OOP Project Report – Group 5

Daniel Rachev, Rūta Giedrytė, Una Jacimovic, Konstantin Kamenov,
Antonio Lupu, Alexandru Fazakas

1 INTRODUCTION

The following report describes the findings of the heuristic evaluation of our developing application, TALIO. Having already worked for quite some time on developing and familiarizing ourselves with the software, we have become oblivious to some otherwise glaring problems or inconveniences which a novice user may encounter. As such, we have decided to conduct an evaluation with reviewers who were previously unfamiliar with the application, with the main objective of collecting feedback on the usability and consistency of the program.

The software we have given our evaluators to review was a "barebones", yet functional prototype of our future product. The principles behind the prototype were the same as those of the anticipated release. When opening the application you are prompted to choose a server to connect to, a username and an (optional) admin password - this can be seen in figure 1.

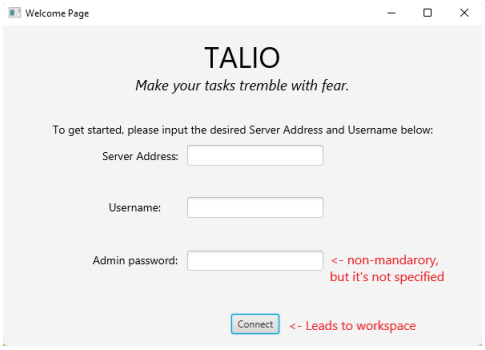


Figure 1: The Welcome Page

The user is then redirected to a "workspace" (figure 2) where they can create, join, open and leave boards. This screen also has an option to disconnect, which redirects the user to the previous "welcome" page.

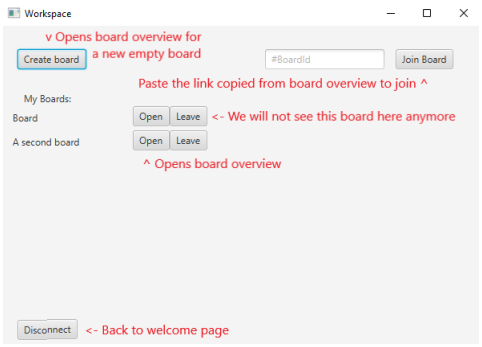


Figure 2: The Workspace

Upon opening a board (figure 3), the user has the option to add, edit and remove lists and cards to their liking. The user will also see any previously created lists and cards when reopening the boards. They also have the possibility of copying an invite code to share with other users so they can also connect to their board.

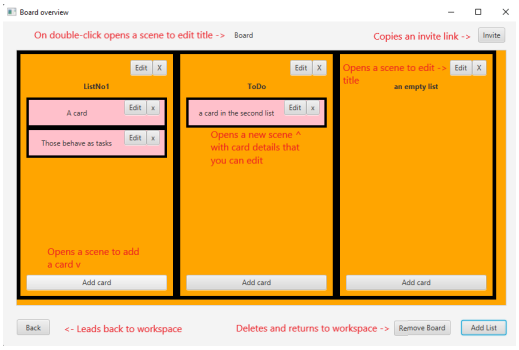


Figure 3: The Board Overview

2 METHODS

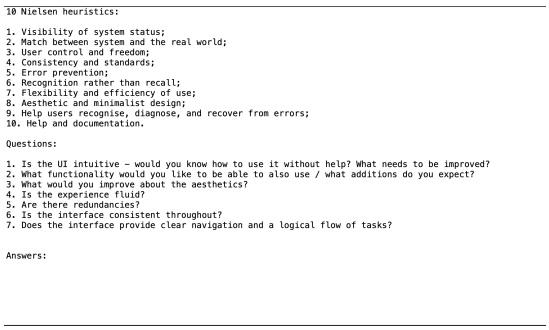
To acquire feedback, four experts were recruited. They were emerging software developers, studying computer science at the Technical University of Delft. We organized a meeting where all four of them could evaluate our team's application demo. Prior to the meeting, we informed them that we were seeking their feedback based on Nielsen's ten heuristics.

When we met, before they had started, we informed them that they would get to try our application on their own, but under supervision by one of our team members. Together with the working demo, we provided them with the list of ten heuristics inside the text document. The document also included a list of questions which were supposed to serve as guidance in their evaluation, besides the list of ten heuristics, and additional space where they were able to answer the questions and write their thoughts which can be seen on the figure 4.

However, they were told that before or after answering the given questions, they should also add anything they believe is important to note that may violate points from heuristics. They were able to directly type their thoughts and answers to the document while using the application. This way we tried to make sure nothing gets forgotten.

However, they were told that before or after answering the given questions, they should also add anything they believe is important to note that may violate points from heuristics.

For the demo of our application, they were given the starting window from figure 1. Experts were also told that at the moment they only have one server available to connect to which is the



```
10 Nielsen heuristics:

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 recognise, diagnose, and recover from errors;
10. Help and documentation.

Questions:

1. Is the UI intuitive - would you know how to use it without help? What needs to be improved?
2. What functionality would you like to be able to also use / what additions do you expect?
3. What would you improve about the aesthetics?
4. Is the experience fluid?
5. Are there redundancies?
6. Is the interface consistent throughout?
7. Does the interface provide clear navigation and a logical flow of tasks?


Answers:
```

Figure 4: Text document

localhost on port 8080. Further goals given by the supervisor were the following:

(1) start the application by connecting as a user
(2) create new board
(3) create new list
(4) create card inside the list
(5) rename the board
(6) rename the list
(7) edit the card
(8) move card inside and outside the list, in another words drag and drop the card
(9) try to join existing board

However, the supervisor did not give instructions on how to do so. Experts were expected to enter the server address by the information on the available server, and the username of their choice. Moreover, they were expected to gain understanding that one username relates to one user, which can be started by any other expert entering the same username. After the instructions, they were left to perform actions on their own, further testing what they did previously.

At the end of the evaluation, we collected their answers from the text documents and unified them in one document where we performed analysis. The goal was to measure how our application aligns with all of the ten heuristics, as we aim to create user-friendly and intuitive applications. Additionally, we wanted to see what was common among the evaluators. Therefore, we split the answers according to the violated heuristics as we wanted to have a clear understanding of what needs to be improved and what our priorities should be.

3 RESULTS

After conducting a heuristic evaluation of our prototype, we received valuable feedback from the four evaluators. One of the main feedback points we received is that some parts of the system are unintuitive or not well explained. Additionally, some parts of the application are not aesthetically pleasing. The following problems were selected based on how many evaluators noticed them during the heuristics evaluation process – for example, the first and most important problems were found by all four of the four evaluators, and the latter ones were noticed by fewer evaluators. We decided to target the following specific set of problems and leave other ones out, based on these criteria. We left out some of the problems found by just one evaluator so that we could focus on the ones that were the most noticeable and that affected the flow of the app the most. We consider the problems that we left out to be more "superficial", not needing as much attention since they don't directly and heavily impact the usability of the app.

A list overflows when many cards are added to it, and it does not support scrolling. This violates the "Consistency and Standards" principle. Users expect that they can scroll through a list when the number of items exceeds the visible space. Inconsistent behavior can create confusion and frustration for users. This can be seen when in the board overview, we add too many cards. Since we did not make the lists scrollable, it does not function as expected.

A mechanism that prevents lists with no names or cards with no names to be created does not exist. This violates "Error Prevention" since without feedback, users may not realize that they have left a required field blank, leading to errors or confusion. For example, when you click the "Add List" button, a new scene pops up, and you can choose to leave the name empty, which should not be possible.

Drag and Drop does not work intuitively enough. The evaluators stated that when trying to drag and drop a card from the middle part – where the title of the card is – some issues would appear. Additionally, when trying to drag and drop a card to a list that is unreachable (because of the scrolling), the app does not support scrolling. This violates the "Consistency and Standards" heuristic since the application should allow the users to manipulate objects in a way that is intuitive.

The functionality for renaming a board is hard to find. This violates the "Visibility of system status" because the system fails to provide clear information about the board renaming functionality. Therefore, it can lead to wasting time searching for this particular functionality. In our application, a board is renamed by double-clicking on its title, in the board overview. But this functionality is not explained or suggested anywhere, so the user might have difficulties grasping it.

There is no label that explicitly says that the admin password is optional. This problem violates "User control and freedom" since the system does not provide the users with enough information so that they have full control and freedom over their actions. Our application does not specify explicitly that the admin password can be left blank and is therefore optional.

Joining an existing board is not self-explanatory. "Recognition other than recall" is violated, because the system requires users to remember or figure out how to join a board. Additionally, the 'Invite' button in the board overview is not suggestive enough. The evaluators stated they didn't know the invite was supposed to give the board id since the board id is not visible. This increases frustration and decreases usability and user experience.

The current color palette is not appealing. This violates the heuristic of "Aesthetic and minimalist design" because the system does not present an aesthetically pleasing color scheme. For example, our board overview's color palette seems rushed and not thought through, which is not appealing to the users.

When editing card details, the old values are not autofilled, therefore you have to type everything again even for changing one word. "Flexibility and efficiency of use" is violated because it makes the process of editing cards slower and more cumbersome.

When editing a large number of cards without the auto-filled old text, you have to retype lots of text which can be difficult and time-consuming.

4 CONCLUSIONS AND IMPROVEMENTS

The input that we got from the four experts when doing the heuristics evaluation has been an eye-opening experience for our development team. They helped us reflect on our process and some of the flaws it could have as well as the application itself. The main problems that our evaluators found can be summarized as follows:

(1) **Some parts of the application are not intuitive and lack explanation.** This problem, in our opinion, partially stems from the backlog being too specific on some occasions and too general on others. We are trying to follow it as closely as possible even if we agree that some of the features are not intuitive (like double-clicking to edit). The admin and password functionalities on the other hand lack some context so we ended up improvising and getting a lackluster result that is hard to use. We as developers spent a lot of time discussing our features. Therefore, we rarely find them hard to use since we have mutually agreed on how they should work. However, for a random user, it could be a daunting task to work with our app.
(2) **The application is not aesthetic.** The sources of problem number two are some of our coding and testing practices. The orange color, which was disliked by experts, was supposed to be there just for testing but it stuck there as a part of the product in the end. All layout-related issues are a result of us developing the interface in a rush, focusing more on functionality. And finally, errors being displayed in different and sometimes unhelpful ways happened because it was never discussed how the user should be notified of errors and everyone put in whatever they found fitting.

Concerning **the former issues**, the team has decided to implement the following changes concerning the heuristics and the results from the evaluation.

(1) A scrollbar will be added to each list of tasks since the evaluators reported difficulties when adding multiple cards. Previously, multiple cards would overflow the list and even cover the add list button. This is going to bring the application more in line with the heuristic *"Visibility of system status"* and will allow users to have a greater number of tasks for a specific list.
(2) The team will add labels containing warnings, e.g. "List name cannot be left blank.", which will indicate to the user that it is required to fill in a certain field. Before, users either did not have any way to know why the system would not let them create a new list or card or they would get a popup with a http error. As a result of the changes, there would be less ambiguity when using the application and the *"Error Prevention"* heuristic will be better enforced.
(3) A feature is going to be implemented such that the mouse cursor will change when hovering over the name of a board as it is currently challenging to identify how to use the rename functionality - no indication is given. This will assist future users in working with the application more easily and it will improve the *"Visibility of system status"* heuristic.
(4) Additionally, more descriptive text will be added to the welcome page as the evaluators reported that it is not properly relayed which fields are mandatory (no explaination was given in the app). Confusion regarding input will be minimized and the *"User control and freedom"* heuristic will be followed in a better manner.

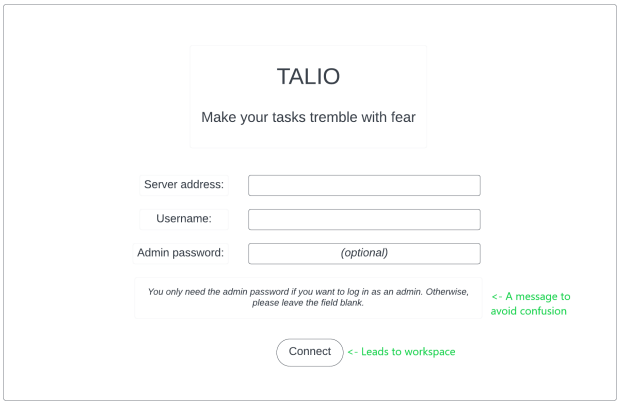


Figure 5: Improved connect page

With **second order of priority**, the developers will append the full board name and id to the invite text label since currently too much is expected of the user to utilize the functionality. This would abide by the *"Recognition other than recall"* heuristic and allow for the user to have an overall more pleasant experience. Furthermore, the team is going to create a more visually attractive design for the application as the present version was considered displeasing. As a result, the *"Aesthetic and minimalist design"* heuristic will be improved upon and users will enjoy engaging with the product more. Next, a card's previously stored title and description will be automatically filled in the corresponding input fields inside the "edit card" scene since it is not user-friendly to ask users to provide the whole text, especially for small changes. This would align the application better with the *"Flexibility and efficiency of use"* heuristic and would simplify users' workload.
Finally, it has been decided that the application will not be resizable in the future and that the size of the window for each separate scene will stay consistent. The *"Aesthetic and minimalist design"* heuristic will be more properly upheld and users view the product as more visually pleasing while working. Moreover, specific buttons, e.g. "cancel", "back", and "save", are going to stay in the same positions, when appropriate, relative to the different windows. This strongly enforces the *"Consistency and standards"* heuristic and will reduce potential frustration for users. Then, the team is going to implement a way for text fields to be cleared when changing scenes as currently, it could lead to undesired results. Consequently, the *"Error prevention"* heuristic would be improved upon and the chance of users making mistakes would decrease. Lastly, the evaluators reported that error messages are not helpful enough, which is going to be addressed by describing in greater detail the potential issues

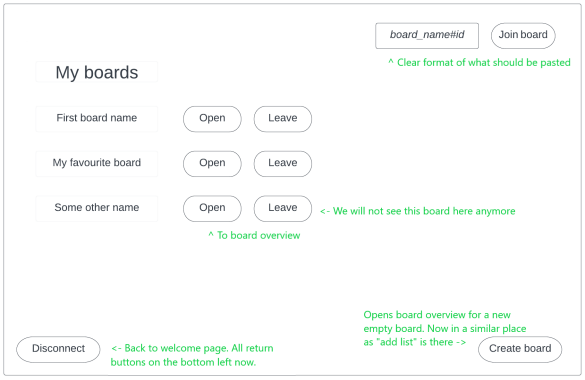


Figure 6: Improved Workspace

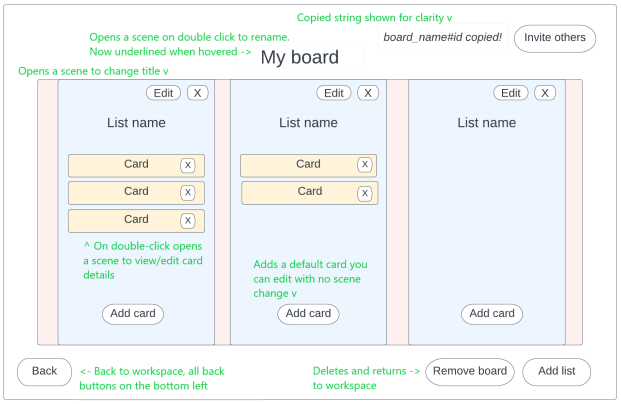


Figure 7: Improved Board Overview

a user might encounter. Naturally, the application would resonate more with the "Help users recognize, diagnose and recover from errors" heuristic and would provide a more pleasant experience to users in general. Figures 5-7 reference our updated vision for the application's interface and functionality, taking into account all of the planned improvements.

Overall, the heuristic evaluation of our team's Talio prototype has highlighted some key areas for improvement. It is essential that we prioritize the identified issues to ensure that our application is intuitive to use, efficient and aesthetically pleasing.