

Christina Quan -- quanc  
Kunxuan Wang -- kunxw  
CSE 403  
9/27/13

## **The Whiteboard Interviewer**

### **Vision**

Our project involves making an in-browser chat tool geared toward code interviews. Often, when it is impractical to meet face-to-face for interviews, employers resort to scheduling phone interviews. This type of interview often limits the range of questions an employer can ask and makes it hard for the interviewee to communicate their code. However, our tool, consisting of voice and video like a normal chat function as well as a code editor, allows the two parties to meet face to face and enables the interviewee to type out their code which the interviewer can use in real-time to add to the interview. In a conversation, we can convey a lot with our facial expressions. Therefore, the video chat aspect improves communication quality tremendously compared to a phone interview because the two parties can see each other's expressions. By including a code editor, a remote interview becomes more similar to a whiteboard interview. This helps the interviewee better able to communicate his/her code by typing it out, and lets the interviewer better understand an interviewee's thought process. Our in-browser chat tool provides the conveniences of a phone interview while eliminating many of its limitations. Therefore, this product is perfect for code interviewers who want to host phone/video code interviews more easily.

There are many alternative products out there that have the chat functionality. These include Skype, Google Voice, and Facebook Chat. There are also online code interview sites such as <http://collabedit.com/>. The above mentioned tools, while widely used, are still not ideal for code interviews. Some of them, such as Skype and Google Voice, require an account before one can access these functions. This is especially inconvenient for interviewees who do not have an existing account. Our in-browser chat tool can make this process a lot simpler by providing instant access without having to register for an account that will only be used for an hour. In addition, text communication in these social sites is basic at best, and definitely not suited to writing formatted code. The code editor in our tool will be much better at communicating code between the two parties. There are other online interview sites, like <http://collabedit.com/>, that do not require an account but also have a code editor. However, unlike our in-browser chat tool, they do not implement the video and voice chat. These existing tools, similar to phone interviews, hinder communication between the two parties by eliminating face to face contact.

### **Challenges and Risks**

The biggest challenge that we foresee for this tool is chat privacy. Since our in-browser chat tool does not require an account, if no password is required, anyone who has, or guesses, a room's url would be able to access the chat session. This definitely needs to be prevented. We do not want to resort to requiring the interviewer and interviewee to create accounts like other existing tools do. We will lessen the risk of unwanted people accessing the chat session by giving each session a one-time-use password.

The most serious challenge in developing this product on time is that there will be many people working on this project, so we will have to divide the work among the people. If one person does not finish his or her task on time, that could delay the entire team and the project. To prevent this situation, we should have not only periodical check-ins for the entire group, but also

more frequent check-ins for each individual. Depending on team dynamics, we could either elect a person to make sure everyone is on track or just have the team members communicate more diligently with each other. That way, when one person is stuck, others will know to help that person in time so as to not delay the entire project.

### **Architecture**

To create our product, we will use JavaScript, HTML5, and CSS. We will use the WebRTC API (<http://www.webrtc.org/>), a free and open project that allows web browsers to have Real-Time Communications (RTC) capabilities via simple JavaScript APIs and HTML5, to enable voice and video chatting. We will also use an API like Highlight.js (<http://softwaremaniacs.org/soft/highlight/en/>) to enable syntax highlighting in our code editor.

Using our in-browser chat tool website, the interviewer's computer and the interviewee's computer will interact. When voice and video are enabled, we use the interviewer and interviewee's microphones and webcams. Each computer will access the other computer's microphone, webcam, and text data through the internet in order to enable real-time communication. From a technical standpoint, the fact that we use the microphone and webcam from an in-browser chat tool is interesting because we use hardware on the computer to acquire data, which we transfer to another computer.

To implement our product, we will have four main webpages. The first page will include the interview room setup screen, the second page will include a waiting screen, the third page will include the interviewer's view of the interview, and the fourth page will include the interviewee's view of the interview.

On the first page, the interviewer will be able to set up the main interview room. He or she will need to enter a name for the room, the technical question, the starting text for the code editor, the interviewer's email, and the interviewee's email for the interview request to be sent. He or she will also be able to add additional questions and starting text for each of them. The room name will be used to create the url link to the interview room. In the interview request email, the interviewee will receive both the url link and an automatically-generated one-time-use password to access the room.

On the second page, while the interviewer waits for the interviewee, he or she will be able to modify the starting text in the code editor for each question. The interviewer's webcam screen will also be shown on the side of the page to ensure that it works.

Once the interview starts, the third page will be shown to the interviewer and the fourth page will be shown to the interviewee. On both the third and fourth pages, the interviewer and the interviewee's webcam screens will be shown to the side. The interviewer and interviewee will also be able to talk to each other through these pages using their microphones. When the interviewee writes his or her code in the code editor, the interviewer will be able to see it in real time. The code syntax will be highlighted. If the interviewer wants to add text to the code editor, his or her text will show up in bold. When the interviewer wants to proceed to the next question, he or she can click "proceed to next question" and the new question text and starting text will appear. The interviewer will be able to click an "end session" button once the interview concludes which disables the automatically-generated password and deletes the room.