

OOP Project Report – Group 57

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

1 INTRODUCTION

1.1 Objective

The purpose of the heuristic evaluation is to identify, evaluate and 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 a product. The prototype is not functional - it is only composed of images which should resemble the front-end design of the program. Therefore, mock-ups used for the heuristic evaluation are presented.

1.2 Evaluated prototype

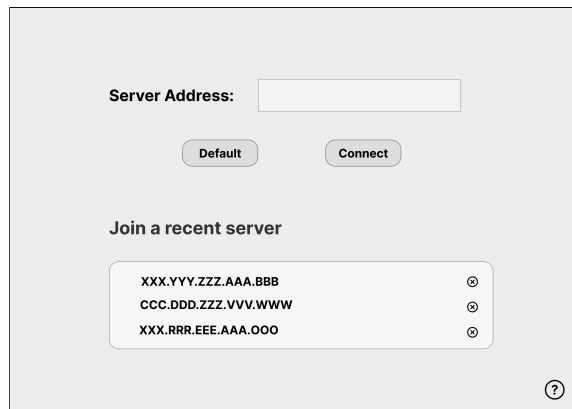


Figure 1: Connecting to a server

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

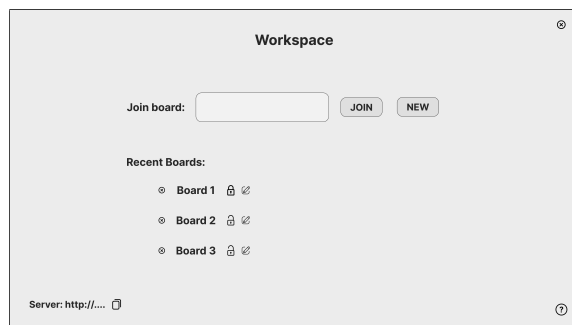


Figure 2: Workspace

Upon successful connection, the workspace scene is displayed. Here the user can choose to join an existing board or to create a new one. Furthermore, a list of recent boards is also displayed.

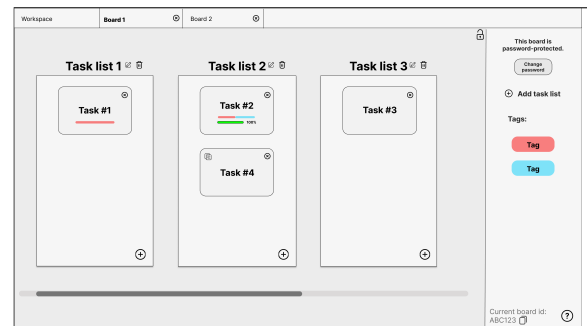


Figure 3: Password protected board with editing access

Boards can be password protected, i. e., some boards require a password to be edited. Passwords are not required for viewing the board. Whenever a user has editing access to a board, an "unlocked" icon is displayed in the top-right corner of the screen.

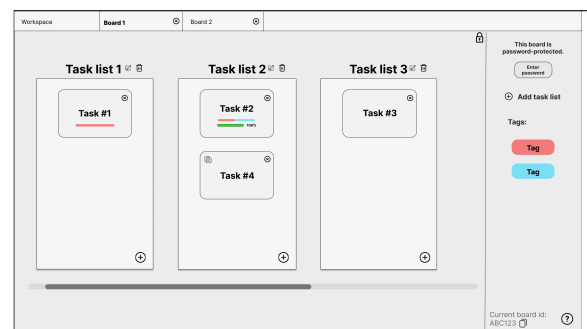


Figure 4: Password protected board without editing access

If a user does not have editing access to a password-protected board, a "locked" icon is displayed in the top-right corner. Moreover, if the user attempts to edit a board without first entering the password, an error message is popped up.



Figure 5: Error message

The error message informs the user that they do not have editing permissions.

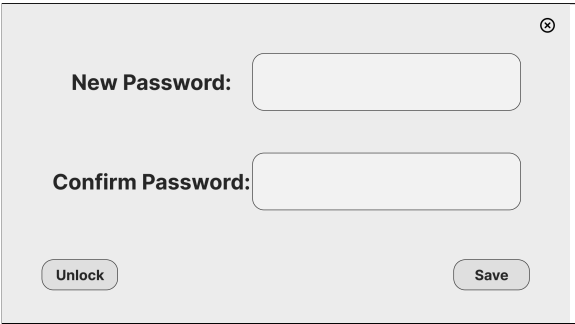


Figure 8: Change password

The user can also change the password of a board.



Figure 6: Removing a password

A password can be removed from a password-protected board. That is called unlocking a board. Unlocking a board removes the board’s password and allows all users to edit it.

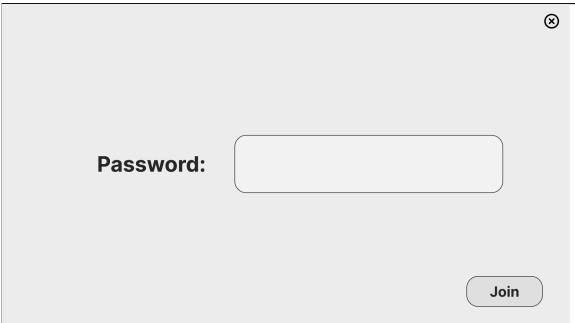


Figure 9: Obtaining editing access

Whenever a user wishes to edit a password-protected board, they should click on the "Enter Password" button, after which they are prompted to type the password.

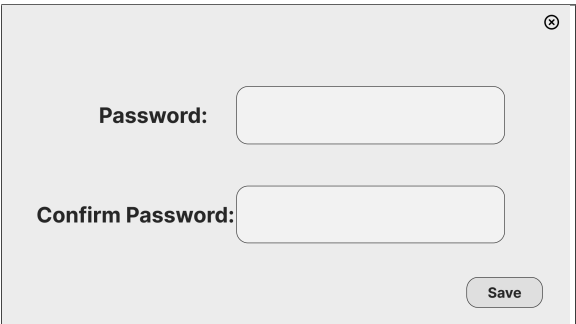


Figure 7: Creating a password

Furthermore, boards can be locked using the same principle. In order to lock a board, the user has to create a password for it.

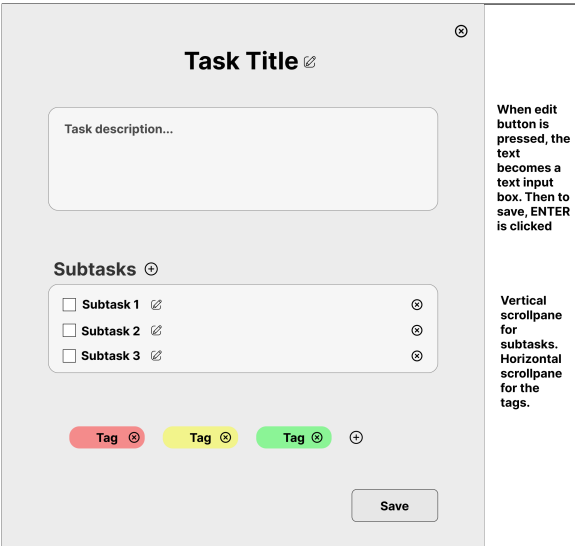


Figure 10: Task view

When a task is clicked on, a popup is shown which enables the user to edit the task and see its extended information. The design allows the user to change the title, description, tags and subtasks of the selected task.

Figure 11: Editing subtasks

Furthermore, the description of the subtasks can be changed using the edit button next to them. Clicking the edit button enables the subtask to be edited and saves the subtask if clicked again.

Figure 12: Adding and editing a tag

Tags can be created and edited. The color for a tag is chosen using the color wheel. Moreover, existing tags from the current board are displayed so that the user could choose from them.

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 informed of the manner in which the evaluation will be handled. They were told that they would only receive 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, the heuristics that were provided to each evaluator are presented. They are the heuristics used by Nielsen [2].

- (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. This format is presented in a later section.

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 showcase through images can be verbally explained by the observer, instead of showing which scene would be loaded.

Furthermore, if at any point in time the evaluator ran into issues, the observer would provide some small hints about the usability of the application. If these minor interventions would not help, the encountered issue would then be considered a usability problem, and would get noted down by the observer.

At first, the evaluators were instructed to briefly pass through the interface, in order to become familiar with it. During this first glance, some problems would already be identified and written down, in order to be further addressed by the developers.

Following this, the experts were advised to approach the prototype more rigorously and begin looking for problems in each scene separately. They were instructed to first observe the overall view and search for basic errors, after which they should evaluate 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 entire 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 in 2003 [1].

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. 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 be eliminated. This task was greatly simplified by using the specific questioning format, described in section 2.3.

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

All results were evaluated using an impact-frequency metric - for each issue identified by the experts, a score from 1 to 7 was assigned for the impact (m_i) and the expected frequency (f_i) of the problem.

Furthermore, a score was calculated for every problem. It was computed 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 i th problem. The intention in using this metric is to classify the issues in a way that would highlight the problems with the highest score as the most severe ones.

Finally, 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 issues that were discovered by the evaluators 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 access its 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 a 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.

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, 5, 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

4.1 General conclusions

Overall, exactly 20 problems were identified in the design of the "Talio" application. Moreover, 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 program.

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.

4.2 Improvements

In order to improve the initial design of the application, the compiled list of usability problems was used. Since a limited amount of resources was available, a prioritization system had to be applied, in order to minimize the impact of the issues perceived by a potential user. Therefore, only the 5 most severe usability problems were fixed, according to the prioritization described in section 3.2.

A list of introduced changes is presented next. Each point in the list corresponds to a usability problem identified by the evaluators (see section 3.2). Furthermore, the words written in *italics> denote the previously discussed heuristics. The left part of every illustration denotes the design before the improvement, and the right part shows the new design.*

- (1) The button which allows closing the workspace was removed. This change improves the design in terms of *error prevention*, since a non-intended and possibly application-breaking functionality was removed.



Figure 13: First improvement

- (2) In order to minimize confusion about the feature of password protection, the lock icon was removed. It was replaced by a text label, with text values "Board editing disabled" and "Board editing enabled", which would be shown whenever the board was locked or unlocked, respectively. This change is intended to significantly reduce confusion about password protection. Therefore, it improves on the 6th heuristic - *recognition rather than recall*.

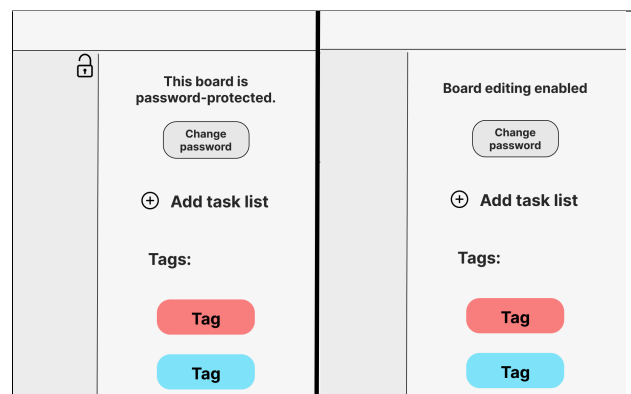


Figure 14: Second improvement

- (3) A text label which stores the title of the board was added to the board view. Once this label is clicked, it is transformed to a text field, which allows the user to edit the title of a board. Also, for consistency purposes, the text that used to be contained within the tabs was changed to now store the ID of the board. This improvement enhances the design in terms of *user control and freedom*, since now the user is allowed to choose a title for any board.

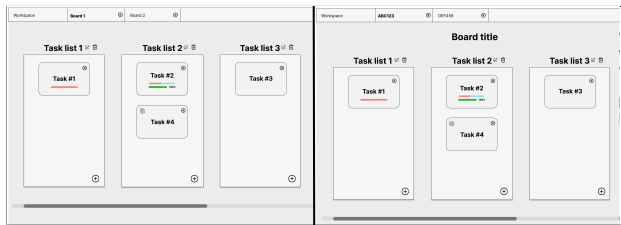


Figure 15: Third improvement

- (4) Tab pane bar was added to the workspace scene. This brought a triple improvement to the usability of the application: the *visibility of system status* was improved, the *user control and freedom* was enhanced and the level of *consistency* was increased.



Figure 16: Fourth improvement

- (5) The percentage field next to the progress bar was changed by a text label " x/y ". This label indicates that x out of y subtasks have been completed. The reasoning behind this change is the following: percentages do not indicate clearly how many subtasks have been completed and how many of them are left. Therefore, this change made the application significantly more user-friendly by increasing the *consistency* and the *match between the system and the real world*.

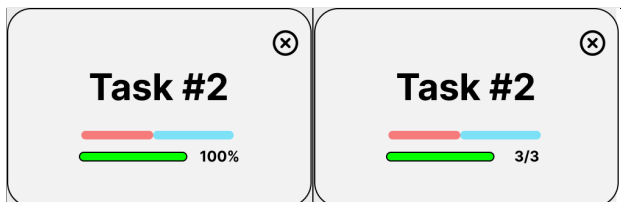


Figure 17: Fifth improvement

4.3 Final design

Lastly, the final improved design is presented.

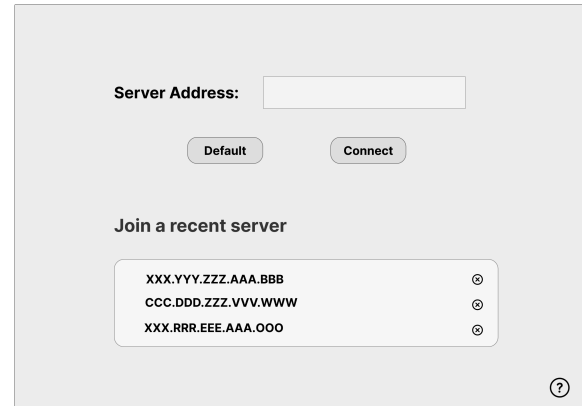


Figure 18: Connecting to a server

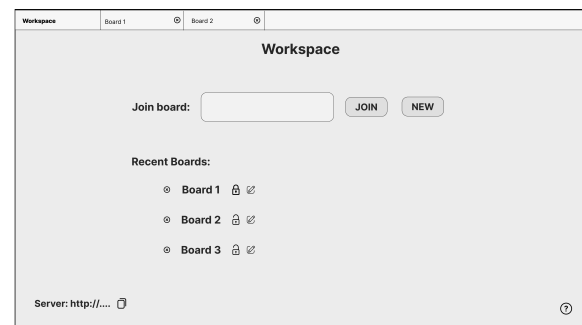


Figure 19: Workspace

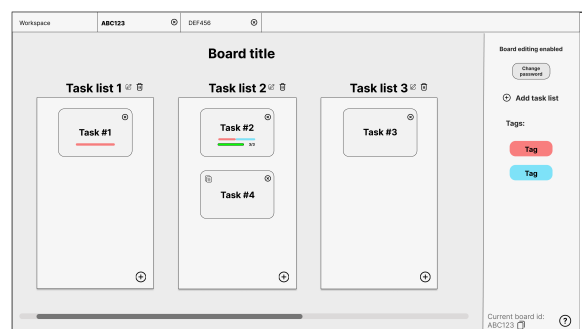


Figure 20: Password protected board with editing access



Figure 21: Password protected board without editing access

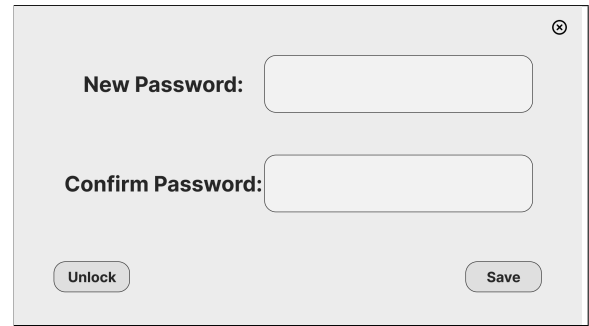


Figure 25: Change password

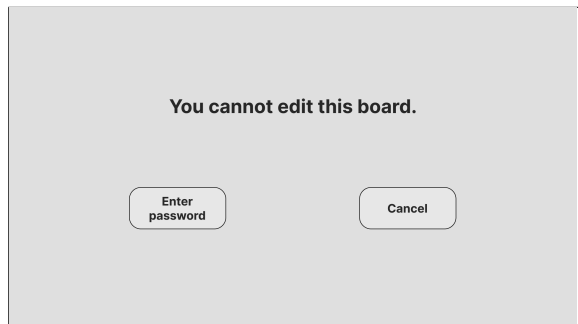


Figure 22: Error message

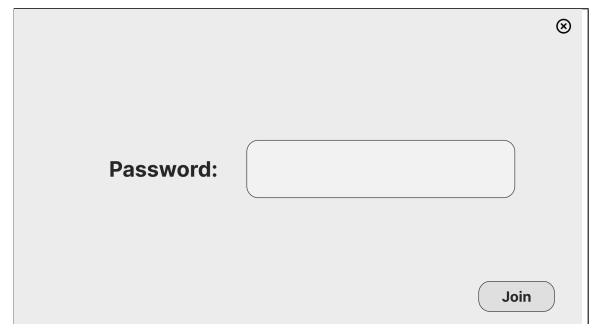


Figure 26: Obtaining editing access



Figure 23: Removing a password

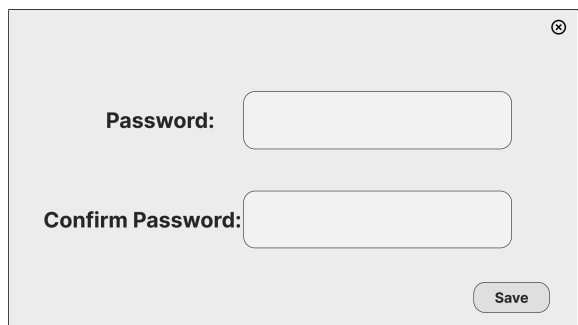


Figure 24: Creating a password

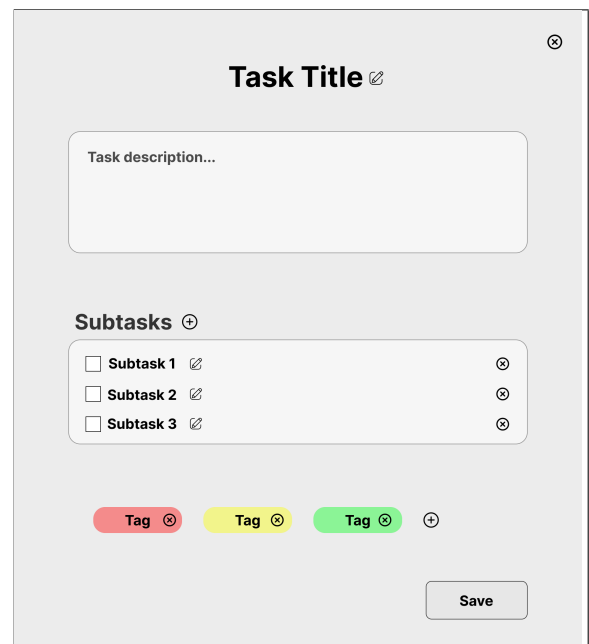


Figure 27: Task view

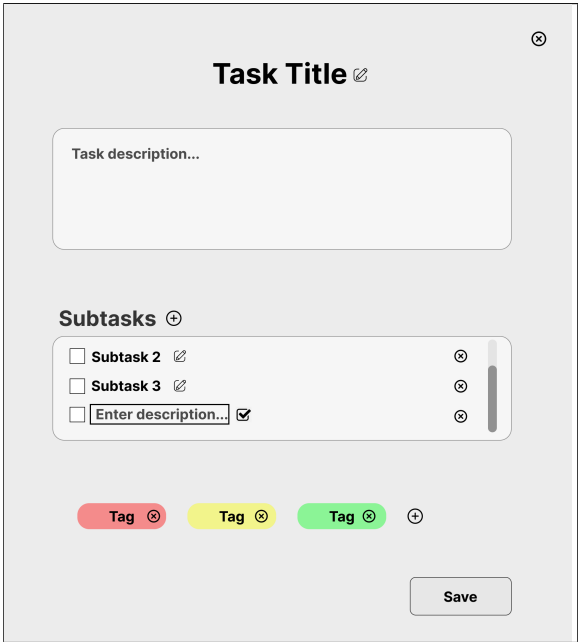


Figure 28: Editing subtasks

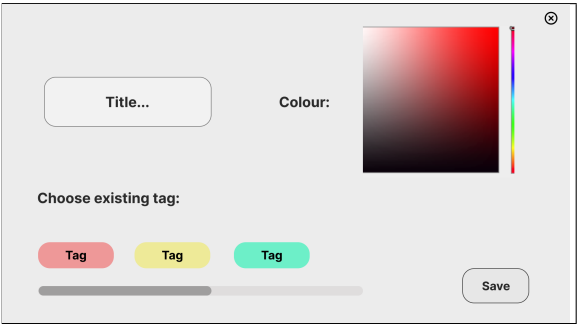


Figure 29: Adding and editing a tag

REFERENCES

[1] Gilbert Cockton, Alan Woolrych, Lynne Hall, and Mark Hindmarch. 2003. Changing Analysts’ Tunes: The Surprising Impact of a New Instrument for Usability Inspection Method Assessment. In *People and Computers XVII — Designing for Society*, Eamonn O’Neill, Philippe Palanque, and Peter Johnson (Eds.). Springer London, London, 145–161.

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

Word count: 2381