OOP Project Report – Group 57

Augustinas Jučas, Zhuoyue Ge, Radu Burchel,
Atanas Donev, Simona Lupşa, Yashaditya Sehgal

ABSTRACT

1 INTRODUCTION

1.1 Evaluated Objective

The purpose of the heuristic evaluation is to identify, evaluate and possibly solve problems that are present in the front-end design of the Talio application.

Evaluators will receive the design of the application, created using "Figma", an online tool used for mock-ups and prototyping the design of an application. The prototype is not functional - it is only composed of images which should resemble the front-end design of the application. Therefore, mock-ups used for this heuristics report are inserted:

1.2 Evaluated Prototype

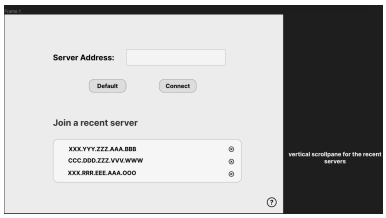

Figure 1: Connect to server

The server connection frame is the first displayed thing when the application is started. Here the user can choose a custom server URL or proceed with the default option. If the provided URL is not valid then an error message is shown.

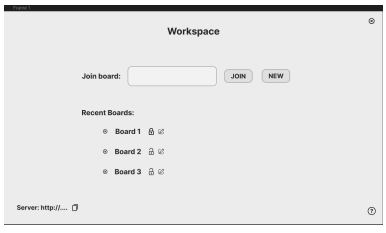

Figure 2: Workspace

Upon successful connection, the workspace frame is displayed. From here the user can choose to join a recent board, join new board or create one.

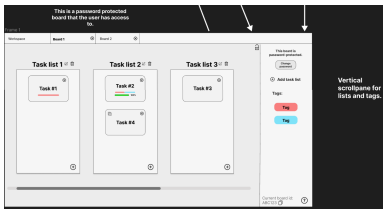

Figure 3: Password protected board with access to it

Boards can either be password protected or not, meaning that some boards require a password to be changed. Password is not required to view the board.

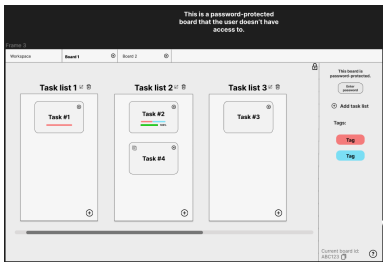

Figure 4: Password protected board without access to it

If the user wants to edit a board without entering the password. Then an error message shows up.

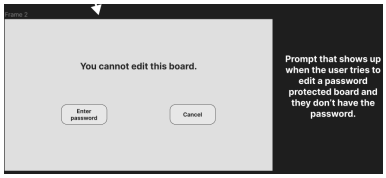

Figure 5: Error message

This error message informs the user that they do not write permissions unless they provide a valid password.

Figure 6: Remove password

Of course, boards can be unlocked. This may sound ambiguous, however, unlocking a board means removing its password and making it available to be edited by everyone.

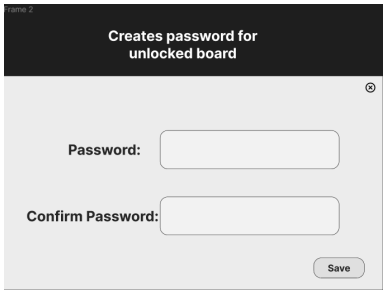

Figure 7: Create password

Respectively, boards can be locked via the same principle. The only thing a user needs to do is create a password to lock the board.

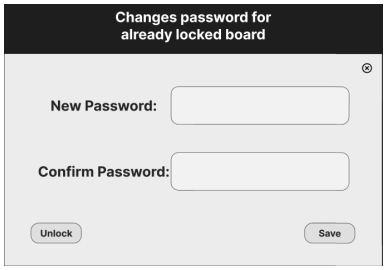

Figure 8: Change password

If the user desires to change the password of the board, that is also possible

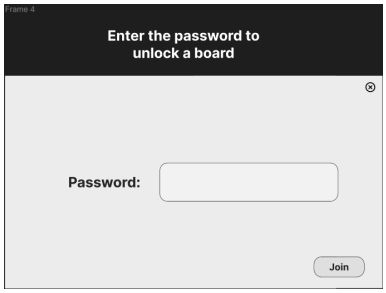

Figure 9: Unlock board

Of course, boards can be accessed and edited without removing the password. However, the rights to edit a board remain only while the user has the board opened. Should the user close the board, they need to unlock it again via password.

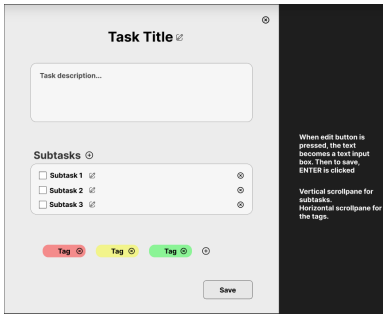

Figure 10: Task view

Whenever a task is clicked on, a popup is shown which enables the user to edit the task. The design is simple and allows the user to change the title, description, tags and sub tasks of the selected task.

Figure 11: Editing sub tasks view

Furthermore, the description of the sub tasks can be changed via the edit button next to them. Upon clicking the edit button, it enables the sub task to be edited and saves the sub task if clicked once again.

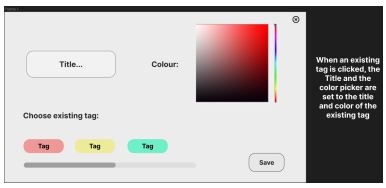

Figure 12: Add/Edit tag

Tags can be created and edited. The color is chosen via the color wheel. Moreover, existing tags are displayed from which the user can choose.

2 METHODS

2.1 Experts

For the heuristic evaluation, five experts were selected. All of them are first-year Computer Science and Engineering students at TU Delft and took part in a Heuristics Evaluations course, taught by an expert in the field. Furthermore, at the time of questioning, all selected students had been working on the same project for a month. Therefore, all of them had seen various versions of the design that they had to evaluate.

Due to the aforementioned circumstances, the selected students qualify as a suitable choice for this evaluation. Furthermore, they demonstrate an adequate level of expertise in the field of web-page usability.

2.2 Procedure

The following procedure was followed for each of the five evaluators. A single observer, which is part of the developing team, was assigned to each evaluator. The expert was given a short presentation of the way the evaluation will be handled. They were told that they will be given only the non-functional design of the application and were presented with the list of heuristics that they would need to assess the prototype against.

Here we provide the heuristics that were presented to each evaluator. They are the heuristics used by Nielsen [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 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.

Evaluators were presented with the format which they would use in order to report their findings. We present this format in a later subsection of the report.

Each expert was interacting with the prototype from inside of the "Figma" project. They were only analyzing images, which had no functionality. Therefore, the role of the observer was to walk the evaluator through the flow of the application. This interaction between evaluator and observer allowed for more flexibility, as an action that would be difficult to show through images can be verbally explained by the observer, instead of showing which scene would be loaded.

At first, the evaluators were instructed to briefly pass through the interface, in order to get familiar with it. During this first glance, some problems would already be identified and written down, but that was not the main goal of this interaction.

After that, the experts were instructed to be more rigorous and start looking for problems in each scene separately. They were instructed to first look at the big image and look for basic errors and then get into the details of each scene - separate buttons, text fields, etc. Each time an inconsistency against the provided heuristics was found, the evaluator would describe the problem to the observer, and the observer would note it down in the format that had been discussed earlier.

Depending on the evaluator, the whole process would last from 45 minutes to 1 hour and 30 minutes.

2.3 Measures (Data collection)

For each of the problems found, the evaluators were supposed to name the following:

(1) Problem description
(2) Likely/actual difficulties
(3) Context of the problem
(4) Assumed causes

This specific list is used in order to more easily identify when two experts are reporting on the same problem. It was introduced by Cockton, Woolrych, Hall and Hindmarch

Before the meeting with an expert, each observer had prepared an empty table. The columns of that table were exactly the 4 items in the list. And for each new problem an expert found, a new row was inserted into the table.

After all experts were questioned, a document with all tables was compiled. That was exactly the raw data that was used for this heuristic evaluation.

3 RESULTS

3.1 Filtering and evaluating the results

After all five evaluators were questioned, 36 potential problems were reported. Some reports were duplicates, since some people reported on the same problems. Therefore, duplicates had to eliminated. This job was greatly simplified by using the specific questioning format, described in section 2.3.

Also, most of the reports were shortened and simplified, so that the list would be more concise and presentable.

All results were evaluated using a impact-frequency metric - for each issue identified by the experts, a score 1-7 was assigned for the impact (m_i) and the expected frequency (f_i) of the problem. Also, some issues that were named by the evaluators, were identified as false-positives.

After this, a score was calculated for every non false-positive problem. The score was calculated using the following formula:

$$score_i = m_i + f_i$$

,

where m_i denotes the impact score and f_i denotes the expected frequency score of the ith problem. The intention in using this metric is to evaluate the problems in such way, so that the problem with the highest score would be the most severe one.

Then, all problems were sorted by their score in descending order, i.e., they were prioritized from most to least severe.

3.2 The final list

The compiled list of non false-positive problems, that was received from the users is as follows:

(1) A button for closing the workspace is available, when such functionality should not be allowed.

Score - 12. Impact - 7, frequency - 5.

(2) It is unclear how password protection works - there are both the "Unlock" buttons, and lock/unlock icons, which is confusing.
Score - 11. Impact - 7, frequency - 4.

(3) It is impossible to specify (or change) the name of a given board.
Score - 10. Impact - 5, frequency - 5.

(4) When the workspace is selected from the board overview, the tab pane disappears.
Score - 10. Impact - 6, frequency - 4.

(5) Displaying completed subtasks in the overview scene is not user-friendly because it is implemented using percentages instead of showing the number of completed tasks.
Score - 9. Impact - 4, frequency - 5.

(6) It is impossible to return to the server connection screen from the workspace without closing the application.
Score - 9. Impact - 7, frequency - 2.

(7) The application does not allow the user to delete a board from the server.
Score - 9. Impact - 7, frequency - 2.

(8) The actual "Help" scene is not presented in the design, even though such functionality is implemented.
Score - 8. Impact - 5, frequency - 3.

(9) When a user deletes a task list, no warning is given and it is immediately deleted. This can cause a user to delete a task list by accident.
Score - 8. Impact - 7, frequency - 1.

(10) In the board overview, it is unclear what the lock icon is supposed to do.
Score - 7. Impact - 4, frequency - 3.

(11) In the board overview scene, it is not obvious how to select a task and get to it's details. There is no explicit button for that, and it is not self-evident that the box of the task should be clicked.
Score - 7. Impact - 6, frequency - 1.

(12) It is hard for the user to figure out how to unlock a board.
Score - 7. Impact - 6, frequency - 1.

(13) Different tags can have the same description and the same color. This could be confusing for the user.
Score - 6. Impact - 3, frequency - 3.

(14) In the overview if too many tags were created for a task, there would be visual problems.
Score - 6. Impact - 5, frequency - 1.

(15) In the task view scene, when creating new subtask the "Save" icon can be misidentified as a checkbox.
Score - 6. Impact - 4, frequency - 2.

(16) The button that adds a new task list is relatively hard to find. It is not obvious where it is, even though it is the most important button in the board overview.
Score - 5. Impact - 3, frequency - 2.

(17) In the workspace, the "New" button can cause confusion, because it is next to the "Join" board button.
Score - 5. Impact - 2, frequency - 3.

(18) It is not specified what happens, when a board is being unlocked and a wrong password is entered.
Score - 5. Impact - 2, frequency - 3.

(19) The user cannot change the background color of the application.
Score - 4. Impact - 2, frequency - 2.

(20) The "Join" and "New" buttons in the server connection scene have different sizes
Score - 3. Impact - 1, frequency - 2.

Also, the following problems were provided by the experts, but were identified as false-positives:

(1) The recently joined boards are not sorted by recency.
(2) The user cannot add a tag to a task from the overview page.
(3) Boards created by the user are not clearly separated from the recent boards.
(4) In the workspace scene, the lock logo, which indicates whether or not a board is password-protected, is redundant.
(5) In the board overview, a task has tags, which do not have text inside of them. That is inconvenient since each tag has to be looked up.
(6) Task viewing and task editing scenes are not separated. The user does not have a separate scene in which they can simply view the tasks they want to achieve.

So overall, 6/26 problems that were identified by the experts were false-positives. So the accuracy of the experts is approximately 23%.

3.3 Heuristics

Finally, it should be noted which usability heuristics were violated by which problems. Therefore, this list is provided:

(1) Visibility of system status: 4.
(2) Match between system and the real world: 5.
(3) User control and freedom: 3, 4, 6, 7, 19.
(4) Consistency and standards: 1, 4, 8, 13.
(5) Error prevention: 1.
(6) Recognition rather than recall: 2, 10, 11, 12, 16, 17.
(7) Flexibility and efficiency of use: 6.
(8) Aesthetic and minimalist design: 14, 15.

(9) Help users recognize, diagnose, and recover from errors: 9, 18.
(10) Help and documentation: 8, 20.

4 CONCLUSIONS AND IMPROVEMENTS

Overall, exactly 20 problems were identified in the design of the Talio application. Also, 4 of these usability issues received a severity score higher than 10, which suggests that there was a notable amount of negligence involved in the first design of the front-end of the application.

Additionally, the majority of the usability problems identified had relatively high impact, but small frequency - that is possibly the main reason why they were not spotted in the initial process of developing the design for the application.

Furthermore, 23% of the total problems found by the experts were identified as false-positives.

5 CITATIONS

To make the references appear, make sure to compile the latex sources, then bibtex, and then latex twice.

REFERENCES

[1] Jakob Nielsen. 1994. Heuristic evaluation. In *Usability Inspection Methods*. John Wiley & Sons, Inc., n.p., 25–62.