CS5760

Graduate Human-Computer Interactions & Usability

Evaluation Assignment 2

Heuristic Evaluation

Jiban Adhikary (Grad 9)

Team 2 (Blu Team)

February 20, 2018

1 Introduction

This documentation provides a detailed heuristic evaluation of a software application being developed. First, it provides a summary of the software that is being developed by a team of undergraduate students, the **Blu Team**. Then it identifies the user interface (UI) domain and lists out the heuristic usability principle for the design's UI domain. It also finds out the usability problems generated from the heuristic evaluation. Further, it identifies the critical usability characteristics that we are concerned about and depicts the concerns with some short stories.

2 Undergraduate System

Blu Team is working on an app which is basically a Volunteer Monitoring app. Volunteers are the backbone of Oklahoma Blue Thumb program, the eyes and ears of creeks and streams across the state. By using the app the volunteers will monitor the creeks monthly. They will assess the physical habitat, test sample of the creek's water. The monitoring will take place at the creek, and all the data related to environmental or physical factors affecting the creeks will be collected. It will also involve collecting a sample of water which will later be chemically tested in the lab. The chemical tests are supposed to be held on a comfortable time the volunteer prefers to do. Previously all of this data were sent out to Blue Thumb using traditional media like fax, email, or snail mail. This app will ease the monitoring process by facilitating volunteers to submit data using their smartphones and email them to the quality assurance officer.

3 Identification of the UI domain and short description

The UI domain of the app being developed follows a simplistic interface design. It contains some buttons and some forms. In the beginning of the app the system asks the user if a new log is to be recorded or the user wants to work on a previously recorded log. If user hits the ***New Log*** button, then a series of forms appear sequentially and the user logs the form observing the weather and site conditions. The user can fill up the chemical test results related form after collecting a sample of water from the water bodies and conducting lab experiments. The system also allows user to record his credibility as a volunteer and submit filled in logs. A user can come back later and work on these logs further. The form pages also provide some useful information related to the terms. A flow of initial design page to the end is depicted in figure 1.

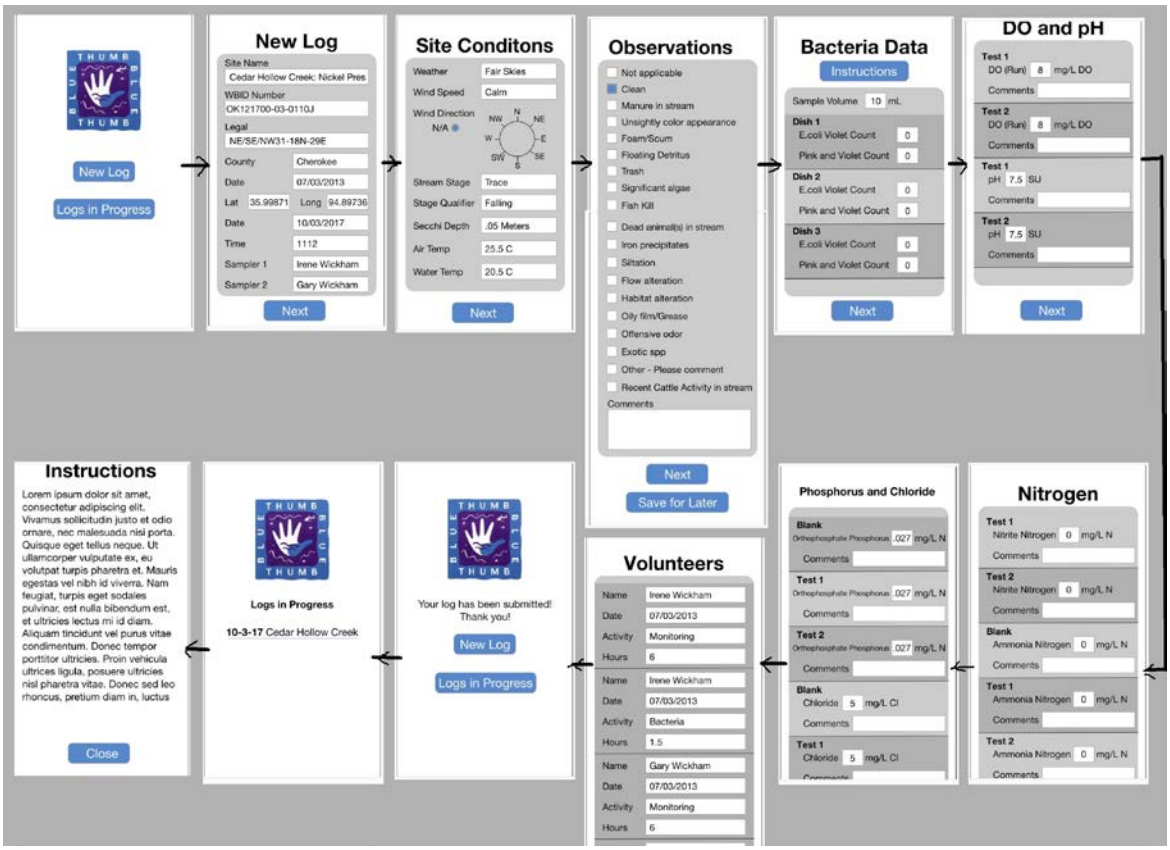

Figure 1: Primary UI of Bluthumb app

4 List of heuristic usability principles for the design's UI domain

In this section some general principle for design are discussed. Don Norman suggested 6 general rules[1] for designing a system. These six general rules are discussed in subsection 4.1 in terms of our UI domain in question. Apart from these design principles, Nielsen[3] provided 10 heuristics for evaluating a UI design. His heuristics are discussed in subsection 4.2. This subsection is followed by a few other secondary rules that help to make a complete and fulfilling UI. It is notable that there is similarity between the principles given by Norman and Nielsen.

4.1 Norman's six design principles

4.1.1 Visibility: This principle suggests the visibility of the functions and information in the UI domain. A well structured UI domain of an application may fail if the functionalities are not visible. The app being developed is a smartphone app and from the detailed analysis of the personas we are sure that volunteers of different ages will use the app. If the buttons and fields in the app are not carefully designed, for example, if they are too small to see, then volunteers with limited visual abilities may struggle interacting with the app. Also if proper information about the functionalities of the components (buttons, text fields, labels etc.) are not provided then it can also make the volunteers confused.

4.1.2 Feedback: An interactive system provides proper feedback upon the user's interaction. Feedback is used to engage users and explain things and enhances user satisfaction. Feedback is provided in the form of text, animation, change of visual effects etc. For example, a menu item changes color when it is selected or the cursor or pointer is placed over it.

4.1.3 Constraints: Constraints are the rules to restrict various type of invalid user interactions. A well designed system interface predicts about the invalid interactions and sets rules so that any type of mishap can be avoided. For example, when providing information in a form user may willingly or unwillingly input invalid data such as entering alphabetic characters in place of numerals, out of range values, invalid data format and not filling up the compulsory form fields etc. In these scenarios if constraints are not properly managed then the system can act arbitrarily.

4.1.4 Mapping: This suggests that controls must be mapped to their respective functions. A system is prone to changes and if controls are not properly mapped to functions then any alteration in the controls may lead to massive design faults due to dependencies. In this case, before a UI is implemented it is better to design a hierarchical structure so that controls and functions have explicitly sorted out dependencies and minimal changes are needed to adapt design alterations.

4.1.5 Consistency: Consistency suggests to incorporate design with controls so that similar operations and tools are used for similar types of works.

4.1.6 Affordance: Affordance is a property of an object that allows users to know how to use the object.

4.2 Nielsen's 10 Usability Heuristics for User Interface Design

4.2.1 Visibility of system status

A system should always make the functions and information visible to the user to let the user know about what is happening around. It also should provide appropriate feedback based on the interaction and within a reasonable amount of time.

4.2.2 Match between system and the real world

The system should be designed so that a user can easily apprehend the words, phrases and concepts used in the system i.e. the terms used in the system should not be arcane or just only system-oriented. Information should also appear in a natural and logical order.

4.2.3 User control and freedom

A system should allow user to explore the system freely. It might happen that a user navigates to an unwanted page while exploring. The system should have fall back mechanism so that the user can easily recover from this type of situations. It suggests a system should allow undo and redo options.

4.2.4 Consistency and standards

A system should be unambiguous. It should not have identical or similar terms and options for different actions.

4.2.5 Error prevention

A system should have the mechanism to prevent errors. The design should be planned in such a way that an unprecedented error is suspected in the first place. This calls for getting rid of the error-prone actions and providing functionality for reconfirmation when committing a critical task such as read, write or update.

4.2.6 Recognition rather than call

A system should be designed in such a way so that a user does not have to memorize the courses of action. The system should provide readable and easily retrievable information so that the user can act accordingly.

4.2.7 Flexibility and efficiency of use

A system should adapt itself so that both of the experienced or inexperienced users can easily cope themselves with their infrequent and frequent type of actions.

4.2.8 Aesthetic and minimalist design

A system should not contain irrelevant information that would mess up with the relevant information. So, the information provided should be concise and to the point in terms of usability.

4.2.9 Help users recognize, diagnose, and recover from errors

Easily comprehensible terms should be used in a system to indicate when an error is occured. There should also be instructions to diagnose an error. The system should also provide fail safe mechanism if a critical issue appears.

4.2.9 Help and documentation

Though it is not a must to have a documentation for a well designed and well working system but a documentation strengthens the system by providing minute details about the system. Also Help functions provide assistive information about the systems and the terms it uses.

4.3 Other Secondary principles

4.3.1 Control: This means how comfortable the user feels when operating the system. It involves the question if the system holds the user as prisoner or if the user has control over the system. Often time it is seen that for complicated design, the user is confused about the overall system and he finds a hard time to explore the system.

4.3.2 Robustness: A system should be robust. It should adapt to and recover from the mistakes done by the user.

4.3.3 Responsiveness: A system should provide response to user's interaction in a proper and timely manner.

4.3.4 Predictability: The system should also be predictable. The user should have a intuitive feeling about the progression of the functionalities of the system.

4.3.4 Security: A system should also be secure to use. It must ensure that it protects a user's privacy and confidentiality. Most of the systems uses a login process to authorize users connect to certain information and different encryption methods to protect their credibility.

5 List of usability problems generated from the heuristic evaluation

Nielsen's [2] severity rating for heuristic evaluation has been used to rate the problems found after the heuristic evaluation of the design documents. Nielsen's rating is given below. Here ☆ indicates a value 0 and ★ indicates a value 1.

☆☆☆☆ = Not a usability problem at all
★☆☆☆ = Cosmetic problem only: need not be fixed unless extra time is available
★★☆☆ = Minor usability problem: fixing this should be given low priority
★★★☆ = Major usability problem: important to fix, so should be given high priority
★★★★ = Usability catastrophe: imperative to fix this before product can be released

Problem: Lack of instructions for filling up the chemical test results
Category: Visibility
Severity: ★★★☆
Description: The design does not provide instructions how to conduct the chemical tests in the lab. Before filling up the chemical test results it is pretty intuitive to check if a user has gone through the proper procedure for conducting the experiments. Absence of these information suggests that users have to recollect the procedure from the training they had taken previously. It is impractical to think that the users will recollect everything they had learnt in the training.

Problem: Lack of visual effects
Category: Feedback
Severity: 
Description: It may not be a critical design issue but the UI does not highlight buttons or form fields being clicked on or filled in. Users should get the impression about which button he is currently on and which field of information he is currently providing.

Problem: Absence of navigation/slide bar
Category: Feedback
Severity: 
Description: The design lacks the presence of navigation/slide bars. Some forms in the system are quite long extending to multi pages and some form fields can take large string. No slide bars have been used to indicate the extension of the forms or fields.

Problem: Absence of Back button
Category: Visibility-Affordance
Severity: 
Description: One major design flaw to the system is absence of 'Back' buttons. It is very critical in terms of usability and functionality. Users should be allowed freedom to navigate through previous and next pages without any problem. It also asks for catering undo and redo capabilities.

Problem: Form inputs are not validated
Category: Constraints, Security, Prevention of error
Severity: 
Description: This is also a major flaw in the design. It was not clear in the cognitive walkthrough process if constraints have been applied on the form fields where an invalid input can result into catastrophic consequences. This is very crucial because if constraints are not applied then the database will contain invalid input that may lead to garbage data and misuse of human hours. It may also lead to severe database and security concerns which are pretty common nowadays.

Problem: Support for autofill data
Category: Mapping
Severity: 
Description: It is unclear form the design that which form fields will be auto filled. How GPS data will be collected, how the wind direction will be measured etc have not been adequately described.

Problem: Absence of search bar/ help functions/faqs
Category: Visibility
Severity: 
Description: The design does not provide searching for logs recorded in an earlier time, creek names and locations. It does not provide a help function to the user to walk him through. It also does not contain a faq section where commonly asked questions have been answered.

6 Identification of critical usability concerns

From the problems mentioned above we can sort out the critical usability concerns by evaluating the severity scales. The problems with a severity rating of 3 and 4 can be termed as critical usability concerns. The sorted out concerns are listed below:

(i) Lack of information for filling up the chemical tests results.

(ii) Absence of Back button

(iii) Absence of navigation/slide bar

(iv) Forms input are not validated properly

(v) Absence of search bar/ help functions/faqs

7 Illustration of the critical usability concerns with a short story

A couple of scenarios are presented below to illustrate how a user can get confused while using the UI the undergrad students have developed.

"I wish if I were the Heisenberg of Breaking Bad, meh"

This is quoted by a volunteer named Clara after she had trouble conducting the lab experiments. Clara is a college graduate and has a degree in Mathematics. She volunteers for Blu Thumb and last week she was conducting a site inspection for the first time. Earlier she has taken the training about how to inspect the creeks and run the chemical tests. But when conducting the inspection, she had a hard time to recollect some information about the test processes. She looked into the app but it was not that helpful providing the information she needed. She got unusual results

which were not close to the ranges provided in the app. She wished if the app had been more informative.

"Where is the NEXT button?"

Mr. Brown is a retired manager of a sells store. After retirement he was not finding anything reasonable to do. When he heard about Blu Thumb, he willfully come forward as a volunteer. Mr. Brown is less habituated about the current technologies and used his smart phone just for calling and texting. When he was introduced with the app he thought it should be easy. The other day when he was conducting an inspection to the 'Bird's song creek', which is 2 miles away from his home, he landed into the ***'Observations'*** page. The ***'Observation'*** page contains a form extending more than a page. And a '***Next'*** button is situated at the end of the page. The undergrad design does not provide any slide bar to give an indication that there is more information below. Mr. Brown as an infrequent user of such apps was finding a hard time to locate the '***Next'*** button. He was also struggling to get back to the previous page (the problem is depicted in the next story). He could not ask for help since he was alone and away from people. He had to postpone the inspection and terminate the app. He was very frustrated.

"It does not let me edit my previous entry, what a shame!"

Robert is a high school student and due to his enthusiasm for volunteering works he wanted to work as a volunteer for Blu Thumb during the summer. He was very prompt when taking the training and since he plays smartphone games a lot he was pretty comfortable using the app. But for having a young mind he is restless and always in a hurry in doing things. When he was filling up the forms last week while visiting a small creek in the neighborhood, he realized he had done a mistake in the previous page. He wanted to correct the information provided. But he found out that the app was having a major design flaw - it does not have a '***Back'*** button. He was very frustrated. He was wondering how come an app does not have a '***Back'*** button.

References:

[1] Norman, Donald A. The Design of Everyday Things. New York: Doubleday, 1990.

[2] Jakob Nielsen. 1992. Reliability of severity estimates for usability problems found by heuristic evaluation. In Posters and Short Talks of the 1992 SIGCHI Conference on Human Factors in Computing Systems (CHI '92). ACM, New York, NY, USA, 129-130. DOI=http://dx.doi.org/10.1145/1125021.1125117

[3] Nielsen, Jakob. "10 usability heuristics for user interface design." Fremont: Nielsen Norman Group.[Consult. 20 maio 2014]. Disponível na Internet (1995).

Appendix

1. Course website (http://cs4760.csl.mtu.edu/2018/)

2. Undergrad team website (http://www.csl.mtu.edu/classes/cs4760/www/projects/s18/group2/www/)