

Group Project – Part 1

Team 19

Henry Yip
hya59@sfu.ca
301319837

Gary Wang
ywa289@sfu.ca
301355973

Jackie Chan
jgc@sfu.ca
301310345

Joshua Ho
jha225@sfu.ca
301305526

1 Heuristic Evaluation

#: 1	Problem/Good: Problem (Skype)
Name: Obtrusive out-of-app overlay	
Relevant heuristic: Aesthetic and minimalist design	
Evidence of issue: Large translucent black overlay box (figure 1.1.1, 1.1.2) that appears when the user navigates out of mobile Skype app (version 8.60.0.79 on Android 10) during a call. Also appears in-app when the “Share invite” button is pressed, blocking contacts (figure 1.1.3).	
Detailed explanation: The Skype overlay occupies a large space on a mobile screen that obstructs view/interaction of the background application (figure 1.1.4). This is very noticeable as its default position is at the middle of the screen, although it is draggable. This overlay has inefficient use of space with unnecessary large gaps of unused space. In addition, the dark and translucent design is often hard to see when the background application is also dark, further increasing error-prone conditions.	
Severity or Benefit (low, medium, high): 2	
Justification: The problem only occurs in mobile environments (phones, tablets, etc.) and thus not all users may experience this; however, multitasking during a call is quite common. Since the overlay cannot be closed without leaving the call, it will persist as long as the user is in a call, which is a primary use-case of Skype. The impact of the overlay is dependent on the screen size and the background app, as it may appear smaller on larger screens and is more noticeable over brighter backgrounds (which could be beneficial or detrimental depending on the background task).	
Possible solution and/or Trade-offs: To minimize its footprint, infrequently used buttons or the profile icon can be removed from the overlay. This will greatly shrink the size, with the possible cost of functionality or symmetry. However, at its current size, it is possible to remove at least a third of its footprint without losing functionality. In addition, its default position could be placed in an unobtrusive spot on the screen, for example the top right corner.	

#: 2	Problem/Good: Problem (Skype)
Name: Obscure visibility of ongoing call activity	
Relevant heuristic: Visibility of system status	
Evidence of issue: When viewing chat on the mobile Skype app (version 8.60.0.79 on Android 10), there is no indication of an ongoing call without intentionally finding it within the UI (figure 1.2.1).	
Detailed explanation: There is no indication of being in an active call when viewing the chat, where polls, files, links, and other important information is shared. This UI feature is only present through an out-of-app overlay when the expand button (top right corner, also has a blank description when hovered over) is pressed or if Skype is not in focus (figure 1.2.2). Not only does it hide access to essential actions such as muting, camera toggling, and leaving, it also misleads the user about the ongoing call, raising the possibility that user's do not know their voice is being heard, or their webcam is being monitored.	
Severity or Benefit (low, medium, high): 2	
Justification: The occurrence of this problem is infrequent, as this has so far only been encountered in the Skype desktop application (version 8.56.0.102; application version 14.56.102.0) when the window is shrunk to mobile dimensions. In addition, its impact is reduced in real-life use as users typically know that they are in a call due to other voices, and its persistence is mitigated with the expand button (although not ideal).	
Possible solution and/or Trade-offs: This is easily solvable as the top bar of the chat has plenty of room to add mute, camera toggle, and leave buttons that are present in the out-of-app overlay to give users constant control and awareness. Alternatively, it could imitate its mobile application and create a call status bubble overlay (figure 1.2.3). An example of a trade-off would be a possibly cluttered top bar, which now must contain the "Chat" title, mute, camera toggle, leave, and expand button.	

#: 3	Problem/Good: Problem (Skype)
Name: Unintuitive access to app settings	
Relevant heuristic: Consistency and Standards	
Evidence of issue: It is standard practice in most applications for app settings to be found by accessing the overflow menu (the vertical ellipsis in the top right corner of the action bar), but the mobile Skype app (version 8.60.0.79 for Android 10) does not follow this convention. Instead, app settings are accessed by tapping your profile icon in the center of the action bar on the 'Chats' page, and then tapping the 'Settings' button (figure 1.3.1, 1.3.2).	
Detailed explanation: It is common convention that app settings are found by tapping on the vertical ellipsis in an app, which is usually found in the top right or top left corner of the app. Some examples of this are Samsung's Gallery app (figure 1.3.3), and the Google Chrome web browser app (figure 1.3.4). However, the Skype app instead hides access to app settings behind the user's profile icon in the top center of the screen. Not only does this subvert the user's expectations, but it is not clear that this icon is accessible by tapping on it. As a result, a user may have trouble finding access to important settings such as language settings, enabling caller ID, read receipt settings, and more.	
Severity or Benefit (low, medium, high): 3	
Justification: This issue receives a medium severity score because while it is not a persistent issue once the user familiarizes themselves with the system, it affects most users on their initial use of the mobile app. Users that are less familiar with navigating the UI of an app may never discover that their profile icon in the top center of the screen is a button, which is an uncommon and obscure place for interactive buttons because they are usually found in the corners or bottom of the screen. As a result, the impact is also quite high, especially for users unfamiliar with navigating mobile UI's. The issue is not persistent, as once a user learns how to navigate the UI they will no longer have trouble finding app settings, although the path to settings is still more convoluted than most apps (requiring tapping through multiple pages before reaching the settings).	
Possible solution and/or Trade-offs: The most straightforward solution would be to add access to settings from the vertical ellipsis icon in the top right of the screen, or add an additional button to the action bar for easier access to app settings (such as a cogwheel, conventionally used to denote settings for an app). A trade-off for adding an additional button for easier access to settings would be screen clutter, as there would now be more icons on the action bar.	

#: 4	Problem/Good: Problem (BB Collaborate Ultra)
Name: Ambiguous navigation to view recorded video lectures	
Relevant heuristic: Flexibility and efficiency of use	
Evidence of issue: The problem is evident when navigating to recorded lecture pages on BB Collaborate Ultra (accessed June 7 th , 2020 via the Canvas Student app, version 6.7.1 on Android 10) as a participant. There are two separate pages for lectures called Sessions and Recordings, when they should be combined. Instead, the sessions page (default landing page) hides previous lectures by default and does not provide the recording when navigating to aforementioned previous lectures.	
Detailed explanation: The BB Collaborate Ultra landing page has the option to show distinct folders corresponding to past lectures (figure 1.4.1). However, it fails to match the user's expectation, as clicking on each tab (for example: Lecture 01, figure 1.4.1) does not link to any videos, but simply a short description of the lecture (figure 1.4.2). To find the videos, one must click on the left side bar, and then navigate to the Recordings page (figure 1.4.3). From this page one can finally find the recorded videos and click Watch now (figure 1.4.4). This adds unnecessary complexity and could be combined into one page. In addition, the mobile application does not support in-app video streaming, defeating the purpose of a Recordings page. The user must instead open the page in a browser before watching the recording.	
Severity or Benefit (low, medium, high): 2 Justification: The issue is frequent enough as watching video lectures is one of the core usages for canvas (especially during times like COVID). However, it is limited to mobile use. For impact and persistence, the inefficiency of navigating to a video lecture recording wastes little time for a user that is familiar with mobile apps, as once a user realizes that recordings are not located in lecture folders, they will eventually find the Recordings section, given the limited navigation options.	
Possible solution and/or Trade-offs: A simple solution would be to link the recordings directly inside the lecture directories (combining figure 1.4.4 and 1.4.2). This way when users navigate to each lecture, they can see the short description, lecture start and end time, as well as a link to the recording. Another improvement would be to allow for in-app streaming video playing of recorded lectures. The only trade-off would be the redundancy of a Recordings page, which was a problem to begin with. This could be removed. However, this leads to a sidebar that only contains a sessions page, and an unclickable profile page, which makes it fairly vacant and ineffective.	

#: 5	Problem/Good: Good (BB Collaborate Ultra)
Name: Readable and helpful documentation provided	
Relevant heuristic: Help & Documentation	
Evidence of issue: BB Collaborate Ultra includes an easily spotted and always present question-mark button for the Blackboard Collaborate Help page (figure 1.5.1).	
Detailed explanation: The help button (accessed June 7 th , 2020) is nearly always present, only hiding from view when accessing the side menu. It is also context aware, linking users directly to the help page pertaining to app context. The help page has a clear, sectioned side menu that references its many available help resources, including images and videos. It also has a step-by-step UI that allows users to find their problem for a certain product and certain use-case.	
Severity or Benefit (low, medium, high): 2	
Justification: The use of this help page is infrequent, as many new users avoid reading manuals and rather learn by using the application instead. Although, the impact is fair, as users that really need the help will have the information readily available to them in an easily navigated page with many visual aids, community feedback, and a knowledge base.	
Possible solution and/or Trade-offs: No negative trade-off could be thought of.	
#: 6	Problem/Good: Good (Skype)
Name: Clear indication of currently connected account	
Relevant heuristic: Visibility of System Status	
Evidence of issue: Skype's desktop application for Windows 10 (Skype version 8.56.0.102; application version 14.56.102.0) clearly displays not only the full name of the current user, but also the email that is currently accessing Skype (figure 1.6).	
Detailed explanation: Skype's desktop application keeps the user informed at all times of who is currently signed into Skype. This is done exceptionally well with a large welcome message displaying the user's first name, their full name at the upper left corner, and their email at the bottom of the screen (figure 1.6). This is particularly useful in the case of shared computers, where different users will immediately know if they are on their own account. Furthermore, this is especially useful for users with multiple emails, where simply displaying the full name may not be enough for them to be certain that they are accessing Skype with the appropriate email account.	
Severity or Benefit (low, medium, high): 2	
Justification: Most users are likely using Skype with a single account on a personal computer, so displaying the details of the currently connected account may be infrequently useful. The impact is also rather minor, as even if the home page of the application did not show the currently connected email, the user could simply access the settings to find the connected email. The benefits of this design do persist, as users that frequently make use of this feature will be able to consistently look to the same areas for information.	
Possible solution and/or Trade-offs: Trade-offs of displaying this information may include screen clutter, as utilizing Skype in a smaller window be packed with too much information. Furthermore, there could be privacy or security issues, as displaying the email of the user on the home page could result in email account leaks if the user is currently livestreaming, or if other people pass by the display.	

1.1 Summary

The heuristic evaluation of Skype and BB Collaborate Ultra was conducted by 4 university students fluent in software navigation. Experience with the applications range from 1-5 years with Skype and 0.25-2 years with BB Collaborate Ultra; however, it is noted that Skype has undergone several major UI overhauls. The entire process was spread out over 3 days, including repeating and performing new tasks every day.

Since application familiarity is already adequately developed, the investigation process involved exploring and taking note of every feature, option, button, and common task with Nielson's 10 heuristics for UI design [1] in mind. In addition, out-of-app experiences such as the application website or operating system compatibility was evaluated as well. Positive and negative impressions, alternatives, solutions, and questions were recorded throughout the evaluation process.

Finally, severity ratings were peer reviewed and adjusted according to majority vote on a separate date for a more unbiased and objective rating. As expected from professional applications used by businesses and institutions worldwide, heuristic violations are few and far between, with no critical issues.

2 Design Requirements Specification

2.1 Context Identification

Group projects are a familiar routine to many university students, and some may experience several, sometimes simultaneous, group projects in their student life. These activities are often done through an online medium through a computer, tablet, phone, or in-person. Typical meeting locations include the school campus, libraries, or at home. Considering the agenda of a full-time student, scheduling group meetings may be difficult due to time or commitment conflicts between members. Moreover, effective cooperation requires networking, communication, time management, idea-sharing, and task assigning. The absence of any of these elements may evidently lead to suboptimal productivity. The intended use of the product is to provide a collaboration platform that incorporates common activities and tools that a group of users may use to enhance and streamline group performance.

2.2 User Identification

Daniel Lee [2] is a 21-year-old college student majoring in Interactive Arts and Technologies at SFU. He is in his 3rd year and consistently takes full course loads every semester, sometimes taking courses at different school campuses. Most of his free time during the day is spent on campus either studying or working on projects and assignments, often with classmates and friends. Despite his studious nature, he lives an unhealthy lifestyle due to lack of exercise and proper meals [3]. Daniel also stays up late at night to play video games and barely balances his work-hard-play-hard lifestyle. He is looking for a video-conferencing application that can host his multiple group projects, meetings, and efficiently communicate his ideas.

Jane Smith [4] is a 19-year-old college student majoring in Psychology at SFU. She is in her 1st year and takes a light course load every semester, as well as working at a part-time job. She enjoys socializing with friends and playing sports. She eats healthy, sleeps, and wakes up early. With COVID-19 on the rise, her preferred in-person collaboration with group mates for projects is no longer valid. She is looking for an application that can facilitate project collaboration virtually, easy to use as a beginner, and is able to compliment her tight schedule.

2.3 Functional Requirements

1. The system must allow users to video call, voice chat, text chat, and share common file types simultaneously.
 1. One, a combination, or all the above features should be compatible upon the user's discretion.
 2. Shared files should be accessible through the text chat interface.
2. The interface should provide the option to leave, mute, and toggle video during a call at any time.
3. The system must have application features to organize project materials, assign tasks, schedule meetings, and communicate ideas graphically.
4. The application must include a user profile page that contains basic user information such as username and a simple networking system to add contacts.

2.4 Non-Functional Requirements

1. Users can only participate in at most one live video or audio conference at any time, and each conference can support up to 50 simultaneous users.
2. Files shared have a 500 Mb size limit, removable, and do not expire.
3. Real-time video streaming quality should closely match the quality of the users' webcam, in terms of resolution and frames per second.
4. The mobile application must comply with Apple's App Store guidelines [5] and Google's Developer Content Policy [6].

3 The Next Step

The end goal of this project is a UI design for a video-conferencing program for university students to collaborate with each other on projects for multiple courses. After evaluating designs of similar programs, specifying user and application context, and establishing requirements, the next step described by the cyclical user-centered design process is to create a design solution, namely, a prototype for our UI. This is planned to be done after a 3-day group brainstorm, using a wire-framing tool provided by the facilitator of this project, while abiding our specifications. Next, the design will be evaluated again with user evaluation, conducted virtually with students of varying demographics. If the project is to pursue further development, the UCD will be repeated.

4 Appendix

Figure 1.1.1

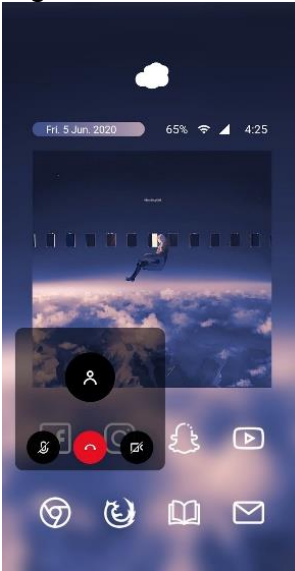


Figure 1.1.2

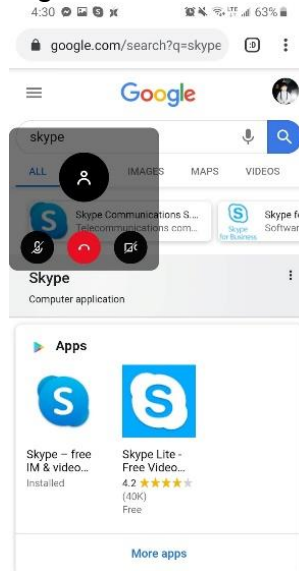


Figure 1.1.3

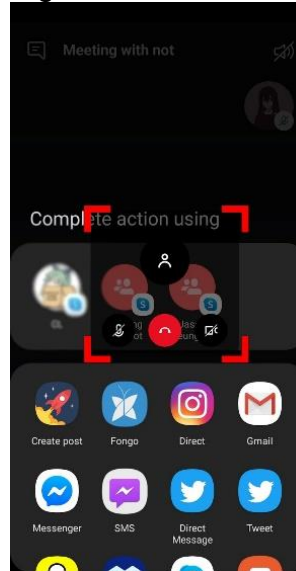


Figure 1.1.4



Figure 1.2.1

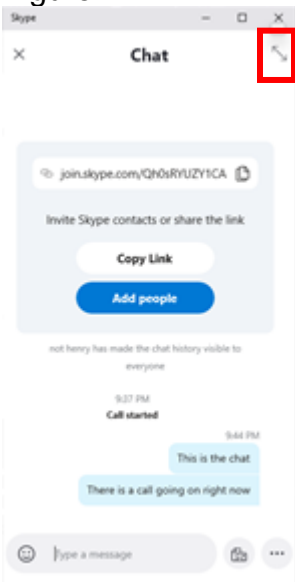


Figure 1.2.2

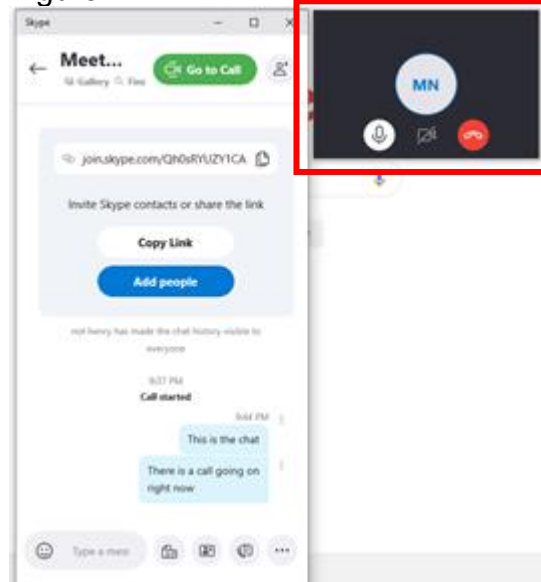


Figure 1.2.3

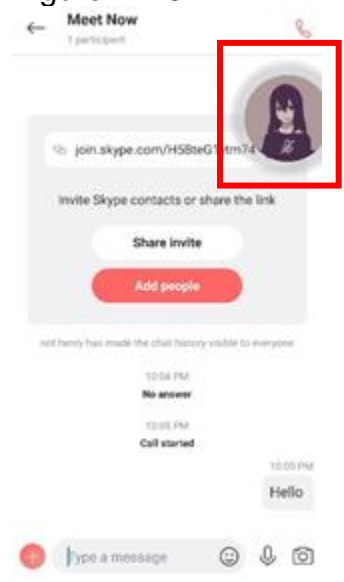


Figure 1.3.1

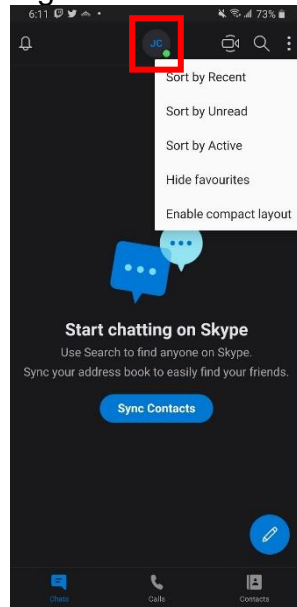


Figure 1.3.2

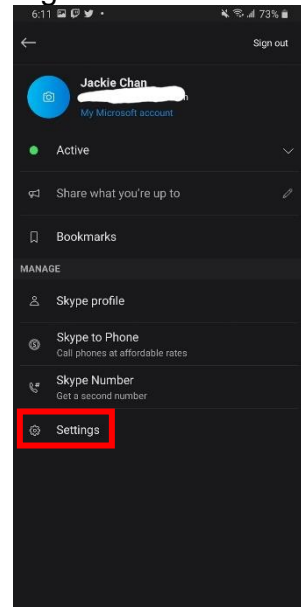


Figure 1.3.3

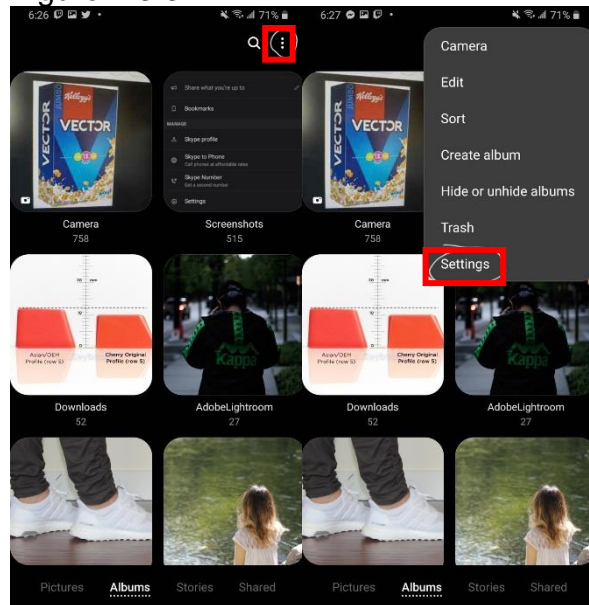


Figure 1.3.4

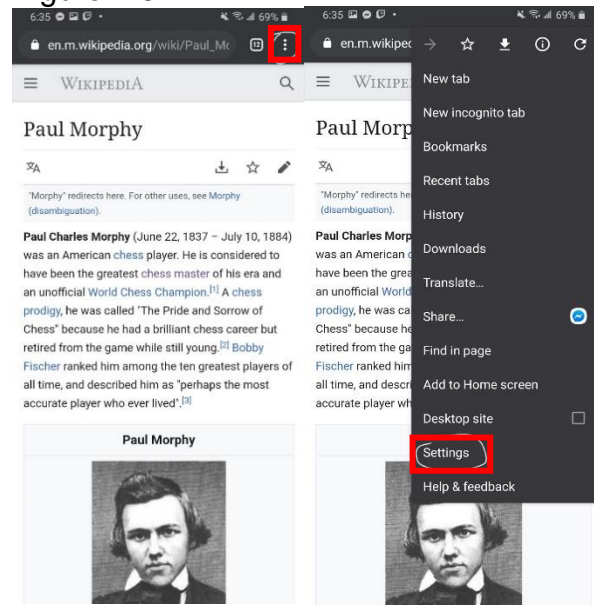


Figure 1.4.1

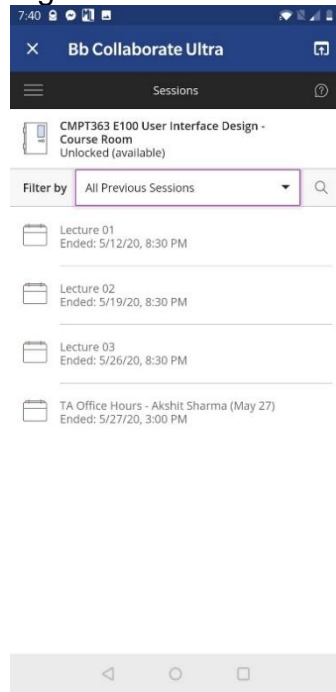


Figure 1.4.2

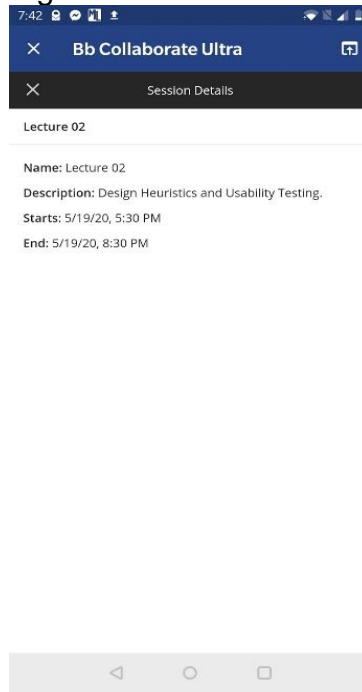


Figure 1.4.3

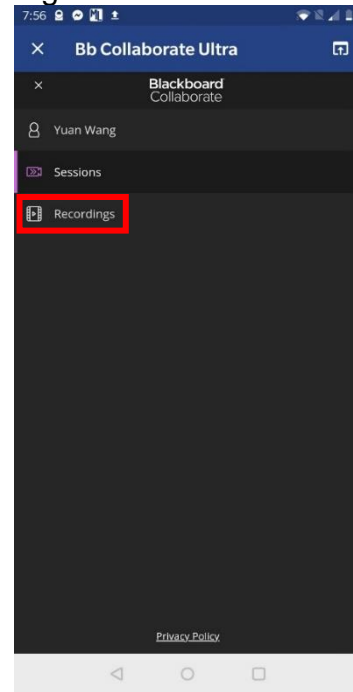


Figure 1.4.4

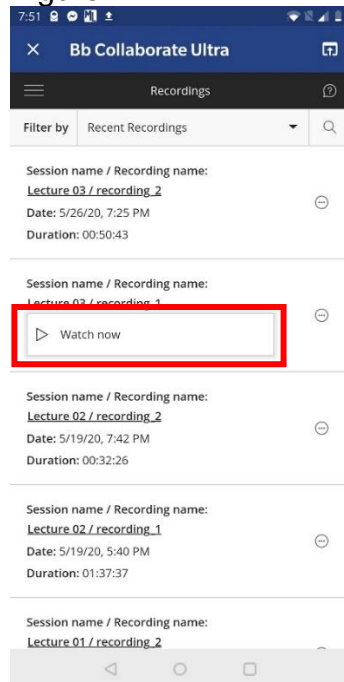


Figure 1.5.1

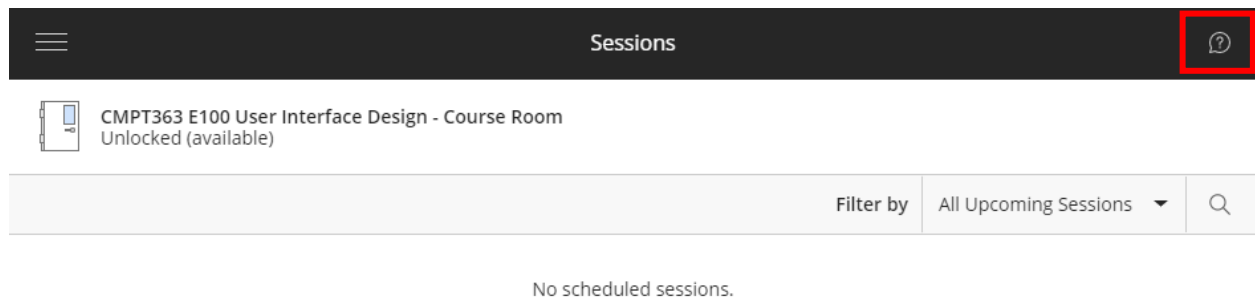


Figure 1.5.2

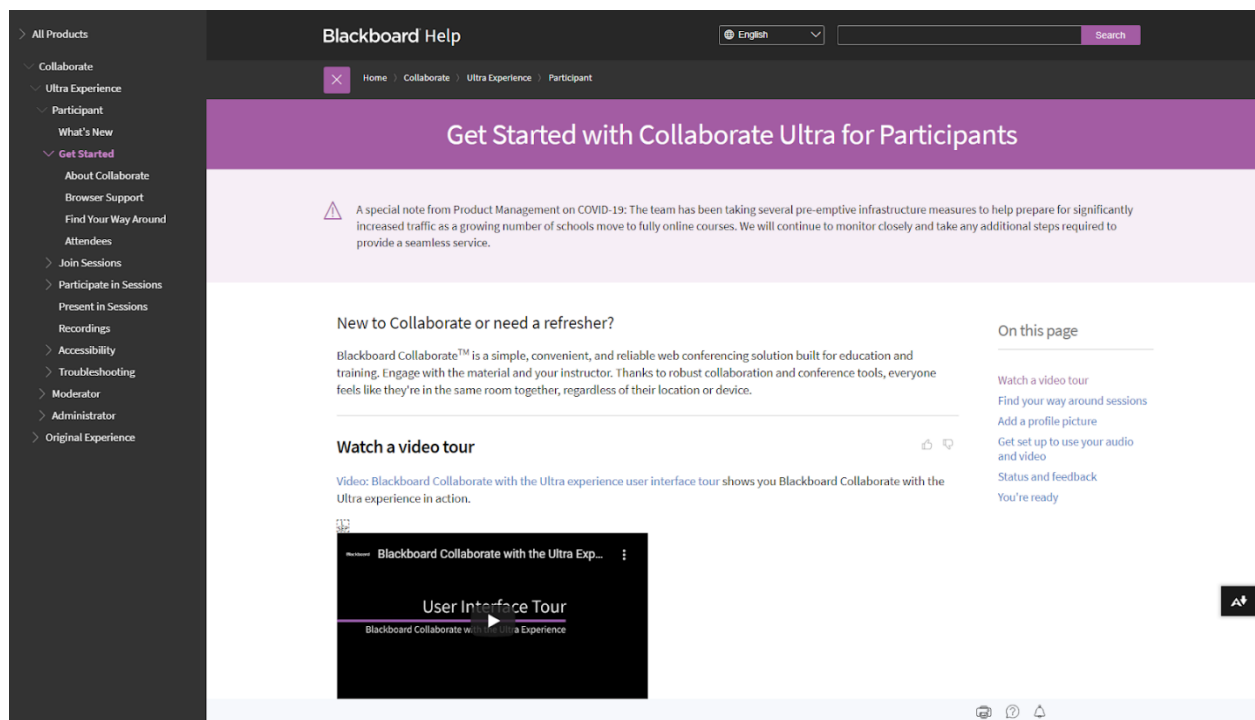
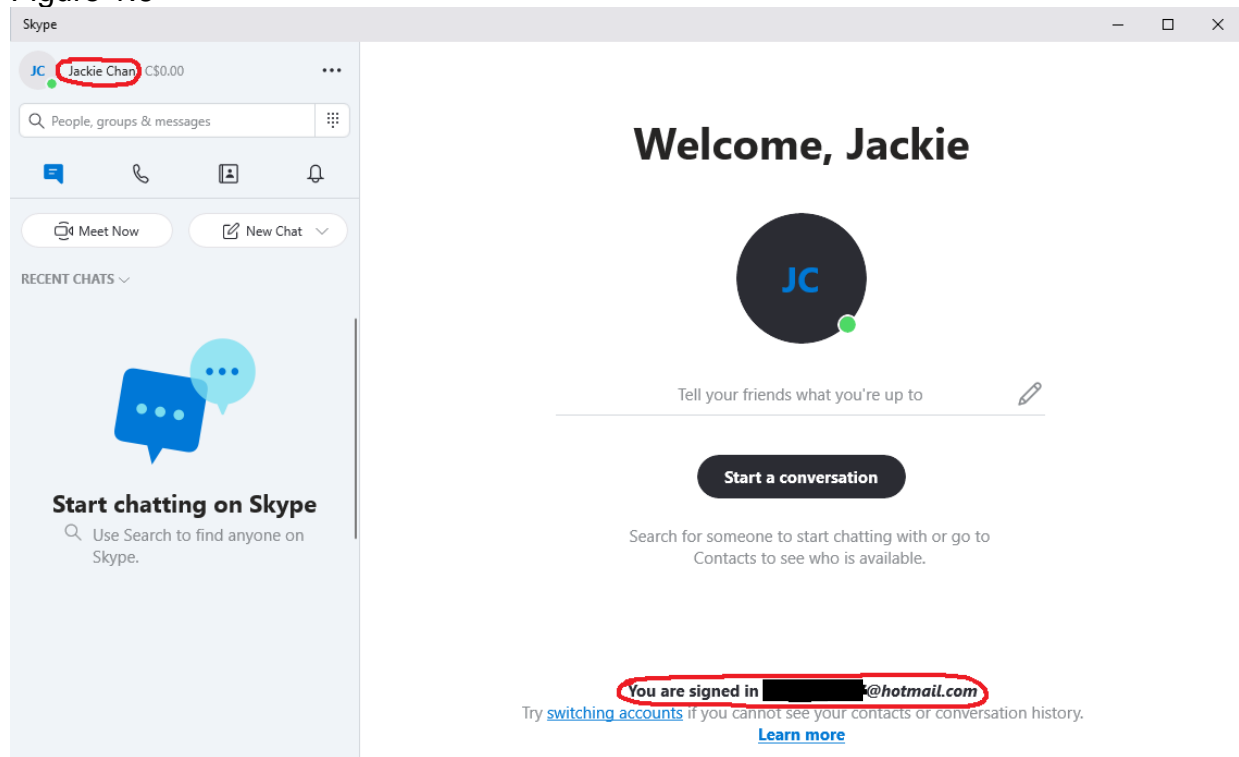


Figure 1.6



5 Citations

- [1] <https://www.nngroup.com/articles/ten-usability-heuristics/>
- [2] <https://www.shutterstock.com/image-photo/young-asian-man-headset-working-on-218228803>
- [3] <https://www.shutterstock.com/image-photo/fast-food-concept-greasy-fried-restaurant-138061871>
- [4] <https://www.shutterstock.com/image-photo/portrait-mixed-race-college-student-campus-62245576>
- [5] <https://developer.apple.com/app-store/review/guidelines/>
- [6] <https://play.google.com/about/developer-content-policy/>