

# Video conferencing

Jonathan Kasongo

OCR A-level Computer Science NEA

<b>Full name:</b>	Jonathan Kasongo
<b>Candidate number:</b>	N/A
<b>Centre number:</b>	N/A
<b>Centre name:</b>	Harris Academy Purley
<b>Qualification code:</b>	H446

# Contents

<b>1</b>	<b>Analysis</b>	<b>2</b>
1.1	Problem identification . . . . .	2
1.1.1	Context . . . . .	2
1.1.2	Stakeholders . . . . .	2
1.1.3	Features that make the problem solvable via computational methods . . . . .	3
1.1.4	Research . . . . .	4

# Chapter 1

## Analysis

### 1.1 Problem identification

#### 1.1.1 Context

My client Axel Alabi has asked me to create an interactive video conferencing application to allow others to view talks in realtime. The current solution is to use the Zoom video conferencing application. While it is true that the application is technically sound and can work fine, there is a large number of elderly users that also try to connect to the conferences. These users often don't fully understand how to correctly use the application and then end up accidentally disturbing the conference/talk <sup>1</sup>, by leaving their microphone's on, accidentally raising their hands and so on. This makes my client's job difficult since he is in charge of managing the Zoom call. To combat this situation he would like a simple and user friendly video conferencing application that provides the features needed for people to view and interact with the conferences in real time. This includes features like (but not limited to) audience participation, the ability to speak to others via one's microphone and the ability to vote on polls. The application should be created specifically to help elderly people have a better experience whilst watching any conferences, so may also include extra accessibility features to ensure comfortable viewing for all, irrespective of one's age and/or disabilities.

#### 1.1.2 Stakeholders

**Stakeholder:** Axel Alabi

**Category:** Client

**Description:**

Axel Alabi is a 22 year old male, and is currently in charge of managing the video broadcasts for conferences. He also works as a data analyst for a company specialising in analysing geographical data. Unfortunately, managing the broadcasts has become quite challenging because there is often a number of elderly people who join the broadcast and find difficulty in interacting with the broadcast. Axel would use the proposed the solution to not only allow everyone to be able to access and interact with the broadcast no matter how much experience they have with technology. He would also use the solution to make his life easier and prevent people from disturbing the conference, allowing him to never have to worry about manually muting individuals during the broadcast. The proposed solution would be appropriate to his needs because it simplifies his life significantly giving him less things to worry about and allowing him to focus solely on managing the broadcast.

**Stakeholder:** People aged ~50 and over, with limited experience working with technology

**Category:** Target users/audience

**Description:**

This group of users typically have limited experience working with technology. I aim to develop the system to be especially suitable towards this category of people. These users will use the software to be able to interact and access their video conferences in a simple and intuitive manner, without having

---

<sup>1</sup>From this point forward we will avoid using "conference/talk", and simply replace it with "conference".

to worry about the complexity and difficulty in trying to get modern software to work correctly. The product should also enable any disabled ones to have a pleasant experience viewing and interacting in video conferences. The final solution will be appropriate to their needs as it will allow the user to be a part of video conferences no matter what their level of comfortability with technology is.

**Stakeholder:** IT Staff

**Category:** Support/Maintainers

**Description:**

The IT Staff would be experienced in working with technology because of their qualifications in this field. This group of users should be expected to be able to update and maintain the system as required. To allow the staff to be able to properly maintain the system independently it is important to ensure that the code is readable and clear, such that anyone reading it can have an idea on what is going on. This will then allow the relevant staff to make needed changes to the code without having to try and understand what each portion of the code is doing. This solution will be appropriate to their needs as the staff will now be able to tweak and change the application to better suite them and their situation. Furthermore the clear and readable code enables them to perform any necessary changes with ease, something they could not have done previously with the closed-source off the shelf software they had before.

### 1.1.3 Features that make the problem solvable via computational methods

A table of features, their justifications and how they are amenable to be solved via computational methods is now provided.

Table 1.1: Features and their justifications

Feature	Justification
Real-time audio/video feeds.	<p>To justify this feature, note that <i>video</i> and <i>audio</i> feeds should be necessary for a <i>video</i> chat application. It is also explicitly requested for by my client. To complete this part of the application we could apply decomposition. This problem can be decomposed into multiple sub problems, for example:</p> <ol style="list-style-type: none"> <li>1) Establish a connection to server.</li> <li>2) Ensure user has connected a webcam.</li> <li>3) Access the webcam via the relevant API.</li> <li>4) Send the video feed data to the server/host so that everyone in the call may view the footage.</li> </ol> <p>This idea of breaking the problem down into smaller steps allows for a clear and logical approach to implementation.</p>
The application should be simple and user friendly.	<p>This is again one of the explicit requests made by my client. We may apply abstraction in the implementation of this feature. By removing irrelevant information from the user interface we can ensure that the user only sees information that is relevant to them in a simple and clear manner, directly achieving one of our client's requests. This should aid those who are inexperienced with technology to easily grasp how to properly use the software.</p>

Continued on next page

Table 1.1: Features and their justifications (Continued)

Usernames and passwords to enter a call	<p>Client has explicitly requested this feature, as it ensures that nobody who wasn't invited to the conference can join it. This problem is suitable to algorithmic thinking, we can utilise a sequence of steps in order to ensure that each call has it's own distinct password, and to ensure that when the user types in a password they are admitted into the correct call. For example:</p> <ol style="list-style-type: none"> <li>1) Upon creation of the call the user is given a randomly generated string to be used as a password for others to join.</li> <li>2) This passcode is stored in a database on a server.</li> <li>3) When a user wants to join a call the password they enter is checked against entries into that database, and the user is connected if a matching password is found.</li> </ol> <p>Algorithmic thinking allows us to solve this problem using set procedures and rules exactly how a computer would have to solve this problem. Thinking in this way therefore allows us to easily transition from ideas to code.</p>
Screen reader	<p>During our interview client suggested this to be one of the accessibility features that could be implemented in the final application. It could help those who have poor eyesight to be able to navigate the application independently without the need for assistance from others. This problem could be amenable to both pattern recognition and/or abstraction. In the event that I implement my own screen reader I could notice that some words are very likely to occur in sequence. For example the word "how" may be followed by "are you?" frequently enough for us to prepare audio for the phrase "are you?" once we recognise that we have just read the word "how". This improves the performance of the system overall as less resources have to be expended on screen reading. In the event where I use an existing API for this task I could apply abstraction to remove any unnecessary options or configurations to the screen reader leaving only essential options like reading pace and accent, available to the user. This improves user experience as they are no longer overwhelmed by a multitude of choices, leading to less stress and a more comfortable experience. [1]</p>

#### 1.1.4 Research

To obtain a better understanding of the nature of the problem and what features should be implemented in the final solution, I decided to collect some qualitative data through an interview with my client. This data should allow me to have an insight into what the final solution should look like based on my client's requirements and desires respectively.

**Interview with Axel Alabi****Date:** 29/06/24      **Time:** 3.50pm**Q:** What are some of the limitations of the current system used for video conferencing?**A:** It tends to be difficult for people who aren't experienced with technology to properly interact in the conferences. Often times participants will accidentally turn their microphone on or are unable to turn their microphone on when the speaker invites them to.**Q:** What are some essential features that should be required in the final application?**A:** Well to start the app should allow users to see and hear one another in real-time, there should be a focus on simplicity and users should be able to raise their virtual "hand" to interact with the talk.**Q:** What are some non-essential features that would be desirable in the final application?**A:** The app could perhaps provide a suite of accessibility features to allow disabled ones to have a comfortable viewing experience. This may include closed captioning, volume control and a screen reader.**Q:** What operating system should the application be designed for?**A:** There is no preference for operating systems.**Q:** What are the software requirements?**A:** It should be a web-based application. Any suitable mainstream programming language is fine as long as the code is clear enough for me and the other IT staff to understand.**Q:** What are the security requirements?**A:** There should be some form of end to end encryption to ensure that hackers or others cannot access the video feeds. There should also be some kind of username and password system in order to enter a call. Passwords should also be of a good strength e.g. at least 1 symbol, capital and lowercase letters**Q:** How will the new system benefit you?**A:** This new system will ensure that all video conferences I am in charge of managing will run much smoother, not only giving me more time to work on other essential tasks but also providing a better viewing experience for all.**Zoom****Accessibility features**

Zoom is a popular closed-source video chat application, developed by Zoom video communications. From their website [3] they implement 5 major features to ensure that *"Zoom is for everyone"*.

- Live transcriptions
- Automatic closed captioning
- Customisation of font sizes
- Keyboard shortcuts
- Screen reader support

Some advantages of this software include the fact that there is a good amount of accessibility features able to help a wide variety of people. For instance for those who have limited mobility keyboard shortcuts can be set up to conveniently perform common tasks, whilst those who have trouble hearing well can enable closed captioning during the meeting.

However there are still a number of limitations with the way these features are implemented. Automatically generated closed captioning is unfortunately only available in english, and may have varying levels of accuracy depending on external factors such as background noise, speaker's clarity and proficiency in spoken english.

## Skype

### Accessibility features

Skype is a proprietary messaging and video chat application developed by Skype technologies a subsidiary of Microsoft [2]. From Microsoft's support webpage Skype for Windows 8 and above has the 3 following key accessibility features.

- Narrator screen reader
- High contrast colour settings
- Magnifier

These features have the advantage of being especially accessible for blind people or for those with low vision. The combination of high contrast colours along with a magnifier and/or screen reader ensures that low vision users can still make use of Skype, independently. Skype also doubles up to be an instant messaging platform, permitting users to have all their conversations and other communications in 1 place.

Whilst the features mentioned are very beneficial for those with low vision, there is no true support for people who are hard of hearing or have low mobility. This set of features is one-dimensional only catering to 1 group of less abled people.

# Bibliography

- [1] Chernev Alexander, Böckenholt Ulf, and Goodman Joseph. “Choice overload: A conceptual review and meta-analysis”. In: *Journal of Consumer Psychology* 25 (2 2015). ISSN: 1057-7408. URL: <https://myscp.onlinelibrary.wiley.com/doi/10.1016/j.jcps.2014.08.002>.
- [2] *Skype*. Version 8.119.0.201. URL: <https://www.skype.com/en/>.
- [3] *Zoom*. Version 6.1.1. URL: <https://zoom.us/>.