SOFTENG 350 Assignment 3

Prototype Description

Scenario and User Needs

A group of human-computer interaction (HCI) undergraduate students are working together on a project that requires them to share files remotely for the duration of the project. The project involves coding using multiple different languages such as HTML, CSS, JavaScript, etc. which results in a variety of file types used for the project. The group members want a system that is capable of handling all these different file types. The project deadline is readily approaching with only a week left until the due date. The students have the following user needs:

Interaction need: "Need for running and compiling shared code files"

Compiling code online is generally not an optimal way to test code which means that the project must be run on the computers of the students. The students would need the files on their computer for them to run them; hence a download function is needed on the cloud service. Since the project is coding based, it would mean that most files are connected and therefore, would require most files to fully run and test the project code. This would mean that the download functionality should also allow the students to download multiple files at once, as downloading each file separately can be time consuming and inefficient. This could be done through an option to download a folder with another option to export the project to the users' local IDE for convenience. Additionally, the functionality of downloading each file separately could also be helpful when the student only needs to update one of their local files.

Emotional need: "Need for comfort about having clear tasks and goals"

With the time constraint of the project, the students are likely to be under stress during the project. This stress may partially be due to being overwhelmed by all the different tasks they would need to keep track of during the project. With the project files being on the cloud system, a task list would be useful to have alongside them as it would make it easier to keep track of progress as everything would be in the same place. The students would be able to manage their tasks and set manual deadlines which can help with the management of the project and potentially ease the stress. The students might also want to include some team tasks so they can all work towards a common goal and feel more connected when working on the project.

Ecological need: "Need for ensuring equal participation"

Group projects commonly involve some of the members not contributing their fair share of work and end up burdening the other members. This lack of participation can be a huge factor that contributes to the overall quality of a group project. To ensure that everyone in the group is participating, the students may want a function that allows them to delegate tasks and keep track of their progress. Additionally, a way for members to update each other on their progress through a chat feature would also be useful. This would ensure that everyone in the group is contributing and participating in the project as individual tasks will be clearly laid out.

Interaction need: "Need for updating shared files safely and efficiently"

The purpose of a file share is so that the different group members can collaborate on the same code files remotely which would, therefore, lead to multiple files being edited/updated after they have been uploaded to the cloud service. This would result in multiple versions of the same file on the cloud services, which can take up unnecessary space and pose some problems when downloading the project. The students would potentially want an option for merging the file that is being uploaded to an existing file on the cloud. This would make sure that all the files are up to date, and no space is being wasted on outdated files. Updating files on the project can potentially cause some of the code to break if an uncompilable file was uploaded. The students would find it helpful if the system warns them before they upload a file that can cause this issue.

System Functionality

As the user scenario is based on the requirements of sharing within a group setting, the webpage has been designed with this in mind. Therefore, the webpage includes a plethora of functions that support the student in achieving their needs. The various system functionalities of the website enhance the experience of working in a group. These range from enabling communication between group members to displaying an aesthetic view of the files in the project.

It is known that an HCI course consists of many group work assignments, and so it is imperative for the cloud data service to be able to create new projects to handle group activities. This is why one of the core system functionalities of the website is the ability to create new projects. A new project can be created with one or more collaborators as a way to instil collaboration across team members. Furthermore, a student may have several group projects happening concurrently. The cloud service accounts for this by listing out all projects the student is part of. The result of this feature allows the student to jump from project to project so that management and participation in multiple simultaneous group projects are supported.

As with any other cloud data services, this cloud service designed specifically for university students requires users to have an account. This account-based design enables users to know with certainty who the other team members are, which is essential for maintaining security in an online data environment. This level of security enables students to collaborate on projects safely knowing that their data is secure and inaccessible to unauthorised users. The website includes the function to add collaborators which is fundamental to any group project. This provides the option to share access to a project to others, growing the size of the team. Moreover, the cloud service presents the ability to restrict the permissions a group member has. This can serve as an implementation of a role system within the group, where for example, some members are reviewers while others are editors. To build on this, the student is able to merge and upload files to share with team members, who then can provide feedback on the work, fostering a positive group environment.

In group work, ensuring that group members deliver on their promises is crucial to the success of the team. One of the user needs is ensuring the participation of every member in the group, which is why this cloud service includes a To-Do list feature. The To-Do feature allows team members to coordinate and delegate tasks to various team members. The assignment of tasks ensures that no two group members are working on the same task, therefore eliminating duplicate work and so the maximum efficiency of the group can be met. Communication is vital to the success of the group. The development team of this cloud data service understands this and has implemented a comment log embedded in the website. The comment log enables the communication between team members and logs all merges and uploads of any file in the project. The latter achieves a basic form of version control, providing the ability for the team to roll back the project to a previous state. This is critical for any project when things go wrong and especially useful for code-based projects.

With projects being the primary use of the cloud data service, the ability to ensure that large projects with a vast number of files remain tidy is paramount. The design of the website features a folder hierarchy to account for large amounts of files. With a folder structure, the interface is tidied, and project files are displayed in an orderly and intuitive fashion. This ensures that all group members are able to locate files quickly and effectively. In addition to the file structure, it is the feature to sort files. Sorting files gives the user versatility in the way they want to view files. Pairing this with a last modified attribute associated with every single file paint a visual representation of group collaboration.

Visual Design Choices

Several visual design decisions were made for this website to enhance user experience and improve the user's workflow while in use. The Gestalt Principles of Perception were mainly used to determine the grouping and the layout of different objects within the website.

After accessing a certain project, the main objectives for the user on this homepage are to explore the various directories or to update and download several files within the project. The Surroundedness principle was utilised to emphasise the file directory system as the central figure within the homepage. The side navigation bar on the left and the comment bar on the right was placed to surround the centre page to achieve this effect.

One main focus of the design was to help users recognise the different features of the website upon first use, as all users of the website will be newly introduced users. Firstly, all clickable features were designed following the similarity principle, within a rectangular shape with a drop-down menu. After coming across a clickable feature to use, the users will be able to easily recognise the other features and thus progress through the website with ease. This is visible in the design of the navigation sidebar, the 'collaborators' menu, the 'sort by' feature and the 'merge and download' buttons. These features were also colour coded in dark-grey to show the similarity between them. Also, the pop-up boxes have been designed with consistent layout and colours, offering the same effect as the clickable features through the similarity principle.

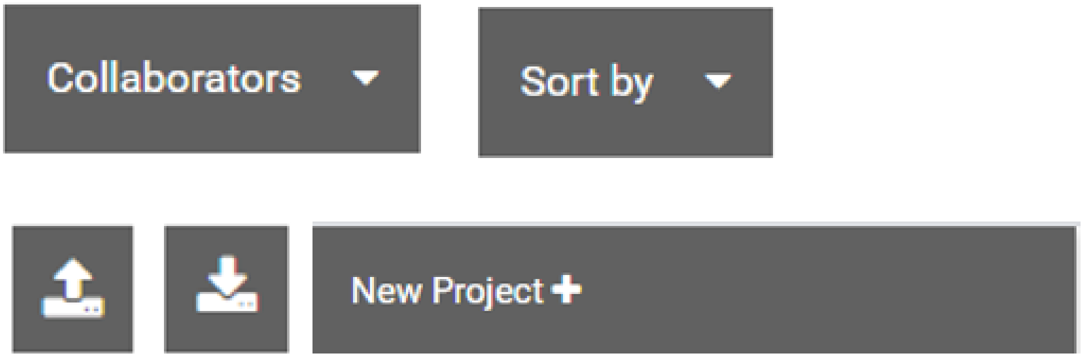


Figure 1: Similar Design of Clickable Features

It was important that the unique features of our website meet the specified user needs and were easily noticeable and locatable by all users. Therefore, a pragmatic approach to using icons was taken to enhance the recognizability (as opposed to recall) of the system. Icons shown in Figure 2, which would be recognisable to university students from their previous usage of similar systems, were utilised to describe the download and upload functions, and to let the users reach the logout feature.

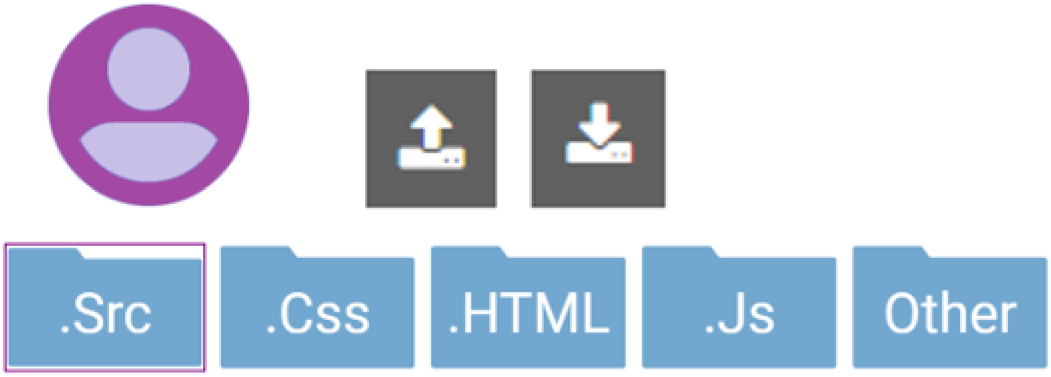
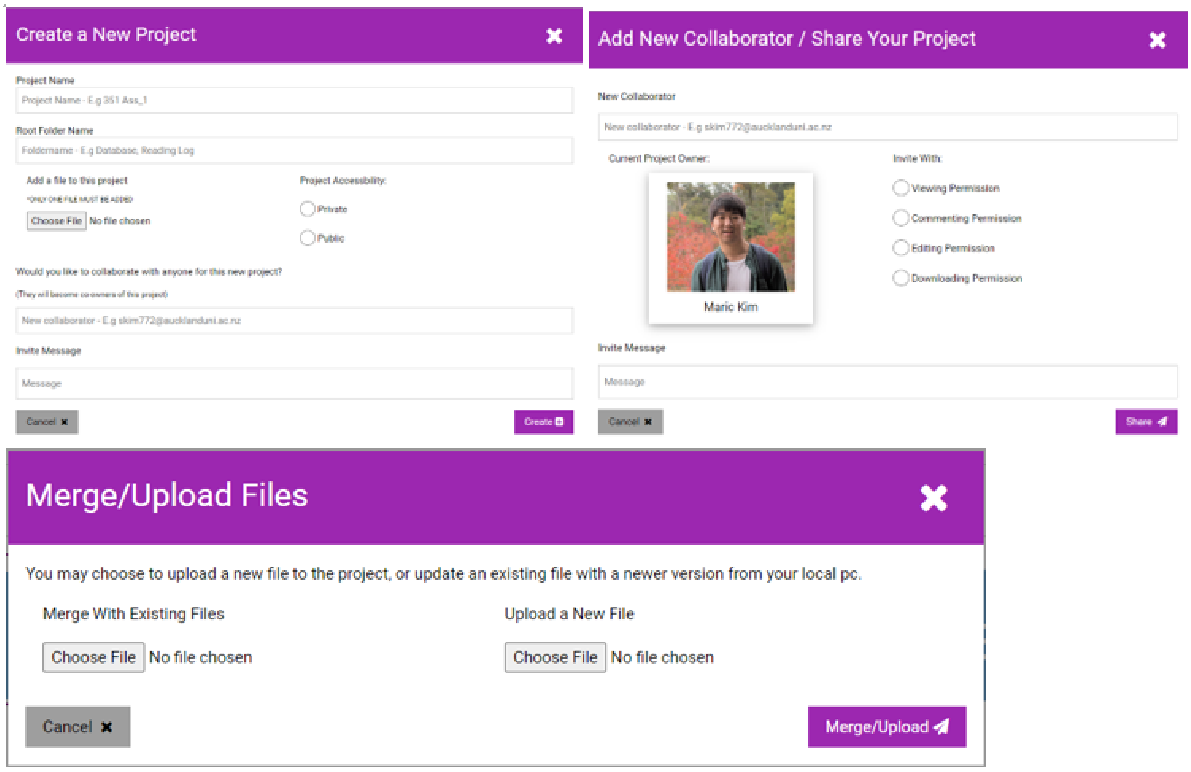


Figure 2: Familiar Icons

Whilst pop-ups are convenient for implementing new functionality into the system, it may visually confuse several users as it introduces a new interface to the screen. Change in the visual environment for first-time users normally results in an inefficient workflow as they readapt to system layout. To aid this issue, pop up pages have been centrally placed, and the titles have been designed with vivid contrasting colours for emphasis. This was done to reduce the time the user spends with their gaze unfocused, and to quickly lead their attention to the top of the pop-up, guiding their workflow direction downwards to fill out all the required input.



Create a New Project

Project Name

Project Name - E.g 351 Ass_1

Root Folder Name

Foldername - E.g Database, Reading Log

Add a file to this project

*ONLY ONE FILE MUST BE ADDED

Choose File No file chosen

Project Accessibility:

○ Private

○ Public

Would you like to collaborate with anyone for this new project?

(They will become co-owners of this project)

New collaborator - E.g skim772@aucklanduni.ac.nz

Invite Message

Message

Add New Collaborator / Share Your Project

New Collaborator

New collaborator - E.g skim772@aucklanduni.ac.nz

Current Project Owner:

Maric Kim

Invite With:

○ Viewing Permission

○ Commenting Permission

○ Editing Permission

○ Downloading Permission

Invite Message

Message

Merge/Upload Files

You may choose to upload a new file to the project, or update an existing file with a newer version from your local pc.

Merge With Existing Files

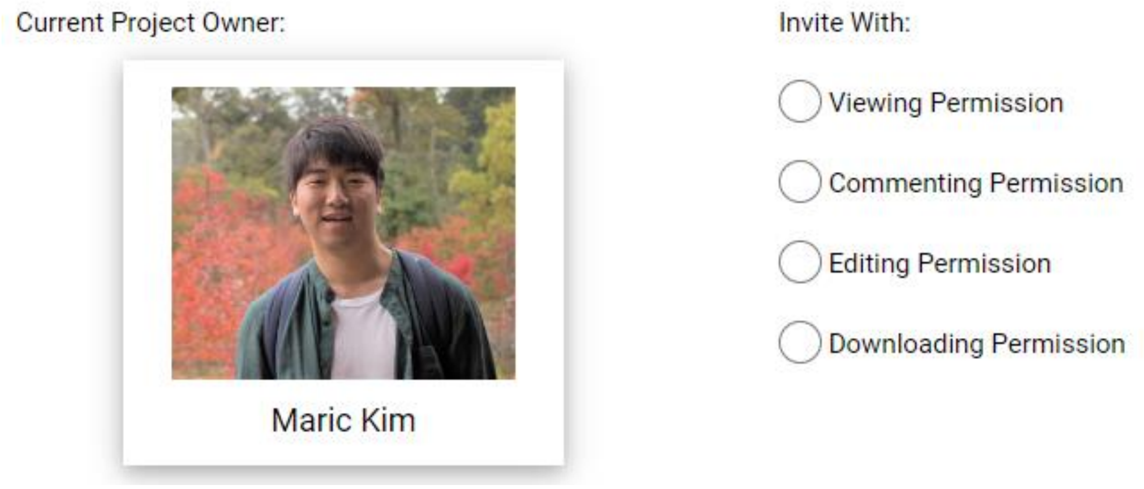
Choose File No file chosen

Upload a New File

Choose File No file chosen

Figure 3: Matching Designs of Pop-up Pages

For the sharing function, it was crucial that none of the required input sections were missed out by the user. So equal weight was applied to each screen element in the layout, other than the title, to achieve balance between them.



Current Project Owner:

Maric Kim

Invite With:

○ Viewing Permission

○ Commenting Permission

○ Editing Permission

○ Downloading Permission

Figure 4: Equal-weighted Screen Elements

User interface operation screen recording.

https://youtu.be/X2V3kFSm1Ks

Protocol Description

When designing the test protocol, we wanted to ensure that the entire scope of the system was tested and furthermore, that we test whether each user requirement is achievable by the user. Thus, the justification for each prompt will be with regards to one of these goals.

The first prompt, to add a collaborator and specify their permissions, was chosen because it tests one of the critical features of the system - to be able to share content with other people. If it were the case that the process of sharing a project in our system was unintuitive, then this would be a significant user experience issue on the website and thus testing it is crucial. Moreover, the ability to add a collaborator relates to the user requirement of ensuring participation, and as we aim to test all aspects of the user requirements, it was essential to test this functionality.

The second prompt, to download a specific file, was chosen for two reasons. Firstly, it is an essential piece of functionality in the system and thus whether or not it is easily understandable by the user is essential also. It was important for the test to include choosing a specific file as this was a potentially confusing aspect of the system. Secondly, one of the user requirements was to be able to run/compile code and our implementation of this requirement was enabling content to be downloaded from the shared cloud project onto the users' local drive.

It was a user requirement of the website that a user be able to ensure the participation of collaborators on a given project. The implementation of this requirement manifested in a 'to-do list' and a 'comment log' to keep track of progress and individuals' contributions. As these features relate directly to the user requirements, and also constitute two key elements of the system, we decided it was necessary to test.

The idea of 'projects' is a central concept in our system, as we have structured the website such that each feature mentioned thus far exists within a given project. Because of this, if a user is unable to create a project, then they would be unable to use the website at all. Thus, it was important that we designed a prompt that tested the user experience of creating a new project.

Uploading a file into a specific folder was chosen as a prompt for two reasons. Firstly, It was a user requirement to be able to safely upload a file into the shared cloud. Secondly, ensuring a file is uploaded into the correct folder is a potentially confusing aspect of the system as it requires multiple clicks, so usage testing was crucial for this prompt.

We included a prompt of downloading a specific folder followed by exporting the entire project to ensure that users would not confuse the process of downloading a file with the process of downloading a folder/exporting the project. Similar to the rationale for prompt 2, being able to download content from the system was a user requirement, and thus all aspects of it were important to usage test.

As the process of logging out requires the user to interpret our use of iconography correctly (understanding that the logout button is located on the profile icon), we wanted to test whether this interpretation occurred intuitively for the user.

Usage Testing Results

Hypothesis, task instruction, informed consent form, raw data are all in the appendix.

Key Characteristics of participants:

Participant 1:

Participant 1 is a 20-year-old female student studying Software Engineering at the University of Auckland. She is comfortable with technology as she uses both a mobile device and laptop on a daily basis. However, her usage of cloud data services is limited to once a week.

Participant 2:

This participant is a 20-year-old female student at the University of Auckland. She is studying a Commerce and Bachelor of Arts conjoint degree. She has a significant amount of experience with standard cloud services, such as google drive, and believes herself to be just above average with regards to general tech-savviness. She uses both a computer and a smartphone every day.

Participant 3:

This participant is a 20-year-old male student at the University of Auckland. They are studying software engineering, so they are very proficient with technology use. They are also very familiar with cloud services like google drive as they have used it for multiple group projects before.

Analysis of Usage Test Observations

Protocol 1:

This protocol is intended to test how intuitive the process of adding a collaborator is on our website. We hypothesised that users might take some time to complete this task. This is because the 'add new collaborator' function is not immediately visible on the page and is instead nested in a drop-down box. The data collected from participants showed us that even though the function was not immediately apparent, people were able to figure out where it was quickly. This was because participants could see related buttons and assumed that the function was in or around that. The data did not support our hypothesis due to the ability of the human brain to perform reasoning and quickly deduce from related elements the location of the required function. As a result, the participants took a reasonable amount of time to complete this task.

Protocol 2:

This protocol was designed to test if users would have trouble with the process of downloading a file and whether this process was easy to learn. We hypothesised that users would not have trouble finding the correct files but might spend some time to locate the download file button. However, we expected that the second time would not have any issues. Right from the start, all participants made the connection that the .Js folder contains JavaScript files. However, some participants struggled with downloading files - some clicked the upload button without the tooltip then corrected themselves to click the download button. Others hovered over buttons they were not sure about to read the tooltip before moving on. The second part of this protocol determined if the process, once learnt, was easy to apply again. The participants showed behaviour aligning with our hypothesis, meaning the process of downloading a file can be learnt very quickly. Overall, the participants were able to complete this task in an acceptable amount of time.

Protocol 3:

The intention of this protocol was to test whether the checklist and the comment features were intuitive to use and to see whether any unexpected user behaviour would occur when attempting to use these features. We hypothesised that the users would not struggle with this protocol as both features are likely familiar to the user. As we saw in the usage testing, our hypothesis was largely correct as all users quickly found the relevant features and interacted with them accordingly. Users immediately understood what the task list was by noting the checkboxes next to the listed items - "I'm guessing this is the task list cause it's got checkboxes". A common behaviour among the users was to press the 'ENTER' key on the keyboard as opposed to the on-screen 'Send' button. From this information, it would be beneficial for our site to ensure that the 'ENTER' key is implemented as an alternative to the 'Send' button.

Protocol 4:

The goal of this protocol was to test whether the process of creating a new project was fluid and intuitive for users. We hypothesised that users would likely not struggle with this task as locating the 'New Project' button would not take long, and the rest of the process is guided by textual prompts. As expected, users had a little issue with the process, and in the case of participant 2, the speed of locating the 'new project' button appeared to indicate that its placement on the screen was expected and intuitive for the user.

Protocol 5:

This protocol tested the ability of the user to upload a file to a specific folder on the cloud service. We expected the user to take some time to navigate to the specified folder, but the observations show that the users could easily navigate through the different folders on the prototype. With the upload button being just an icon, we expected the users to spend some time correctly identifying it and this posed to be true for 1 of the participants as they had some confusion between the download button and the upload button. The tooltips helped in this situation as it assured the participant which was the correct button. None of the participants had any trouble with selecting a file to upload from the pop-up. Overall, all the participants were able to correctly complete the task without having much difficulty and in a reasonable amount of time.

Protocol 6:

This protocol had two parts, both relating to downloading the project. Part one involved the user downloading a specific folder from the project. We expected that some users might download the wrong folder or have difficulty finding the download button. Most of the participants had enough experience with the website at this point, so it was relatively easy for them to execute this task. One of the participants did get confused with the "download file" and "download folder" buttons saying

"I could accidentally press this (download file button) thinking I'm downloading the HTML folder". This was a point that we had not considered and will look to improve. The second part of this prompt was to export the project. We expected this to be a straightforward task for the users, and the observations show that it was, as all participants were very quick when completing this task.

Protocol 7:

This protocol was the final task for the user who was to log out of the cloud system. The logout button uses a profile icon as we expected the user to make use of previous experiences with the icon and understand its implication. We expected the participants to take a short time to find the log-out button, but from the observations, it seemed to take longer than expected. The common problem seemed to be that the participants were not "sure that the icon was a button" which caused the increase in time taken to complete this task. The participants had mistaken the button for just a picture which is likely due to its positioning or sizing. With this in mind, we aim to improve the appearance of the profile button.

Conclusion

Overall, the usage tests demonstrated that the majority of the SkyShare system is intuitive for users to use and navigate through. From the gathered data, we concluded that the most needed improvement is clarifying the different download buttons. We may be able to do this by reconsidering their placement on the screen. Alternatively, we may be able to merge the functionality of the download buttons such that one button is used instead of two. This would require implementing a pop-up or drop-down menu that allows the user to specify what is being downloaded. Another issue that occurred related to the perception that the profile icon was not a button. From participant 2, we saw that this might be due to the buttons size and position on the screen, so to improve this, we would have to trial different sizing and positionings.

Appendix:

Greeting

Hi there, some colleagues and I have developed a prototype software application that allows for file sharing via the cloud and we would like you to partake in a usage test of the application. This is to help us further develop the application, so it better meets the needs of users. If you are willing, please sign this Informed Consent form. **shows consent form**

Informed Consent Form

Informed Consent for Participant of Development Project

SkyShare - **Version one**. Title of Project: **SOFTENG350 Assignment 3**

Project team member(s) directly involved: **Justin Teo, Maric Kim, Blake Hattingh, Gurpreet Singh.**

I.THE PURPOSE OF YOUR PARTICIPATION THIS PROJECT As part of the **SOFTENG350 Assignment 3** project, you are invited to participate in evaluating and improving various designs of **SkyShare**, **a cloud share application for university student projects.**

II.PROCEDURES You will be asked to perform a set of tasks using the **SkyShare application**. These tasks consist of **testing the functionality and usability of various aspects of the prototype.** Your role in these tests is to help us evaluate the designs. We are not evaluating you or your performance in any way. As you perform various tasks with the system, your actions and comments will be noted, and you will be asked to describe verbally your learning process. You may be asked questions during and after the evaluation in order to clarify our understanding of your evaluation. You may also be asked to fill out a questionnaire relating to your usage of the system.
The evaluation session will last no more than four hours, with the typical session being about two hours. The tasks are not very tiring, but you are welcome to take rest breaks as needed. If you prefer, the session may be divided into two shorter sessions.

III. RISKS There are no known risks to the participants of this study.

IV. BENEFITS OF THIS PROJECT Your participation in this project will provide information that may be used to improve our designs for **SkyShare**. No guarantee of further benefits has been made to encourage you to participate (Change this if a benefit such as a payment or a gift is offered). You are requested to refrain from discussing the evaluation with other people who might be in the candidate pool from which other participants might be drawn.

V. EXTENT OF ANONYMITY AND CONFIDENTIALITY The results of this study will be kept strictly confidential. Your written consent is required for the researchers to release any data identified with you as an individual to anyone other than personnel working on the project. The information you provide will have your name removed and only a subject number will identify you during analyses and any written reports of the research. The session may be recorded. If it is recorded, the recordings will be stored securely, viewed only by the project team members and erased after three months. If the project team members wish to use a portion of your recording for any other purpose, they will get your written permission before using it. Your signature on this form does not give them permission to show your recording to anyone else.

VI. COMPENSATION Your participation is voluntary and unpaid.

VII. FREEDOM TO WITHDRAW You are free to withdraw from this study at any time for any reason.

VIII. APPROVAL OF RESEARCH This research has been approved, as required, by the Institutional Review Board for projects involving human subjects at **SkyShare**.

IX. PARTICIPANT RESPONSIBILITIES AND PERMISSION I voluntarily agree to participate in this study, and I know of no reason I cannot participate. I have read and understand the informed consent and conditions of this project. I have had all my questions answered. I hereby acknowledge the above and give my voluntary consent for participation in this project. If I participate, I may withdraw at any time without penalty. I agree to abide by the rules of this project.

Usage tests - Raw Data

Protocol 1.

Hypothesis: The add function is not immediately visible. The user may click a few drop downs before reaching the 'add collaborator' button.

Task: Add a team member to the project with the email test1@HCI.com with editing permissions

Prompt: "Imagine you are the leader of a project and would like to invite your teammate to the project using the email test1@HCI.com. Because they will be contributing to the project you want to give them editing permissions"

P1: Quickly scanned the whole page and from left to right and found the collaborator drop-down. Clicked it and saw Add new collaborator. "I'm assuming this is where I add a person". Inputs the example email in the correct place. Clicks the correct check box for editing permissions. Clicks share. "Oh cool there's a prompt"
P2: Immediately clicked collaborators drown menu - "it's probably here". Then filled in all details easily and clicked share.
P3: Started off by clicking on navigation tab on the left- "just familiarising myself with the website", discovers the collaborators button while browsing and clicks add collaborators "oh i found it",fills details easily and clicks share.

Protocol 2.

Hypothesis: User will quickly navigate to the .Js folder and click on app.Js but will spend some time to find the download file button. Similar to before, user will navigate to the .HTML folder and click on login.html but will now quickly identify the download button since they have done it before.

Task: Download the javascript file called app3.js and html file login.html

Prompt: "Now let's say you need to work on a javascript file called app3.js. This means you would need to download it first, how would you do that?" [wait till they have succeeded] Now you want to also work on login.html, can you do that?"

P1: Scans the folder directory. "From the file folder thing I'm assuming it's this one" Clicks on .Js folder. Scans the files in there. "Then i'll select this one" Selects app3.Js. Mouse over upload button and read tooltip. "So this one is upload then this one is download". Clicks on download button. Closes pop-up. "Using the same procedure I guess I can". Goes to .HTML and selects login.html and

clicks download. Second time downloading file took fractions of the time compared to the first.
P2: "well you said javascript so it's probably in here". User clicked the .js folder very quickly. User then finds the correct file in the folder.
User then clicks the 'download folder' button - perhaps didn't read the button name.
After realising the mistake user skims mouse over the download and upload buttons and finds the 'download file button'.
Was then able to download HTML file instantly without error.
P3: Quickly clicked through the different folders from left to right and stops .js "There's the javascript files"

Didn't read the folder names.

Reads through the different files and selects on app3.js

Accidently clicks on merge/upload button without reading the tooltip "whoops this is to upload not download"

Closes the pop-up, clicks on the download button not reading tooltips again "there it's downloaded"

For login.html file- user more familiar with the website, instantly goes to the html folder and selects the file and clicks download without any mistakes.

Protocol 3.

Hypothesis: User will immediately recognise the task list panel on the left and find the item and press the check box to mark it as done. Then the user will pan to the right and type the comment and click "Send". User will not have an issue with this prompt.

Task: check "comment the code" off the task list, and message the group saying "I have finished commenting code"

Prompt: "Assume that you are a member of the team and have just completed the task of commenting the code. You want to check this item off the task list and message your team members saying "I have finished commenting the code" How would you do that?" [task finished once they click 'send']

P1: Navigates to the left bar "There seems to be a task list on the side". Clicks another task to figure out the checkbox system. "Ok it doesn't immediately vanish when I click". Then clicks the task to check it. Goes to the right side and clicks text box and starts typing and press enter "oh ok enter button not working". Clicks send button.
P2:" I assume this is the task list" - user quickly navigates to the task after a quick look around the screen, skims through the various tasks and and checks off the correct one. Understands immediately where and how to enter the message and does so quickly.
P3: mouse goes straight to the task list "Im guessing this is the tasklist cause its got checkboxes" Ticks the task. Assumed that clicking on the whole button would tick it. Manages to find comment box instantly, Types in message and presses enter. Enter button not implemented. Clicks send button after realising that.

Protocol 4.

Hypothesis: User will take a short time to find the new project button at the upper left hand corner. Users will then easily fill in the required details.

Task: create a new private project, called myProject that has a root folder called root1, with a collaborator with the email test1@test.com and upload a file.

Prompt: "Now imagine you want to create a new project, called 'my project'. You want to add the colaborator 'test1@test.com", you want the project to be private and you want the root folder to be root1. How would you do that?

P1: Scans the website and finds New Project+. Clicks it. "Ok a pop-up box" Fills the details accordingly. "Choose a file Oh god" The opening of the windows file explorer scared the participant. Clicks private accessibility. Clicks create. "This seems pretty straight forward"
P2: "probably go here", user immediately finds the 'new project' button - seems as though they intuitively knew where it would be.
Quickly fills out the information and adds the correct privacy setting. Clicks create button.

P3: briefly looks around the website. "There it is"- finds new project button
Clicks button and fills out details on pop up. Clicks on "Private" text rather than tick box "Oh you cant click on the text to tick", clicks tick box and clicks create.button.

Protocol 5.

Hypothesis: User may try to upload with a wrong folder selected. User might take a small time to find the upload button.

Task: Upload a new file into the src folder

Prompt: "Now you want to add a new file into the Src folder of the HCI project (it doesn't matter what file). How would you go about doing that?"

P1: "I would upload something" clicks upload button straightaway because participant hovered over it before and thus knew where it is. Opens the prompt "Whats merge? I don't know what that is" "Upload file choose file". Chooses file from disk and clicks upload.
P2:" I'm assuming thats the src folder" - clicks source folder. "Then probably go here" - quickly finds the upload button - seems that user remembered where it was from skimming over buttons during task 2. Reads the two ways to upload files (merge and upload) and immediately knows which is the correct one to click. Chooses a file and presses upload.
P3: User seems very familiar with interface at this point, Clicks on src folder
"Which one was upload again?" - hovers over upload button and reads tooltip "yea knew it was this one" - clicks it, skim reads the pop-up and uploads file.

Protocol 6.

Hypothesis: User might try to download with the wrong folder selected. User might take a small time to find the download button. User won't have trouble in finding the export button.

Task: Download HTML folder, then export entire HCI project

Prompt: "let's say you want to have a separate version of the HTML folder on your own computer, how would you download this folder?" [wait until the folder is downloaded] "Now let's say you are on a different computer that does not have the project so you want to export the entire project on your computer, how could you do that? "

P1: Clicks HTML folder. Navigates download file button. Then notices the Download Folder button "wait why is there two different download button? Whats the difference?" "I could accidentally press this thinking im downloading the HTML folder" *gesturing at the download file button* clicks the download folder button and downloads folder. "OK exporting where is that … right here next to the project name" Quickly finds the export button and clicks it to export.

P2: Has learnt how to open a folder and so quickly clicks the correct folder. Then remembers where the download folder button was from the mistake in task 2 and clicks it. "So just go download folder".

User briefly scans over the webpage with the mouse until the 'export project' button was found. Then clicks it.

P3: goes to the HTML folder , clicks download file "that was easy",

Quickly locates the export button, "Oh the export button is on the top" - maybe didn't expect it to be there. Clicks export.

Protocol 7.

Hypothesis: User may not find the logout option immediately as it is within an icon. Eventually user will click the profile icon and see the option to logout, then click the logout button drop-down

Task: User is to log out from the system

Prompt: "You have finished what you needed to do on the project, all that is left to do is log out of the website, how would you do that?"

P1: Scans around the screen looking for the word "log out" but does not find it. Proceeds to open every drop-down box then eventually finds that the profile icon is clickable, "Oh see the mouse changes means I can click this" clicks and finds the logout option. Clicks it to log out.

P2: "okaaay" - user scans the website with the mouse and after approximately 5 seconds clicks the profile icon - it was the first button the user clicked. User then clicks log out. User remarked that "I wasn't sure that the icon was a button - it wasn't totally clear". - perhaps the icon's size is larger than most profile buttons and this caused confusion.

P3: "oh am i logged in" - might not be obvious that user is logged in (off task), navigates straight to the profile icon and clicks it. Clicks log out from the drop down menu "There it is".