



EVALUATION ASSIGNMENT 6

CS5760 – Graduate Human-Computer Interaction

Abstract

A final report summarizing the usability testing procedure and resulting data, as well as any conclusions, findings, or insights for future development related to the application based upon analysis of the test data.

Briana Bettin – Team 2

April 21st, 2017

Introduction

The drinking water inspection app is designed for use by Wisconsin DNR representatives who survey drinking water facilities. These inspections require a form documenting any concerns or violations of standard to be noted, and must be submitted for review. The application takes the paper form and places it in a digital format via a web app, which the representatives can then complete and download or print to be submitted.

The undergraduate team's final documentation, work, and code for this application can be found at this web link: <http://www.csl.mtu.edu/classes/cs4760/www/projects/s17/group2/www/>

The goal of testing was to ensure that the application could perform the functionality necessary for the DNR representatives in the field, be a proper substitute for their paper form, as well as to test the ways the application extended from the original paper form with flexibility of useable device as well as additional assistive design to aid in the correctness of filled out forms.

To this end, the tests were conducted as a roleplay of testers walking through a DNR inspection, in which they were told various observations they would have made about the fictional plant in order to see how the adequacy to which the application form could be utilized to denote these observations. Participant actions were observed after each scenario was explained to them and their action path and subsequent commentary or intermittent actions aided in informing justification for the utility of the application as well as recommendations for future application alterations.

Test Plans

For the testing, a singular protocol was devised and utilized among all participants. This protocol was devised such that all major potential functionality scenarios could be observed with each participant, to gauge overall success as well as success in individual tasks. As these forms may not be filled out in a linear fashion, among half of the participants two pairs of scenarios that occurred traditionally one after the other were "flipped", so that half of the participants had a nonlinear order of events to measure if the expectation of linearity deterred their capability to utilize the app to complete the scenarios.

Each scenario was specifically designed with an expected "goal" of functionality the user might achieve or fail based on completion of the test. This task goal was not explained to participants as it may give away the underlying intent of the scenario and tell them what to do, as opposed to the participants exploring and drawing their own conclusions. Thus, participants were told only the scenario, but each scenario was designed specifically with the task preceding it in mind, as well as with the overall narrative goal of completion of a form in its entirety.

"Elements" in the tasks denote the sections of the form, as the form is broken into specific categories based on observational areas and equipment. As stated, these may not be linearly noted depending on the layout of the plant or movement of the inspector. Each element of the form has a similar layout in

terms of possible functionality, thus choosing specific elements to test specific tasks is not indicative of a unique trait of that element, but rather a means to test various permutations of necessary information selection in the elements with a single form completion.

The tasks and their accompanying narrative scenarios are as follows. Tasks 3 and 4 were swapped in order for four of the six participants, and tasks 5 and 6 were additionally swapped for one participant, in order to compare linear completion results to nonlinear completion:

- 1. Task:** Fill in general information
Scenario: You are a water system inspector on your annual call to a plant in Waukesha county. The DNR plant's ID number is 1 and it is a Water Treatment system. You decide to fill in these crucial details on the form before beginning the inspection, so you don't forget later.
- 2. Task:** No Issues in Element #1
Scenario: You begin your inspection of source issues, checking and double checking all the casings, ventilation, grout, and so on in plant. You determine nothing appears to be out of order, and decide to note your findings before moving on to the next inspection section.
- 3. Task:** Find an Issue, Element #2
Scenario: You move to inspect the pumps, and notice some leakage occurring around the pipelines. Realizing this could be a problem, you make sure to take note of it on your evaluation form.
- 4. Task:** Fill in another with text, Element #3
Scenario: As you move on to checking the storage tanks, you notice an array of electrical wires near the tank, precariously close to an area where they may come into contact with water. This setup seems a major issue for the safe design of the storage area, and you want to ensure it is brought to someone's attention.
- 5. Task:** Choose N/A, Element #4
Scenario: You move to check the treatment system, but realize this particular plant feeds into a larger plant with the filtration system housed there. As such, there is nothing to check here.
- 6. Task:** Choose multiple issues, Element #5
Scenario: You begin your final checks on the cross connections for the distribution system, and notice some plumbing lines that go nowhere. You also notice the distribution piping appears to be leaking.
- 7. Task:** Add a corrective action(s)
Scenario: As you are finishing your evaluation of the plant, you recall several of the issues you noticed, including leaking pipes, unsafe electric wiring, and lines leading to nowhere. You want to make a note of steps this facility must take in order to be up to code, so that your suggestions appear in their evaluation and progress can be tracked for next year.
- 8. Task:** Finalize Document
Scenario: You place the last finishing touches and your evaluation, and feel as though it is ready for completion. You decide to move on to the next step and take a look at your completed form.
- 9. Task:** (If errors) Correct Errors
Scenario: You are alerted that something seems amiss in your completion of the form. You

decide to investigate and attempt to correct whatever has gone wrong so that you can finalize your inspection form.

10. Task: Review & Print Form (PDF)

Scenario: You arrive at the preview of your completed form before it is printed. You decide to look it over one last time before you call the work done. After your perusal, you realize you left your portable printer back at the office! You decide to print the document to a PDF for now, so that you can save the work you've done to print out later.

Results

Of the six tests, three participants tested on a desktop computer with a browser window resized to simulate a mobile phone screen size. The other three participants tested on an Android phone, with a casting application transmitting the screen display to a computer for the testers to view actions as well as record in-application behavior via recording software.

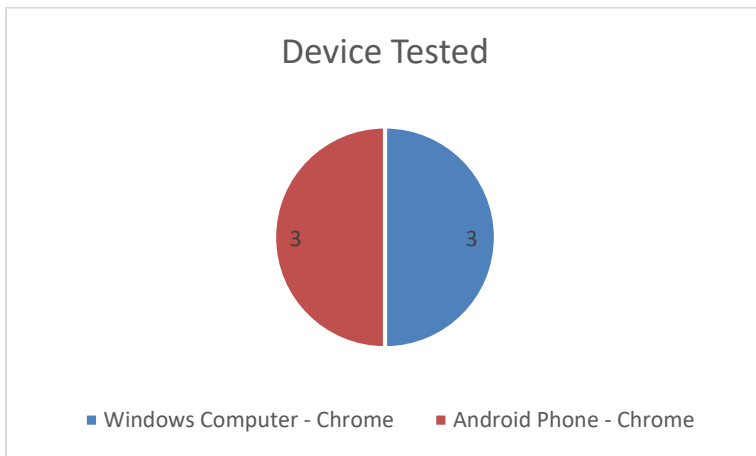


Figure 1: Devices utilized in testing sessions

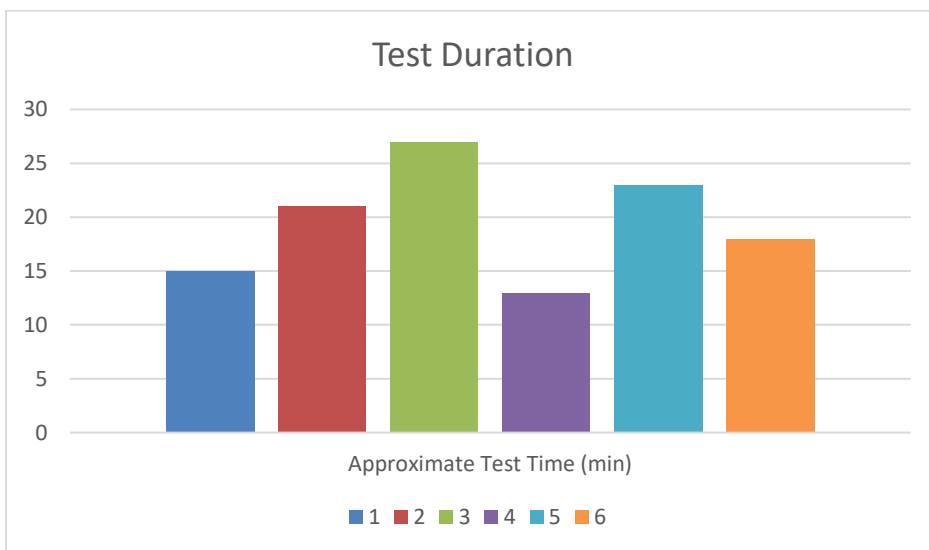


Figure 2: Duration of testing sessions

All six testers were male, without any awareness of color blindness, visibility impairments, or accessibility needs in their typical technology usage.

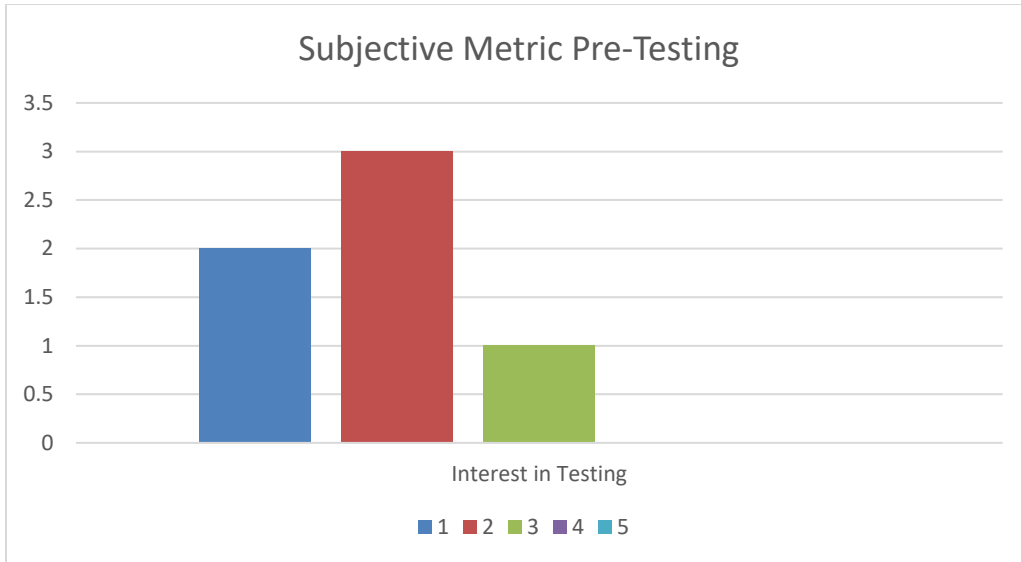


Figure 3: Participant responses to subjective pre test measures (note: one participant response of a 2.5 in “Interest in Testing” was rounded to a 2 for a whole number scale)

Only one tester had ever surveyed anything before, and stated they had inspected and surveyed computer equipment in the past.

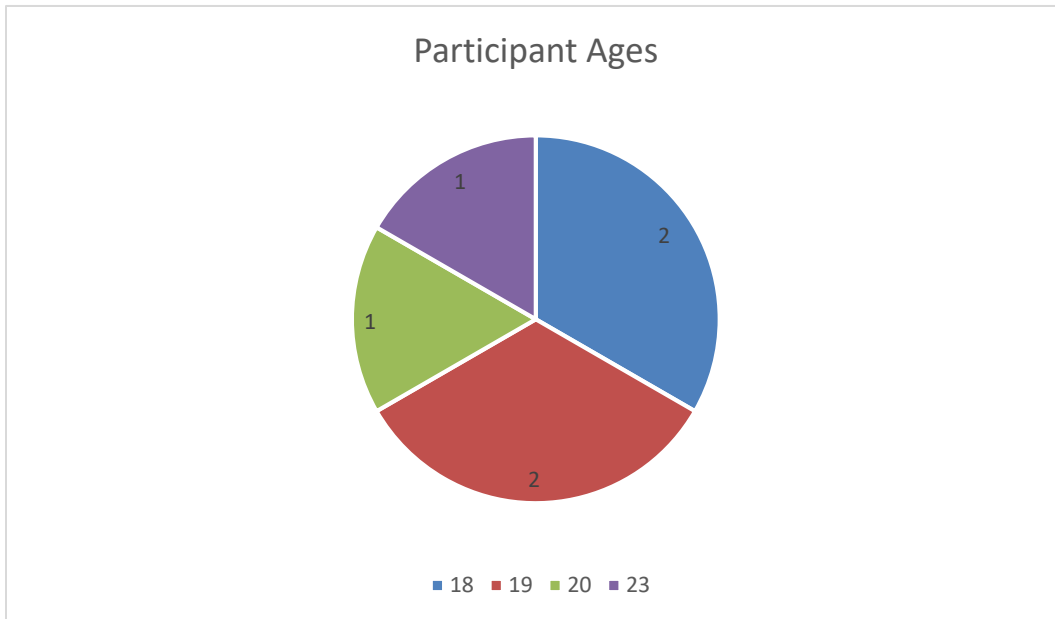


Figure 4: Breakdown of Participant Ages

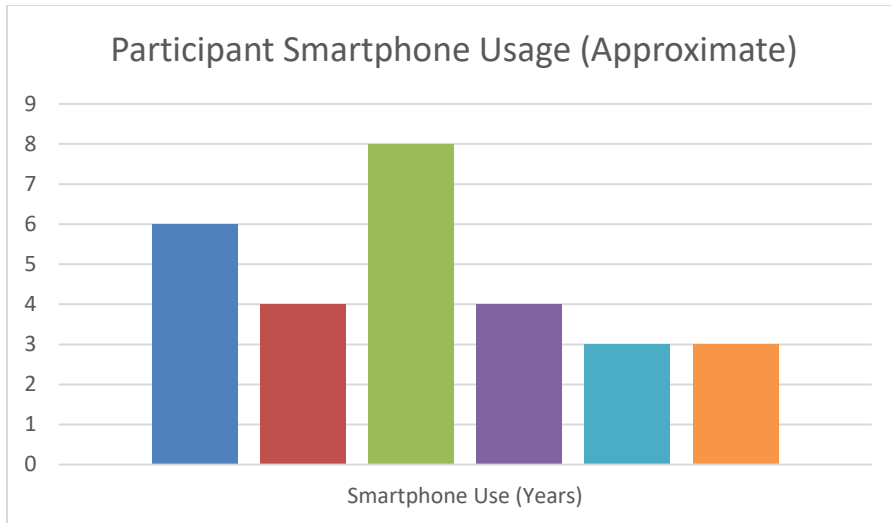


Figure 5: Smartphone Usage Time of Participants

All six participants were able to complete the task related to each scenario as desired within the application. Despite all participants having no training versus the DNR representatives who are the target audience, these users were still able to navigate the scenarios and fill out the form as expected based on the scenario given. Even if users had confusion about a scenario or did not initially complete a task correctly, they were able to course correct and revise as testing went on without any prompting or attention being drawn to their action by the tester.

To this point, of the four users who had task order swapping occur, three initially assumed linearity of the form and filled out or opened the wrong element first for the given scenario (i.e., Element 3's scenario was given before Element 2's, and thus they opened or filled out Element 2 instead). However, these users managed to course correct given the next scenario or within that scenario while reviewing the application text. This revised their assumption of linearity in the application, and they thusly changed their form to the desired task result without being informed of the mix-up.

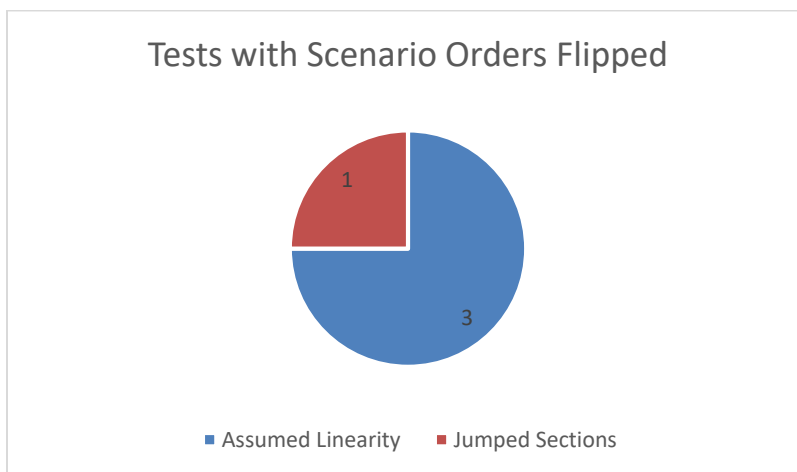


Figure 6: All tests with scenarios in a different than linear order, and participant initial action

This correction trait also carried over in the “Issue Present” radio bubble selection as well. Users consistently corrected their bubble choice to match their element accordion selections after the fact if they had forgotten or not noticed the bubbles prior. Often, this happened after the assumption of linearity was shattered – it appears that these users after that point became more aware of the text content within the application and thought more contextually about it once the order of the content became potentially fluid.

Three of the six users specifically pointed out the color coding in the application headers, commenting how it made sense or was a surprising feature they personally enjoyed as they completed the scenarios.

All participants had some apparent confusion or difficulty understanding corrective action, even if they could complete the general concept of the task. Four of the six participants created a single action and lumped all their issues into the one text box, one additional participant created two actions (three total issues were specified), and the tester who did create three complete actions had confusion at first regarding if the “Add” button was a button, stating it looked like a highlight and the color choice was odd in that location. While participants clearly understood the general theory of the task and thus completed the action desired (ability to create a corrective action and note something related to an issue they had specified prior), the difference of this section coupled with a lack of terminology and background by the participants truly showed through here.

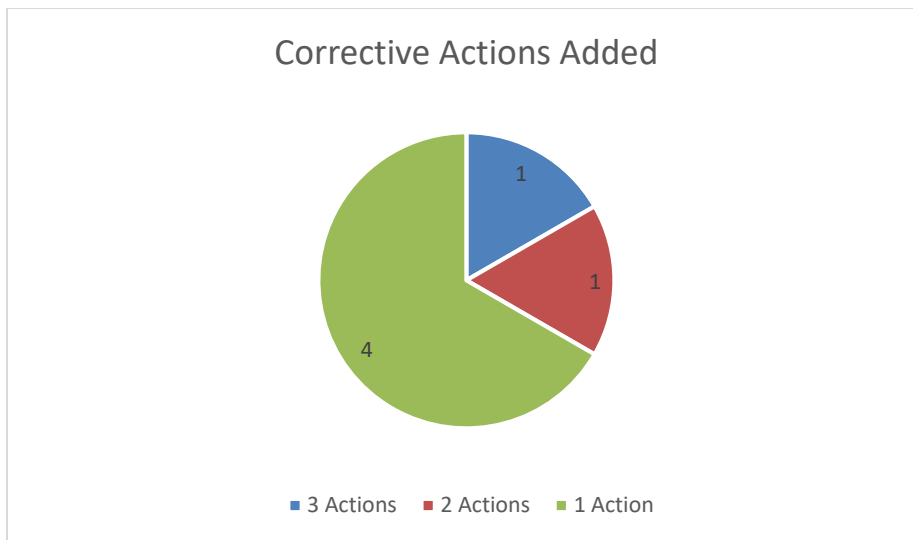


Figure 7: Corrective Actions added by Participants in this section

However, the corrective action was also one of the sections in which participants gave plenty of feedback regarding their thoughts and expectations from the section. These recommendations are further discussed in the conclusions, but it was a pleasing observation that despite all testers finding some difficulty with the task, it also yielded the most positive insight and all testers were still able to conceptually complete the task.

At least half of the participants very clearly recognized and toyed with the tab toggle and radio button clicks affecting tab toggle – whether they vocalized or not or continued to tinker until their mental

model understood what functionality was occurring, obvious actions indicating a level of curiosity or surprise at the initial functionality were displayed in at least half of the participants. Participants who were further curious about how and why the tabs were toggling clicked between the buttons more often, and were more likely to denote unexpected errors of accordion open/close or coloration of the header through this curiosity toward the functionality.

Of the three mobile tests, all three users had difficulty printing the preview to a PDF on the device. With the utilization of an Android device, the device wanted to place the file in Google Drive, but users wanted to place the file in Downloads – and even when they tried to allow placement to Google Drive, the Android interface would still behave awkwardly. This is not an issue of the app, but does point out a consideration that must be taken and a case that may need to be further tested and optimized for with mobile phone users, as printing from a phone or saving a PDF to a phone in this way may not be a common feature a user and result in this confusion. Desktop testers did not experience this issue, likely due to the fact that print preview and saving a PDF are more traditionally understood and completed functions in a computer based setting.

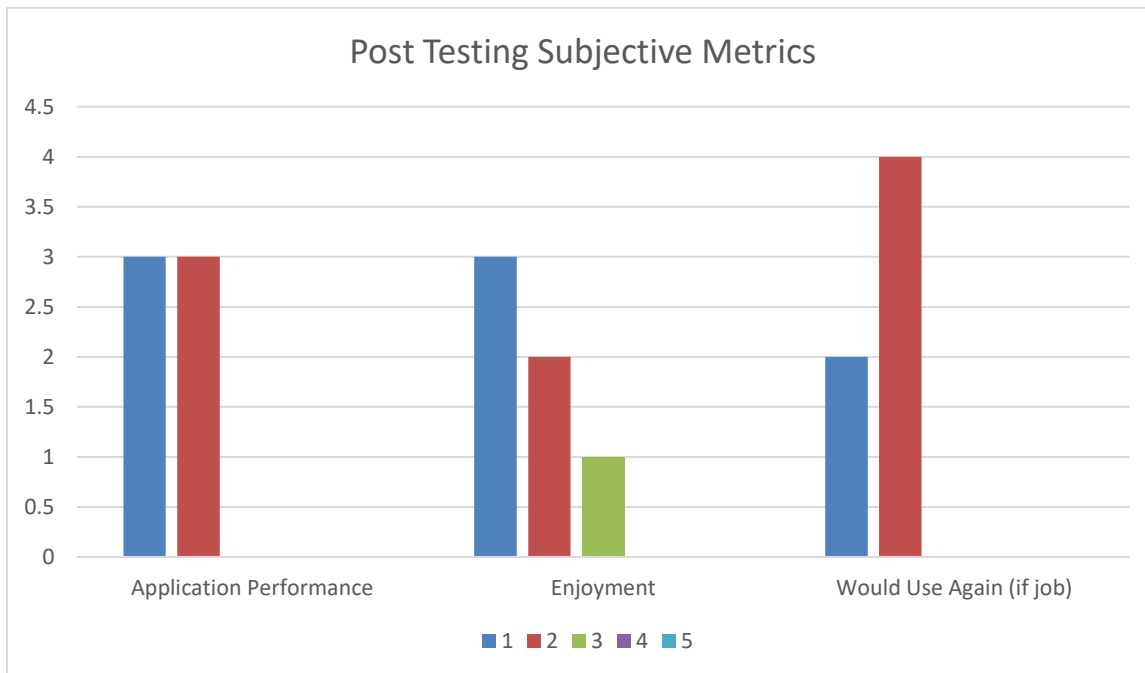


Figure 8: Subjective metric results for participants post-testing (note: one 1.5 stated by a participant in “Application Performance” was rounded to a 1 for the purposes of keeping a whole scale)

Conclusions

Overall, testing provided strong evidence that the app is intuitive given realistic scenarios a representative may encounter, and appears the function overall as anticipated with little participant confusion or errors being made. Testing also provided several insights as to recommendations for change to the design and functionality of the application in future iterations.

A notable feature for future versions was brought up via inquiry by the testing participants, which was what would happen should they exit the browser, refresh the page, or otherwise leave the form. At present, there is no system in place to save the representative's work or to return to a form that was started earlier – it is based on the device's caching with little control on the user's end. Some form of system for returning to begun forms, or saving the status of a form may prove useful in many scenarios where a user must close or restart their device but does not want to lose their work. This may take the form of a randomized share link that the user can copy down and access, a login system, or some form of cookies and caching beyond the default behavior, to name a few potential approaches.

Several participants also noted that the headers opened and closed what appeared to be randomly at first. A few participants toyed with the behavior a bit to recognize their radio button selections toggled the accordion open and closed, but their efforts to understand the behavior by toying with it, as well as their feedback verbally regarding it indicated this behavior was not what they had anticipated. A recommendation for standard or expected behavior would be to have the headers open with general header selection – that is, the body of the header itself or the arrow to the side of the header. On radio button clicks, if an issue is present is selected then the body should be kept open or expanded if not open, and on no or N/A it may collapse or stay collapsed. Toggling open closed without regard for the selection has confusing effects for users.

When participants preview the form, it is shown exactly how it would print to any standard printing device or to a PDF. One participant in particular pointed out that the text boxes were not expanding to the size of the text that was typed, and as a result information was being lost and unseen on the final form. In the printable version, these text fields should resize in some way (or potentially transform from field areas to labels in the final rendering) to ensure that all typed text is visible on the final form.

In the present design, when a participant selects "Yes, issue present", the header of the section turns red to indicate an issue exists in this section; likewise, if "No" is selected, the header turns green as everything is in acceptable condition. The "N/A" option leaves the header its normal blue (or a slightly darker shade). However, if "Yes" or "No" were selected first and then "N/A" is selected, the header stays the previous color (green or red). This is confusing and takes away the value of the color cue for the user. Ensuring that "N/A" repaints the color of the header to blue, just as "Yes" and "No" repaint to green and red, would resolve the confusion this selection sequence can create.

In the version of the application presented for testing, errors regarding unexpected user entries or application failures were not implemented. The listing of all counties in Wisconsin as a dropdown selection in general information was also not implemented, with only the first four counties being pulled. Implementing these features would be useful for ease of use and user understanding of any issues that may arise during use.

While the representatives surveying these sites are likely to be professionals in this field, forms can still be rather confusing, especially in a stressful situation. Several participants recommended some sort of terminology glossary for phrases that may be confusing or unintuitive, such as "PWSID". To not take

away from the general design of the application but still provide this information inline, tooltip “?” icons may be implemented that provide additional information when hovered or clicked.

Within the logic of the application, one participant noted it was odd that they were allowed to select “No issue present” but still check issues available within the accordion under the header. This may be mitigated when errors are implemented in the application, but some sort of logical condition stopping a user from selecting issues within an element if they have stated “No Issues Present” or “N/A” could ensure unnecessary errors are not made. Further, if a user selects “Yes”, they should not be allowed to submit the form without selecting an issue, as they have stated one must be present.

Finally, several users commented on not quite understanding the Corrective Action section. While this may be again due to the lack of experience the testers have relative to the representatives, they also provided interesting feedback regarding how this section could be better designed. Participants noted frustration with having to scroll back through the form to find all the issues they had selected, only to have to go back down and add a new action to type them in again. While the corrective action is meant to be action taken toward the issue, not a restatement of the issue, the sentiment pointed to a solid design revision. As users select issues that are present, corrective action should automatically generate an action “stub” related to this issue. This should include the description of the issue the user selected, along with the date and text field sections that presently exist. This would keep users from having to continually scroll the entire form to fill in issues, and instead focus on the issues within the section itself. The text box having some sort of label stating “Action Required Regarding to Resolve Issue” may clarify what is expected within the text field if there was any confusion as well.

Further, the remove button in the Corrective Action section seems superfluous, as it is not functional in the current application. In this model, no add or remove button should be needed, as the auto generation should auto add (or subsequently auto remove if an issue is unchecked). If add and remove are still desired however, add should give a list of potential issues and auto-check the issue within the necessary element of the form so users do not have to scroll to revise. Related, remove should remove the check marks from the affected areas, and remove should exist either as a button attached to each action, or as a button that when clicked allows a user to select via checkbox multiple issues and confirm for removal.

To conclude, the application performed quite well in testing and testers seemed to enjoy testing the application and felt its functionality was useful and functional given the scenarios. Testing yielded many interesting insights as to potential changes and fixes to be made in future iterations, and these changes if added should serve to further improve the usability and user satisfaction of the application.

Appendix

APPENDIX A: UNDERGRADUATE GROUP MEMBER ATTENDANCE

Four to five of the six undergraduate team members were present at each testing session. I did not take attendance but nearly the entire team attended each session (and had mentioned prior if conflicts existed for a session they could not make). The team was prompt and helpful if testing needed to be

done on cords or set up, and stuck around after testing to discuss preliminary thoughts and results. Their attendance and value within that attendance was not only satisfactory, but exemplary.

APPENDIX B: BUG REPORT

Bug #	Name	Location	Description	Severity
1	Header Toggle	Form Header	The accordion expands and collapses whenever a radio button is clicked – if users clicked on the header prior to open it, then click a radio button for if the issue is present or not, it will auto close.	2
2	County Listing	General Info Section	Only the first four counties in alphabetical order are listed (other counties can be typed manually)	3
3	PWSID Character Limit	General Info Section	The PWSID# is known to be 8 characters in specifications, but is not being properly bounded	3
4	Print Text Fields	Print Preview	On the print screen, text fields do not expand to size of input during printout, resulting in words outside the box bounds being cut off	1
5	Error Presence	Preview Submission Click	Errors are not currently implemented, so could not be tested	1
6	About:Blank on Preview	Print Preview	When Print Preview is selected, page spawns a new “about:blank” page that is a replication of the previous page. Can easily confuse users who do not realize this is not their original browser window.	4
7	Dummy "Remove" Corrective Action	Corrective Action Section	Remove Corrective Action button does not do anything	2
8	Header Color N/A	Form Header	Color does not change back if user selects Yes or No to turn the header green or red, and then selects N/A, which is usually dark blue.	2