

OOP Project Report – Group 65

Sare Öztürk, Lia Petrova, Andreas Shiamishis,
Thomas van Weert, Cosmin Vasilescu

ABSTRACT

This report aims to describe the procedure and analyze the heuristic evaluation of the application currently developed by our team - *Talio*.

1 INTRODUCTION

The evaluation, ran according to the procedure described by the Nielsen Norman Group [2], aims to evaluate the app's GUI, making use of Nielsen's 10 usability heuristics 2.2.1. This evaluation is conducted with a group of experts, who, according to our own methodology, have run an independent analysis to identify and report any features that could be improved upon, to come up with a list of enhancements that would refine the overall user experience.

The evaluators are presented with a prototype of the app (appendix A.1), which aims to help its users organize different tasks in boards and lists (4) in an intuitive and user-friendly manner, incorporating natural actions like drag and drop (13, 14 for cards; 8 for lists) and keyboard shortcuts. For each component, they are provided with the basic CRUD operations (for cards: 15, 10, 12, 11; for lists: 5, 7, 9, 6). Furthermore, collaboration is encouraged, allowing multiple users to work simultaneously in the same environment, by connecting to the same server and accessing the same board (2, 3).

2 METHODS

2.1 Experts

Given that a single person's findings are not usually representative of the entire set of problems, and different typologies (demographics, psychological etc) of people look for and identify different types of issues, it has been decided to choose as evaluators another team consisting of 6 students with very little experience in front-end development, all participants in the OOPP course. Thus, the overall level of expertise of our project's evaluators was relatively low. However, they demographically matched the profile of a potential user of *Talio*.

For an inclusive and authentic environment, each evaluator conducted their research independently, having reported to the observer that they have gone through the interface at least three times.

2.2 Procedure

The experts were tasked with analyzing the application based on a live demo, by comparing the design against accepted usability heuristics. To have the analysts explore all aspects of the application, including its intuitiveness and usability, we provided a variety of tasks to be completed without prior knowledge.

2.2.1 Heuristics. In our evaluation we have used the following usability heuristics, of which detailed descriptions can be found in the related article [1]:

- (1) **Visibility of system status**
- (2) **Match between system and the real world**
- (3) **User control and freedom**
- (4) **Consistency and standards**
- (5) **Error and 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 last part of the procedure consisted of a debriefing session, enabling the evaluators to report to the observer - member of the development team - on positive aspects, which wouldn't be included otherwise.

2.2.2 Scenarios and Evaluation. The evaluators were assisted by an observer, part of our design team, able to provide assistance with certain tasks, e.g., starting the server and (multiple) clients from an IDE present on a provided laptop. Nevertheless, the evaluators were only provided help when they were clearly unable to continue - such situations indicating fundamental usability issues.

The evaluators were given access to our current, provisional version of the application (which only included features that had been already approved and merged into the main branch). They had the opportunity to interact with it in real time. As they each navigated through the interface, they were instructed with different scenarios to follow in order to spot any mistakes or inconsistencies. We repeated these scenarios four times total spread over the sessions to ensure complete coverage. The scenarios provided are the following:

- When initially starting the application the evaluators were instructed to enter a server. Furthermore, they tried changing the dimensions of the screen to see whether the scaling of the elements is correct. They also attempted to enter different addresses than the correct one as the server to see how the system handled such errors.
- When entering the server, they were initially presented with an empty board. At this point, they were instructed to interact with lists, without entering any cards. This included adding, editing, deleting and dragging lists across the board to change their order. They used the "Add Card List" button

to add a list, entered a name into the text field in the pop-up window and confirmed. They experimented by trying out different names, very long or empty ones. They deleted some lists by clicking the "X" next to the list's name and confirming, or cancelling, the action.

- After creating a couple of lists, they were instructed to create and interact with cards. More specifically, to add a card, name it, delete it, drag it to another list or another spot in its current list and edit its name. Then, the evaluators experimented with the cards by editing them, by clicking on the card's title, deleting them, by clicking the "X" next to the card's name and dragging them across the board, by clicking the drag sign next to the "X".
- Finally, they were instructed to experiment with the multi-user features of the application at which point they opened a second client to see how the board is altered when another client applies changes to the same board. They edited, deleted, added and moved both cards and lists and observed the immediate changes, if any, on the second client window.

2.3 Measures

The data being collected within our evaluation consisted of the feedback provided by the evaluators, to be used to measure the usability and clarity of the design. The experts were asked to vocalise points that were not clear to them at first glance or anything that they found difficult to use or understand, as they went through the different scenarios, followed by a comparison to their initial expectations. The received input was collected in writing by the observer, in the form of notes. We have chosen this way because we wanted to avoid impoverishing the evaluators with any additional tasks, to maintain the focus on the app itself. This method also rapidly provided the results, with all the reported problems being conveniently included in the observer's notes.

The collected feedback from the evaluators was used in the measuring of the usability and clarity of the design. The experts were asked to report features they found unclear

3 RESULTS

3.1 Data adjustment procedures

Given the raw results, we aimed not to alter the content unnecessarily much. Thus, we have compiled the content by first formalizing the issues to better align with our internal terminology and to be more concise, by capturing the main meaning in a purely denotative fashion. Afterwards, we attempted to group together issues that corresponded to similar components of our app, to facilitate the process of reflecting on potential solutions. Finally, for each issue we tried to better emphasize the context in which it appeared.

3.2 Findings

We hereby present the list of the identified usability issues in increasing order of priority, each with a reference to the violated heuristic:

- (1) The application lacked any kind of personalization settings page, to the detriment of users not satisfied with the default design (8), furthermore, hindering their ability to customize the app into a more familiar interface (6).
- (2) When entering the program, no default server is provided, instead making the user responsible for providing the exact address of the server he wants to connect to - a problem for a user not familiar with the terminology (2 6 7).
- (3) Keyboard shortcuts were noted as absent, lowering performance and preventing users from using the program to its full capacity. Such accelerators are of great advantage, especially to users already familiar with the program, allowing more efficient app navigation(7).
- (4) In case a card without a title (an empty one) was created, it could not be edited, as a change would only be possible when the user would click on the text in the box assigned for the name of the card, pointing towards an error in allowing empty titles in the first place (3 5).
- (5) A list with a long title and a user would press the button meant to delete it, the dialog wouldn't scale accordingly, causing the text to overflow and cover the cancel button - a significant issue in the case of an accidental press. (3).
- (6) Creating a list of cards with a long title, instead of adjusting the visual length of this particular list in accordance with it, the title itself appeared as a simple centred block, while the list was still stretched out with a lot of empty space on both sides of the name of the list. This brought about both unpleasant overview of the board and confusion about the purpose of the title, not properly identifying the element (?? 4).
- (7) The lack of a button for going back to entering a server, or any other type of an 'emergency exit', in case something in the course of the program and/or the user interacting with it goes wrong, would inevitably cause trouble for any inexperienced user (3 9).
- (8) Setting the application to full-screen would lead to improper scaling of the window and content. The scene corresponding to entering a new server, the content remained in the left corner with the same size it had before, a similar behaviour being exhibited when resizing the board overview, with the content resizing in an unpredictable and inconsistent fashion (4).
- (9) Even though the application displayed error messages, e.g. on entering a wrong server, they were not fully noticeable and dragging the attention of the user, neither were they entirely specific on how to fix the error. When entering a wrong server address, the subtle warning got displayed, but after the input was fixed, the error message would not disappear until a new address was added and another connection to the server was attempted (9).
- (10) By the time the heuristic evaluation was conducted, the synchronization of the drag and drop for cards and lists would not work properly for multiple users connected to the same board, which caused inconsistency when the app was being used concurrently by more than one person (4 9).

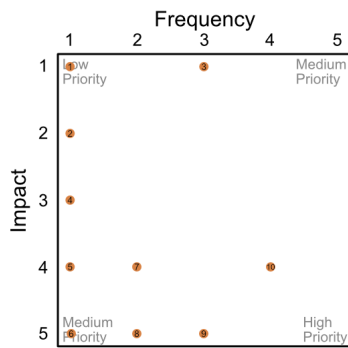


Figure 1: Priority Severity Matrix

As previously mentioned, as part of the debriefing session held with the observer, some positive aspects of the app, which the team intends to keep part of the GUI in the future:

- The fact that every delete button showed a prompt window asking the user for confirmation once more was highly appreciable in accordance to them, because this way many accidental deletions and possibly irrecoverable situations would be avoided.
- The board being scroll-able and the lists maintaining their relative positioning on it was highly valued.
- Running a check not only on the format of the server address, but also its correctness was also appreciated, as it would prevent a user from running into unknown errors by connecting to an address which does not belong to a server actively running the app.
- Certain interface animations were also appreciated, such as those facilitating CRUD operations.

4 CONCLUSIONS & IMPROVEMENTS

While the overall sentiment towards the current stage of our application was positive, there are clearly a lot of improvements to be done. Based on the previously enumerated list of findings, we have come up with a list of changes to improve on the before mentioned findings. The list of changes is in increasing order of priority, thus each corresponding to the respective issue it addresses:

- (1) Personalization features will be introduced. 21
- (2) A default server address - `https://localhost:8080/` - will be provided in the input field that the user has to fill in, and the user will be informed to leave it as it is in case he doesn't want to connect to a server with a particular address. 16
- (3) Keyboard shortcuts will be introduced.
- (4) Form validation will be enabled for creating and updating actions, which will impose non-empty titles. 20

- (5) A limit to the number of characters the title of a card or a list can have will be imposed, and the list's title will make better use of the reserved area. 18
- (6) The confirmation dialog for deletion will be adjusted to completely fit title of the component it refers to. 19
- (7) As part of keyboard shortcuts, the CTRL-Z combination will be enabled, and further button additions will be added to allow users to revert their actions or exit the board.
- (8) The different components of the app will be made responsive to window resizing / rescaling, so that they would both use as much space as needed, and preserve an aesthetic look. 17
- (9) Error message reporting will be more consistent, noticeable and helpful in suggesting potential solutions to solve the problem to the user. 18
- (10) Multi user features will be fully supported to prevent anomalies due to concurrent modifications.

Something to note is that not all improvements however can be paired with visual changes (see appendix A.2). The heuristics describe a set of overall rules for the application, but these are not limited to the visual aspects of an application.

REFERENCES

- [1] Jakob Nielsen. 1994. *10 Usability Heuristics for User Interface Design*. <https://www.nngroup.com/articles/ten-usability-heuristics/>
- [2] Jakob Nielsen. 1994. *How to Conduct a Heuristic Evaluation*. <https://www.nngroup.com/articles/how-to-conduct-a-heuristic-evaluation/>

A APPENDIX

A.1 Prototype screenshots

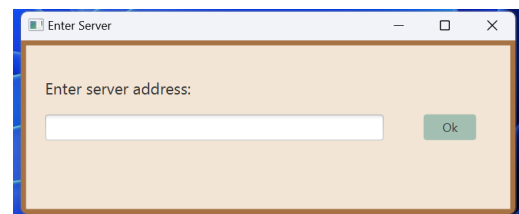


Figure 2: Enter server address

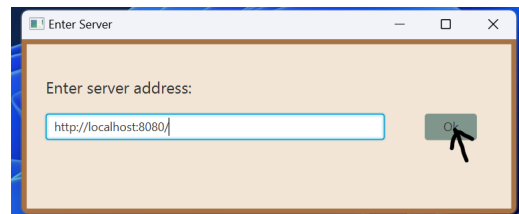


Figure 3: Enter server address - pressed button

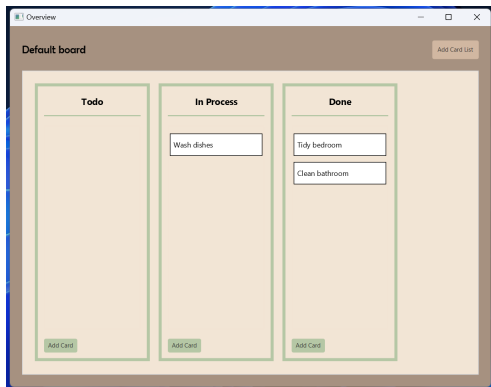


Figure 4: Board overview

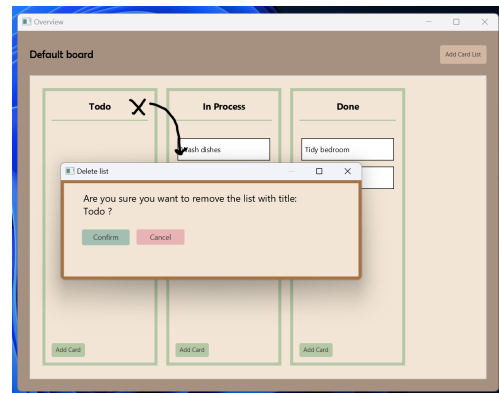


Figure 7: Delete list

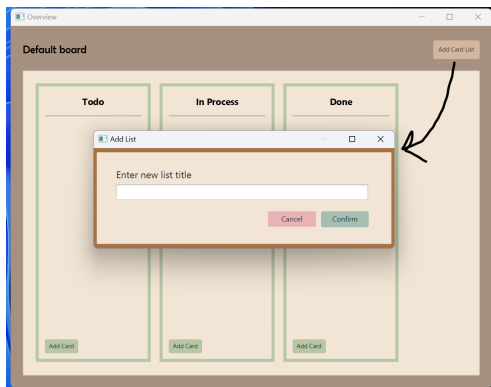


Figure 5: Add list

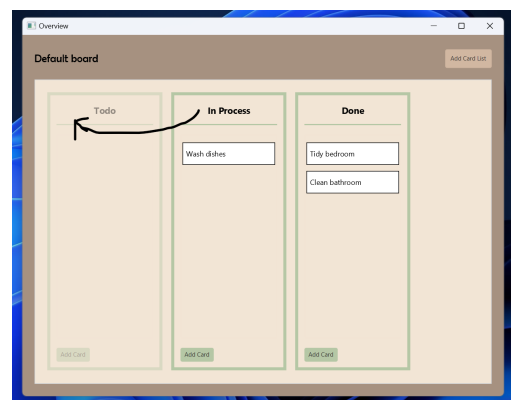


Figure 8: Move list

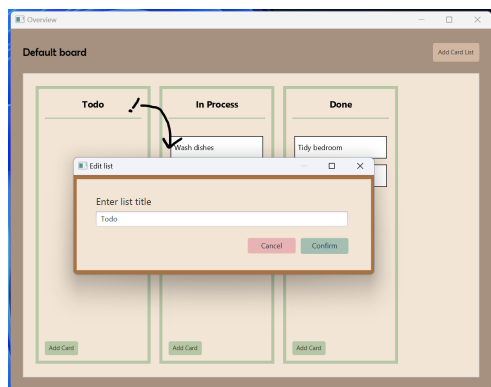


Figure 6: Edit list

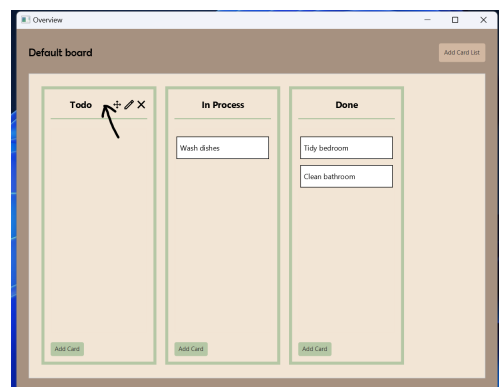


Figure 9: List options

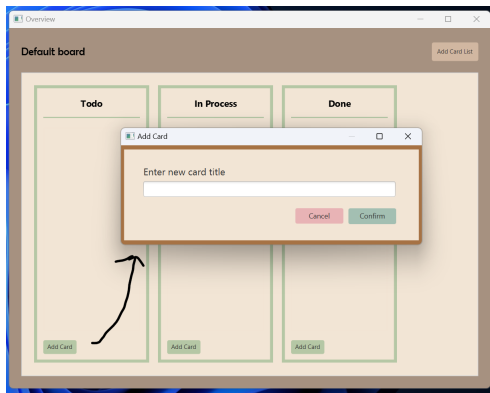


Figure 10: Add card

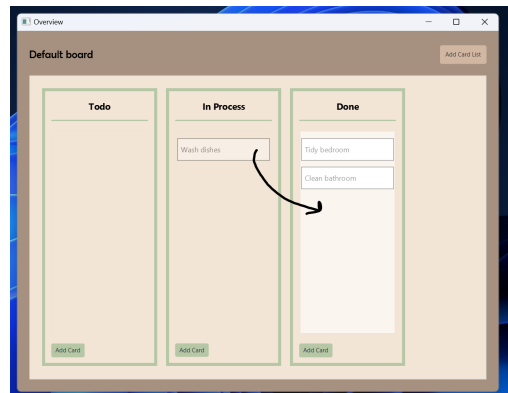


Figure 13: Move card

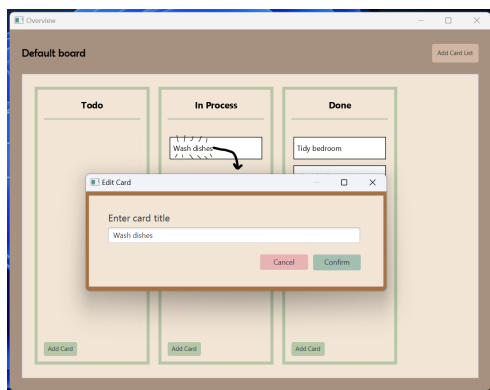


Figure 11: Edit card

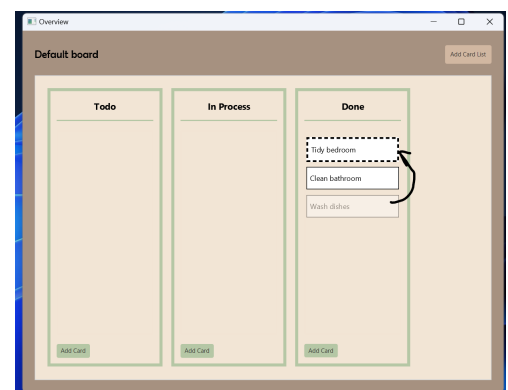


Figure 14: Reorder card

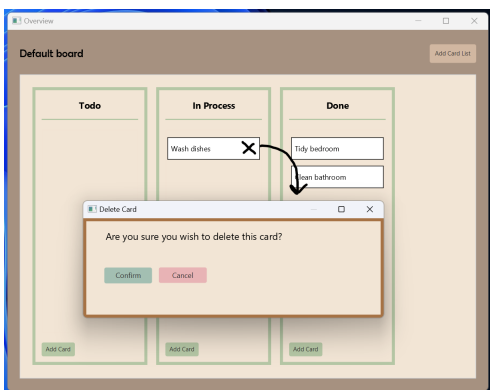


Figure 12: Delete card

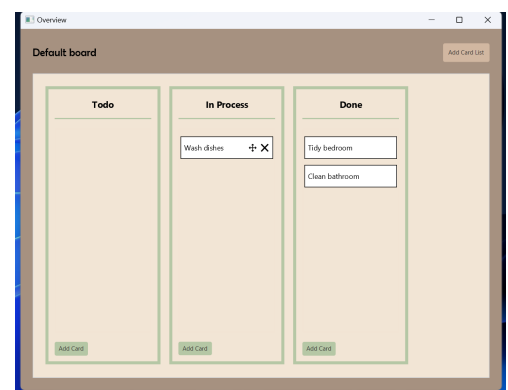


Figure 15: Card options

A.2 Improvement Mocks

Enter server address:

OK

Figure 16: Default Server

Enter server address:

OK

Server is not a valid Talio server

Figure 17: Responsive Design

Enter new list title:

Title is too long, max 32 characters

Cancel

Confirm

Figure 18: Maximum Title Size

Are you sure you want to delete the card list:
Waiting for Merge

Confirm

Cancel

Figure 19: Titles fit in Confirm

Enter new list title:

Title cannot be empty

Cancel

Confirm

Figure 20: Non-empty Titles

Overview

test

Add Card List

Board name

Save

Board id : 1

Outer board color

#99cc99

Save

Reset

Inner board color

White

Save

Reset

Return to workspace

Leave Board

Close window

Custom Color...

Figure 21: Personalization Options