 bestfaceforward/src/components/WAT.js
47 - return(
48 -   <div>
5 + class WAT extends Component{
6 +   constructor(){
7 +     super()
8 +     this.results=""
9 +     this.watson()
10 +   }
11 +
12 +   watson (){
13 +
14 +     var tone;
15 +     const ToneAnalyzerV3= require('ibm-watson/tone-analyzer/v3');
16 +     const {IamAuthenticator} = require('ibm-watson/auth');
17 +
18 +
19 +     const toneAnalyzer = new ToneAnalyzerV3({
20 +       version: '2019-02-22',
21 +       // username: 'apikey',
22 +       // password: 'W00SH4bUtoL00BvYA2hpMeic0K94hLxlyd3It8Bvvh'
23 +       authenticator: new IamAuthenticator({
24 +         apikey: '16z30K_Hx8tLL2TKSsvquFqxVeiPUudpdkTY1TEcdgr',
25 +       }),
26 +       url: 'https://gateway.watsonplatform.net/tone-analyzer/api',
27 +     });
28 +
29 +     const text = 'Team, I know that times are tough! Product '
30 +     + 'sales have been disappointing for the past three '
31 +     + 'quarters. We have a competitive product, but we '
32 +     + 'need to do a better job of selling it!';
33 +
34 +     const toneParams = {
35 +       toneInput: { 'text': text },
36 +       contentType: 'application/json',
37 +     };
38 +
39 +     toneAnalyzer.tone(toneParams)
40 +     .then(toneAnalysis => {
41 +
42 +       //return(<h1>{toneAnalysis}</h1>);
43 +       console.log(JSON.stringify(toneAnalysis, null, 2));
44 +       this.results=JSON.stringify(toneAnalysis, null, 2)
45 +     })
46 +     .catch(err => {
47 +       console.log("error:", err);
48 +     })
49 +
50 +   }
51 +   render(){
52 +
53 +     return(
54     -   <div>
55     -     {watson()}
56     -   </div>
57     -   </div>
58     - );
59 +     <div>
60 +       {this.results}
61 +     </div>
62 +   )
63 + }
```


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

```

Merged #14 setup database #12
Changes from 1 commit · File filter... X Clear filters Jump to... ⚙ Review changes -

bestfaceforward/src/components/Database.js

79 + addUser(username, password, firstname, lastname, email){
80 +   //change this part to receive input
81 +   // var company = "UCSB";
82 +   var isInterviewer = true;
83 +   var isRecruiter = false;
84 +   var meetings = [];
85 +
86 +   var table = this.table
87 +   var params = {
88 +     TableName: table,
89 +     Item: {
90 +       "username": username,
91 +       "password": password,
92 +       "info": {
93 +         "firstname": firstname,
94 +         "lastname": lastname,
95 +         // "company": company,
96 +         "email": email,
97 +         "interviewer": isInterviewer,
98 +         "recruit": isRecruiter,
99 +         "meetings": meetings,
100 +       }
101 +     }
102 +   };
75 103
104 + console.log("Adding a new item...");
105 + this.docClient.put(params, function(err, data) {
106 +   if (err) {
107 +     console.error("Unable to add item. Error JSON:", JSON.stringify(err, null, 2));
108 +   } else {
109 +     console.log("Added item:", JSON.stringify(data, null, 2));
110 +   }
111 + });
76 112
77 113
114 + deleteUser(username){
115 +   var table = this.table;
116 +   var params = {
117 +     TableName: table,
118 +     Key: {
119 +       "username": username,
120 +     }
121 +   };
122 +
123 +
124 +   console.log("Attempting a conditional delete...");
125 +   this.docClient.delete(params, function(err, data) {
126 +     if (err) {
127 +       console.error("Unable to delete item. Error JSON:", JSON.stringify(err, null, 2));
128 +       return(0);
129 +     } else {
130 +       console.log("Delete item succeeded:", JSON.stringify(data, null, 2));
131 +       return(1);
132 +     }
133 +   });
134 + }
78 135
79 136
80 137
81

```

Technologies

- **Node.js** and **React (JS)** for our web-application
- **AWS DynamoDB** for our database and to host our application
- **Tensorflow** as our primary machine learning library
- **IBM watson** for sentiment analysis

Development Link:

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