Usability Report
Team 43 to Team 44
11/19/2016

Our team discovered multiple flaws during our evaluation of Team 44's UI. The most noteworthy of these errors are described below.

A NullPointerException thrown whenever a user attempts to register with empty fields, but no message of any kind is given to user; the screen simply remains as-is, and does not react to the user's action in any noticeable way. This is a violation of the Nielsen heuristic of *Visibility of system status*, because the user is ignorant of what the system reaction was to their action, when in fact a NullPointerException has been thrown behind the scenes. The user is not informed that their attempt failed, nor why it failed, or how to rectify their mistake. While the user can nonetheless continue and successfully complete the registration, if they enter the proper input, they are left to figure out themselves just how to do so.

The recommendation of Team 44 is to display an error message detailing the specific problem; that the user has not entered valid input and their registration has not been completed. This communicates to the user that a specific error has occured due to user input, and how it can be fixed.

This issue was given a Severity Rating of 3 (Major) by Team 44, because it fails to inform the user and has the potential to confuse the user, thus worsening the initial problem of incomplete registration, because the user's actions are unpredictable.

In multiple fields for both a water report and a water purity report, the user can enter arbitrary strings including, or entirely composed of, invalid or nonsensical inputs, and the application will accept, store and display said inputs without protest. Specifically, the fields Type, Condition, Virus PPM, and Contaminant PPM all suffer from this same oversight. This is a violation of the Nielsen heuristic *Error prevention*, becuase the user is implicitly trusted to enter only valid inputs, and thereby explicitly permitted to enter completely arbitrary input at their own discretion. Furthermore, human error is inevitable, and with the current setup, even if a user intends to enter valid input but happens to mistype or misclick, the invalid input will not be checked in any way. This oversight is compounded by the fact that user input for the fields Type, Condition, and both PPMs are visible to other users, which could very well confuse them into believing that incorrect inputs are in fact valid, and thereby perpetuate the problem.

The recommendation of Team 44 is to redefine the trust boundary so that it falls between user and application, and validate the input the user provides in both Reports. This could be accomplished in multiple ways, from checking the strings against a set of acceptable strings, a dropdown menu, radio buttons, etc.

This issue was assigned a Severity Rating of 3 (Major) because it introduces a vulnerability to human error, and has the potential to propagate the error to other users.

As the application currently stands, there is no way to access the historical report functionality, or otherwise view a graphical display of the purity report data. The button is missing from the main screen, and as far as Team 44 can tell, from the entirety of the UI. This violates the Nielsen heuristic of *Match between system and real world*, since (presumably), historical reports have been implemented in some fashion within the application, but the user is left ignorant to their existence, because there is no way to view them.

The recommendation of Team 44 is to add a button onto the main screen, which would allow users with the appropriate authorization to view historical reports, as expected by the project requirements.

This issue was assigned a Severity Rating of 4 (Catastrophic) because it completely ignores a specified project requirement, and omits entirely a key part of the application as a whole.