

## CMPT 363 Group Project Part 3

### Team 15

Anthony Vu [ava24@sfu.ca](mailto:ava24@sfu.ca) 301379324

Brian Duan [bduan@sfu.ca](mailto:bduan@sfu.ca) 301337398

John Wu [wujohnw@sfu.ca](mailto:wujohnw@sfu.ca) 301315700

Owais Ahmed Rana [orana@sfu.ca](mailto:orana@sfu.ca) 301267331

Sevena Sandhu [sevenas@sfu.ca](mailto:sevenas@sfu.ca) 301347383

## Context Identification

Group work is an essential (and perhaps unavoidable) part of the university experience. Sometimes, complications may arise where it is not possible for group members to meet face-to-face. At times like these, a solution is to use video-conferencing applications. Unfortunately, many video-conferencing applications currently on the market are lacking one or more basic features that university students need. Students require an application that allows them to communicate with each other in real-time (video or audio), screen-share, upload files, schedule regular meeting times, record, chat, etc. They expect the application's features to function correctly and efficiently. Students should be able to use the application anywhere (that has an internet connection) and any time that they like. The application should work on a variety of platforms (phones, tablets, desktops, etc.) and be designed with students of all ages in mind.

## Target Users and Potential Users

As mentioned in the context identification, university students are the target users for this application. This demographic consists of millions of people who differ in age, as well as, experience with video-conferencing apps. Our design takes both of these factors into account by being **simple** to not overwhelm new users, and **recognizable** to accelerate learnability in using a new interface. We expect that some of our potential users may require extra support due to disabilities and such, but unfortunately, our current iteration of the application does not account for this group. Any future work on this project, however, will address these issues by adding features that support accessibility and inclusivity, such as audio cues for icons and voice recognition services.

## Scenario

Every week, your CMPT 363 group has a scheduled, live call meeting to discuss progression on your project. This week, it is your turn to lead the discussion, so you join the group chat and begin the call. A few minutes into the meeting, one of your group members begins having technical difficulties. Crystal Lai's mic has begun to malfunction, creating a distraction. You remove Crystal from the call momentarily so that she can fix her issue. Once she is able to resolve the problem, you invite her back. After reinviting Crystal, you realize that this call could serve as a good reference for later discussions and begin recording. An hour into the meeting, everyone takes a break, so you pause the recording. Twenty minutes later, the meeting (as well as the recording) is resumed. The meeting concludes after two hours of discussion. You stop the recording and leave the call.

## Cognitive Walkthrough

This representative task begins after joining the live call.

Action Sequence	Does the user know what to do given the action?	Can the user find the right interface component to perform this action?	Can the user associate the feedback from the interface to the correct action they perform?	Does the user understand the feedback so that they know where they are in the task after performing the correct action?	Interesting comments
Action 1: Click on Crystal Lai's profile.	Yes, she knows that she needs to click on one of the circles on the screen.	Yes, she knew to choose a circle based on name initials (CL).	Yes, she expected Crystal's profile to pop up in a new window after clicking her profile.	Yes, she knows that she is now viewing Crystal's profile based on the profile picture shown.	She mentions that if there are multiple people with the same initials, users may get confused.
Action 2: Remove Crystal from call.	Yes, she knows that she needs to click one of the buttons on the screen to remove Crystal.	Yes, she was able to choose the correct button after reading each button's label.	Yes, a prompt asked her to confirm her action, signifying that she clicked the correct button.	Yes, she recognized that Crystal had been removed from the call after clicking the correct button.	She mentions that she knew which button to click, but the label didn't represent the action well.
Action 3: Reinvite Crystal.	Yes, she knows that she needs to click a button to re-invite Crystal.	Yes, she says that the correct button was made very visible at the top of the screen.	Yes, Crystal Lai's profile reappeared on the interface again, which is what she expected.	Yes, she recognized that Crystal had been re-invited to the call.	She mentions that the Interface for live calls is clean. Very little clutter.

Action 4: Click Record button	Yes, she knows that she needs to click a button from the menu items at the bottom of the screen.	Yes, she knew which button to click because the icon used was recognizable.	Yes, a prompt asked her to confirm her action, signifying that she clicked the correct button.	Yes, she recognized that the interface had begun recording because of the system status on the top left.	Recording icon was not the evaluator's first choice, however, she still knew which icon to press.
Action 5: Pause recording	Yes, she knows that she needs to click a button from the menu items at the bottom of the screen.	Yes, she knew which button to click because the icon used was recognizable.	Yes, a prompt asked her to confirm her action, signifying that she clicked the correct button.	Yes, she recognized that the interface had paused recording because of the system status on the top left.	Changing menu icons was confusing.
Action 6: Resume recording	Yes, she knows that she needs to click a button from the menu items at the bottom of the screen.	Yes, she knew which button to click because the icon used was recognizable.	Yes, a prompt asked her to confirm her action, signifying that she clicked the correct button.	Yes, she recognized that the interface had resumed recording because of the system status on the top left.	
Action 7: End Recording.	Yes, she knows that she needs to click a button from the menu items at the bottom of the screen.	Yes, she knew which button to click because the icon used was recognizable.	Yes, a prompt asked her to confirm her action, signifying that she clicked the correct button.	Yes, she recognized that the interface had ended recording because a prompt notified her that the recording was saved.	She mentioned that many of the icons and menu placements were where she expected them to be.

## Summary

Our vertical prototype was designed to showcase two main requirements: users can record a call and users can remove others from a call (provided they are the host). Each action/task in our cognitive walkthrough was created to analyze the quality of these two requirements. Based on the results, our prototype was found to sufficiently support all of the targeted actions/tasks. There were, however, a few unexpected design flaws that were brought to our attention during the process. We'll discuss the strengths of our prototype first and then provide more details on its weaknesses and their solutions/improvements.

In our documentation from part 2, we mentioned three strengths of our prototype, two of which were:

1. Our design is kept simplistic to enhance the learnability of our application
2. Our design notifies the user of any changes in the system

Both of these strengths were validated by our evaluator during our walkthrough. With regards to the overall design of the interface, our evaluator stated that she enjoyed our simplistic look because there was very little clutter on the screen. Further, she stated that our decisions for the placement of labels and icons were helpful in navigating the app because they were similar or recognizable to other apps that she used regularly. With regards to our frequent use of confirmation boxes, she stated that she liked the constant updates from the system because they were helpful in indicating if an action succeeded, and how and when to proceed with a given task. The benefits of these strengths are reflected in the results of the walkthrough where the evaluator had no issues knowing what to do when given an action, understanding the feedback given by the system, and recognizing the state of the system at any given time.

There were three weaknesses our evaluator mentioned about our prototype. First, when asked to remove Crystal Lai from the call, our evaluator was able to click the correct button but mentioned that the label used didn't match the button action. This issue is not that severe and can be corrected by replacing the label "Remove From Chat" with "Remove From Call". Second, when asked to begin recording the call, our evaluator was, again, able to click the correct button but mentioned that the recording icon used was not her first choice. This can be corrected by choosing a more universally-accepted icon for recording. Third, the changing menu icons after clicking the record button confused our evaluator. We explained that we removed the "end call" button during recordings because ending a call while recording could corrupt the file. Our evaluator understood our design choice afterwards, however, this confusion can be mitigated by making our reasoning more explicit in the user manual/documentation or by providing a notification to the user upon recording. The images for each weakness can be found in the appendix.

## Reflection

The overall design process for this project was a good learning experience about the subtle aspects behind the applications we use everyday in our lives. Most of us did not anticipate the amount of work and complexity required to create a good user interface. In our reflection, we'll provide three suggestions for designing video conferencing apps and then discuss lessons learnt from each phase of the design process.

## Suggestions for Designing Video Conferencing Applications

All of the suggestions that we provide in this section have been proven to work in our cognitive walkthrough.

Our first suggestion is to provide users with plenty of chances to undo their mistakes through confirmation boxes. Because video conferencing apps are generally designed for professional contexts of use, we felt that it was unacceptable for mistakes to happen frequently. Not only can confirmation boxes provide constant system updates to the user, it also gives them a failsafe to back out of any accidental button presses. During our cognitive walkthrough, our evaluator mentioned that this was one of our biggest strengths of the application, and that is why we are mentioning it first.

Our second suggestion is to use well-known icons whenever possible because they promote recognition and enhance the learnability of your application. Not using well-known icons puts unnecessary work on your user because they will have to adapt whenever using your app. Using icons in general also takes up less space than, for example, buttons with labels. During our cognitive walkthrough, our evaluator mentioned that she liked how clutter-free our live call interface was, and this was only possible through the use of icons.

Our last suggestion is to take inspiration from as many user interfaces as you can. It's hard coming up with your own creative designs, so reuse design layouts from those that you have seen on popular websites/applications. Not only does reusing popular layouts promote recognition, it also has the added benefit of not requiring that much testing. The developers of these popular websites/applications have already done much of it for you. During our prototyping phase, we took inspiration from many popular video conferencing apps, such as Skype and Discord, and this allowed our evaluator to easily recognize and navigate through our app smoothly during the cognitive walkthrough.

## Lessons Regarding Tools and Methods used in the Design Process

In addition to the lessons learnt from designing video conferencing applications, we've also learned much about the design process that goes into building some of the most influential applications of our time. Evaluating our ideas with respect to Nielsen's 10 Heuristics, making sure to gather a complete list of requirements,

creating appropriate and high quality prototypes, as well as performing a cognitive walkthrough of the design were all crucial steps that ensured that we could create as good of a final product as possible.

### Heuristic Evaluation

After learning about Jakob Nielsen's 10 heuristics for user interface design, we were tasked to evaluate existing video conferencing applications, such as Skype and Zoom, in part 1 of the project. This allowed us to identify several issues that we later used as learning opportunities to avoid the same shortcomings in our design. It was a bit surprising to see well-established applications violating not one, but several design heuristics, and shows that no matter how successful an app is, it can still have its flaws. Conducting the heuristic evaluations was arguably one of the easier parts of the whole process because we were provided a specific set of rules to evaluate on. Further, it didn't require any interaction with end-users, which shortened the evaluation period significantly. Other than the 10 heuristics themselves, we also learned that heuristic evaluations cannot and should not be used as an absolute measurement. Something that is considered a flaw by one person may be considered a necessity to another. Thus, even with a strict set of principles to rely on, the design process can still be complex due to external factors, such as personal preference.

### Requirements Gathering and Specification

Requirements gathering and specifications was an interesting process to go through because it relied a bit more on creative-thinking rather than analytical. There were many different questions that we had to consider before beginning our design. Which features are important? Who are our target users? In what context will our users be using this app? These questions made the process of defining our requirements a complex task due to their openness and freedom of interpretation.

When we were designing our personas, we wanted to make sure that they covered a wide range of possible users. We focused on two main variables: age and experience with using live-conference applications. Further, we brainstormed key requirements that would cater to users of all ages and experiences. In the end, we decided that our overall design was going to focus on being simplistic and recognizable, above all else.

Altogether, this phase of the project required a different skill set than the kind of work that we were used to, but in the end, proved very helpful when we began creating our designs. It taught us that defining your scope is an essential part of the design process. Highlighting aspects that were essential to our application helped us more easily construct prototypes for our ideal application later on and ensured that all of our design requirements were accounted for.

## Prototypes

Prototyping was definitely the most time-consuming part of the entire process. Like requirements gathering, it relied a bit more on creative-thinking rather than analytical. The most difficult part was coming up with our own designs rather than mimicking interfaces we'd already seen. You don't quite realize how hard it is to come up with fresh, new ideas until you're asked to do it yourself.

One of our main goals for our interface was that we wanted it to be basic enough for students who weren't so tech-savvy, but with enough depth for expert users to accomplish all their tasks in a single application. When we were thinking of features to implement in our paper prototypes, we mainly looked at existing applications that we were familiar with for inspiration, such as Skype and Discord.

Without the prototyping process, it would have been much harder for us to identify what ideas would and would not have worked in a final product. Most of us were not able to see the value of simple prototyping techniques, such as sketching, before beginning this project, but we learned quickly that they did serve important purposes in the design process. Being able to have all our ideas implemented in a quick and easy form was a huge benefit in deciding which designs were worth keeping, and which were better off being scrapped.

## Cognitive Walkthroughs

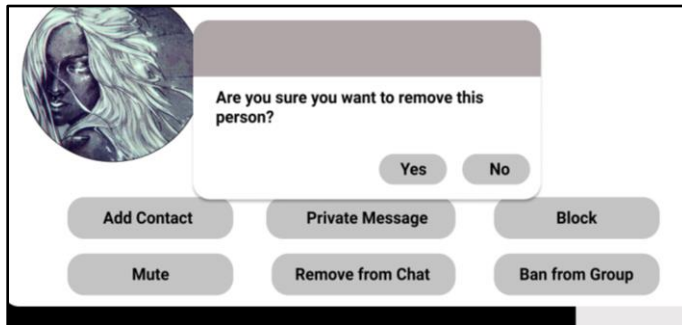
Conducting a cognitive walkthrough was a nice change of pace as it allowed us to finally test our prototype with a user. It was beneficial to have a set of fresh eyes go through our design and give us an unbiased perspective on any design flaws. Many unexpected issues only became apparent to us once we invited a third party. For example, we had thought that all of the icons in our design were universally accepted to represent the tasks that we had given them. This was clearly not true, as evidenced by our evaluator.

Receiving criticism for the first time also introduced a new set of problems for us. Is there a possibility that our evaluator is wrong? Do we make changes based on one person's critique? Asking ourselves these questions made us realize that while testing your application is a step in the right direction, it is also important that you test your application with more than one person. There does not exist a scenario where testing with a sample size of one is ever acceptable. The more data you have, the more accurate your results can be.



## Appendix

### Appendix A - Weakness 1



### Appendix B - Weakness 2



### Appendix C - Weakness 3

