OOP Project Report – Group 61

Wojciech Graj, Yusuf Özdemir, Alex Despan, Paul Anton, Radu Andrei Vasile, Błażej Łytkowski

1 INTRODUCTION

Developing an app that will be used by people of different ages and backgrounds sometimes requires taking a step back and looking at the project prototype through the eyes of an independent evaluator, a person other than a programmer working on it. Therefore, following a set of principles, we conducted a Heuristic Usability Evaluation. We defined its objectives as follows: discovering a standard way users work in the application environment, measuring how well the app functions correlate with user expectations, determining how user-friendly the app is, and how well the design and styling were implemented.

The prototype that the evaluation was conducted on is an early functioning version of our application. It implemented the vast majority of the basic requirements and offered the evaluators a full idea of relations between scenes and objects like cards, card lists and boards. The prototype allowed the users to go through a standard sequence of performed actions including connecting to a server, navigating through the interface, browsing and modifying the presented data. The application presentation layer consisted of a basic layout design and primitive styling created mainly for clarity and simplicity while the project is in the development stage.

2 METHODS

2.1 Experts

In order to conduct the following evaluation, we have recruited a total number of 6 experts, all of them being students at TU Delft, enrolled in the OOPP course, which is mandatory for all first-year students of the Computer Science and Engineering faculty. Their level of expertise is intermediate, having little prior work experience, but demonstrating sufficient knowledge and skills necessary for this field, therefore belonging to the category of more experienced users.

2.2 Procedure

Prior to conducting the evaluation, we scheduled a meeting with the experts, instructing them to evaluate the interface and write down a report, considering a list of 10 usability heuristics. They are tasked with testing every feature of the application, and any questions that may appear will be answered promptly by a developer from our team.

The experts are given the most recent version of the program at the evaluation meeting, which is not completely functional but should have a comprehensive user interface with all functionality. To ensure correct feedback, all sections of the application's graphical interface are functioning, and all pages are linked to one another.

The experts are given an application prototype that precisely represents the final product, and they begin reviewing it on the first page, where they are prompted to connect to a server by inputting a server address. They are sent to a page where they can join a board after providing the correct address. If no boards are available, they may create their own by providing a name, a theme, and a password. If any boards are visible, the user can join them by inputting an invite key or link to an already created or joined board.

If a board is password-protected, the user can view it in read-only mode, in which they cannot change its state. In such circumstances, a warning message appears at the bottom of the page, informing the user of their permissions. If the user inputs the proper password, they will be granted full board access.

After successfully joining a board, the user is taken to the main screen, which includes a navigation bar with five options: disconnecting from the server, creating a new board, joining a board, adding a list, and changing the board's settings. The user views the board's card lists below the navigation bar, each with two buttons allowing them to add a card or change its settings.

Each list has many cards, which are displayed appropriately and can be edited or deleted by the user. Users can also drag and drop cards to other lists, and the changes will be reflected. This sleek and user-friendly layout allows users to easily navigate the application and maintain their boards.

Evaluators should go through the interface at least two times. The first evaluation is meant to provide the evaluator with a sense of the flow of the interface and the general scope of the system. Thus, on the next evaluation, he or she would be allowed to pay more attention to distinct interface components, while possessing more profound knowledge of the way these elements fit into the application. During these evaluations, the evaluator inspects several dialogue elements, comparing each of them with a list of recognized usability principles. Of course, the evaluator can implement his own custom heuristics in order to address specific situations.

In general, they are using Jakob Nielsen's 10 general principles for interaction design [1], which are:

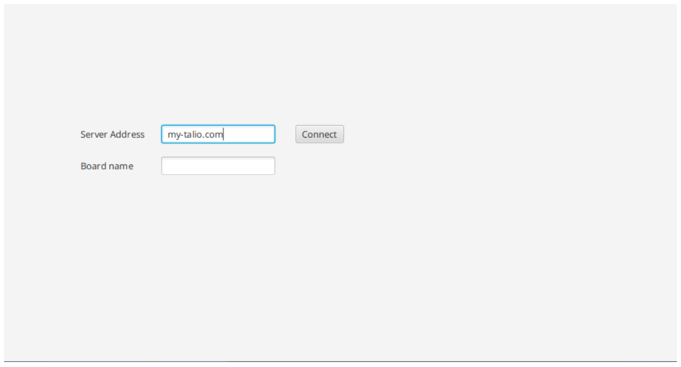
- Visibility of system status: Keep the user informed about the system's current status with appropriate and timely feedback.
- Match between system and the real world: Use familiar language and adhere to real-world conventions to ensure the design effectively communicates with the intended audience.
- User control and freedom: Provide a clear "emergency exit" option to allow users to easily discontinue actions.
- Consistency and standards: Follow accepted industry and platform conventions to prevent ambiguity or uncertainty for users.
- Error prevention: Take proactive measures to prevent errors from occurring rather than just providing effective error messages.
- Recognition over recall: Ensure critical elements, actions, and options are readily visible and accessible to reduce the cognitive burden on users.
- Flexibility and efficiency: Incorporate shortcuts for experienced users to enhance interaction efficiency and accommodate different levels of expertise.

- Aesthetic and minimalist design: Avoid including irrelevant or infrequently required information to maintain the prominence of relevant information.
- Help users recognize, diagnose, and recover from errors: Use clear and straightforward language in error messages, avoid technical jargon or codes, and provide guidance on how to address the issue.
- Help and documentation: Ensure the software is intuitive to use with easy-to-follow documentation readily available to users.

Measures (Data collection): Evaluators should report on problems and measure them, for example, on a severity scale. A report on a problem should have the following format:

(1) Brief and concise explanation of the problem
(2) The anticipated difficulties that the user will encounter as a result of this problem
(3) The specific circumstances in which the problem may occur
(4) An accurate description of the cause(s) of the problem

2.3 App presented to evaluators



my-talio.com

Figure 1: Connecting to server

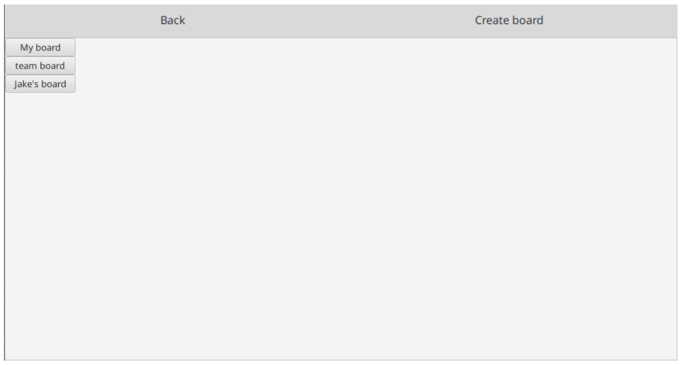
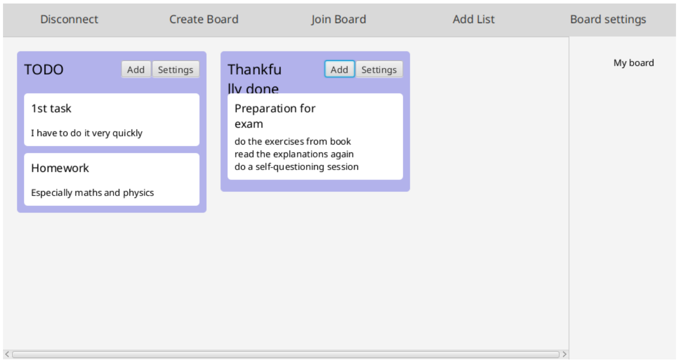
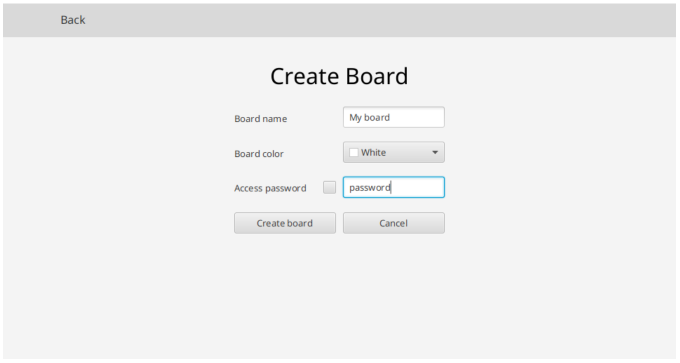


Figure 2: Joining board



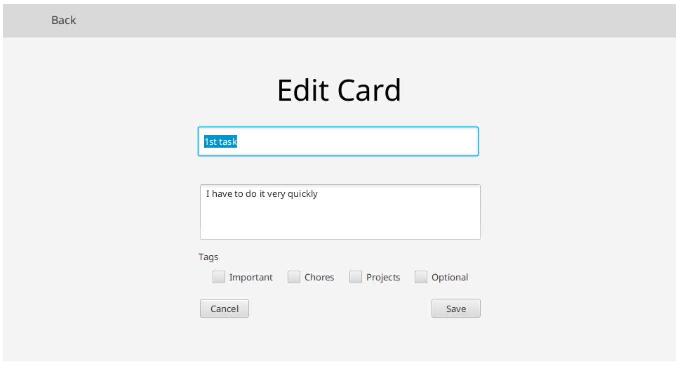
1st task
I have to do it very quickly
Homework
Especially maths and physics
Preparation for exam
do the exercises from book
read the explanations again
do a self-questioning session

Figure 3: The main view



Board name: My board
Board color: White
Access password: password

Figure 4: Creating new board



1st task
I have to do it very quickly

Figure 5: Editing a card

3 RESULTS

The evaluation of our product by another team was successful, as our demonstration showcased all aspects of the program and was quite thorough, thereby allowing the evaluators to locate a maximal amount of usability concerns. The evaluators compiled a list of feedback, which was grouped by the ten heuristics mentioned above. In order to process the feedback into a list of concise and actionable issues, we first split the feedback into individual points and then removed positive feedback as it did not point out problems that require solving.

To determine the priority of each usability concern, we employed a prioritizing severity matrix and placed each issue within the said matrix, allowing us to assign a severity based on the impact and frequency of the issue. Unfortunately, we did not receive feedback from each individual, but instead from the entire group, so we estimated both the impact and frequency ourselves. This yielded the following table of issues:

No.	Usability Concern	Heuristic	Impact (1-3)	Frequency (1-3)	Severity (1-4)
1	No customization or personalization	Flexibility and efficiency of use	2	3	3
2	After connecting to the server, it is unclear if the user should join a board	Visibility of system status	2	3	3
3	No feedback when transitioning between scenes leads to confusion about actions that occurred	Visibility of system status	2	3	3
4	No keyboard shortcuts	Flexibility and efficiency of use	2	3	3
5	Users cannot change the name of a board after creation	User control and freedom	2	2	2
6	Buttons to modify lists and tasks have an inconsistent style	Consistency and standards	1	3	2
7	Button to modify task is difficult to see	Recognition rather than recall	1	3	2
8	Error messages are too technical	Help users recognize, diagnose, and recover from errors.	2	1	1
9	Checkbox for board passwords is redundant	Aesthetic and minimalist design	1	2	1
10	No documentation about keyboard shortcuts	Help and documentation	1	2	1

As can be seen in the above table, there is a fairly equal distribution of severities in the 1-3 range, and there are no concerns with the highest severity. Additionally, while there aren't any heuristics that are particularly over-represented, we noticed that certain heuristics tended to have higher severities than others. As such, we counted the number of usability concerns for each heuristic and weighed them with their severity to identify the most problematic areas of our program. Heuristics #1 (Visibility of system status) and #7 (Flexibility and efficiency of use) proved to be the biggest areas of concern. '

4 IMPROVEMENTS

4.1 App after the changes

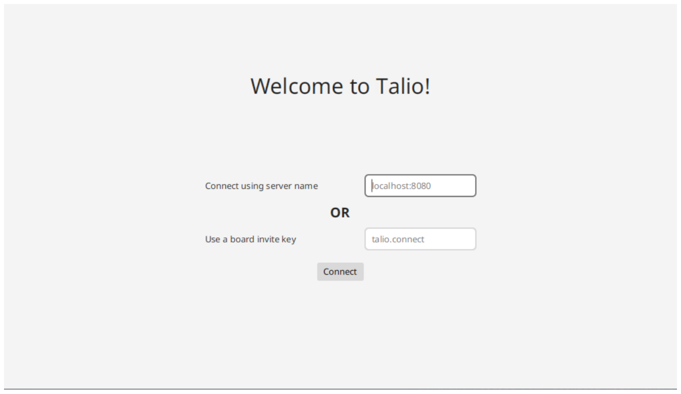


Figure 6: Connecting to server

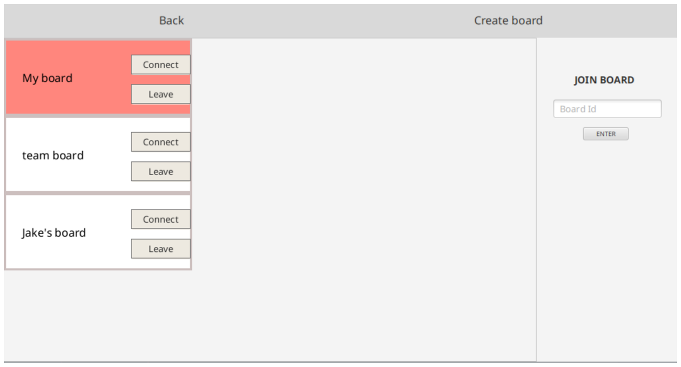


Figure 7: Joining board

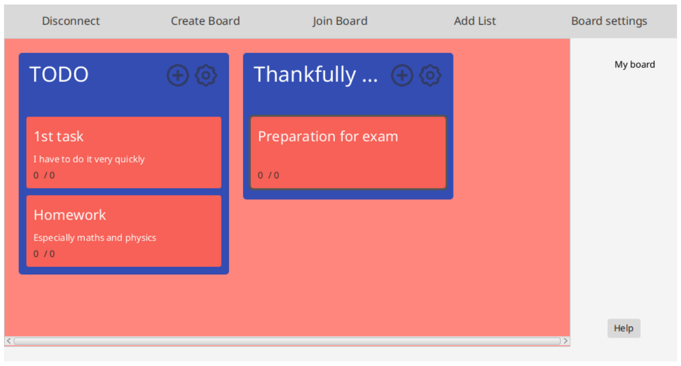
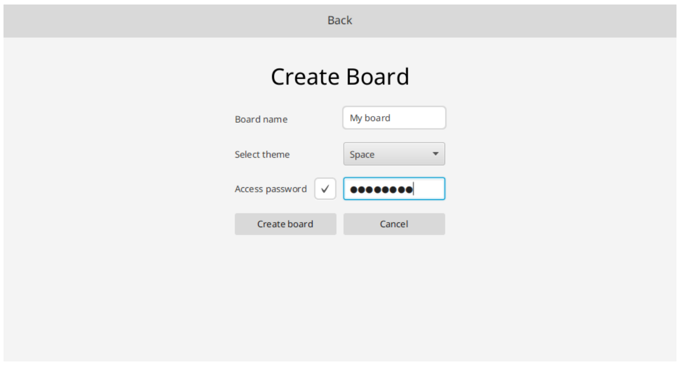
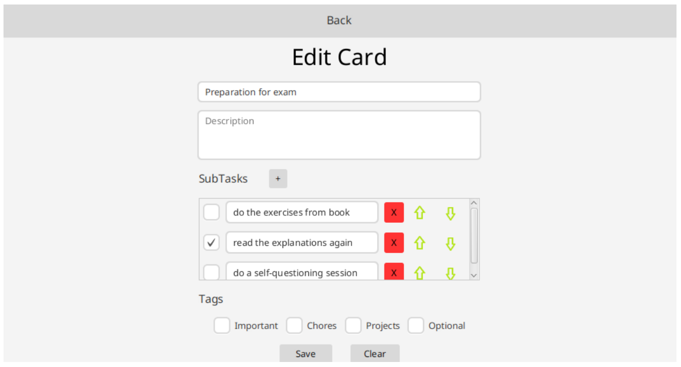


Figure 8: The main view



Board name: My board
Select theme: Space
Access password: ✓ ●●●●●●●●

Figure 9: Creating new board

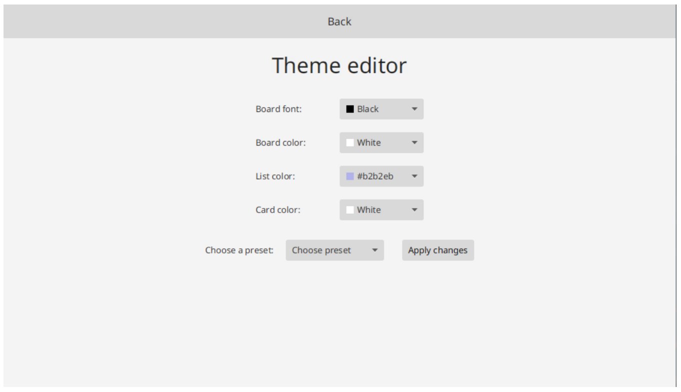


Preparation for exam
Description

- [ ] do the exercises from book
- [✓] read the explanations again
- [ ] do a self-questioning session

Tags: Important, Chores, Projects, Optional

Figure 10: Editing a card



Board font: Black
Board color: White
List color: #b2b2eb
Card color: White
Choose a preset: Choose preset

Figure 11: Theme editor

4.2 Styling changes

The heuristic evaluation showed us that our app before the changes was inconsistent and lacked customization in many areas. The evaluators pointed out several flaws in various heuristic categories, ranging from user comfort to aesthetics. We linked the observed problems with concrete elements of our app and introduced changes modifying their looks and behaviour.

The heuristic we took a look at first was "Buttons to modify lists and tasks have an inconsistent style" motivated by inconsistent style across our app. The buttons were really only a hint of irregular style across each of our scenes. In order to solve it, we created only a few general-purpose styles for components such as the navigation bar, buttons and input fields, which we applied throughout the whole application. We also strived to keep to similar design principles from one component to another by for example mostly using rounded corners (seen in Fig.9) and toned colors.

The next one on our list was "Button to modify task is difficult to see". We decided that they should also adhere to the set of styles we just created and found appropriately looking icons to represent them. When hovering over the icons, they change opacity which helps the users notice them and understand their role.

The evaluators mentioned the error messages were too technical. In order to make the errors more user-friendly we've implemented custom messages in multiple instances, such as when editing a card that was deleted or creating a board with an already taken name.

Initially, our prototype lacked any customization options. However, we've made significant improvements by introducing a centralized Theme Editor (Fig.11), which can be accessed through the Board Settings menu. This editor enables users to modify the color of their cards, card lists, and the board, as well as the font color. To make things easier, we've also included several pre-defined themes, including "Dark Theme," "Cherry Theme," and "Default Theme," which users can quickly choose from.

4.3 Functional changes

Our evaluators also noticed a few occasions on which it was inconvenient or not obvious how to carry out some tasks in the application. Some of the things users expected as granted weren't even possible like changing the board's name or modifying their appearance. After discussions regarding those problems, we agreed on and implemented several new functionalities and improved many others.

Our prototype did not have keyboard shortcuts, which prevented the fast-learning users from speeding up their workflow. Wanting to address this nagging issue and create a friendly user experience, we implemented a couple of very important ones. We included shortcuts like swapping the cards in one list, editing them and generally controlling the interface.

With the creation of shortcuts, we also introduced a help button (visible in Fig.8), mentioned by "No documentation about keyboard shortcuts". We thought that such a feature is essential and with a low effort from our programmers' side, it greatly benefits the user.

We noticed that the user could be lost when first opening the app. In order to improve the user's first experience, an explicit Join Board section (in Fig.7) was added to the platform's user interface, enabling him to easily create and manage their boards. Additionally, displaying previously joined boards on the left provides him with quick access to their existing boards on upcoming Talio usages.

Another change the evaluators mentioned is removing the password tickbox on creating or editing a board (Fig.9). After carefully discussing this subject with our team, we've decided to exclude this heuristic from our improvements, since we've agreed the tickbox would make it clear for the user that they have to enter a password.

The last piece of feedback we tackled was "No feedback when transitioning between scenes...". We decided that with the new colorful customization options and with the other scenes clearly labelled with a big title, not many changes are actually needed. Our only modification was adding a big "Welcome to Talio" message in our Server Join scene (Fig.6) to improve the user's first experience.

5 CONCLUSIONS

Consequently, the evaluation from the other team provided us with an opportunity to improve our product. The team has identified several areas for improvement, including the visual settings, system visibility, keyboard shortcut implementation, the cohesiveness of the edit button with the list style, and the clarity of error messages. The team will make the necessary changes to address these issues and ensure that the client can clearly see and understand the board on which they are operating. Overall, these enhancements will improve the user experience and make the application more manageable to use.

REFERENCES

[1] Jakob Nielsen. 1994. 10 Usability Heuristics for User Interface Design.