OOP Project Report – Group 44

Rebecca Nys, Matei Hristodorescu, Marius Frija,
Matthias van Vliet, Zico Krassenburg, Yulia Waarsenburg

1 INTRODUCTION

The aim of this report is to evaluate and improve the prototype of the User Interface of our Task List Organiser application. To achieve this goal we performed a Heuristic Usability Evaluation to gain a meaningful understanding of how our application prototype adheres to the 10 heuristics proposed by Nielson [1] and how we could possibly improve our User Interface design to produce a better user experience.

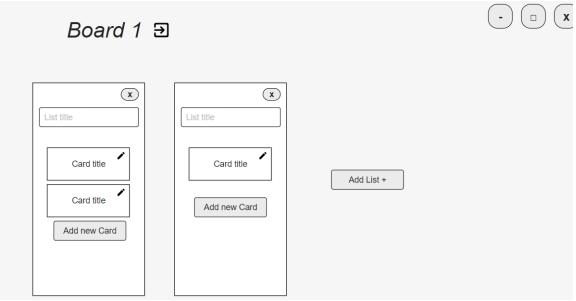

Figure 1: Sample overview of a board

1.1 *Task List Organiser*

The Task List Organiser, which we will from now on call TaLiO, is an application which allows users to make so-called boards on which lists of tasks can be added to. This serves the users as a useful tool for many purposes, for example to keep track of any tasks that should be done, are being done, or have been completed in the past.

Some of TaLiO's key-features are:

- The option to create boards which have a certain board title
- The option to add lists to existing boards which in turn have their own list title
- The option to add cards to existing lists which have their own title and description
- Built-in functionality to edit or delete any of the existing elements
- The ability to collaborate in real-time with other users to work together on the same board
- The persistence of boards, so that boards are saved in a database and can be retrieved by users at a later time

Figure 2: Login screen of the prototype

1.2 *Prototype*

The prototype that was used in this Heuristic Usability Evaluation has been produced using Moqups, an online tool for quickly making a working prototype of a GUI - a graphical user interface. It was largely based on the GUI of the application at that point in time. That way any issues that were found in the prototype could also be improved upon in the real application. The link to the designs we used for our evaluation can be found here [2]. However, we'll also give an impression of most states of the prototype with the following figures.

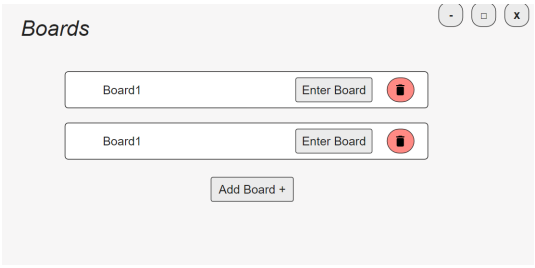

Figure 3: Board selection menu to select a board to view

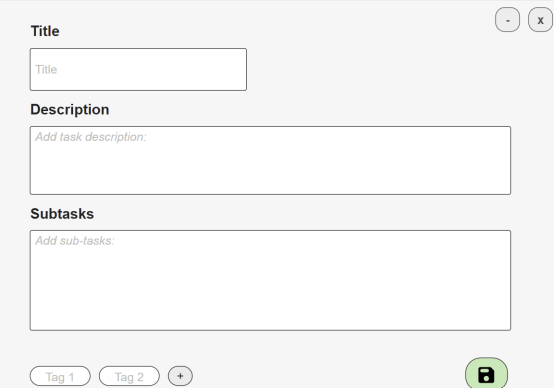

Figure 4: The overview of a single task card

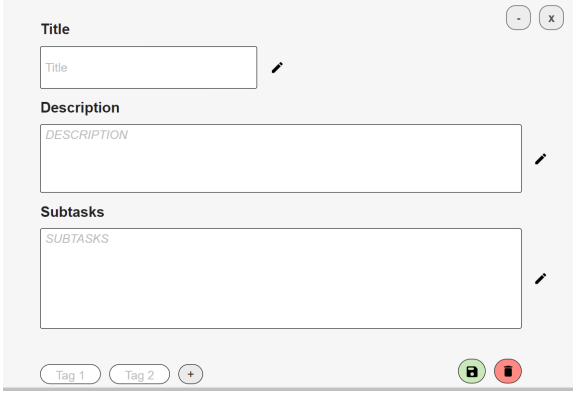

Figure 5: The overview of the edit menu of a single task card

Figure 6: Board before additional lists are added

Figure 7: A list containing cards

2 METHODS

2.1 *Experts*

A good heuristic evaluation is never done with only a single individual, because one person will never be able to find all the usability problems in an interface. Different people will find distinct usability problems, thus improving the evaluation significantly. Since Nielsen stated in his research that his recommendation was to include 3-5 evaluators [1], we conducted a meeting with another group of students, consisting of 5 people, to which from now on, we will refer as experts or evaluators. Their level of expertise is between novice and advanced. Each of the experts, performed the heuristic evaluation alone, since everyone's level of expertise, was above Fundamental Awareness (basic knowledge), we didn't have an active observer during the evaluation.

2.2 *Procedure*

For the heuristic evaluation we had the choice to either let the evaluators verbalize their comments to an observer or to let them give their findings via a written report. Since the group, we were doing the evaluation with, didn't have their evaluation yet prepared, we decided to do the evaluation with the aid of Google Forms. This kind of written report was the best suited for us, since it didn't require us to wait for the other team and to fully customize the kind of questions we wanted to ask the other team. Another benefit was that each individual evaluator could inspect our interface alone, so there wasn't any communication between them at all (we also specified to them, that they should answer the questions individually). In this way we ensured that there were unbiased evaluations.

2.2.1 Instructions.

In our Google Forms, we wrote an extensive instruction, on how the evaluators should use our prototype and that the mock-up they were presented with, was interactive, meaning that all the buttons, presented as elements of the GUI, triggered specific actions. We also provided instructions on the format we expected the evaluators, to answer our questions. For example, in the case of identifying any specific issue, the instructions informed the evaluators to structure their answers according to the following criteria:

(1) Give an elaborate explanation of the problem/issue you identified.
(2) State the anticipated difficulties that the user will encounter as a result of the problem.

(3) In which context may the problem occur?
(4) Give a description of the causes of this particular problem/issue.

We wanted to provide to the evaluators as little guidance as possible, since everyone had some experience with the application. In case that they needed some aid, we stated that they could use the "Hotspots" functionality of Moqups. We also stated, that this should only be used as a last resort.

2.2.2 Prototype.

Since we already had some functional scenes in our project, we decided to create a prototype, which would be the closest possible to the current state of our application. We made use of https://my.moqups.com to create the prototype. We only added some minor modifications to the slides, to show some more functionality which we decided to implement. The final result was that we provided our evaluators with 11 slides, each slide represented a different state of the application which would occur during its usage.

2.2.3 Steps.

The evaluators were instructed to access the prototype which was provided for the evaluation. In order to make the necessary assessments, they were informed to visualize each slide of the prototype either in the pre-established sequential order which best simulates the user experience of the final product or in any particular order they desired.

2.3 *Measures (Data collection)*

In our heuristic usability evaluation we are measuring how well our application follows Jakob Nielsen's 10 broad rules of thumb [1]. These are general principles for a user interface design. The 10 heuristics are listed below.

(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

Experts need to report any issues and improvements of our application which correspond to the heuristics listed above. We record their responses using a Google Form. We asked them to give our application a score ranging from 1 to 5 based on how does the application comply with each heuristic separately, a score of 1 meaning "Very bad" and a score of 5 meaning "Excellent", and an explanation of their rating.

3 RESULTS

3.1 *Overall description of the results*

The Heuristic Usability Evaluation was performed by a number of 5 evaluators. Therefore, we have obtained a complete set of results based on the previously described format (see 2.2.1).

3.2 *Compliance with the Heuristics*

According to 2.3, a set of score values has been collected for each of the heuristic principles. Each set of collected values was statistically analyzed, the mean value of the score rounded to 2 significant digits and its absolute error being presented. Therefore, the final results are as following:

- Visibility of system status: 2.8 ± 0.6
- Match between system and the real world: 4.4 ± 0.5
- User control and freedom: 2.4 ± 1.1
- Consistency and standards: 4.2 ± 0.3
- Error prevention: 2.8 ± 1.0
- Recognition rather than recall: 3.8 ± 0.7
- Flexibility and efficiency of use: 3.6 ± 0.9
- Aesthetic and minimalist design: 4.8 ± 0.3
- Help users recognize, diagnose, and recover from errors: 2.2 ± 1.1
- Help and documentation: 1.6 ± 0.7

Mean Score: 3.4

3.3 *Identified Issues*

For the vast majority of the evaluated heuristics, very similar usability issues were found by the evaluators, thus making it possible to aggregate the explanations into generalizing descriptions for each of the 10 Heuristics.

Identified issues with regard to the compliance with the Heuristic Usability Principles

(1) **Visibility of system status**
- *Explanation:* the user is not able to visualize all the available boards and switch between them while navigating a selected board
- *The resulting difficulties for the user:* the user would not have a clear overview of the contents of the application and would be forced to access the main overview in order to assess the available boards and access a particular one, thus losing the visual focus on the current working board
- *The context of occurrence:* during the regular use of the application
- *Causes of the issue:* no overview of the list of available boards and no means to switch to another one are provided in the scene corresponding to an individual board

(2) **Match between system and the real world**
- *Explanation:* the "Add new list" and "Add new card" dialogue windows do not have a clear and descriptive titles
- *The resulting difficulties for the user:* the user would get confused not knowing exactly what scene does he interact with
- *The context of occurrence:* the moment that the user wants to add a new list or a new card to the board
- *Causes of the issue:* no titles/labels are displayed for the "Add new list" and "Add new card" scenes

(3) **User control and freedom**
- *Explanation:* lack of the ability to revert the creation or the deletion of a task/list

- *The resulting difficulties for the user:* the user would have to re-add the deleted card/list which would be time-consuming, inconvenient and an unpleasant experience
- *The context of occurrence:* when the user would like to revert the deletion of a card/list
- *Causes of the issue:* the lack of the necessary implementations regarding reverting the deletion of an existent task/list

(4) **Consistency and standards**
- *Explanation:* the design of the cancel buttons for the "Add new list" and "Add new card" are not fully intuitive
- *The resulting difficulties for the user:* the users might be misled by the aspect of the close button, having the impression that it would close the application rather than canceling their actions
- *The context of occurrence:* when the user would like to cancel the action of adding a new card/list
- *Causes of the issue:* The lack of a specific "Cancel" button and the presence of "X" close button in the related scenes

(5) **Error prevention**
- *Explanation:* user is not asked to confirm the execution of certain accidental or unwanted actions
- *The resulting difficulties for the user:* the user would not be able to revert the effect of certain deliberate/accidental actions
- *The context of occurrence:* when the user accidentally deletes data elements, cancels certain actions or exits the application
- *Causes of the issue:* the lack of a pop-up confirmation window linked to specific high-risk actions

(6) **Recognition rather than recall**
- *Explanation:* there is not an extensive use of icons
- *The resulting difficulties for the user:* the user experience is not as intuitive and pleasant for inexperienced users
- *The context of occurrence:* the regular use of the application
- *Causes of the issue:* the reduced number of icons and visual elements used in the GUI

(7) **Flexibility and efficiency of use**
- *Explanation:* there are no existing shortcuts for user actions
- *The resulting difficulties for the user:* an experienced user would not get a more efficient and productive use of the application
- *The context of occurrence:* during the regular use of the application
- *Causes of the issue:* the lack of the implementation of shortcuts

(8) **Aesthetic and minimalist design**
- *Explanation:* the GUI elements do not cover the majority of the space of the application window
- *The resulting difficulties for the user:* the visual experience of the user would not be as pleasant
- *The context of occurrence:* the regular use of the application
- *Causes of the issue:* the inaccurate alignment and setting of the dimensions of the GUI elements

(9) **Help users recognize, diagnose, and recover from errors**
- *Explanation:* the user does not receive any error messages for attempting unpermitted actions
- *The resulting difficulties for the user:* the user would not understand the causes of the application errors which will negatively impact his experience
- *The context of occurrence:* the occurrence of internal errors based on the actions of the user
- *Causes of the issue:* the lack of pop-up text messages that would inform about a particular error

(10) **Help and documentation**
- *Explanation:* there is not any informational support for the user regarding the usage of the application
- *The resulting difficulties for the user:* the user could not receive any form of informational support if needed, leaving the user problems unresolved
- *The context of occurrence:* when the user encounters certain problems or does not understand specific features of the application
- *Causes of the issue:* the lack of any documentation regarding the operation of the application and of means of accessing it

3.4 *Prioritizing Severity Matrix*

The issues identified by the evaluators (see 3.3) were put into perspective by analyzing them relative to 2 parameters: the impact of the issue on the usability of the application (*impact*) and its frequency of occurrence (*frequency*). This way, we will assign each issue to a specific category based on the bound values of the 2 parameters: *Low* or *High*.

The criteria that were took in consideration when assigning a tuple of values to each issue are:

- The average rating score offered by the evaluators to the corresponding heuristic (see 3.2)
- The functional and non-functional requirements of the application (i.e. the backlog)
- The amount of resources required for solving the issues (determined by the team)

The processed data is displayed in a diagram, which is divided in 4 sectors based on the values of the parameters, each issue being labeled in accordance with the ordering of the issues in the section 3.3.

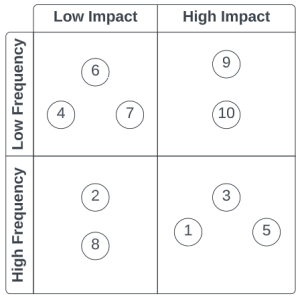

Figure 8: The Prioritizing Severity Matrix

4 CONCLUSIONS

The results from the Heuristic Usability Evaluation conducted by five evaluators showed that the compliance of the application with the ten Heuristic Usability Principles varied. The compliance with the principles of Match between System and the Real World, Consistency and Standards, and Aesthetic and Minimalist Design were rated highly. However, the principles of User Control and Freedom, Visibility of System Status, Error Prevention, and Help and Documentation were rated lower. The identified issues were relatively consistent across the evaluators, with descriptions and examples provided for each of the ten principles. Overall, the evaluation found that there were usability issues with the application that should be addressed to improve user experience and efficiency.

5 IMPROVEMENTS

Based on the conclusions and analyzing figure 1, the most significant improvements to the application's usability relate to heuristics 1, 3, and 5. According to section 3.4 these heuristics have a greater impact on the usability of the application as indicated by the higher frequency of occurrences in the results. Prior to demonstrating the improvements, figure 9 demonstrates the initial appearance of the board-overview.

Figure 9: Board overview-before

5.1 Visibility of system status

To incorporate the given feedback by the evaluators regarding the visibility of system status our new prototype includes an additional drop-down menu, where the client can switch between boards and immediately can see which boards are available without losing the visual focus on the current working board [3]. Figure 11 demonstrates the ability to choose & switch between different boards.

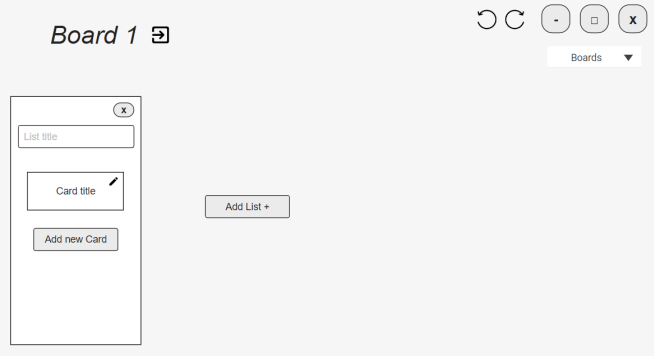

Figure 10: Improved board overview

Figure 11: Selection of different boards

5.2 User control & freedom

To enhance user control and freedom, additional undo/redo buttons will be incorporated into the board-overview. This feature enables users to reverse and/or undo their previous action. Figure 10 shows the improved board-overview, while Figure 12 illustrates a list with the same additional buttons.

Figure 12: List with additional undo/redo buttons

5.3 Error prevention

We also made improvements to the error prevention aspect of the application design. The design includes additional pop-ups that serve as extra confirmation when the user attempts to delete lists/cards, or wants to disconnect to the server. These pop-ups help to prevent accidental actions and greatly enhance the overall user experience. Figure 13, 14 show examples of a disconnection & deletion confirmation pop-ups which we included in the improved application design.

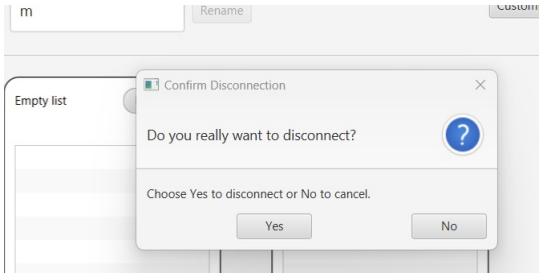

Figure 13: Disconnection confirmation

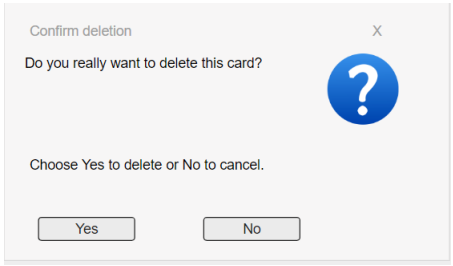

Figure 14: Delete card confirmation

Finally the improved application also includes some informational support regarding the usage of the application. We added an additional button to provide the user how the drag & drop in the application works (figure 15). This feature significantly enhances Help and Documentation

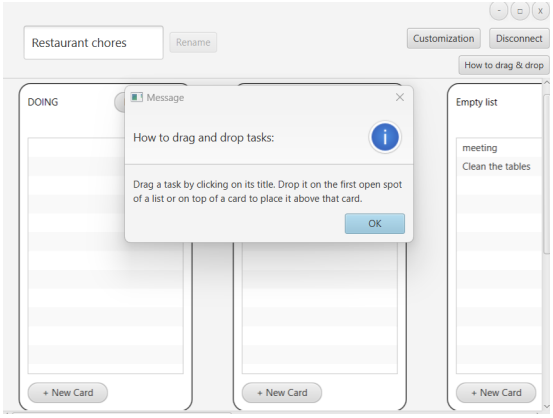

Figure 15: Disconnection confirmation

REFERENCES

[1] Jakob Nielsen. 1994. How to Conduct a Heuristic Evaluation. https://www.nngroup.com/articles/how-to-conduct-a-heuristic-evaluation/
[2] S.C Evercoder Software S.R.L. 2023. Prototype TaLiO before. https://app.moqups.com/rFhmAZHdjt95QffbaMcxEa3yX2TrJWDU/view/page/a81a2eca6
[3] S.C Evercoder Software S.R.L. 2023. Prototype TaLiO improved. https://app.moqups.com/B6a1ZlbIs3KlZI0N1CXDOy71m6sijvnj/view/page/a250b9965