OOP Project Report – Group 29

Alexandru Tabacaru, Dorian Herbiet, Edsard Grisel,
Matej Bavec, Rares Iordan, Teun Bosch

ABSTRACT

This report documents the process of utilizing Jakob Nielsen's usability heuristics to evaluate and improve the user interface of the Talio application prototype, an initial iteration of a to-do list management tool designed for task distribution within organizations. Our primary goal is to expose usability challenges and establish clear improvement actions to refine the application's user experience. The report first defines the starting prototype, which covers basic functionalities such as server connection, task organization in lists, real-time collaboration among multiple clients, and task and list creation. Expert evaluators from Group 22, including Mikołaj Gazeel, Vlad Dragutoiu, Matīss Bērziņš, and Bogdan Micu, individually tested the prototype following Nielsen's usability heuristics to ensure independent and unbiased assessments. They were provided with a detailed procedure and instructed to report any issues and offer recommendations for improvement. We analyzed the evaluators' feedback, identifying crucial usability concerns and prioritizing suggestions based on their severity and impact. The report presents the evaluation procedure, results, and implemented or planned modifications supported by heuristic principles. We conclude by outlining how and why we will improve our application based on the results and showcasing the revised UI design.

1 INTRODUCTION

Managing duties effectively and remaining organized is more important than ever in the fast-paced world of today. Our team developed Talio, a to-do list management solution, to help businesses streamline task distribution and increase productivity. Using Jakob Nielsen's usability heuristics, we set out to evaluate and enhance the Talio application prototype's user interface in this study. The goal is to pinpoint usability issues, design specific improvement strategies, and eventually improve the application's user experience based on evaluation results.

It is crucial to fully describe the prototype before starting the evaluation process. The client-server architecture of the Talio application is currently created utilizing Spring/JavaFX. A single board may be hosted by the server and used concurrently by all clients that are connected. The application has a single scene that displays a summary of all lists and the tasks assigned to them. Users may add, update, and remove lists, tasks, and labels from this overview. They can also drag and drop tasks to reorder them or transfer them across lists that already exist.

For addressing issues like timeouts, incorrect server addresses, or unexpected errors, there are various screens provided. The primary scene, which is the board overview, is displayed to the user after they have successfully connected to the server. Three to-do lists—to-do, in progress, and done—are first added to the board. At the bottom of each list is a "Add Task" button, which takes the user to a screen where they may enter the task's title, description, and labels. Each list also has a settings wheel at the top where users may rename or delete lists. The buttons to create a list or add a task are located at the bottom of the board and direct the user to the appropriate screens.

2 METHODS

We have recruited a team of 4 experts for our Heuristic Usability Evaluation by carefully selecting them based on their compatibility with our team's standards, commitment levels, and goals. They are experts by their formal education in heuristics and by their work on an application similar to ours.

The experts are seeing a prototype of our application, wherein we have instructed them to use the prototype as if they were new, yet technically competent, users who are completely unfamiliar with the project. To prevent redundant feedback, we also communicated the limitations of our project such as the lack of certain otherwise expected features to the experts. To open the application and gain full access to our prototype, they first needed to run the project and the server locally on their machine. This would grant them access to the server page, and they would be able to connect to their server by entering the word "localhost" or the machine's IP. Once these steps are complete the experts have unrestricted access to all of the features of our prototype.

Once the experts have gained full access to the prototype, they each look at the interface of the application and note down the issues they spot as they use the application. The fact that there are multiple experts is crucial as it will allow them to find the greatest number of issues that others may have missed. The experts use the Nielsen heuristics evaluation method to find potential problems with our user interface and app design. These heuristics include ensuring visibility of system status, using language that matches the user's understanding of the real world, providing user control and freedom, following consistency and standards, preventing errors, minimizing the need for user recall, providing flexibility and efficiency of use, using a minimalist design, helping users recognize and recover from errors, and providing helpful documentation. By applying these heuristics, designers can identify and improve usability issues and enhance user experience.

In the end, the experts will submit their report in the form of text feedback which will ideally subdivide their evaluation by different scenes in our project. For example, there may be a section for the login screen and a different section for the task creation screen. Under both sections, they will indicate usability problems which they have encountered, both aesthetic and functional.

3 RESULTS

For the "match between system and the real world" category, the experts would like some clearer ways to display error messages to users as they could find it difficult to find out what's wrong when it just displays 'Error, For input string: "' ' when no list name is given. Therefore it will be useful to either not allow empty names or

display an error saying the name can't be empty. Furthermore, not giving the user an idea of what sort of address format is needed can also confuse so perhaps adding a default address already provided in the input field will make it easier for users.

Regarding "user control and freedom", the experts felt that the user is limited in terms of what is possible in the applications home screen as renaming and deleting lists, reordering the tasks in a list, and editing tasks weren't functional. Furthermore, in the create task window, adding labels also didn't work. This meant that some of the core features of the application weren't functional, significantly hindering user experience.

In the "error prevention" category, an issue pointed out by the experts was that the app was prone to errors occurring due to missing information such as the board id which is required to create a list, however, it wasn't displayed anywhere so the user can't know that board id is 1. Without the board id, the user can't use the create list functionality but we are already working on removing the need for a board id input on list creation. Finally, title fields for tasks or task lists can be left empty which may confuse users as well as issues in code.

In the "recognition rather than recall" category there was only one issue found by Matiss. He noticed that the description of the task is no longer visible after the task's creation.

Regarding the flexibility and efficiency of use, the app has several issues. The first one, according to Matiss, Vlad, and Bogdan, is that on the login screen, when focused on the input and pressing enter, it doesn't automatically submit. Another issue discovered by Matiss is that dragging and dropping tasks works but it's overwritten by the old state (before the drag and drop). This has been fixed in the meantime. The final issue in this category is that, currently, to rename a list, a user has to use a drop-down menu, then click on rename. Mikołaj thinks that being able to rename the lists straight from the board screen would improve user experience.

The app has several issues regarding the aesthetic and minimalist design category. One important issue is that it doesn't resize properly on any of the scenes, which has been reported by everyone. On the Linux platform, the connect screen has a problem with the text not being displayed correctly, as noted by Matiss, Vlad, and Bogdan. Additionally, when creating a tasklist with a name that is too long, the TaskList itself becomes bigger, making a vertical scroll bar causing it to not be aligned with the others anymore, which has been pointed out by Matiss and Mikołaj. Similarly, according to Matiss and Mikołaj when making a task with a long task name, it overflows the card. Mikołaj also proposed a solution of either limiting the name length or making some system for displaying longer task names in a wieldy fashion. Matiss has also noticed that the green "Add Task" button in the board overview scene is redundant. In the login scene, the bottom margin is too big according to Mikołaj. Moreover, Mikołaj suggests that the create buttons would look nicer as plus icons, while the settings icon and down arrow look like two separate buttons, but they are only one, this could be the cause of confusion to the user.

In the category of "Help users recognize, diagnose, and recover from errors" during the heuristic usability evaluation, the following issue was found: the error message when a wrong board id was given for creating a list was unhelpful and outright confusing (the message received was «for input string ""»).

Furthermore, there were no issues found by experts in the "help and documentation" or "visibility of system status" category but we think that we should add a help button that shows a pop-up with instructions or FAQs to help the user navigate and use the app and that we should have some confirmation popups after certain events such as renaming or deleting.

In terms of how we are going to prioritize the fixes suggested, we will first make sure to fix the issues related to editing tasks and lists as these are essential for the user to apply this application and these sorts of comments were made very frequently by the experts. Our second priority is fixing the points made in terms of error prevention. Although they were not very frequently mentioned, we believe that these are issues of high severity and should be prioritized over "high-frequency low severity" issues such as issues with styling and resizing of the window.

4 CONCLUSIONS

To summarize, the Talio prototype's heuristic evaluation has led our team to gain a better overview of the application's usability issues. The primary conclusions from the results suggest that the most important factors for improving the user experience are concerned with error prevention, user control and independence, and aesthetic and minimalist design.

Based on the findings, we will make the following improvements to our application:

Improve error prevention by providing clearer error messages and addressing missing information, such as displaying the board id and preventing empty task titles. Increase user control and freedom by ensuring core features such as renaming and deleting lists, reordering tasks, and adding labels are functional. Fix problems with text display, task, and list organization, and resizing to improve the minimalist look. We will also take into account using more aesthetic icons for things such as settings and buttons.

These improvements, grounded in the heuristics and evaluation results, aim to create a more user-friendly and efficient application. The improved version is expected to be more responsive, provide better feedback to users, and ultimately, enhance the overall user experience.

In the revised UI design, we will incorporate the suggested changes, focusing on a cleaner layout, improved navigation, and more effective error handling. By addressing these concerns, we are confident that Talio will be better equipped to meet the needs of its users, providing them with an intuitive and smooth experience. This conclusion presents our final GUI design, which incorporates the proposed improvements. (insert photos of the design/application in the final version)