

# OOP Heuristic Usability Evaluation – Group 16

Alexis Fimeyer, Max Eradus, Ewout de Dobbelaar, Rares Ţerban,  
Farhan Taufiqillah Lingga Djaya, Samuel van den Houten

## CONTENTS

Contents	
1 Introduction	1
2 Methods	1
2.1 Experts	1
2.2 Procedure	1
2.3 Measures	1
3 Results	2
4 Conclusion & Improvements	2
4.1 Usability	2
4.2 Buttons	2
4.3 Error Messages	3
4.4 Missing menus	3
4.5 Added Functionality	3
4.6 Final design	3
References	3
A Design before	4
B Design after	5

## 1 INTRODUCTION

In this report we will be evaluating a prototype for the client of our task board application. We have recruited a team of experts to be our evaluators, to provide new insights we wouldn't have gotten by looking at the prototype ourselves. The end goal of this process is to improve the usability of our design and by extension our final application.

Our prototype consist of a series of wireframes, see appendix A.

## 2 METHODS

### 2.1 Experts

For our Heuristic Usability Evaluation, we recruited a group of experts from the Computer Science and Engineering department at our university. The group consisted of five individuals who are currently in their first year of the program. While they may be considered relatively new to the field, they have recently completed a course on Web and Database Technology and have gained foundational knowledge on usability and user-centered design principles. Additionally, they are highly motivated to expand their knowledge in this area and have demonstrated a keen interest in improving the user experience of digital products. Although They are not professionals in the field yet, we are confident that the insights and feedback provided by this group of experts will be useful in identifying usability issues and making recommendations for improvement.

### 2.2 Procedure

Our exerts were provided with a PDF document containing all our wire-frames. The experts were also provided a list of the heuristic principles and video's about how to conduct a heuristic usability evaluation. In these videos the experts were told to do the review on an individual basis. In the videos it was explained that the experts should examine the wire-frames over a period of around 2 hours. During which they should look over the entire wire-frames multiple times to try to find all the points of improvement. During the heuristic usability evaluation the experts were only able to view the wire-frames we had provided and the list of heuristic usability values. The list that they had to use was:[2]

- (1) Visibility of system status
- (2) Match between system and the real world
- (3) User control and freedom
- (4) Consistency and standards
- (5) Error prevention
- (6) Recognition rather than recall
- (7) Flexibility and efficiency of use
- (8) Aesthetic and minimalist design
- (9) Help users recognize, diagnose, and recover from errors
- (10) Help and documentation

The reviewers were asked to evaluate the mock and write down a brief description of the problem, the anticipated difficulties that the user will encounter as a consequence of the problem, the specific context in which the problem could occur and a description of the cause of the problem. After this step the experts were asked to discuss in a group to merge all the found problems into one list of problems and to add the specific heuristic usability value that the problem breached.

### 2.3 Measures

Our experts were asked to structure their review as a list of problems. Once they had all created their individual lists, all lists were merged into one (removing duplicates). This created a final pdf file, which we received from the experts. We asked our experts to deliver their problems in the following format:[1]

- (1) **Problem Description:** a brief description of the problem
- (2) **Likely/actual difficulties:** the anticipated difficulties that the user will encounter as a consequence of the problem
- (3) **Specific contexts:** the specific context in which the problem may occur
- (4) **Assumed causes:** description of the cause(s) of the problem

### 3 RESULTS

In total we received 21 problems from our experts. The problems can be roughly categorized into the following categories:

- Lack of functionality/UI
- Bad terminology
- Lack of descriptive error messages
- Function of UI element is unclear
- Miscellaneous

We received 8 problems describing missing UI elements or scenes. Some important examples are screens for tag management and appearance customization. Various navigation buttons are also still missing.

We also received 5 problems describing bad or confusing terminology. Some, like "TaskList" and "Board ID" make sense in the context of software development, but are confusing when used in a user-facing application. Others, like "choose" and "select" become confusing when paired with the wrong input type, like a textfield.

3 issues describe the lack of (descriptive) error messages. They mention only 1 generic error message being present in the prototype, which could create confusion about the type of error that has appeared.

We received 4 problem describing unclear functions of UI elements (1 of is also included in "Bad terminology"). These describe UI elements that look like other UI elements, which makes their use ambiguous. Some examples include button that looks like tags and underlined text that doesn't function like a link.

Finally, we received 2 miscellaneous problems: the app mostly uses low-contrast shades of gray which makes the application difficult to use for visually impaired people and there does not exist a confirmation screen for deleting cards, which may lead to accidental deletions.

We placed the problems we received into a priority matrix, which gives us a clear view of ones that need to be fixed most urgently.

#### Identified problems:

- 1. Unclear error message (Heuristic 1)
- 2. TaskList deletion option missing (Heuristic 3)
- 3. Confusing "Finished" & "Delete" buttons (Heuristic 3)
- 4. No board editing overview (Heuristic 3)
- 5. Inconsistent TaskList naming (Heuristic 4)
- 6. Board deletion option missing (Heuristic 3)
- 7. Missing "Cancel" button (Heuristic 3)
- 8. Confusing "Choose" term (Heuristic 4)
- 9. Ambiguous "board ID" (Heuristic 4)
- 10. Confusing "Change" & "Disconnect" buttons (Heuristic 4)
- 11. No error message for empty input (Heuristic 5)
- 12. Vague error message (Heuristic 9)
- 13. Ambiguous underline function (Heuristic 6)
- 14. Unclear "Edit" buttons (Heuristic 4)
- 15. Missing color/tag modification (Heuristic 6)
- 16. Unclear "Select a board ID" message (Heuristic 6)
- 17. No customization option (Heuristic 7)
- 18. Missing subtasks & tags section (Heuristic 7)
- 19. No Tag Management page (Heuristic 7)
- 20. Low contrast gray shades (Heuristic 8)
- 21. No deletion prevention (Heuristic 9)

**Table 1: Priority Matrix**

Frequency \ Impact	1	2	3	4	5
1	<b>13</b>	<b>11</b>	—	—	—
2	<b>5</b>	<b>12, 14, 19</b>	<b>15, 16</b>	—	—
3	<b>10</b>	<b>3, 4, 7, 8</b>	<b>9</b>	—	<b>21</b>
4	—	<b>17</b>	<b>2, 6, 18</b>	<b>1</b>	—
5	—	<b>20</b>	—	—	—

### 4 CONCLUSION & IMPROVEMENTS

A whole lot can be improved about the wire-frame now that the results are in. There were twenty-one issues to look through and see how practical they would be to implement.

#### 4.1 Usability

First of all it can be said that a whole lot of issues lay in the area of usability. This will include changing some prompt-text. This is a very good improvement which wouldn't have found if we just reviewed the wire-frames ourselves. These changes will make the user less confused and more familiar with the app. these changes include:

- Problem 5: Change TaskList into Task List in the creation screen
- Problem 8: Change the "choose a ...." to "name of ..."
- Problem 9: Change term "Board Id" to "Board key"
- Problem 10: Change the text on the change server button to : "Join new Server"
- Problem 13: Delete the underline and make the text bold so that it is still obvious that it is a header.
- Problem 16: Change the "Select a boardId" to "Enter a board key".

#### 4.2 Buttons

Then there are the issues with misplaced / unnatural buttons. This is also something we should immediately improve upon. This includes repositioning, re-texturing and the addition of buttons. The issues regarding the buttons are and their implied changes are:

- Problem 2: The addition of a "Delete" button for a TaskList.
- Problem 3: Re-style the buttons for "Finish" and "Delete" in card view to be inline with the rest of the application.
- Problem 6: The addition of a "Delete" button for a Board .
- Problem 7: The addition of an "exit" button in "Add Task List".
- Problem 14: Re-style edit buttons to be just a pencil, such that it does not imply such importance as it does now.

### 4.3 Error Messages

The next section of Problems had to do with error messages. These are really important for a good working application so these should have really high priority. The amount of issues in this category can also be attributed to just making a general error message wire-frame, and not one for every possible error in the system. But if the application is finished, the application does definitely need an error message for every possible situation, so this is more of a list of all the error messages that need to be crafted.

- Problem 1: Make the error message customize-able for every possible situation.
- Problem 11: Make a error message that appears when the field is empty and the user tries to submit.
- Problem 12: Again, make the error message modular so that the wire-frame can be used everywhere but the message can be changed.
- Problem 21: Make a error that lets you double check if you want to delete something.

### 4.4 Missing menus

This section of problems has to do with missing screens. Screens are necessary for requesting information from the user and creating and editing elements in the program. The issues in this section are characterized by an oversight when creating the mocks. These screens are must-have's in our application.

- Problem 17: Make Customization page, for changing Board and UI colours.
- Problem 18: Make a screen to add sub-tasks and tags to cards
- Problem 19: Make a screen for tag management
- Problem 20: This problem is connected to problem 17 in that it points out that the colors used in the application might be hard to discern for visually impaired people, since the color of buttons was only slightly darker than the background, this could be solved in a color picker or by making the buttons higher contrast.

### 4.5 Added Functionality

There were also problems with the clearness of some functions. This is taken together in this category with functions that were simply missing, though they are advanced requirements so its not the biggest problem. But the problems which deal with the clearness of certain functions do most certainly need to be addressed.

- Problem 4: The name of the Task List and BoardName need to be made editable if the user clicks on edit, this is a "in-place" edit. This could also be made a bit clearer by making the button a pencil instead of making it say "edit".
- Problem 15: The functionality to add tags to cards needs to be added, and in the same vain, the functionality to change the color of a specific card.

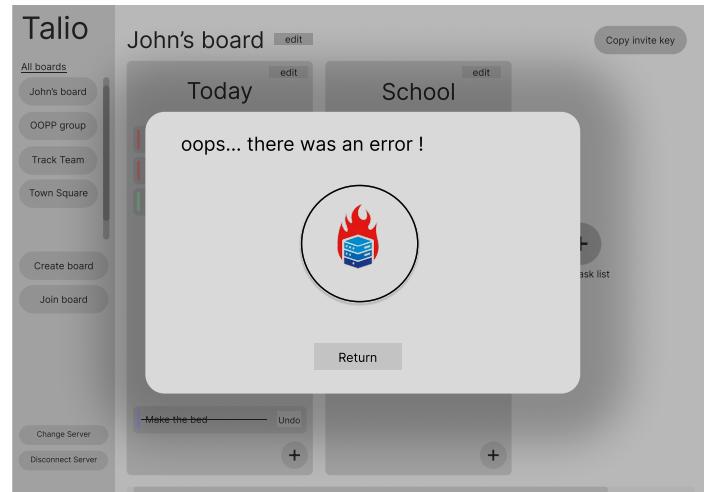
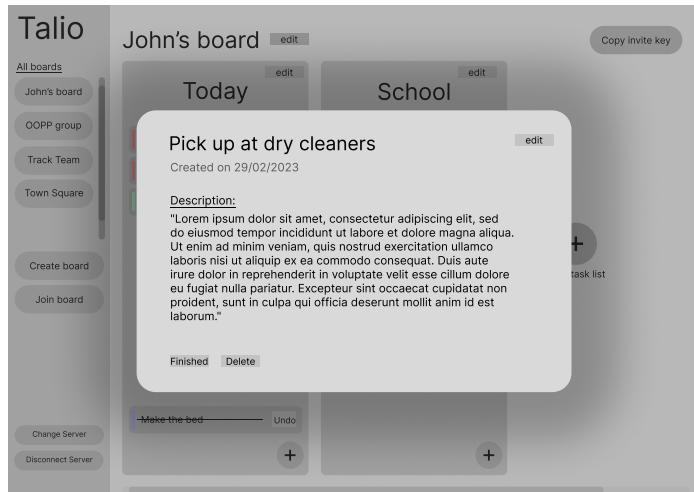
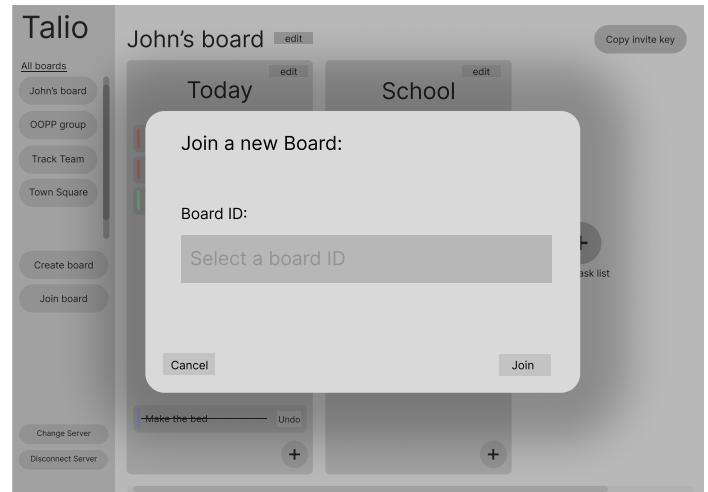
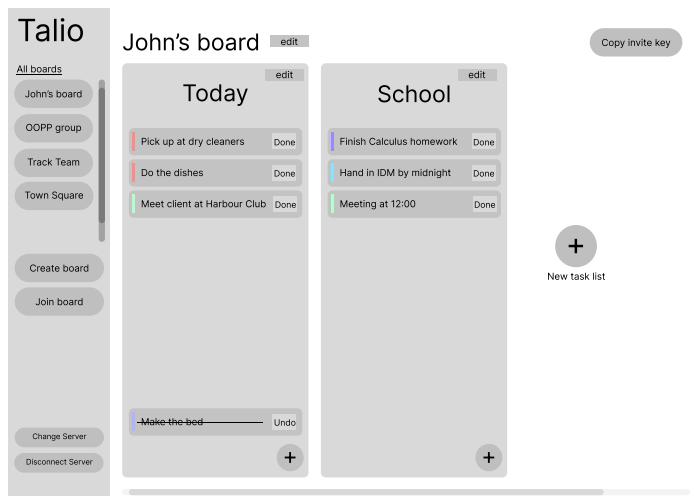
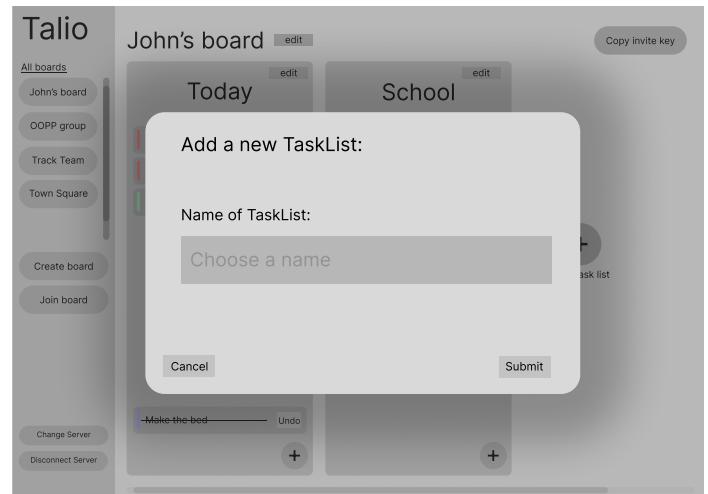
### 4.6 Final design

Our final GUI design can be found at appendix B.

## REFERENCES

- [1] Cockton, Woolrych, Hall, and Hindmarch. 2003.
- [2] Jacob Nielsen. 1994. Heuristic Evaluation. In *Usability Inspection Methods*.

## A DESIGN BEFORE



## B DESIGN AFTER

