

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	2
3 Results	2
4 Conclusion & Improvements	2
4.1 Lack of Functionality	2
4.2 Bad terminology	2
4.3 Lack of descriptive error messages	3
4.4 Function of UI elements unclear	3
4.5 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 representing scenes in our final application, 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 our application prototype. This prototype consists of a series of wireframes, which each depict a screen that will be present in our final application. Figure 1 depicts a connection screen, where a user can input a server address to connect to. From there, the user will enter the Board Overview (Figure 2), where the content of the current board will be displayed and switching to other boards is possible via the sidebar. Double-clicking one of the tasks (e.g. "Do the dishes") will lead to a "Task details" popup (Figure 3), which show additional information about the task. Similarly, pressing the "Add task list"-button will lead to a popup which allows the user to create a new named task list (Figure 4). The "Join board" button in the sidebar leads to a popup which allows the user to join either a new or existing board (Figure 5). Finally, in the case of an error during any of these actions, a popup will appear informing the user about this (Figure 6). This prototype includes the basic requirements, in addition to the multi-board and card details requirements, as described in the example backlog. 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. The experts should examine the wire-frames over a period of around 2 hours. During this time they should examine the application flow depicted by the prototype multiple times and note any usability problems and areas of improvement from a user perspective. 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. The experts were not supervised or assisted by us during any part of this review process.

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 problems 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

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

4 CONCLUSION & IMPROVEMENTS

With these results we can see that our original prototype had very many problems with our functionality. We miss functions or scenes and that is a big problem. These scenes are the features that the client is going to use and they need to be as good as they can be. This is why these are the main problems that we want to work with.

4.1 Lack of Functionality

The lack of functionality sounds like the biggest of our problems. But most of these functions we lack are not mandatory so it is not the biggest problem. There are some problems that do underline the lack of some functionality. One of these would be that we do not have a button for deleting task list(Problem 2). This is a very important function and can really not be forgotten(see Table 1). This is because if the user can not remove boards, his workspace will get cluttered with old boards. So we improved this in the Design (See appendix B: 8). Another of these issues would be that the user can not change the task list name and board name(Problem 4). This is important because this prevents that small typos will ruin user experience. We fixed this by making a pop-up wire-frame in which the user can change the name of the task list(See appendix B: 14).

4.2 Bad terminology

First of all it can be said that very many issues lay in the area of terminology. This will result in us changing some prompt-text. This is a very good improvement which wouldn't have been found if we just reviewed the wire-frames ourselves. These changes will make the user less confused and more familiar with the app. Some of these changes are: changing the prompt text in the task list creation screen form "choose a" to "enter a ..." (Problem 8)(See appendix B:

10). Another one would be not so much terms but icons: we made the text that said "edit" into an icon of a pencil(problem 14). This will make it at first glance obvious what it does. This helps with usability(See appendix B: ??, 14).

4.3 Lack of descriptive error messages

One of the most important aspects of an application is its error handling. If the errors are ambiguous the user will never understand what they mean. And in turn he may never know what the problem caused. This is why we will change at least some error messages. The ones that we will changes are these: The most Problems were complains about not having enough error messages(Problem 1, Problem 12, Problem 21). These are very important problems, as is visible from there position in the priority matrix(Table 1). But this is almost a trivial thing to improve. But we can not really show all of them in the appendix because the appendix would then be filled with just error messages. So we just made a new error popup, and the text in it is modular. So you can assume that all errors will be shown like this(See appendix B: 12).

4.4 Function of UI elements unclear

This is when we have functionality, but it is not clear what it does. This can be because of an unclear color or shape, or because of earlier experience with similar systems systems that had other functions mapped to certain designs. A prime example of this is the underlined "all boards" title(Problem 13). In most systems an underline means that it is clickable. This can create confusion for the user. That is why it is also high in the matrix(see table 1). We fix this by just removing the underline and making the text bold instead(See appendix B: 8).

4.5 Final design

When opening the application, the user will again be presented with the connection screen (Figure 7). As with the initial prototype, this leads to the board overview (Figure 8). The "task details", "add task list" and "join board" wireframes (Figure 9, 10, 11) are also unchanged in function and method of access. The "error" popup (Figure 12 still appears in the same scenarios as before, but the error text now contains a description of the error that occurred. Clicking the "edit" button in the task details popup will allow the user to edit that task (Figure 13). The pencil icon adjacent to the card title and description, when clicked, allow the user to edit those specific fields. At the bottom of the popup, the user is also able to select the color the task will be tagged with. The "Submit" button will make these changes final, while the "cancel" button discards the changes. Both buttons bring the user back to the board overview. From the overview, clicking the "edit" button on the task list will bring the user to its editing screen (Figure 14). Here, a list of the contained tasks will be displayed. Click the "X" one will permanently delete the task. The editing of the title and the cancel and submit buttons work similarly to the "edit task" popup. Task lists themselves can also be deleted by clicking the "X" that appears at the top left of a list in the overview. Both the deletion of a task and a task list will prompt the user with a confirmation request (Figure 15, 16). Clicking "No" will bring the user back to the previous scene, while clicking "Yes" will make the change permanent.

REFERENCES

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

A DESIGN BEFORE



Figure 1: Connect

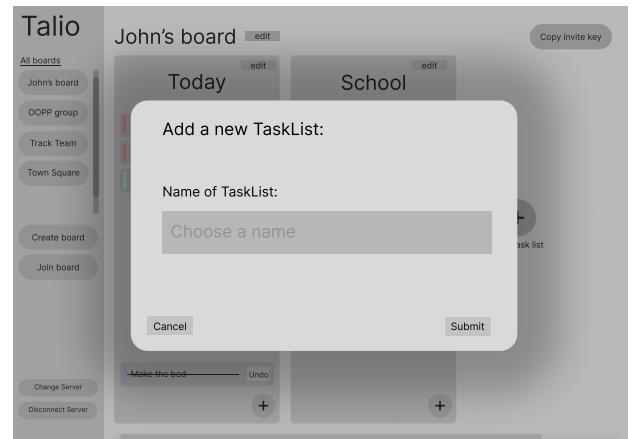


Figure 4: Add task list

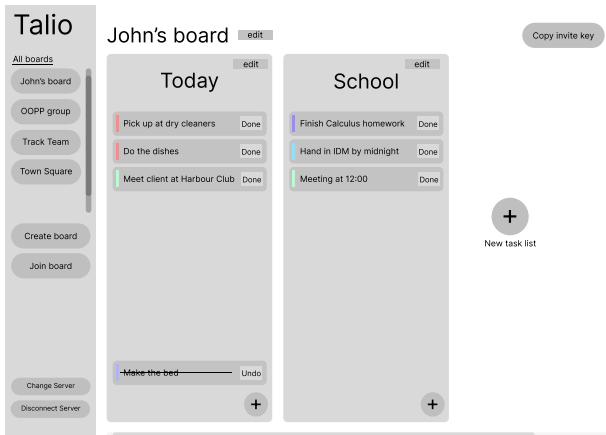


Figure 2: Board overview

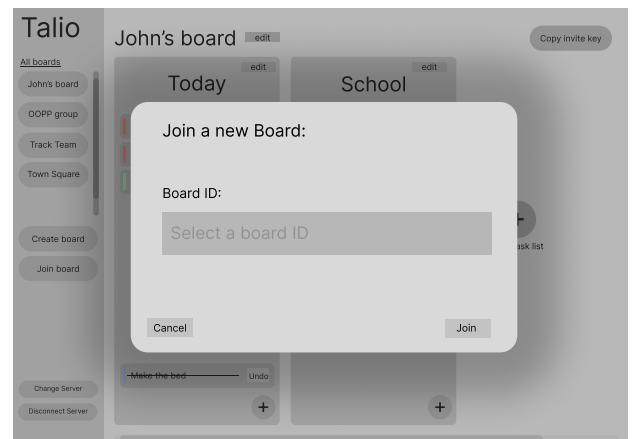


Figure 5: Join board

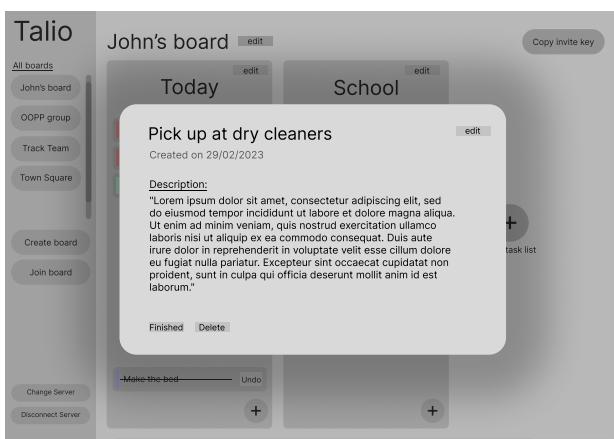


Figure 3: Task details

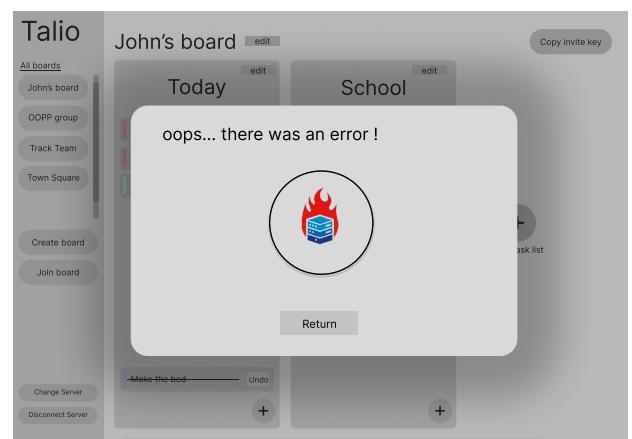


Figure 6: Error

B DESIGN AFTER



Figure 7: Connect

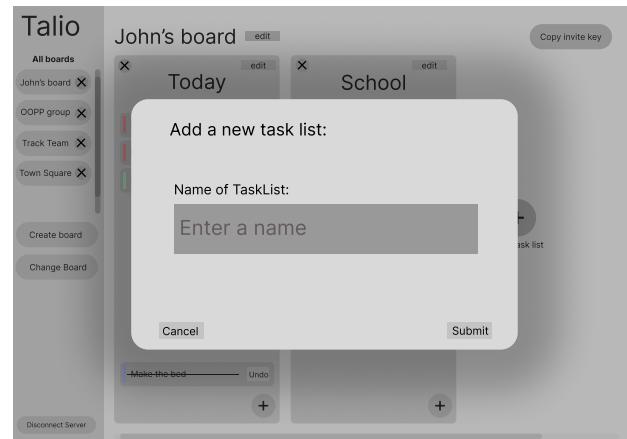


Figure 10: Add task list

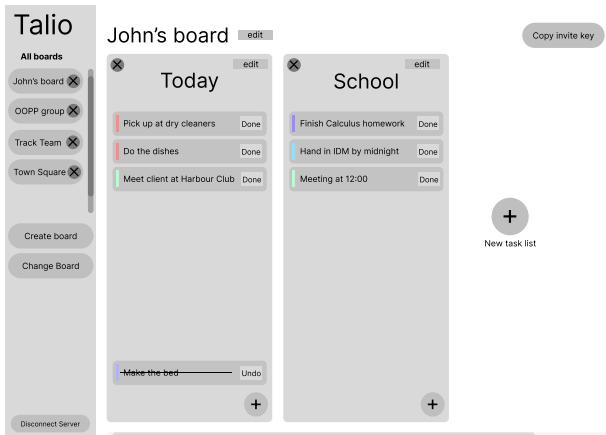


Figure 8: Board overview

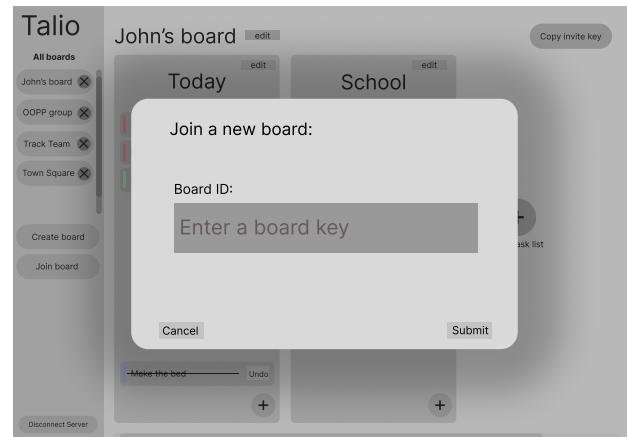


Figure 11: Join board

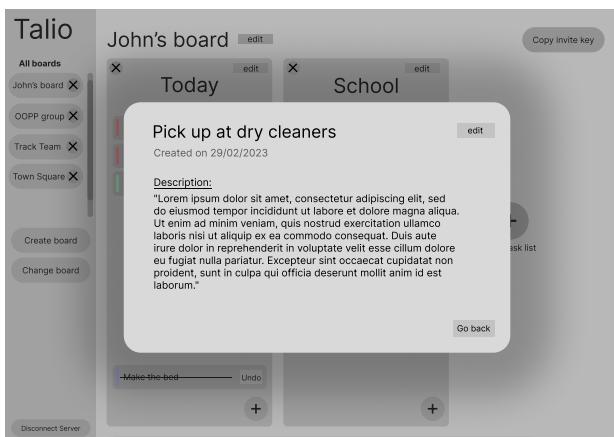


Figure 9: Task details

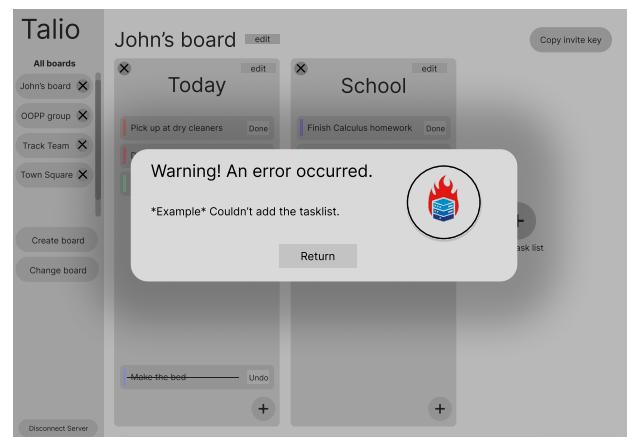


Figure 12: Error



Figure 13: Edit task

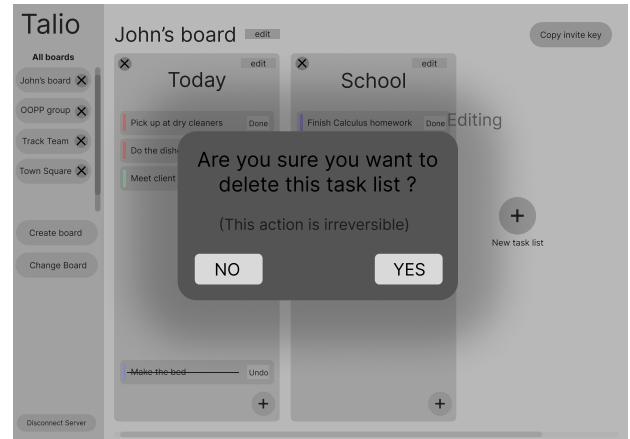


Figure 16: Delete confirmation task list

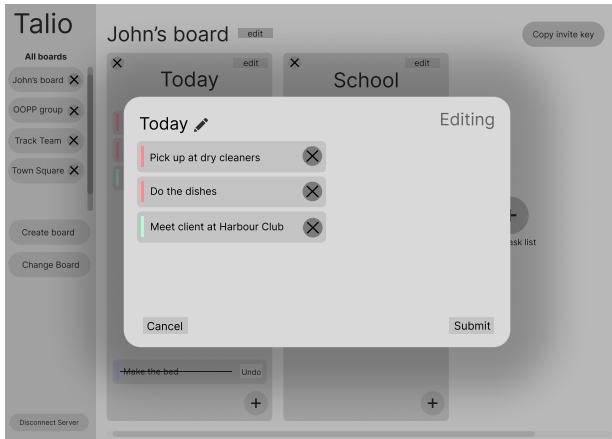


Figure 14: Nested tasks



Figure 15: Delete confirmation task