OOP Heuristic Usability Evaluation – Team 50

Team 50

ABSTRACT

A clear and well-documented LaTeX document is presented as an article formatted for publication by ACM in a conference proceedings or journal publication. Based on the “acmart” document class, this article presents and explains many of the common variations, as well as many of the formatting elements an author may use in the preparation of the documentation of their work.

1 INTRODUCTION

Software user experience design is a critical aspect of creating a successful product. To ensure a positive user experience, it is beneficial to evaluate the usability and effectiveness of the interface during the design process. One widely used method for conducting such an evaluation is the heuristic evaluation, which involves a group of evaluators systematically reviewing the interface against a set of usability heuristics.

In this report, we present the results of a heuristic evaluation that was conducted on a prototype of a software user interface by a group of evaluators. The evaluation aimed to identify usability issues and provide recommendations for improving the design of the actual product.

Our prototype was made in Figma, and had multiple interactive elements, navigation on button clicks and keyboard hits. Although all major states of our application were represented (e.g. admin / client view) with all the states of our components (selected, focused, hovered), the interface was not fully functional, e.g evaluators couldn’t type into input fields, and some popup menus were opened by a button click and closed by leaving the popup with the mouse (instead of closing automatically when one clicks outside of it), since we didn’t find a more suitable solution. Furthermore, we didn’t implement the Help popup and the button interactively, we just inserted a comment in the evaluators’ instruction that contained the popup’s design that would appear in that case.

Our full designs are accessible through the following link: LINK

And the prototype can be run using this link: LINK

Next we will introduce our prototype in this document.

When a person enters our application, they will see a login screen where they need to provide a server url and a name, or in case of an admin, a server url and a password. Users can switch between the admin/end-user login screen using the lower button.

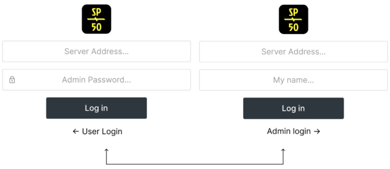


Figure 1: Login screen

Next, they are presented with the Workspace View in which end-users can see their boards that they have already opened, and admins can see all boards. This makes it possible for the admin to interact with all boards and also to use the Clear Server button on the bottom that removes all existing boards in the database of that particular server.

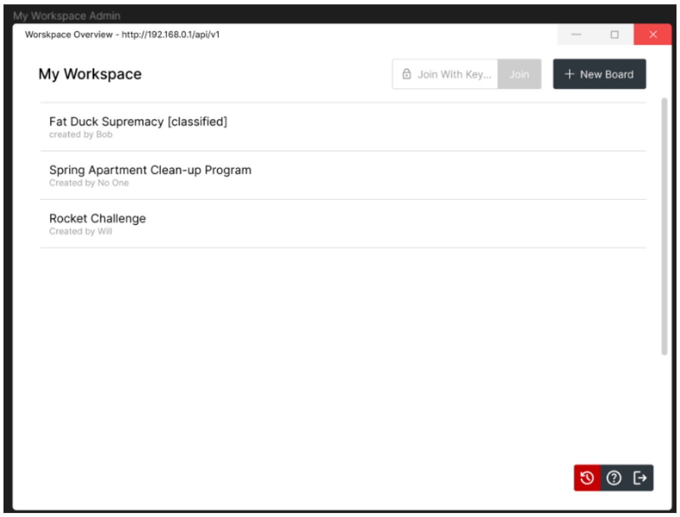


Figure 2: Workspace View

When one clicks on a board, the app redirects them to the board overview scene in which evaluators can try out some interactions of our application, like hovering over an icon and seeing its hovered state or opening a popup by clicking on an icon or a task.

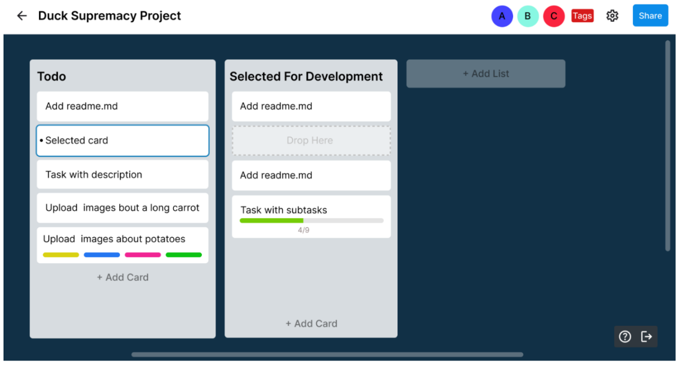
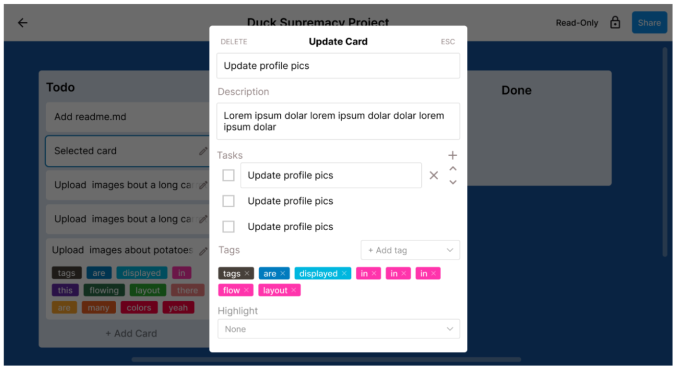


Figure 3: Board Overview

Specifically, when they click on a task, the task details popup is shown which can be then closed by pressing escape or by clicking on the button.



Update Card

Update profile pics

Description

Lorem ipsum dolar lorem ipsum dolar dolar lorem ipsum dolar

Tasks

- □ Update profile pics
- □ Update profile pics
- □ Update profile pics

Tags + Add tag

tags × are × displayed × in × in × in × flow × layout ×

Highlight

None

Figure 4: Details View

If users would liket to edit the colours of tag, or add/remove them, they can do it in the tag manager popup that can be opened by clicking on the red Tag button in the header.

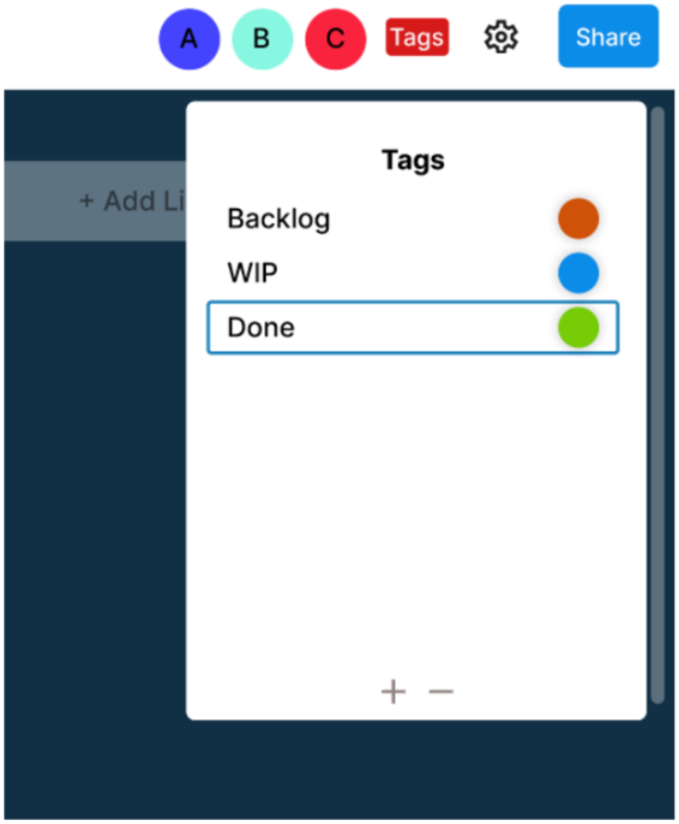


Figure 5: Tag Manager

When users would like to customize the board’s look, they can do so in the Customization popup which can be opened by clicking on the gear icon.

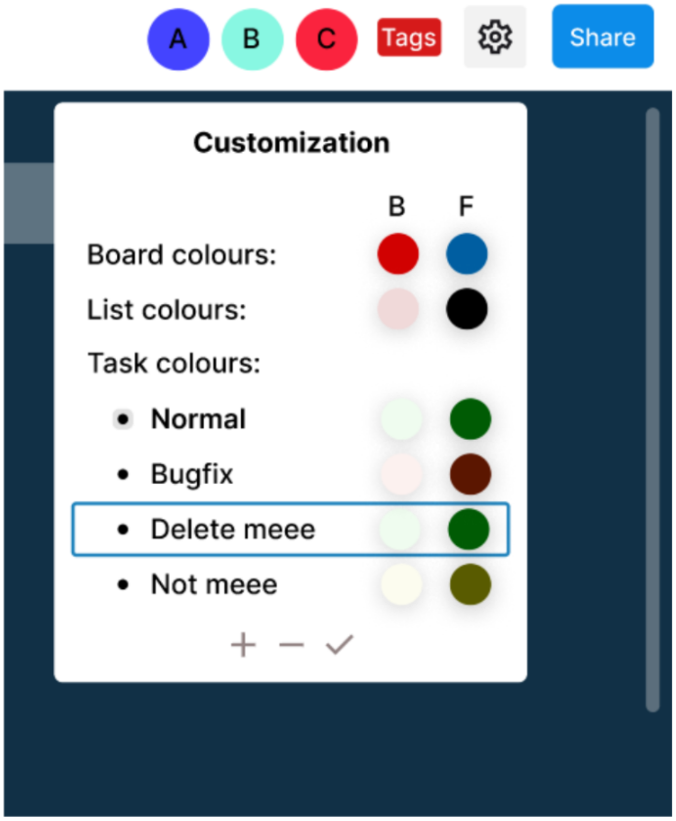


Figure 6: Customization View

2 METHOD

2.1 Evaluators

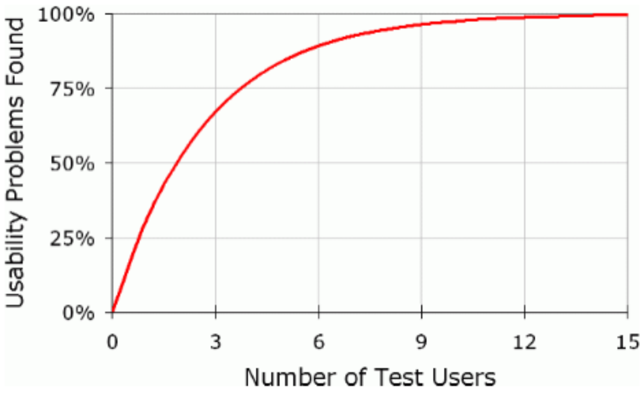


Figure 7: (Nielsen 2000)

Our team enlisted six novice evaluators who possess a solid understanding of human-computer interaction, heuristic evaluation techniques, and our specific product. Their familiarity with both the prototype's requirements and the heuristics used for assessment were expected to ensure an almost comprehensive evaluation process. According to Borys (2014), novice evaluators who are well-versed in HCI heuristic evaluation criteria can perform on par with experts. Thus, we anticipated a high number of usability issues to be identified, as suggested by the trend depicted in the figure above.

2.2 Procedure

To maintain consistency and prevent potential bias, evaluators were explicitly instructed to conduct their analysis independently. The evaluators were asked to spend 1 hour searching for and documenting problems in our prototype, so that they could make a thorough analysis.

We provided Nielsen's list of 10 problem types (see below) to look for during a heuristic evaluation, and we provided descriptions and examples of those problems, so that the evaluators could get an idea what to look for and how to categorize them. We asked them to go through the application multiple times, and each time select maximum 3-4 problem types to look for.

We also asked them to go through these situations multiple times, as we expect high demand for these scenarios.

- Log in to the application with your name, create a board with a custom name and create multiple lists and cards. After this, try to reorganize the board by drag and drop and by keyboard shortcuts. (In the prototype they needed to imagine this).
- Open an existing board with a given key that you haven't opened previously. Assume it's password protected, so you can't make changes to the board but you should be able to open the details section and the popups too. Try to find how you can enter a password (entering just theoretically), and evaluate how easy the procedure is.
- Open an existing board with a given key that you haven't opened previously. Assume it's password protected, so you can't make changes to the board but you should be able to open the details section and the popups too. Try to find how you can enter a password (entering just theoretically), and evaluate how easy the procedure is.

As this was only a Figma prototype, some of the interactions were naturally not possible but the evaluators could still navigate between the different states of the application sometimes imagining that the actions took place (e.g. a drag and drop).

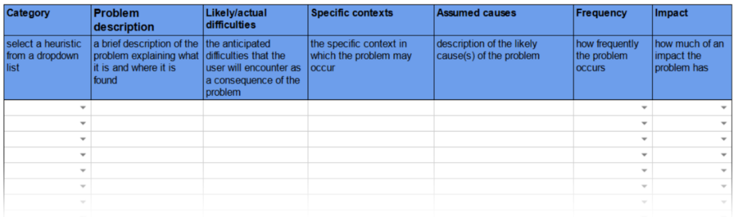
In this heuristic evaluation, a problem is characterized as a design aspect or functionality that infringes upon one or more usability heuristic principles detailed below. Evaluators were asked to report each identified issue by filling out the provided template sheet (this can be seen in the next section).

2.2.1 *Usability Heuristics:*

- Visibility of system status
- Match between system and real world
- User control and freedom
- Consistency and standards
- Error prevention
- Recognition rather recall
- Flexibility and efficiency of use
- Aesthetic and minimalist design
- Help users recognise, diagnose, and recover from errors
- Help and documentation

3 DATA COLLECTION

The evaluators are requested to submit the issues they come across using the format depicted in 8:



Category	Problem description	Likely/actual difficulties	Specific contexts	Assumed causes	Frequency	Impact
select a heuristic from a dropdown list	a brief description of the problem explaining what it is and where it is found	the anticipated difficulties that the user will encounter as a consequence of the problem	the specific context in which the problem may occur	description of the likely cause(s) of the problem	how frequently the problem occurs	how much of an impact the problem has

Figure 8: Form Used for Evaluation

After documenting identified usability problems, evaluators were prompted to assess these issues based on two factors: frequency and impact. They needed to assign a score ranging from 1 to 5, allowing for the results to be displayed on a severity matrix in figure 9.

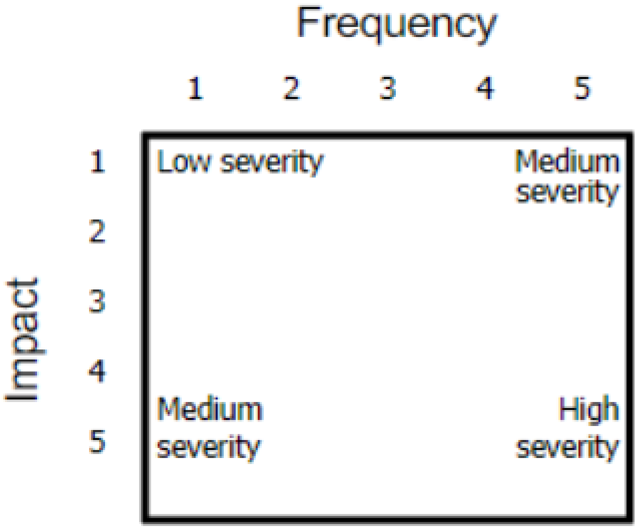


Figure 9: Severity Matrix *(Nielsen 1993)*

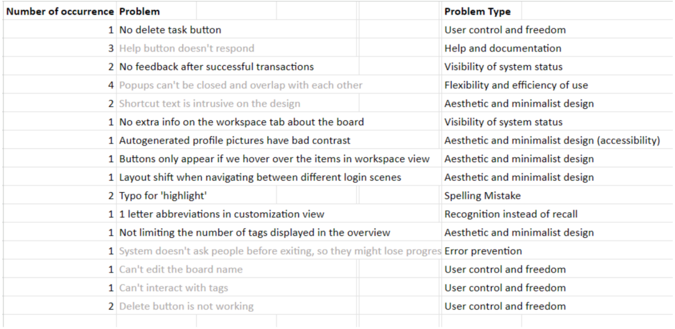
4 RESULTS

4.1 Raw Data

Our raw feedback data can be accessed through the following link: link

4.2 Analysis

After gathering the results, we read through them carefully, and identified issues that were reported by multiple people.



Number of occurrence	Problem	Problem Type
1	No delete task button	User control and freedom
3	Help button doesn't respond	Help and documentation
2	No feedback after successful transactions	Visibility of system status
4	Popups can't be closed and overlap with each other	Flexibility and efficiency of use
2	Shortcut text is intrusive on the design	Aesthetic and minimalist design
1	No extra info on the workspace tab about the board	Visibility of system status
1	Autogenerated profile pictures have bad contrast	Aesthetic and minimalist design (accessibility)
1	Buttons only appear if we hover over the items in workspace view	Aesthetic and minimalist design
1	Layout shift when navigating between different login scenes	Aesthetic and minimalist design
2	Typo for 'highlight'	Spelling Mistake
1	1 letter abbreviations in customization view	Recognition instead of recall
1	Not limiting the number of tags displayed in the overview	Aesthetic and minimalist design
1	System doesn't ask people before exiting, so they might lose progres	Error prevention
1	Can't edit the board name	User control and freedom
1	Can't interact with tags	User control and freedom
2	Delete button is not working	User control and freedom

Figure 10: Issues by number of occurrence

4.3 Processing the raw data

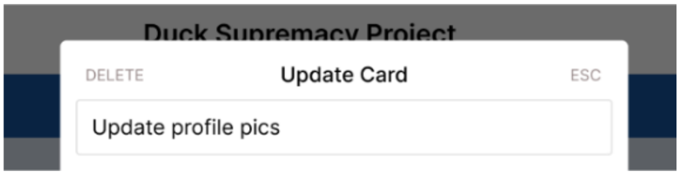
Unfortunately, we needed to discard multiple problem reports, as they were only relevant to the fact that the prototype wasn't fully functional or that the help button was not showing the popup - from which we wrote a comment before the evaluation started but was seemingly ignored by some people. Reading through the problems, we noticed that around 50% of the reported problems were marked with an incorrect heuristic categorization, so we corrected them when aggregating the data. We also noticed that we needed to introduce a new heuristic, *accessibility*, since it was not in our original list

In the following pages, we describe the identified issues that were not discarded because of the reasons above, and reflect on them by showing how we would solve them.

4.4 No delete task button. (user control and freedom)

Reported by 1 person.

There was no delete button for a task in our design, partially because we assumed people would just select the card and delete it using the delete key but partially also because we forgot it. In the updated design, we added the button in the details section.



Update profile pics

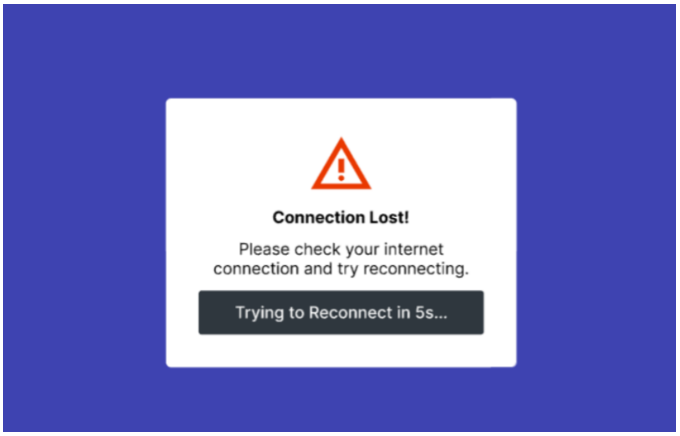
Figure 11: Added Delete Button

4.5 No feedback after successful transactions (visibility of system status)

Reported by 2 persons.

The problem the participants described was concerning that our application doesn't give feedback about successful entry creation / deletion, and their concern was that people might not be sure whether the app made a successful transaction.

Our opinion on this is that the fact that people can see the changes instantly in the application is a perfect positive feedback, since the application auto-saves all work, and would instantly report any kind of errors. Users might still not be sure about this fact, so several applications display status messages in the header, and others just display a message when people press Ctrl+S, informing them about the autosave feature. We think that the later option would be more appropriate for us, since it doesn't introduce unnecessary popups and doesn't take up space from the window.



Connection Lost!

Please check your internet connection and try reconnecting.

Figure 12: Error popup

4.6 No extra information on the workspace tab about the boards apart from the title (visibility of system status)

Reported by 1 person.

The Talio task organizer app currently lacks additional information about the content of individual boards in the workspace tab. This absence of detail can make it difficult for users to quickly assess the status of their boards and prioritize their tasks. Therefore, there might be a need for a more comprehensive and informative overview of each board in the workspace tab, including the number of lists and tasks.

We tried to keep our application as minimalistic as possible but some kind of preview of the board might be helpful for quickly identifying the board. The evaluator suggested displaying the number of cards but we think that it is not relevant / helpful enough, so a screenshot of the board showing its custom background color might be more appropriate. However, considering the amount of time we have to finish the project, this and the (1/5) impact that the evaluator assigned to the issue, this will not be a priority for us to implement in the following weeks.

4.7 Auto-generated profile pictures might have bad contrast (aesthetic, accessibility)

Reported by 1 person.

The background colors of these status indicator icons are automatically generated but the foreground remained the same (black) regardless of the generated color's brightness.

We will therefore make sure our color generator only varies the brightness of the colors in a particular range, such that the color is

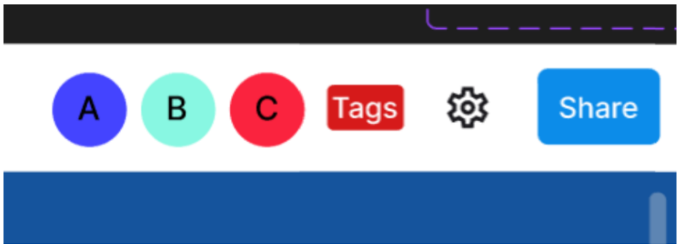


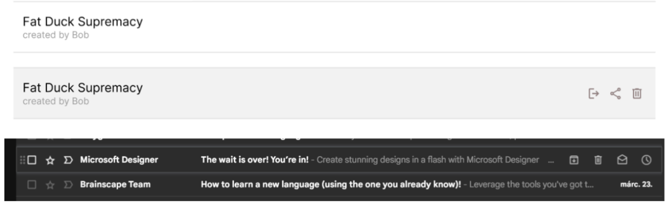
Figure 13: Low Contrast

not blending into the header's background, nor would the initial letter become unreadable even for people with visual impairments.

4.8 Buttons only appear if we hover over the items in the workspace view (aesthetic)

Reported by 1 person.

The evaluator expressed apprehension regarding the appearance of action buttons exclusively when hovering over an item on the workspace page. Despite this concern, it is worth noting that this design pattern is commonly employed in widely used applications such as Gmail to reduce visual clutter on the screen and display buttons solely on relevant items. Therefore, the prevalence of this pattern in large and popular applications serves as evidence for its applicability, rendering further research unnecessary.



Fat Duck Supremacy
created by Bob

Fat Duck Supremacy
created by Bob

Microsoft Designer The wait is over! You're in! - Create stunning designs in a flash with Microsoft Designer ...

Brainscape Team How to learn a new language (using the one you already know)! - Leverage the tools you've got t... márc. 23.

Figure 14: Comparison between Gmail and our application

4.9 Layout shift when navigating between the admin and the user login scenes (aesthetic)

Reported by 1 person.

There was a slight translation of the elements between the two scenes because the elements were only manually centered. We corrected the mistake and now the navigation is seamless in the prototype. The centering of elements in the application is ensured programmatically.

4.10 Typo for 'Highlight' in the task details view. (spelling mistake)

Reported by 2 persons.

We corrected the spelling mistake. (Figure 17, 18)

4.11 One-letter abbreviations in customization view are unclear

Reported by 1 person.

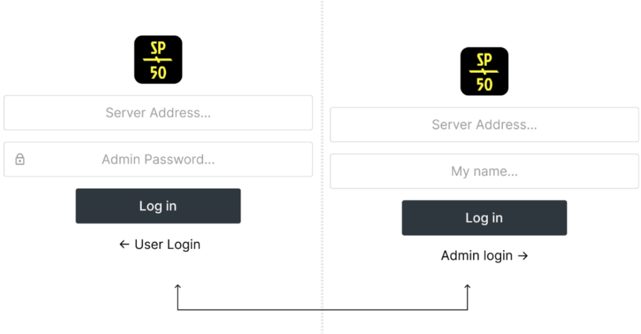


Figure 15: Before

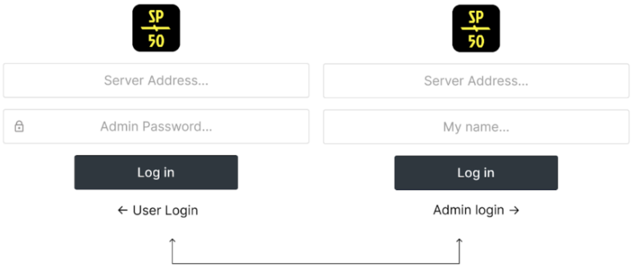
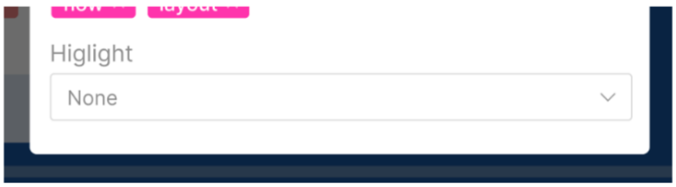


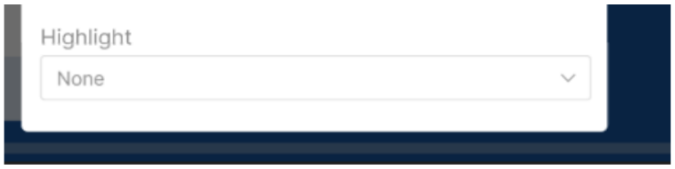
Figure 16: After



Higlight

None

Figure 17: Before

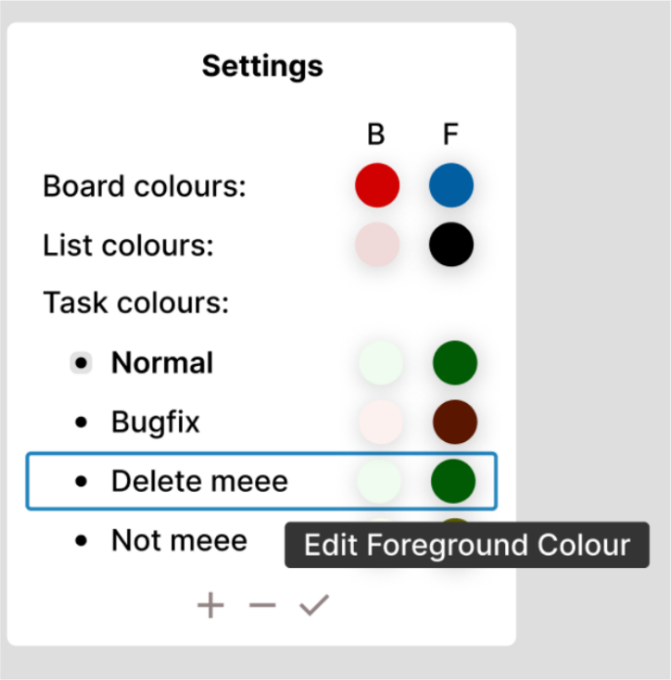


Highlight

None

Figure 18: After

The one-letter abbreviations in the customization view (B for background, F for foreground color) can be unclear. Our solution is to include tooltips for these letters, and also for the color picker circles to clarify their meaning when the user hovers over them with the mouse.



Settings

B F

Board colours:

List colours:

Task colours:

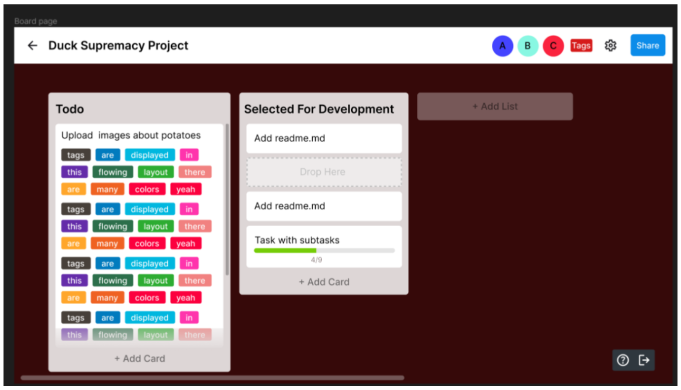
- Normal
- Bugfix
- Delete meee
- Not meee

Figure 19: Input Elements with Tooltip Text

4.12 Not limiting the number of tags displayed in the overview (aesthetic and minimalist design)

Reported by 1 person.

The evaluator is concerned that we don't limit the number of tags displayed on the board overview, and it may seem that the list comprises only one card. Though, while reproducing / illustrating the problem, it's clear that this is outside the scope of the application's normal use cases, it showed made us realize we did not have a vertical scroll bar on the lists, and it could be a better idea than having it on the whole scene, as this way the + Add Card button is always visible.



Board page

Duck Supremacy Project

Todo

Upload images about potatoes

tags are displayed in this flowing layout there are many colors yeah tags are displayed in this flowing layout there are many colors yeah tags are displayed in this flowing layout there are many colors yeah tags are displayed in this flowing layout there

Selected For Development

Add readme.md

Drop Here

Add readme.md

Task with subtasks 4/9

Figure 20: Card with lots of tags

5 PRIORISATION

Since we did not receive an overwhelming amount of feedback, we could correct all our mistakes that we thought were actually mistakes. We ignored issue #3 regarding adding extra information to the workspace scene since it is not in the requirement list of the client, and it would require a considerable amount of extra work to complete. We did not need to prioritize across the remaining issues but we created a severity graph to see what was the distribution of the frequency and impact of the issues we had.

When implementing the actual application, we prioritized accessibility and aesthetics, and we also tried to prevent errors happening e.g. when we try to navigate to with the left arrow on the very first card, nothing happens, as expected instead of getting an error that the user might find difficult to comprehend.

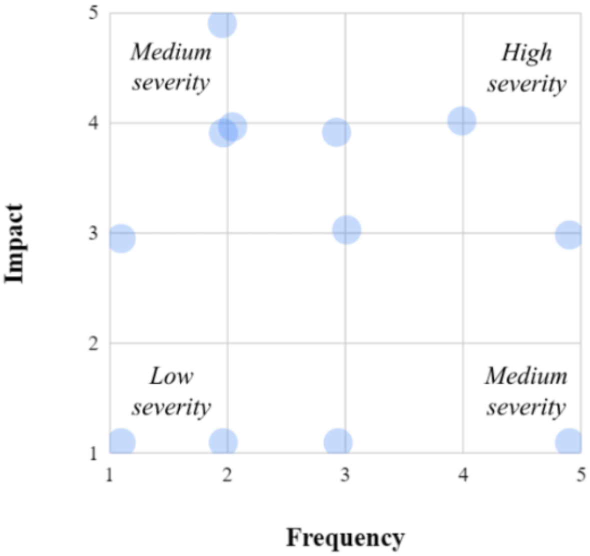


Figure 21: Severity Matrix Based on the Results

Based on the severity matrix in Figure 21, the severity of issues are evenly distributed and the most significant issue is issue #2 No feedback after successful transactions. Additionally, (Fig. 22) showcases the number of issues found by the evaluators for each usability heuristic.

6 CONCLUSIONS, IMPROVEMENTS

After gathering the results, it was noted that around 50% of the reported problems were marked with an incorrect heuristic categorization, and multiple problem reports were discarded because they were only relevant to the fact that the prototype was not fully functional or the evaluators did not read the instructions correctly. However, several issues were identified that were valuable to the product design, including the absence of a delete task button, a spelling mistake and that the customization view might be unclear.

Overall, this heuristic evaluation provided valuable remarks regarding the usability of the prototype but statistically did not provide enough useful information to draw any conclusions about

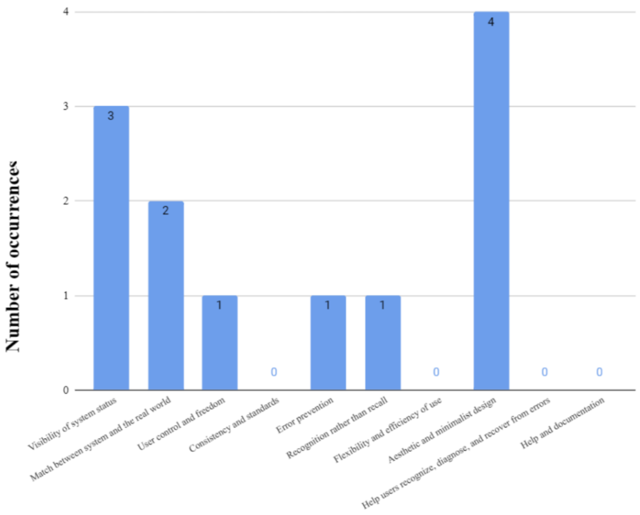


Figure 22: Number of issues for each heuristic

specific areas or improvements or the actual severity of problems because of the low number of identified issues and the lack of overlap between found issues.

Although the low number of reported issues might be caused by the already highly polished interface, we think that repeating the experiment with the collaboration of another group who spend more time with the experiment and read the instructions carefully could lead to a statistically more reliable result set.

REFERENCES

- Borys, Magdalena Laskowski, Maciej. (2014). Expert vs novice evaluators: Comparison of heuristic evaluation assessment. ICEIS 2014 - Proceedings of the 16th International Conference on Enterprise Information Systems. 3. 144-1
- Nielsen, J. (2000, March 18). Why You Only Need to Test with 5 Users: Article by Jakob Nielsen. Nielsen Norman Group.
- Nielsen, J (1993). Usability Engineering. Oxford, United Kingdom: Elsevier Science. ISBN 9780125184069.