Master Thesis

Czech
Technical
University
in Prague

F3

Faculty of Electrical Engineering
Department of Computer Graphics and Interaction

Redesigning a collaborative planning system

Adam Loucký

Supervisor: doc. Ing. Zdeněk Míkovec, Ph.D.
January 2025

MASTER‘S THESIS ASSIGNMENT

I. Personal and study details

Student's name: **Loucký Adam** Personal ID number: **498850**

Faculty / Institute: **Faculty of Electrical Engineering**

Department / Institute: **Department of Computer Graphics and Interaction**

Study program: **Open Informatics**

Specialisation: **Human-Computer Interaction**

II. Master’s thesis details

Master’s thesis title in English:

Redesign of the collaborative planning system

Master’s thesis title in Czech:

Redesign kolaborativního plánovacího systému

Name and workplace of master’s thesis supervisor:

doc. Ing. Zdeněk Míkovec, Ph.D. Department of Computer Graphics and Interaction

Name and workplace of second master’s thesis supervisor or consultant:

Date of master’s thesis assignment: **03.02.2025** Deadline for master's thesis submission: ____________

Assignment valid until: **20.09.2026**

Head of department’s signature

prof. Mgr. Petr Páta, Ph.D.
Vice-dean´s signature on behalf of the Dean

III. Assignment receipt

The student acknowledges that the master’s thesis is an individual work.
The student must produce his thesis without the assistance of others, with the exception of provided consultations.
Within the master’s thesis, the author must state the names of consultants and include a list of references.

Date of assignment receipt Student’s signature

CTU
CZECH TECHNICAL UNIVERSITY IN PRAGUE

MASTER‘S THESIS ASSIGNMENT

I. Personal and study details

Student's name: **Loucký Adam** Personal ID number: **498850**

Faculty / Institute: **Faculty of Electrical Engineering**

Department / Institute: **Department of Computer Graphics and Interaction**

Study program: **Open Informatics**

Specialisation: **Human-Computer Interaction**

II. Master’s thesis details

Master’s thesis title in English:

Redesign of the collaborative planning system

Master’s thesis title in Czech:

Redesign kolaborativního plánovacího systému

Guidelines:

The goal of the work is to design a new user interface for selected parts of an existing planning system for collaborative modification and visualization of data structures capturing resources, events, and tasks over time. First, analyze the existing planning system and identify problems and shortcomings. Using user research, redefine the needs and functional requirements. Then, select parts of the system suitable for redesign. For these parts, design a new user interface. Follow the UCD (User-Centered Design) methodology, i.e., iterative design and testing of prototypes with the target user group. The result of the work will be a final prototype and the corresponding user interface documentation for future implementation.

Bibliography / sources:

1. Elizabeth Goodman, Mike Kuniavsky and Andrea Moed (2012). Observing the User Experience. Morgan Kaufman.
2. Bill Buxton (2007). Sketching User Experiences. Morgan Kaufman.
3. Miroslav Disman (2002). Jak se vyrábí sociologická znalost: Příručka pro uživatele. Nakladatelství Karolinum.
4. Colin Ware (2010). Visual Thinking for Design. Morgan Kaufman.

Acknowledgements

I would like to thank my family and friends for supporting me during my studies, along with doc. Ing. Zdeněk Míkovec, Ph.D. and the Hydronaut team for frequent consultations and valuable advice during the creation of this thesis.

Declaration

I declare, that the contents of this Masters thesis on the topic of ’Redesigning a collaborative planning system’ were created by me alone, using my expertise in combination with the listed literature and sources.

Prague, May 21st 2025

Abstract

This thesis deals with the development of a new design for a collaborative planning system, used both commercially, and in the Hydronaut shuttle mission planning. The system, called CommonTongue, allows for the creation of a hierarchy of tasks, each having its own set of associated data. The goal is to visualize that data and allow the user to efficiently plan their events in collaboration with other users online.

During the course of the thesis, CommonTongue was examined from the perspective of a new user, with three rounds of user tests being used as the basis for the final design. The findings of these examinations were then used to identify fundamental design issues, which negatively impacted the performance of the system.

These issues were addressed with a redesign of the main application view and the main interface used to modify data in the system. The interface changes follow fundamental design principles, and aim to make the system intuitive and readable.

Keywords: Human-Computer interaction, User-Centered Design, User Experience, User Interface, Critical systems

Supervisor: doc. Ing. Zdeněk Míkovec, Ph.D.

Abstrakt

Práce se zabývá vývojem nového návrhu kolaborativního plánovacího systému využívaného pro komerční a výzkumné účely projektem Hydronaut. Systém, pojmenovaný CommonTongue, umožňuje tvorbu hierarchie úkolů, ve které má každá položka příslušnou sadu dat. Cílem systému je vizualizovat tato data a umožnit uživateli efektivní kolaborativní plánování událostí online.

Během tvorby práce byl systém CommonTongue analyzován během tří kol uživatelských testů. Testování probíhalo vždy z pohledu nového uživatele, s cílem nalézt zásadní problémy návrhu, které negativně ovlivňovaly výkon systému.

Nalezené problémy byly řešeny novým návrhem hlavního pohledu aplikace a hlavního rozhraní pro úpravu dat v systému. Provedené změny jsou odvozeny od základních konceptů návrhu uživatelského rozhraní a jejich cílem je rozhraní zpřehlednit a udělat práci s programem intuitivní.

Klíčová slova: Human-Computer Interaction, User-Centered Design, User Experience, Uživatelské Rozhraní, Kritické systémy

Překlad názvu: Redesign kolaborativního plánovacího systému

Contents

Figures

Tables

Chapter 1

Introduction

With the trend of digitizing daily tasks, the average human being gets more and more accustomed to the idea of organizing their days using a computer, be it email systems to organize mail or digital calendars to organize their appointments. It only makes sense that project organization is to follow, with popular applications like Slack[1], Trello[2], Notion[3], or Jira[4] being used daily to streamline organization within teams. A key feature of these programs is the ability to communicate with other people by creating a group workspace, with members receiving notifications in the case of updates to the project. A similar system, called CommonTongue[1], was designed by the research group Hydronaut[2] for the purpose of organizing their missions.

Figure 1.1: Logo of the Hydronaut project

CommonTongue is a system with a specialized use – being able to make a task active or inactive for the user only. The tasks are shared between multiple users, each of whom can make the task passive or active on demand, which toggles between receiving notifications, and muting them. This workflow allows team leaders to access communication between team members at any time they needed, while also keeping notifications for the task muted. This is similar to sending each email as a copy to the team leader, with the added benefit of not being interrupted by constant notifications. With the goal

[1] `http://www.slack.com/`

[2] `http://www.trello.com/`

[3] `http://www.notion.com/`

[4] `https://www.atlassian.com/software/jira`

of making work and personal organization easier, CommonTongue aims to provide an alternative to email communication.

To make such a system usable, CommonTongue must meet usability standards, while also guiding the user into the intended behavior. The goal of this thesis is to identify key issues with the user interface that could limit the performance of the system, choose those that are the most critical, and create a set of changes that address these issues, while also supporting the intended behavior of the user.

The main method used for initial analysis relied on analyzing the behavior of both a first-time user and an expert conducting a set of instructions, for example, creating a plan for a mission or organizing their daily routine. Combining the results of the initial tests with expert analysis, a list of observations was compiled and later used to guide the redesign of the interface. The newly created interface was compared to the original by a second round of user tests, which focused on the initial perception of the system, with the goal of comparing which system features could be intuited from the interface alone. Lastly, the final round of tests focused on finer interaction with the system, where users were tasked with using the system to plan a sample scenario. The final round of testing led to a new design of the main interface used to modify task data, which was evaluated using comparison testing. A brief analysis of similar systems was also conducted to compare the features of CommonTongue with its competitors and guide the new design.

Chapter 2
Research

The research chapter will focus on analyzing the CommonTongue system both as a new user and by observing experts, describing the program's interface and functionality, and then conducting competitive analysis to ensure the system holds against other state-of-the-art project planners.

2.1 Interface analysis

The system was unfamiliar before the start of this thesis and one month was taken to analyze the interface – first from the perspective of a new user and later by consulting and observing experts who are familiar with the program. This chapter will only detail the methods by which the information was gathered and will be referenced in Chapter 3.

2.1.1 Personal analysis

A set of three missions needed to be planned using the CommonTongue system, which was an opportunity for an initial analysis of the system. This analysis was performed personally and consisted of three separate sessions during which three mission plans were created within the system. The result of these sessions was an understanding of the system interface and functionality, which was required to begin working on the thesis.

2.1.2 Observing experts

The goal of observing expert users was to see what is possible in the system and how efficient expert users are in performing their tasks. The expert had a series of tasks that they needed to perform, and they worked with the system undisturbed. The observations from this interaction were noted and used in the later design. Even though the observation wasn't a fully curated test, it still provided valuable insight on the expert usage of the system. Additional discussions were conducted with the authors of the system in order to ensure a proper understanding of the intended functionality of the program and its features.

2.2 CommonTongue

The CommonTongue system is a specialized project planner, used mainly by the Hydronaut research project to organize their events and missions, which makes CommonTongue a critical system – a system that must be highly reliable. While the malfunction of the system doesn't endanger human life, it can negatively influence the mission proceedings (for example, by not displaying a deadline or a time slot correctly) and lead to time loss or financial loss. The application is also used in other commercial areas, which reinforces the need for reliability. This also means the design shouldn't only be focused on satisfying the Hydronaut team, but also fit the commercial needs.

2.2.1 Page Layout

The main interface is divided into six views, each of which provides a different perspective on the same data. They are as follows:

- **Dashboard** - shows a timeline of tasks
- **List** - shows a list of tasks
- **Calendar** - shows a calendar with tasks as time slots
- **Mind Map** - shows the task hierarchy in a mind map format
- **Wall** - shows unseen notifications
- **Gantt**[1] - shows a list of tasks as a Gantt chart

It is important to note that views like *Gantt*, *Calendar*, and *Dashboard* require time data to be filled in; otherwise, the time slot of the task cannot be visualized.

This thesis is focused on the default view, called *List*. This particular view offers the user a list-like perspective, with each of the tasks represented by a separate row. The hierarchy of tasks is visualized by a horizontal offset in the rows.

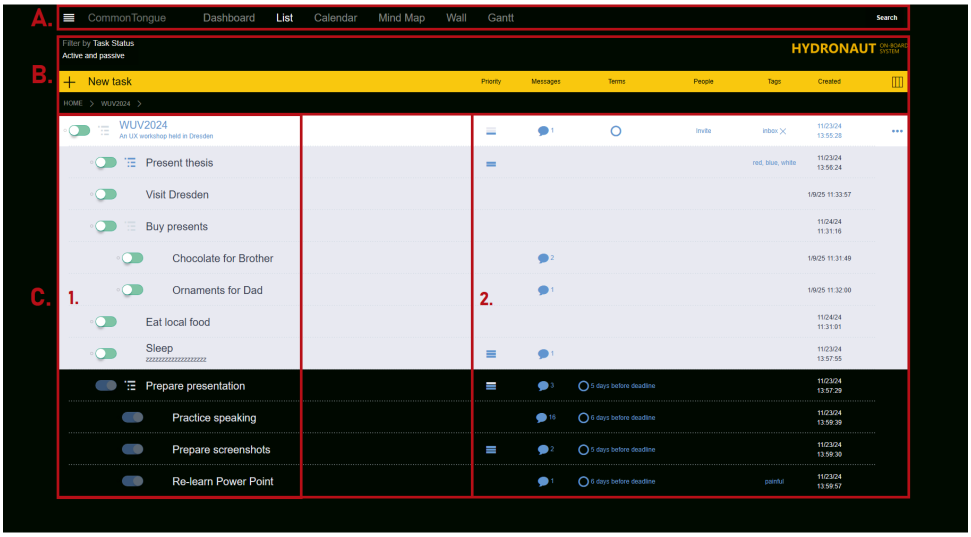
The screenshot in Figure 2.1 is divided into four sections according to their function, which will be described in a *left-to-right* and *top-to-bottom* approach.

Section A serves as the main bar, which is used to access the CommonTongue menu, switch between data views, and search data in the list.

Section B is the table header, which allows the user to change the sorting order of the list, enables the creation of new tasks, and provides labeling for each of the data columns. Underneath the label is the breadcrumb navigation bar, used to visualize the task hierarchy.

Section C represents the list portion of the interface, and consists of two parts, section C1 and C2. Section C1 contains the task activity toggle, hierarchy controls, and task names and descriptions. Section C2 contains

[1] `https://www.gantt.com/`



	Priority	Messages	Terms	People	Tags	Created
WUV2024 An UX workshop held in Dresden		1	○	Invite	inbox ×	11/23/24 13:55:28
Present thesis					red, blue, white	11/23/24 13:56:24
Visit Dresden						1/9/25 11:33:57
Buy presents						11/24/24 11:31:16
Chocolate for Brother		2				1/9/25 11:31:49
Ornaments for Dad		1				1/9/25 11:32:00
Eat local food						11/24/24 11:31:01
Sleep zzzzzzzzzzzzzzzzzz		1				11/23/24 13:57:55
Prepare presentation		3	○ 5 days before deadline			11/23/24 13:57:29
Practice speaking		16	○ 6 days before deadline			11/23/24 13:59:39
Prepare screenshots		2	○ 5 days before deadline			11/23/24 13:59:30
Re-learn Power Point		1	○ 6 days before deadline		painful	11/23/24 13:59:57

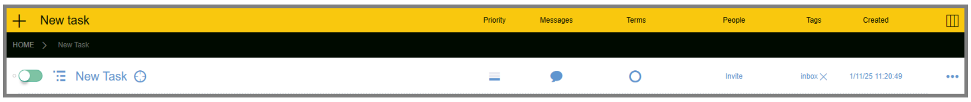
Figure 2.1: Example of CommonTongue interface, A. page header; B. table header; C. task names, hierarchy controls; D. task data

columns of the task data. The default columns contain task priority, messages, and deadlines, but more columns can be added in the table header. Additional columns can display data such as the list of people assigned to the task creation date or the tags assigned to the task. A large amount of whitespace separating the two sections is also visible.

The data used to fill the system in Figure 2.1 contains a sample scenario (which will be referenced as the *conference scenario*), where the CommonTongue user planned their upcoming scientific conference trip, with tasks like *Visit Dresden* and *Prepare presentation* serving as sample tasks in the list. This scenario was specifically created to serve as comparison data for the different project planner applications and will be used in Chapter 2.3.

2.2.2 Tasks

There are two main features of the CommonTongue system. First is the ability to create complex hierarchies of tasks and share them with other users online. The idea is similar to the idea of a shared file explorer in an operating system with tasks acting as both folders and files. Since the tasks themselves can contain metadata (the name of the task, priority, date of creation, etc.), the hierarchy can be organized in a similar fashion to the user's desktop or work folder. A single task can be seen in Figure 2.2.



	Priority	Messages	Terms	People	Tags	Created
New Task			○	Invite	inbox ×	1/11/25 11:20:49

Figure 2.2: New task in the task list

The second feature is the ability to activate and deactivate a task. With this feature, the user can create their personal lists of tasks and mark the completed tasks as inactive (which can be seen in the sample *conference scenario*). The other use becomes clear when working in groups. The main idea is that the leader deactivates themselves in the task until another group member activates them again, thus notifying the leader of the group that the task has progressed and should be checked. This situation is shown in Figure 2.3. The activity toggle is the main distinction between the CommonTongue system and its competitors.

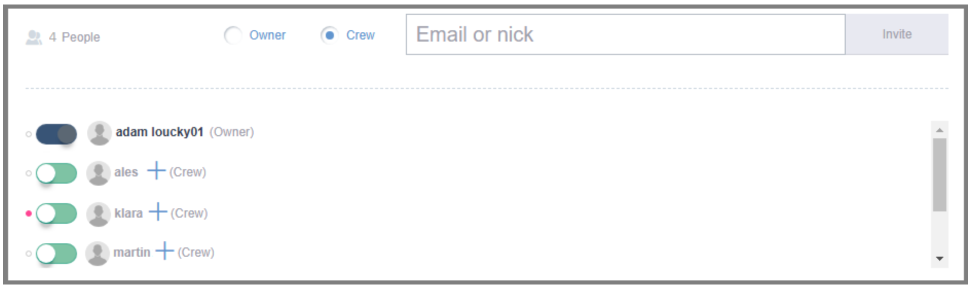


Figure 2.3: Detail of the activity management section

Tasks are added to the list by pressing the *New Task* button, or by pressing *Enter* on the keyboard. After adding a new task, the name of the task must be entered. Adding the name confirms its creation, and then the task becomes fully interactive. If the name isn't entered, the task creation is canceled.

Each of the tasks has a detailed view (visible in Figure 2.4), which allows the modification and adding of task data. There are six main tabs in the task detail, each of which is focused on a different type of task data. The detailed view of a task can be accessed by clicking on the task and is the primary way to modify and add data to the system.

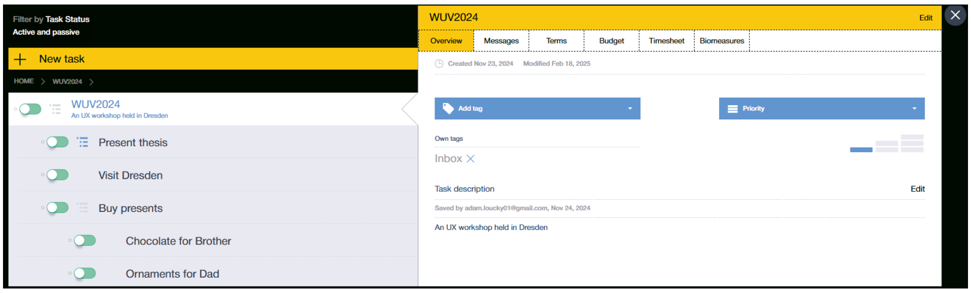


Figure 2.4: Task detail window showing the Overview tab

There are multiple ways to interact with the example task in Figure 2.2, which change depending on the clicked column of the list. The interaction options are (from left to right):

- **Activity button** - toggles task activity for the user
- **Hierarchy button** - toggles visibility of sub-tasks

- **Name** - opens the task detail on the message, double-click allows name editing
- **Target button** - focuses the task and shows its sub-tasks
- **Priority** - opens the Overview tab detail
- **Messages** - opens the Messages tab detail
- **Terms** - opens the Terms tab detail
- **Task menu/Three dots** - opens the task menu, which allows adding sub-tasks or deleting, copying and cutting

2.2.3 Other features

There are three main quality-of-life features in the system that notify the user when changes occur in their active tasks. The first feature is the recoloring of the message and term icons to orange when there are unread messages or upcoming deadlines. This is depicted in the Figure 2.5. Another feature in the figure is the red *unread changes* circle, which is located to the left of the activity button and becomes visible when an active task is modified in any way.

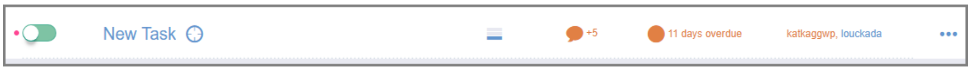


Figure 2.5: Task with unread messages and overdue deadline, both icons are highlighted in orange

The last of the features is related to the People column. People who have unread messages are marked orange; otherwise, their name is blue. This aims to help the user grasp which of the team members read their message and which didn't.

2.3 State-of-the-art analysis

Since *products don't exist in a vacuum* (Kuniavsky [5]) and often influence each other, a competitive research study had to be done before redesigning. Four other project organizer applications were examined. Each of the applications is different and targets different niches, which means that the analysis will focus on describing the core principles of the individual systems while comparing them to CommonTongue. This analysis also reveals what features users expect when using an organizing system.

2.3.1 Slack

Slack (Figure 2.6) is a platform focused on instant communication within a business, with features such as instant messaging threads with voice, image, and video uploads, video call functionality, and direct messaging functionality.

This means that it serves as a communication channel that replaces emails and allows workers within a business to communicate in a fashion similar to those using other personal messaging applications.

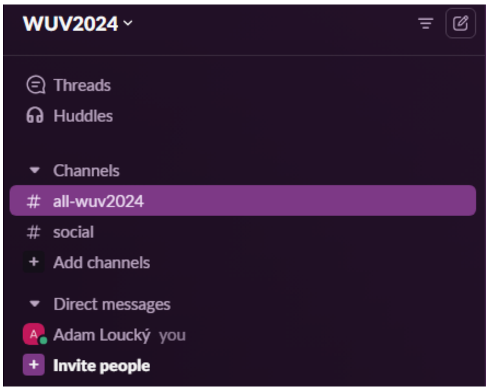


Figure 2.6: Example of a Slack group directory

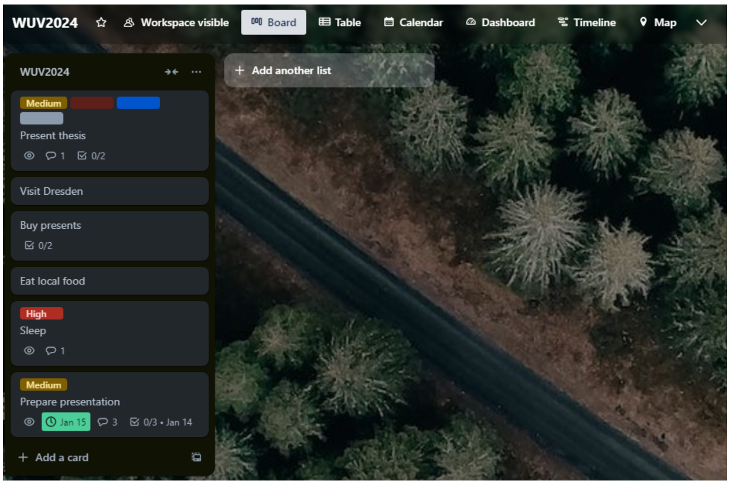
Although the usage of Slack and CommonTongue are different, there are similarities in the communication aspects of both systems. Individual tasks in CommonTongue are similar to channels in Slack, since both allow communication between dedicated groups or teams. Some core ideas are also similar, since both systems aim to reduce email communication and replace it with in-app messages. This also means that Slack could be a valid benchmark for comparing the messaging system functionality of CommonTongue.

2.3.2 Trello

Trello aims to help the user organize their tasks by putting them on a virtual task board. The main view consists of a board with several lists, which contain sets of tasks (Figure 2.7). The tasks in the lists can hold their own data; for example, tags, deadlines, checkbox lists, or text notes to comment on the task. Multi-person collaboration is also a feature, which means that Trello is also usable for team projects.

Trello contains multiple views of the data, which means that a layout similar to CommonTongue's List can also be used, although the options for displaying data and customizing columns are limited compared to the board view. This can be seen in Figure 2.8.

Overall, Trello seems well suited for project organization, although the communication with other users is quite limited, since users only communicate through comments. This is similar to the CommonTongue communication system, where direct chatting isn't possible. The main difference between the systems is the prioritization of a particular view, with CommonTongue favoring the List layout, and Trello favoring the Board layout.



WUV2024

Medium
Present thesis
1 0/2

Visit Dresden

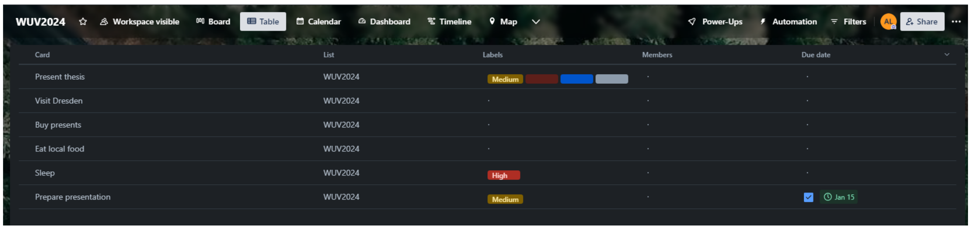
Buy presents
0/2

Eat local food

High
Sleep
1

Medium
Prepare presentation
Jan 15 3 0/3 • Jan 14

Figure 2.7: Example of the conference scenario visualized in Trello's board view

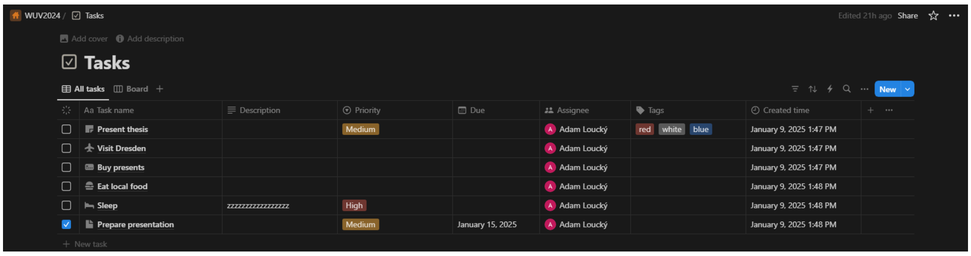


Card	List	Labels	Members	Due date
Present thesis	WUV2024	Medium	·	·
Visit Dresden	WUV2024	·	·	·
Buy presents	WUV2024	·	·	·
Eat local food	WUV2024	·	·	·
Sleep	WUV2024	High	·	·
Prepare presentation	WUV2024	Medium	·	☑ Jan 15

Figure 2.8: Example of the conference scenario visualized in Trello's table view

2.3.3 Notion

Notion focuses on creating a workspace where the user can organize their tasks either by themselves or in large groups. The core idea is similar to a note taking app with a highly customizable dashboard. There are multiple ways to view the same data – for viewing tasks, one can view them as a list (Figure 2.9), calendar or a task board (similar to Trello), which allows multiple different workflows and is in turn similar to the CommonTongue system. The communication in Notion is based on a comment system, where members of the workspace can comment on elements of a document to create message threads, which is different from direct messaging applications like Slack, for example.



Tasks

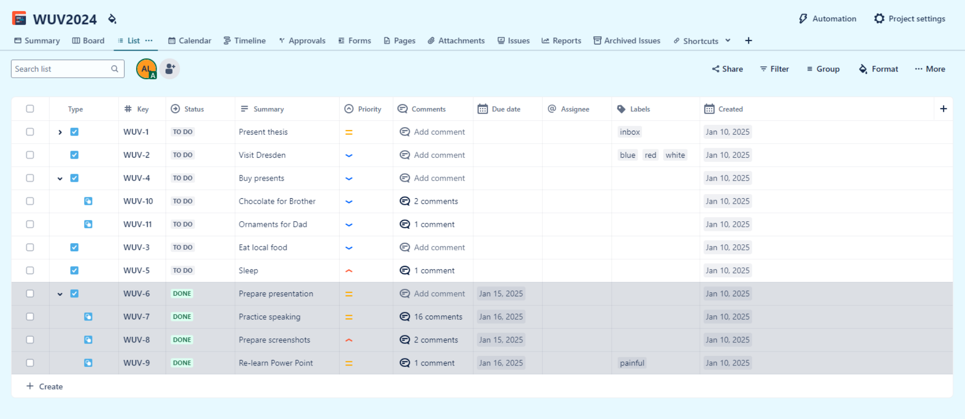
	Task name	Description	Priority	Due	Assignee	Tags	Created time
□	Present thesis		Medium		Adam Loucký	red white blue	January 9, 2025 1:47 PM
□	Visit Dresden				Adam Loucký		January 9, 2025 1:47 PM
□	Buy presents				Adam Loucký		January 9, 2025 1:47 PM
□	Eat local food				Adam Loucký		January 9, 2025 1:48 PM
□	Sleep	zzzzzzzzzzzzzzzzz	High		Adam Loucký		January 9, 2025 1:48 PM
☑	Prepare presentation		Medium	January 15, 2025	Adam Loucký		January 9, 2025 1:48 PM

Figure 2.9: Example of the conference scenario visualized in Notion using the Tasks layout

Notion is an application with a high density of features. Since there are multiple different project types, multiple different views, and multiple customization options for each view, the experience of working with Notion was quite overwhelming. CommonTongue lacks these vast customization options and the interface is quite static. The customization in CommonTongue boils down to adding and removing columns from the list, and creating new tags. Comparatively, the system is significantly less customizable than the options offered by Notion.

2.3.4 Jira

Jira is a software tool designed to manage and track project work. Users create projects with tasks or issues, which can be assigned to members of the project. Similarly to other already examined applications, the tasks can hold their own data and can be displayed in multiple views like a calendar, timeline, task list, or a Kanban board. Jira also offers integration with other Atlassian Corporation systems, like Bitbucket (which allows Git operations) and Confluence (for documentation).



□	Type	Key	Status	Summary	Priority	Comments	Due date	Assignee	Labels	Created
□	› ☑	WUV-1	TO DO	Present thesis	=	Add comment			inbox	Jan 10, 2025
□	☑	WUV-2	TO DO	Visit Dresden	⌄	Add comment			blue red white	Jan 10, 2025
□	⌄ ☑	WUV-4	TO DO	Buy presents	⌄	Add comment				Jan 10, 2025
□	⧉	WUV-10	TO DO	Chocolate for Brother	⌄	2 comments				Jan 10, 2025
□	⧉	WUV-11	TO DO	Ornaments for Dad	⌄	1 comment				Jan 10, 2025
□	☑	WUV-3	TO DO	Eat local food	⌄	Add comment				Jan 10, 2025
□	☑	WUV-5	TO DO	Sleep	^	1 comment				Jan 10, 2025
□	⌄ ☑	WUV-6	DONE	Prepare presentation	=	Add comment	Jan 15, 2025			Jan 10, 2025
□	⧉	WUV-7	DONE	Practice speaking	=	16 comments	Jan 16, 2025			Jan 10, 2025
□	⧉	WUV-8	DONE	Prepare screenshots	^	2 comments	Jan 15, 2025			Jan 10, 2025
□	⧉	WUV-9	DONE	Re-learn Power Point	=	1 comment	Jan 16, 2025		painful	Jan 10, 2025

Figure 2.10: Example of the conference scenario visualized in Jira using the List layout

As can be seen in Figure 2.10, the conference scenario was almost fully replicable in Jira, with all CommonTongue features being represented in some way, even allowing the creation of sub-tasks. Overall, Jira is the most similar to the CommonTongue system and therefore should be looked at as the main competitor.

2.3.5 Conclusion

The interface and functionality of the CommonTongue system was described in detail and then compared to other contemporary state-of-the-art project planning systems. The outcome of this analysis was knowledge of the system's intended usage and also features used to reinforce this usage.

The main competitors in the organizer scene were also discovered, which could serve as motivation to differentiate the system from its competitors by reinforcing the intended workflow using a new interface design. The quality of CommonTongue features could also be compared with other project planners in the future to ensure the system's competitiveness.

Chapter 3
List design

This chapter will detail the issues found while analyzing the List view in the initial analysis in Chapter 2.1, and the changes that were proposed to improve the system to minimize the impact of these issues. There were four main aspects of the interface that stood out in the initial analysis and needed to be evaluated. These key aspects were the *table header layout*, *table spacing*, *inactive task functionality*, and *visual consistency*. Each of these items will be evaluated from the perspective of the intended program usage and usability/data visualization standards, and if they are marked as issues, a proposed solution will be showcased using a design in Figma. The reasoning behind the evaluation will also be justified. The structure of the individual sections will be as follows:

- **Description** - brief description of the area
- **Potential issues** - description of the problematic elements
- **Solutions** - explored solutions to the issues

3.1 Table header

The first evaluated section was the table header area (marked as section B. in the interface description chapter in Figure 2.1). This section contains:

- **Sorting filter** - control for task sorting
- **Table header** - creating new tasks, column labels and a column filter
- **Breadcrumb navigation bar** - shows the path to the selected task

3.1.1 Potential issues

The overall construction of the header is quite similar to the traditional table layout (as seen in systems like Notion or Jira in the previous analysis 2.3), with the header labeling each of the columns and additional controls of the table being above the label. However, there are two main differences.

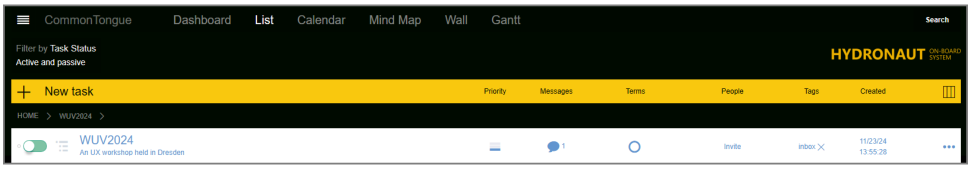
The hierarchy navigation bar is not present in the other project planners, and its placement below the table header may be problematic, since it disturbs one of the 10 Usability Heuristics proposed by Jakob Nielsen [4] – the Consistency and Standards heuristic. In short – external industry conventions should be followed (unless the change is intentional), which in this case means the label of a table should be immediately above the content. In this way, the content and the header of the table are clearly associated, which decreases the cognitive load of working with the interface. The disconnect between the table and the content is also further deepened by the black background of the hierarchy navigation bar.

The second difference is the inclusion of the *New Task* button in the header, instead of the bottom of the list (like in Notion, Jira, or Trello). Unlike the first change, this change makes sense for the use case of CommonTongue. Since the lists of tasks can often be very long and can span off-screen, the addition of a new task button at the top can be beneficial. The issue; however, arises when the *New Task* button is located directly in the header. Its location could disturb the perceived function of the header as a label for the columns and also occupies space where labels for the activity button and task name should be, which means the same usability heuristic is disturbed (this time from an issue of internal consistency).

The space between the task names and the first column is also alarming, as the screen space needs to be used efficiently, but screen space efficiency is a complex issue and will be addressed in its own chapter – Chapter 3.2.

3.1.2 Solutions

In the previous section, two key issues were encountered – the header placement and the *New Task* button placement. These issues will be addressed by a set of solutions, and later a final version, containing the final design, will be chosen. The original design can be seen in Figure 3.1.

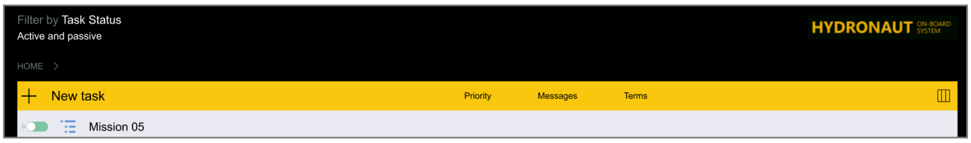


Filter by Task Status
Active and passive

New task	Priority	Messages	Terms	People	Tags	Created
WUV2024 An UX workshop held in Dresden		1		Invite	inbox	11/23/24 13:55:28

Figure 3.1: The table header, marked as section B. in Figure 2.1

The first version of the new design can be seen in Figure 3.2. The table header was simply swapped with the hierarchy bar, which should clearly connect the header to the table. The effects of this change will be observed during testing (Chapter 4), particularly to identify if the hierarchy navigation bar still serves its purpose after the swap.



Filter by Task Status
Active and passive

New task	Priority	Messages	Terms
Mission 05			

Figure 3.2: The swapped version of the table header section

The second set of changes focused on the *New Task* button placement, with the goal of making room for the new column labels and therefore making the header unified in function. The first change can be seen in Figure 3.3 – a new row was added to the header and additional column labels were added. The new row simulates the look of a new task, with the goal of subconsciously telling the user the function of the button, even before an interaction occurs. The description of the hotkey for adding a new task (Enter) was also added to the new row, which should help the user learn to work with hotkeys in the program.

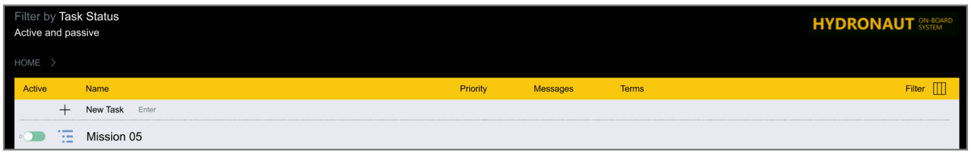


Figure 3.3: The table header section with the *New Task* button in a new row

A second version of the *New Task* button was also created (Figure 3.4), with the button moved to the right side of the header. This version wasn't used in the final design, since it took up space for additional columns. The button would also move off the screen if too many columns were added and horizontal scrolling would be necessary to reach it.

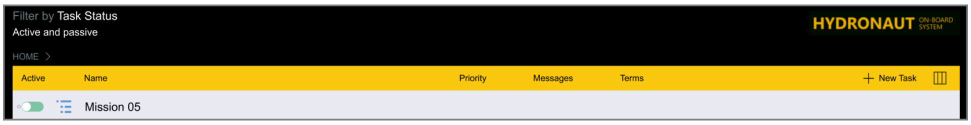


Figure 3.4: The table header section with the *New Task* button moved to the right

After failing to create a viable alternative design, the design in Figure 3.3 was expanded – the *new task* row was recolored to a more yellow tint (Figure 3.5). This made the button a color that was a cross between the header's yellow and the task list's white, and was an important step in the design process. It was decided that the standard Hydronaut yellow would become the color of the control elements of the interface, with the table header, *New Task* button and also the task edit menu all being yellow. This follows a fundamental visualization rule, which states that in order to visualize similarity in function or concept, color can be used to group objects (Ware [3], Chapter 3).

The final design, which was selected to fix the detailed issues, was the design seen in Figure 3.5. The additions made to the interface follow design principles and should remedy the issues found during the analysis. Their effectivity will also be observed during the testing in Chapter 4.

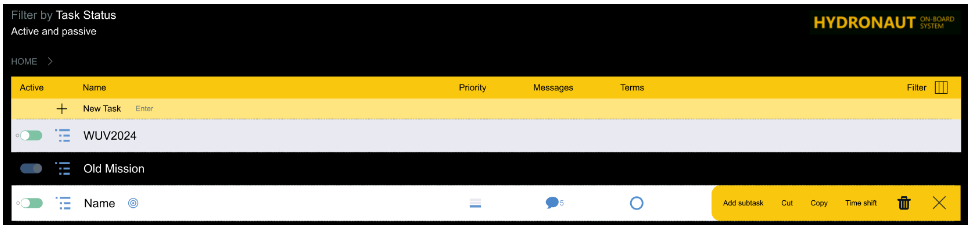
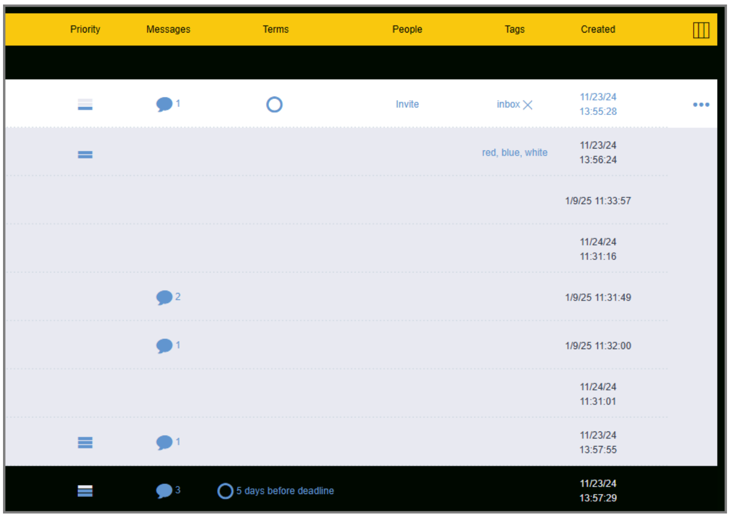


Figure 3.5: Unified color for table header and the recolored *New Task* row and task menu

3.2 Horizontal spacing

The second area to be evaluated is the table itself, which was previously labeled section C2 in the sectioned Figure 2.1. This section's analysis differed from the others, as some issues were already identified by the development team before the beginning of this thesis, the main complaint being that *a small amount of tasks and columns fit the screen.* This was explored during the initial analysis and the claims were confirmed, with six data columns being the maximum before the need for horizontal scrolling. This could be remedied by simply moving the data column section left, which would create more space for data but limit the maximum length of task names. In the end, the columns were slightly shifted to the left, but another approach was also explored in tandem with the shift.

The table in Figure 3.6 contains several columns of data of different types, some visualized as icons and some as text. The visibility of the columns can be toggled using the filter menu in the top right, which means the interface is configurable and needs to be carefully designed in order to work in multiple configurations.



Priority	Messages	Terms	People	Tags	Created
≡	💬 1	○	Invite	inbox ×	11/23/24 13:55:28
≡				red, blue, white	11/23/24 13:56:24
					1/9/25 11:33:57
					11/24/24 11:31:16
	💬 2				1/9/25 11:31:49
	💬 1				1/9/25 11:32:00
					11/24/24 11:31:01
≡	💬 1				11/23/24 13:57:55
≡	💬 3	○ 5 days before deadline			11/23/24 13:57:29

Figure 3.6: The table columns, marked as section C. in Figure 2.1

3.2.1 Potential issues

The overall structure of the table itself (seen in Figure 3.6) is very similar to the table views in all the competitor project planners. The table has distinct rows and sometimes contains icons, and the data in the columns is varied. The main difference is the lack of visible columns or column lines.

Both Notion (Figure 2.9) and Jira (Figure 2.10) divide their tables into distinct cells, creating a traditional table. Trello (Figure 2.8) on the other hand, treats the tasks as individual rows and separates the columns with a vertical gap, which is similar to the way CommonTongue splits up columns. However, the Trello table contains only five distinct columns, which means that separation might not be necessary in Trello. This is not the case for CommonTongue, as the amount of columns on one screen can typically be larger than five.

Vertical gaps are also less space efficient than other separation methods, since the human eye needs more space to separate a set of objects into groups (according to the proximity principle of Gestalt psychology [7], Figure 3.7). Since one of the default complaints was a lack of spatial efficiency, an alternative to the vertical gap should be explored in the following Solutions subsection.

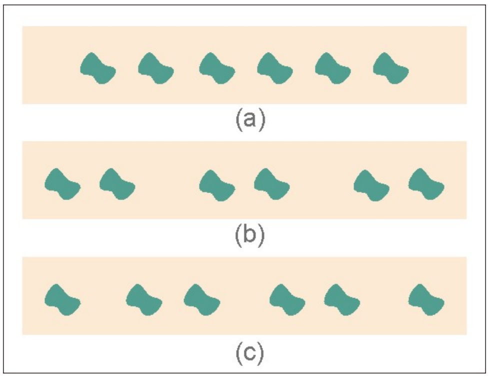


Figure 3.7: Proximity principle explanation by Dejan Todorovic [7], (a) is viewed as a line, (b) is viewed as three couples, (c) is viewed as four groups

Another potential issue is the amount of blank space in the data section. This happens because the icons for messages, priority, and terms appear only if the task is hovered or if the data is filled in. This can be beneficial for the program, since it utilizes the pop-out effect [3], which is a psychological phenomenon that allows for very fast searching if the searched object differs from the surroundings in some way (demonstrated in Figure 3.8). In the CommonTongue case, the icon is either there (it contains data) or not there (it doesn't contain data). This, however, can cause issues with the alignment of columns and rows, since missing icons and data can break up the user's perception of what is and what isn't in a column (according to the proximity principle cited before). This means that the interface should contain another method that would help identify columns.

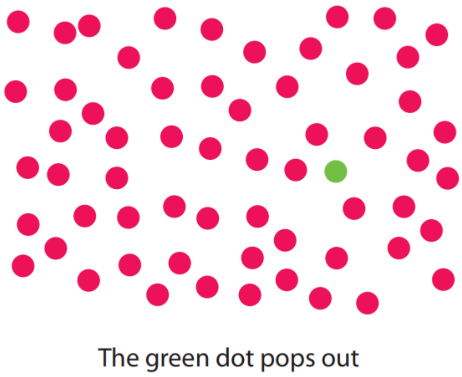


Figure 3.8: Pop-out effect demonstrated by color from Colin Ware [3]

3.2.2 Solutions

Since only a single issue was identified, multiple solutions were created in order to properly explore the options for reducing vertical space between columns. Since the current implementation involved vertical gaps, another idea, originating from gestalt principles, could be used. The common region principle [7] is a principle that says that elements tend to be grouped together if they are located within the same closed region (Figure 3.9).

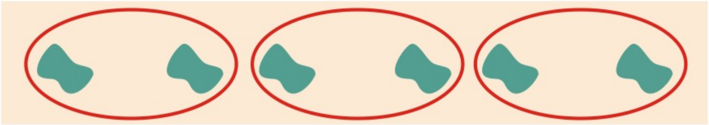


Figure 3.9: Common region principle explanation by Dejan Todorovic [7]

The first version of changes involved cell backgrounds and borders, which would create a system of data cells. The cells could then be grouped closer together, since the data was enclosed in a separate area. This change went through three iterations, in which different combinations of backgrounds and borders were designed. The distance between the columns was also experimented with. The versions can be seen in Figure 3.10.

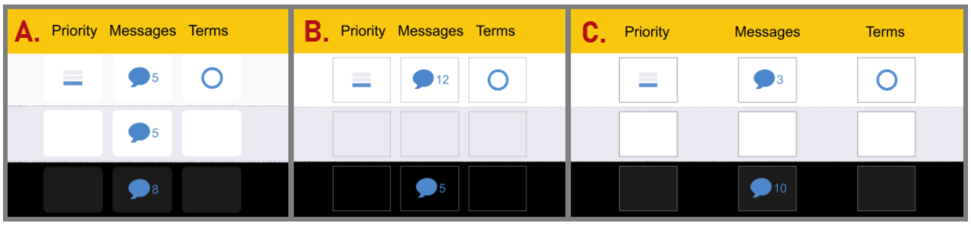


Figure 3.10: Three versions of the data cell change, A. background version, B. border version, C. combination version

The second version abandoned the concept of cell backgrounds and instead used vertical separators, which would hopefully create enough separation for the columns to be viewed as separate. From the perspective of the common region principle, this would mean that each column and row would be its own region, which would support both column searching and row searching. A total of three separator versions were created.

The first is a traditional line separator that spans the entire table vertically and is similarly salient (salient meaning pronounced, visually distinct) to the horizontal separator line. The second version used a checkered design, which is less salient than the horizontal line, which means that rows would be more pronounced than columns. The last version is a combination of both previous designs, where each of the cells has its own separation line. This creates a semi-checkered line, which the user perceives as a full line when focused (according to the closure principle of Gestalt principles [7]).

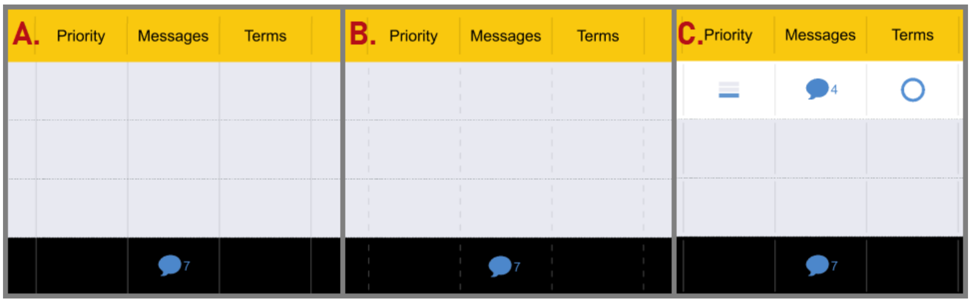
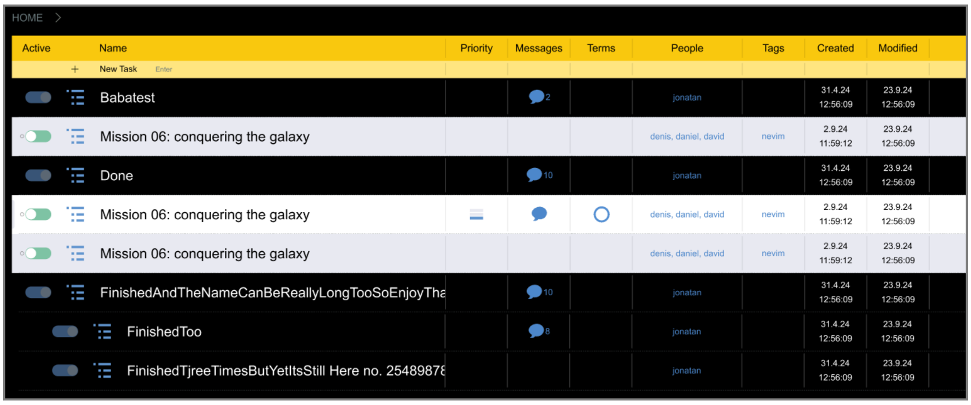


Figure 3.11: Three versions of the separators, A. full line version, B. checkered version, C. semi-checkered version

Finally, a combination of all the changes made so far was created to create a checkpoint in the design development. This version swaps the header and navigation bar, adds a new row for the *New Task* button, moves the columns to the left, and adds the semi-checkered separator. This creates space for two to three additional columns and makes the data more compact and readable, since the separators guide the user's eyes along the columns. This should compensate for gaps in the data and preserve the pop-out effect created by the gaps.

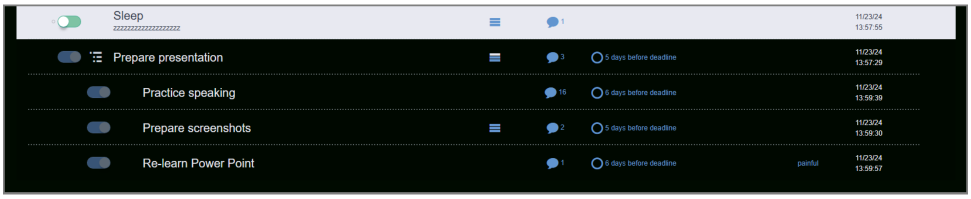


Active	Name	Priority	Messages	Terms	People	Tags	Created	Modified
+	New Task Enter							
	Babatest		💬 2		jonatan		31.4.24 12:56:09	23.9.24 12:56:09
	Mission 06: conquering the galaxy				denis, daniel, david	nevim	2.9.24 11:59:12	23.9.24 12:56:09
	Done		💬 10		jonatan		31.4.24 12:56:09	23.9.24 12:56:09
	Mission 06: conquering the galaxy	≡	💬	○	denis, daniel, david	nevim	2.9.24 11:59:12	23.9.24 12:56:09
	Mission 06: conquering the galaxy				denis, daniel, david	nevim	2.9.24 11:59:12	23.9.24 12:56:09
	FinishedAndTheNameCanBeReallyLongTooSoEnjoyTha		💬 10		jonatan		31.4.24 12:56:09	23.9.24 12:56:09
	FinishedToo		💬 8		jonatan		31.4.24 12:56:09	23.9.24 12:56:09
	FinishedTjreeTimesButYetItsStill Here no. 2548987ٰ				jonatan		31.4.24 12:56:09	23.9.24 12:56:09

Figure 3.12: List interface combining changes from sections 3.1 and 3.2

3.3 Inactive tasks

The third problematic area of the program are the inactive tasks (included in Section C in the overview in Figure 2.1). Inactive tasks represent a task from which the user doesn't receive notifications, and they differ from the active tasks in the background color and text color. Their status is represented by the activity toggle on the left side of the task. Otherwise, inactive tasks have the same structure as active tasks (visible in Figure 3.13).



Sleep zzzzzzzzzzzzzzzzzz	≡	1			11/23/24 13:57:55
Prepare presentation	≡	3	○ 5 days before deadline		11/23/24 13:57:29
Practice speaking		16	○ 6 days before deadline		11/23/24 13:59:39
Prepare screenshots	≡	2	○ 5 days before deadline		11/23/24 13:59:30
Re-learn Power Point		1	○ 6 days before deadline	painful	11/23/24 13:59:57

Figure 3.13: The inactive tasks, marked as section C. in Figure 2.1

3.3.1 Potential issues

The inactive tasks have a different color from active tasks, which is enough to distinguish them as separate (as discussed in Section 3.1). The task background color, however, is poorly chosen. The main issue is that the task color matches the color of the page background. This blends the inactive tasks with the page, and there isn't a clear separation between the table and the background.

Another issue is the saliency of the tasks. Colin Ware states that: *The most important single principle in the use of color is that whenever detailed information is to be shown, luminance contrast is necessary* [3]. This also means that the larger the contrast, the more salient the object is. This, however, is used poorly in the CommonTongue interface. Since the inactive tasks are black and use a white font color, the contrast difference is significantly larger than for the active tasks, which use a very light gray and black. This means that the inactive tasks actually become more salient than the inactive ones, which is counterintuitive for the CommonTongue use case, where the inactive tasks represent muted notifications. It is also important to use colors carefully, especially if an emphasis or hover functionality is implemented. *A common mistake, often seen in PowerPoint slides, is to highlight something using color in such a way that luminance contrast is reduced* [3].

The last detail is apparent when examining the activity toggle on the left side of the task. The task is considered active when the toggle is turned to the left. This is unusual since most activity toggles are activated when turned to the right. Since the standard is different from the CommonTongue implementation, and the difference in left and right doesn't serve a special purpose in the interface, it should be considered an error.

3.3.2 Solutions

The inactive task issues were solved by a simple recoloring. The inactive task background was recolored dark gray, which reduced the contrast between the text and the background. The text was also darkened to a lighter gray color that is still readable, but noticeably muted when compared to the active tasks. The inactive task becomes more salient when hovered, with the text turning white and the background becoming slightly brighter.

The changes aimed to reduce the contrast of the inactive tasks, and also make the tasks discernible from the background, thus creating a clearer silhouette of the table. The inactive task size was also reduced, in order to make them physically feel less salient and to fit more tasks on one screen vertically. The activity button was also flipped to represent the common usage of the activity toggle. The new inactive tasks can be seen in Figure 3.14.

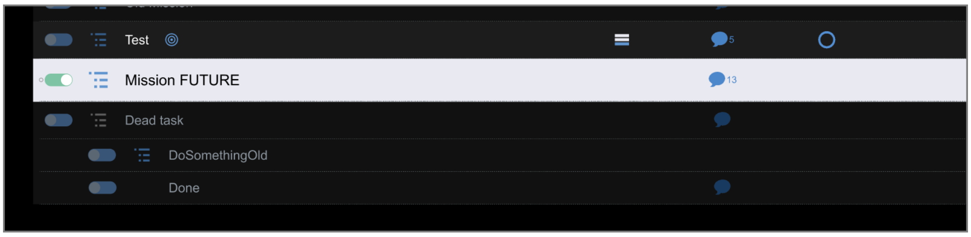


Figure 3.14: Recolored inactive tasks with flipped activity toggle, inactive tasks are 20% smaller than active tasks

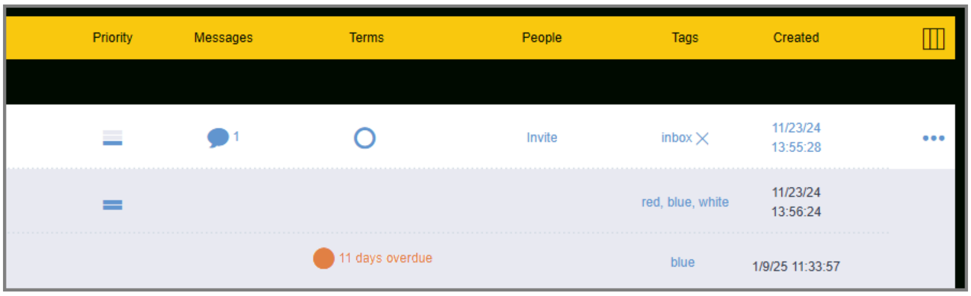
It will be important to observe whether the changes made to inactive tasks are too drastic or not, since there were three different changes that made the tasks less salient.

3.4 Visual consistency

The last identified problem in the original interface is similar to the one in Section 3.3 and concerns visual consistency in the table data region. This region contains various types of data represented by icons, text, or a combination of both. The data is stored in columns and rows and is usually blue in color (visible in Figure 3.15), with occasional orange and black/white.

3.4.1 Potential issues

Blue color is usually associated with hyperlinks in web design, and in CommonTongue, it is used to represent interactive elements. Any coloring, however, only makes sense if used consistently in the entire user interface, which means similar interactive elements should all be colored in the same way, or not colored at all. This way, the internal consistency within the program is preserved. This is not the case for CommonTongue, since all data columns and cells in the interface are interactive, but aren't



Priority	Messages	Terms	People	Tags	Created	▥
☰	💬 1	○	Invite	inbox ×	11/23/24 13:55:28	•••
☰				red, blue, white	11/23/24 13:56:24	
		● 11 days overdue		blue	1/9/25 11:33:57	

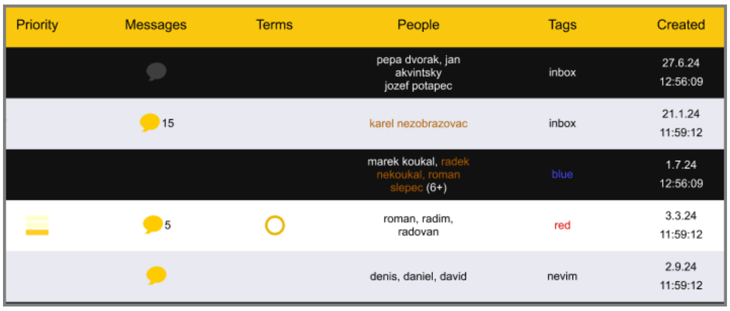
Figure 3.15: An example of coloring in CommonTongue

colored consistently. An example can be seen in Figure 3.15, where the *Created* column contains black-colored text, which should indicate no interactivity, even though the column is similarly interactive to other columns.

Another concern may be the usage of icons, since the interface uses a combination of line and glyph icons (outlines and filled-in icons) and also rounded and sharp icons. This is also visible in Figure 3.15, where the messages icon is a glyph, the terms icon is an outline, and the column filter icon is a thin and sharp outline. In the same spirit of internal consistency, similar icon design should probably be used where possible.

3.4.2 Solutions

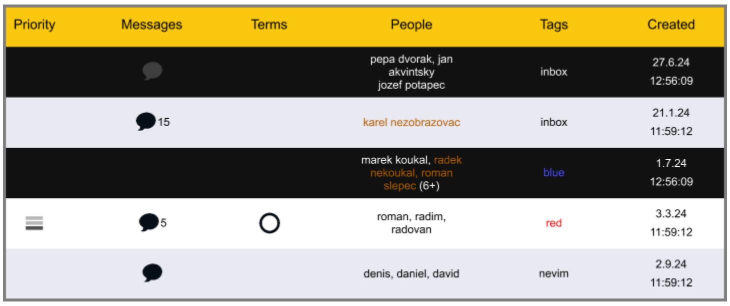
In order to make the interface more consistent, elements of the interface needed to be recolored. First, a combination of yellow for icons and black for data was chosen (Figure 3.16). The reasoning was mainly to preserve consistency with other control elements, but in the end, yellow was a poor choice for active tasks, since the brightness contrast was quite low and the icons weren't salient enough. The color difference between the text and icons also made the text seem non-interactive.



Priority	Messages	Terms	People	Tags	Created
	💬		pepa dvorak, jan akvintsky jozef potapec	inbox	27.6.24 12:56:09
	💬 15		karel nezobrazovac	inbox	21.1.24 11:59:12
			marek koukal, radek nekoukal, roman slepec (6+)	blue	1.7.24 12:56:09
☰	💬 5	○	roman, radim, radovan	red	3.3.24 11:59:12
	💬		denis, daniel, david	nevim	2.9.24 11:59:12

Figure 3.16: Recolored interface, color changed to black and yellow

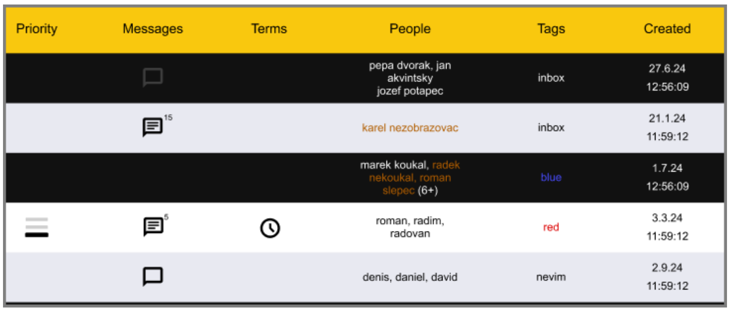
Another version was attempted (Figure 3.17), with the original blue color being replaced by gray or black (depending on if the task was active or not). This way, the text and the icons would both share a color and would therefore be seen as a part of a group of interactive elements. Recoloring the data to black and gray made the interface less noisy, but may cause the user to think that the table is not interactive, which means more thought has to be put into making the interface seem interactive (by other means than color).



Priority	Messages	Terms	People	Tags	Created
	💬		pepa dvorak, jan akvintsky jozef potapec	inbox	27.6.24 12:56:09
	💬 15		karel nezobrazovac	inbox	21.1.24 11:59:12
			marek koukal, radek nekoukal, roman slepec (6+)	blue	1.7.24 12:56:09
≡	💬 5	○	roman, radim, radovan	red	3.3.24 11:59:12
	💬		denis, daniel, david	nevim	2.9.24 11:59:12

Figure 3.17: Recolored interface, color changed to black and gray

A third version (Figure 3.18) with the icons changed was created, which aimed to use a more unified set of icons, as was addressed in the potential issues section. Outlined icons with rounded edges were chosen for the redesign because they were the most similar to the original style of the interface.



Priority	Messages	Terms	People	Tags	Created
	□		pepa dvorak, jan akvintsky jozef potapec	inbox	27.6.24 12:56:09
	▤ 15		karel nezobrazovac	inbox	21.1.24 11:59:12
			marek koukal, radek nekoukal, roman slepec (6+)	blue	1.7.24 12:56:09
≡	▤ 5	◷	roman, radim, radovan	red	3.3.24 11:59:12
	□		denis, daniel, david	nevim	2.9.24 11:59:12

Figure 3.18: Black and gray version of the interface with changed icons

The new icons also allow for more states to be expressed – an example is the message icon (Figure 3.19), which was originally only a one-state icon. The new icon has two versions, one for a state where there are no messages and one for the state where there are messages.

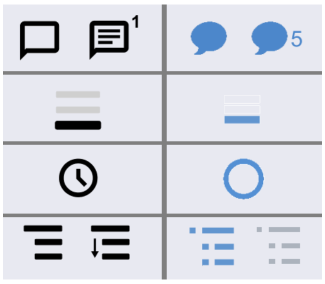
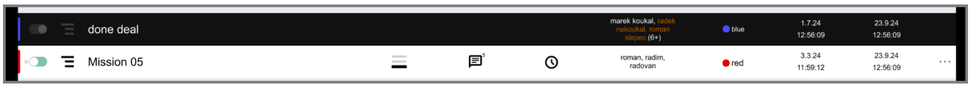


Figure 3.19: New versions of icons (left) compared to old versions (right)

A beneficial side effect of the removal of blue from the interface opened up the possibility of using colored tags. Since the new interface uses only three different hues of color, the design wouldn't be disturbed by colorful tags, and the tags themselves could aid in visual searching (via the pop-out effect [3]). An example of this future functionality can be seen in Figure 3.20.



done deal	marek koukal, radek nekoukal, roman slepec (6+)	● blue	1.7.24 12:56:09	23.9.24 12:56:09
Mission 05	roman, radim, radovan	● red	3.3.24 11:59:12	23.9.24 12:56:09

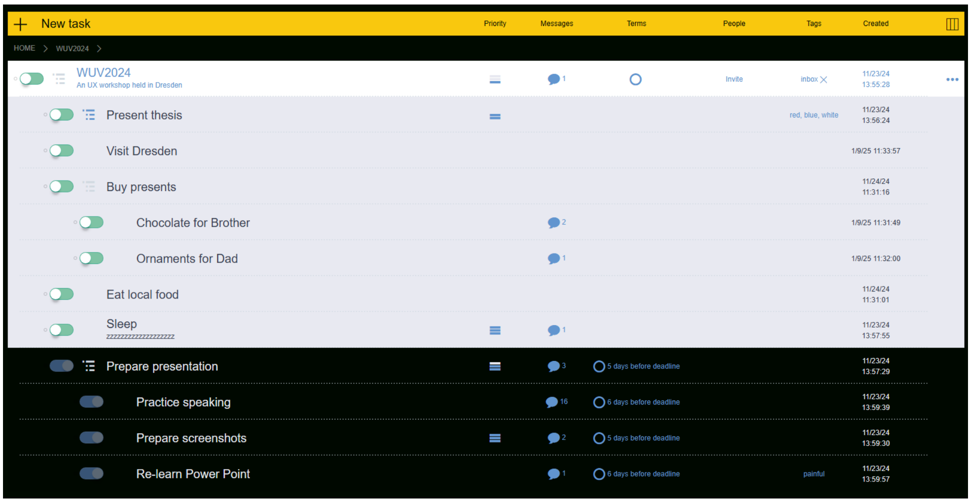
Figure 3.20: Two tasks tagged as blue and red, with ribbons on the left side

3.5 Resulting design

The combination of changes mentioned in Chapter 3 resulted in the creation of a new design for the List section of the website.

The header is now connected to the table, all columns are labeled, and the color of the interface and icons is consistent. The overall profile of the data columns is slimmer both vertically (since the height of tasks was changed) and horizontally (since separators were added). The direct comparison can be seen in Figures 3.21 and 3.22 on the next page.

The design will undergo user testing to fully evaluate its benefits compared to the old design. These tests will be described in the following chapter (Chapter 4).

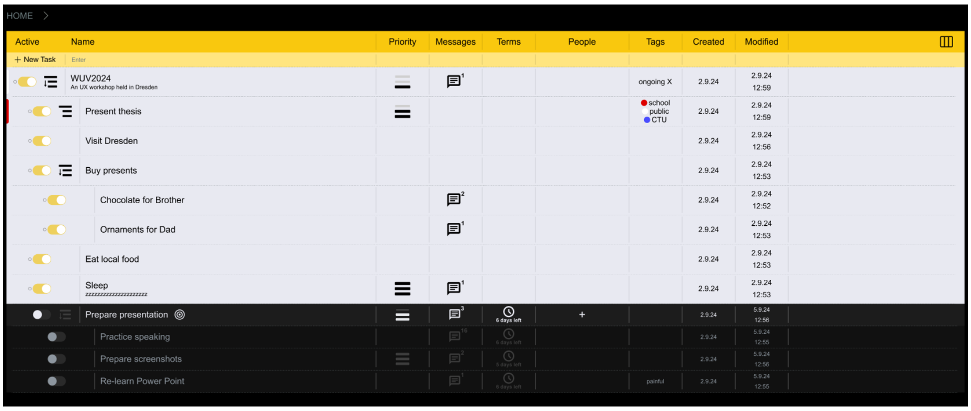


+ New task

HOME > WUV2024 >

Name	Priority	Messages	Terms	People	Tags	Created
WUV2024 An UX workshop held in Dresden	≡	1	○	Invite	inbox ✕	11/23/24 13:55:28
Present thesis	≡				red, blue, white	11/23/24 13:56:24
Visit Dresden						1/9/25 11:33:57
Buy presents						11/24/24 11:31:16
Chocolate for Brother		2				1/9/25 11:31:49
Ornaments for Dad		1				1/9/25 11:32:00
Eat local food						11/24/24 11:31:01
Sleep zzzzzzzzzzzzzzzzzz	≡	1				11/23/24 13:57:55
Prepare presentation	≡	3	○ 5 days before deadline			11/23/24 13:57:29
Practice speaking		16	○ 6 days before deadline			11/23/24 13:59:39
Prepare screenshots	≡	2	○ 5 days before deadline			11/23/24 13:59:30
Re-learn Power Point		1	○ 6 days before deadline		painful	11/23/24 13:59:57

Figure 3.21: The original design, showing the conference scenario



HOME >

Active	Name	Priority	Messages	Terms	People	Tags	Created	Modified
+ New Task	Enter							
	WUV2024 An UX workshop held in Dresden	≡	1			ongoing X	2.9.24	2.9.24 12:59
	Present thesis	≡				● school public ● CTU	2.9.24	2.9.24 12:59
	Visit Dresden						2.9.24	2.9.24 12:56
	Buy presents						2.9.24	2.9.24 12:53
	Chocolate for Brother		2				2.9.24	2.9.24 12:52
	Ornaments for Dad		1				2.9.24	2.9.24 12:53
	Eat local food						2.9.24	2.9.24 12:53
	Sleep zzzzzzzzzzzzzzzzzzzz	≡	1				2.9.24	2.9.24 12:53
	Prepare presentation ◎	≡	3	6 days left	+		2.9.24	5.9.24 12:56
	Practice speaking		16	6 days left			2.9.24	5.9.24 12:55
	Prepare screenshots	≡	2	5 days left			2.9.24	5.9.24 12:56
	Re-learn Power Point		1	6 days left		painful	2.9.24	5.9.24 12:55

Figure 3.22: The updated design, showing the conference scenario

Chapter 4

First impression testing

The design created in the previous chapter had to be evaluated from a user perspective in order to properly assess its usefulness as a replacement of the older design. User testing is also useful for generating new ideas, and the knowledge gained from the tests will be used to further improve the newer design.

The comparison between the old and new design was done using usability tests, a method of user testing based on human interaction with a program. To cite Kuniavsky: *The usability test will tell you whether your audience can use what you've made* [5]. The participants of the testing are usually selected from an initial group based on their backgrounds and their knowledge, in order to satisfy the test requirements, usually to represent the program's target user. These selected participants are then given a task to accomplish and are observed during their work with the system, after which they are interviewed regarding their feelings and questions they may have after completing the task.

This process leads to a better understanding of how the system can be used by someone without previous experience, a user of a competing product, or an expert user.

4.1 Test preparation

Before conducting the test, preparation is needed. A major influence on how the test was prepared was a book by Elizabeth Goodman, Mike Kuniavsky, and Andrea Moed called Observing the User Experience [5], which describes the process of creating a usability test in great detail in Chapter 11. The process is described as a series of steps, with recruiting, selecting the features to be tested, creating tasks for the test, and writing a script being the main preparation steps. The usability tests conducted as a part of this thesis were constructed according to these steps, each of which will be further expanded and described.

The initial impression testing was carried out online, using video chat to communicate with the participants. There are drawbacks to remote testing, but due to the nature of the test, the convenience of online testing outweighed the potential problems. Online testing allowed for easier scheduling of tests, which in turn resulted in more tests being carried out in a shorter time span than would be possible with in-person tests. The convenience of doing these tests remotely later allowed for an additional round of in-person tests, described in Chapter 6.

4.1.1 Recruiting

CommonTongue is a digital project planner and therefore will be used by people who work with computers on a day-to-day basis and are at least semi-skilled at using them. This can be assumed from the workflow of the system, which requires constant computer access – users message each other, read notifications, update information, etc. The selected group of people should therefore contain exactly those typical users – users with at least some computer skills who use their computers for work or in their free time.

Some of the people selected for the test should also have experience with daily organization (for example, having a schedule in their calendar), and some should instead have very little organizing experience. The same diversity should also be applied to criteria such as age and education, since the aim of CommonTongue is to be accessible to a wide array of users.

An initial five-question survey was created in order to gauge previous user experience along with information regarding education and age. The end goal of this survey is to sort the users into groups based on the information they provided. This sorting should create groups from which test participants will be picked. The survey contained the following questions:

- *What is your age?* - 18-25/26-32/33-50/50+
- *What is your education level?* - None/Elementary/Secondary/Tertiary
- *How often do you use your computer for work?* - Never/Several times a month/Once a week/Several times a week/Every day
- *How often do you use your computer in your free time?* - Never/Several times a month/Once a week/Several times a week/Every day
- *What do you use to organize your day?* - Nothing/Calendar/Organizing application/Other

Originally, the test included twelve people, but after seeing diminishing returns during the course of the testing, a test scale of eight participants was deemed sufficient.

4.1.2 Feature selection

The features to be tested in the usability test were selected based on the initial development of the new design. The goal of the test should be to verify that the new updated interface is as intuitive or better than the old interface, while also gauging the effectiveness of some of the individual changes.

Due to the goal being focused solely on the initial impression and readability, and less on interaction, an alternative to the standard usability test was chosen

– a test internally dubbed the *Screenshot test*. The main focus of this test is to show different versions of the interface with different sample data in each of the screenshots to the user. These screenshots capture important milestones in the development, with the first screenshot representing the original version, and the last screenshot representing the final design created in the previous chapters. There were four versions of the interface captured in the screenshots, each of them visible in Figure 4.1.

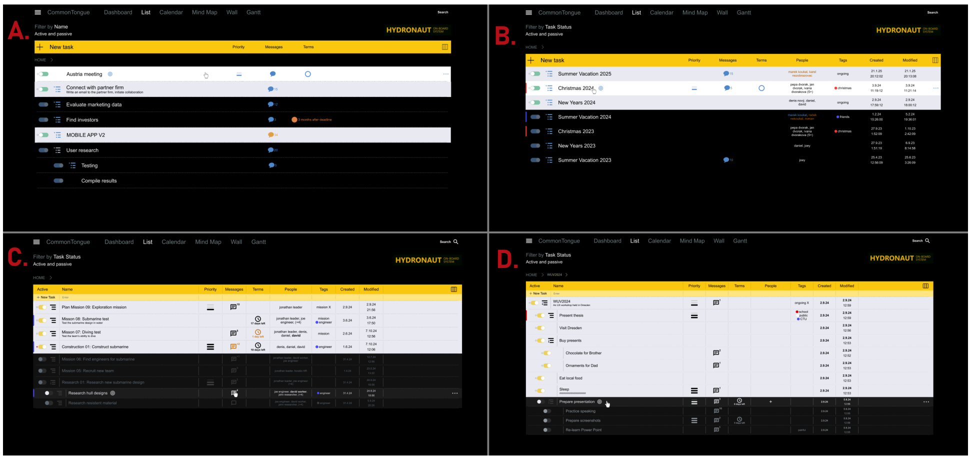


Figure 4.1: The four screenshots used in the test: A. Original version, B. First changes version, C. Semi-final version, D. Final version

The screenshots all contain sample scenarios, which were carefully created with the goal of showcasing different functionalities present within the system (priority, terms, hierarchy, etc.). The topics of the different scenarios focus on family vacation planning, business management, trip planning, and research project planning.

The scenarios for both old (A. and B.) and new (C. and D.) versions are similar in structure – both versions have a complex and a simple scenario. Both versions should also provide a complete view of the program's functionality by themselves. This means that after seeing either of the pairs, the user should have a basic understanding of the interface.

There were four different orders, in which scenarios were presented to the user, based on the Latin square design [8] (each screenshot will be shown in all positions), while also alternating between the new and old versions. This is to prevent errors during testing that could arise from presenting the screenshots in the same order for each of the tests.

The hypothesis was that users who see screenshots C. and D. first will have a better understanding of the program's functionality than users who saw the older screenshots (A. and B.) first. There should also be notable spikes in program knowledge when the newer versions are first shown. In order to properly evaluate the designs and prevent bias from any previous user experiences, each of the different screenshot orders was also assigned to at least one organized and one unorganized user.

4.1.3 Task creation

After setting the goal and the features to be compared, a task had to be created to evaluate the four screenshots. Two pilot tests were conducted, each of which experimented with a different approach.

The first test allowed the user freely talk about the screenshot, which led to the user focusing on rating and evaluating the interface in too much detail. This doesn't happen in a real user scenario, where they want to accomplish a goal as fast as possible. This meant that a task goal had to be set in order to create a proper test.

The second test instructed the user to focus on the scenario data and find as much information about the event or the person organizing the event as possible. These instructions improved the quality of the test significantly; the participant focused on the task at hand, meaning their assumptions about the functionalities of the program could be observed. Instead of observations which focused on the design (*"I feel like this New Task text is way too big."*), the participants explored the functions of the system (*"I see that there are some white and some black tasks, so that must mean that some of them are already done."*).

The second version of the task was selected and a one-minute time limit was also added to make the participant focus on what they considered most important. Any remaining details not covered during the initial minute were discussed after the minute had elapsed to let the user finish their task.

4.1.4 Script writing

The initial idea for usability tests was to have a semi-structured script, which guided the course of the test, while also being flexible enough to adapt to the participant personalities and their approaches to problem solving. The overall structure of the script was created, with an introduction, followed by four blocks discussing each of the screenshots, ending the test with a debriefing and discussion of the user's feelings regarding the program.

The introduction told the user about the system – its purpose, the ability to create hierarchies of tasks, and its use as a planning tool for multiple users. The user was also informed about the nature of the screenshots – they contain an event plan and the user interface in the screenshots changes (although the functions of the interface stay the same). After the introduction, the participant was informed of their objective and its time limit and asked to think out loud during the test. They were also assured that any answer they provide is valid and that they should be honest when giving feedback or speculating about possible functionalities of the interface.

The main workflow revolved around the minute-long interval in which the user analyzed the event. The participants discussed the sample event in the screenshot, and the interface elements whose functions they understood. After the interval, some users preferred to continue talking about the event, guided by questions from the test supervisor, and others talked about their feelings regarding the interface.

During the course of the test, the user was observed and probed for explanations regarding their expectations and opinions. If a relevant part of the screenshot wasn't discussed, the user was asked to analyze it after they finished speaking. An example can be a situation where the user didn't notice a filtering functionality. A valid question would then be: *"If you wanted to sort the tasks according to their name, where would you do that?"*. These guiding questions often revealed valuable information during the test; in this example, the filter option was hard to find.

At the end of the test, the participants were prompted to ask about any parts of the interface whose function they still don't feel confident guessing. The unclear features were then compiled into a list of problematic elements that were not clearly explained in any of the screenshots.

The participants were also asked to pick two screenshots. The first screenshot should be the best-looking design, and the second should be the screenshot that helped them form a better understanding of the interface. The selected screenshots served as a metric for evaluating the users subjective feelings regarding the interfaces.

4.2 Results

The tests were carried out over a one-week period, and the results confirmed the hypothesis: Users who saw the newer designs understood the program better than people with the older versions. Improvements were made in two categories, the initial impression and the aesthetics.

4.2.1 Initial impression

"First impressions of a product are incredibly important," [5], which is why the participants were closely observed when a new screenshot was presented to them. When tasked with finding the goal of the scenario during testing, the users searched the interface for elements that they found important, based either on their previous experience with the program or their experience with similar programs.

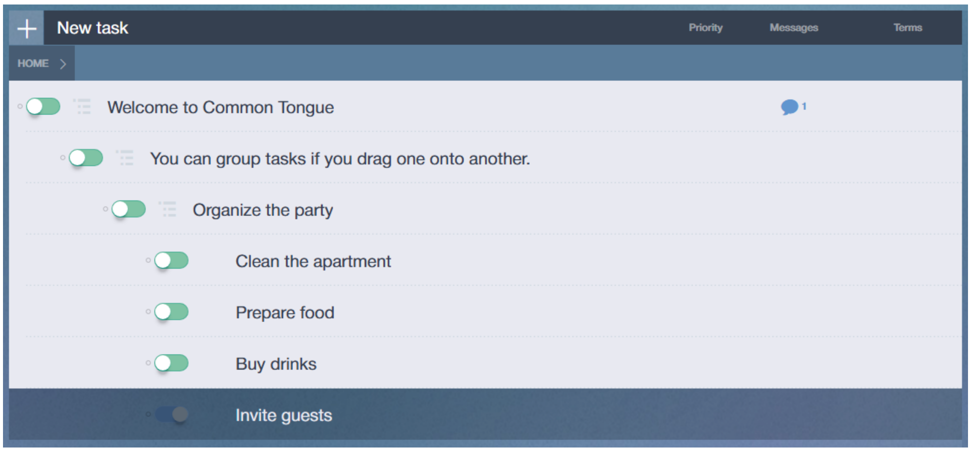
In the case of CommonTongue, the most important areas were the columns for the task name, description, activity, priority, and terms. The names of the tasks were useful for identifying the planned event, while the data columns provided additional information for each of the tasks. The participants worked efficiently and were able to easily identify the function of columns in the newer interfaces when compared to the older interfaces. This was a result of the icon update done in Chapter 3, according to the participants.

It was clear that the users felt lost in the old interface, with complaints like: *"I don't really know what Terms means, there is a circle there but I don't get what it does."*, referencing the old terms icon being a circle. Users who first received the newer interfaces did not struggle with the mentioned data columns, although the column names (for example, Terms and When) felt unclear to users.

Some of the other parts of the interface still performed poorly and were not mentioned or understood by the participants, even when prompted to find them. The problem elements in the interface were mainly the filter, breadcrumb navigation, column filter, tag ribbons, and the focus icon. These problem elements are shown in Figure 5.1 and will be addressed when updating the design in Chapter 5.

An important note regarding first impressions is that the users reacted to a sample scenario with data. This significantly improved their ability to intuit the system functions, as they had a baseline of how the system is operated. If the interface was empty, the results might have been different.

This knowledge should be used to create a suitable default dataset, which would be the default set for every new account and would showcase the functionalities of the program in the best possible way. A version of this base scenario already exists (Figure 4.2), however, it doesn't showcase enough features to properly introduce CommonTongue to a new user. The new scenario should be used to guide the user into the intended behavior, for example, by demonstrating that descriptions can be added.



Welcome to Common Tongue

You can group tasks if you drag one onto another.

Organize the party

Clean the apartment

Prepare food

Buy drinks

Invite guests

Figure 4.2: An intro scenario in CommonTongue Speak

The main features which should be introduced in the scenario are activity, hierarchy, priority, messages, deadline, people, calendar, and budget. The combination of initial impression (Chapter 4) and interaction testing (Chapter 6) showed that users who are introduced to the program through a sample scenario understand the system functionality faster and better than users who start with empty interfaces.

4.2.2 Aesthetics

The aesthetics of the design were evaluated based on the user ratings for each of the screenshots and also by collecting positive feedback regarding the interface functionality, feel, and look. In general, the newer design screenshots were always chosen as best looking, with users rating the new icons, colors, and layout favorably compared to the older designs: *"I like that the inactive tasks don't blend in with the background that much anymore."*. The users also correctly guessed that the interface of the tasks is split into cells which open different task detail tabs based on the column clicked, which is not visible in the older screenshots.

Some of the features in the newer screenshots were rated better in the old designs, namely the hierarchy button and the new task button. The coloring of some of the new interface elements was also inconsistent. The problematic elements should be addressed in an updated design, which is described in Chapter 5.

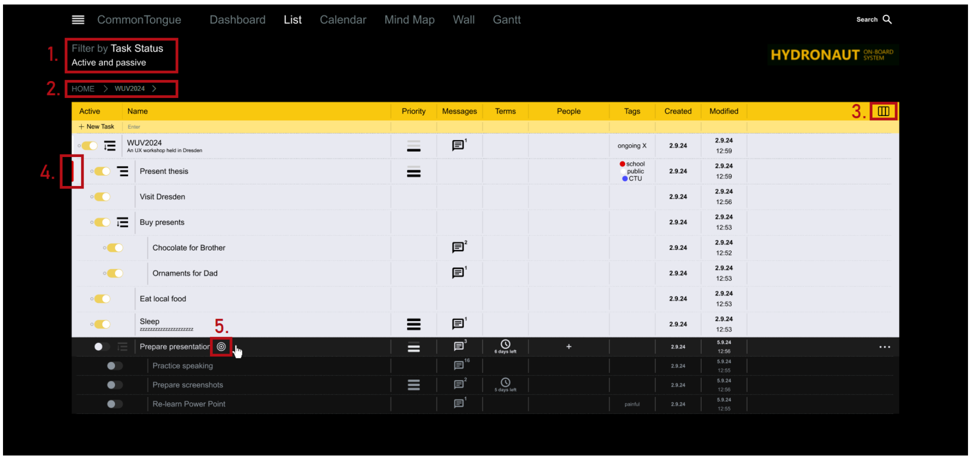
4.3 Conclusion

The interface was tested and evaluated by eight participants. The initial impression and overall visual clarity improved with the new interface design, with users mentioning new icons as a major improvement: *"Ahhh so that is what Term means, its a deadline for the task! I wouldn't get that without the icon."*. Users also preferred the aesthetics of the newer designs, claiming that the interface felt more readable and compact: *"I prefer the first screenshot, it felt like I could interpret the information faster."*. This is the intended result of the layout changes, which reduced the amount of whitespace, and the recoloring, which resulted in less visual noise. Participants rated the new designs favorably compared to older versions, and notable spikes in program knowledge were observed when showing newer designs.

Chapter 5

List redesign

During the tests in Chapter 4, problematic interface elements (shown in Figure 5.1) were discovered, namely the filter, breadcrumb navigation, column filter, tag ribbons and the focus icon. These elements were unclear to users, and their functions weren't explained properly by their appearance. The changes made to these elements will be described in more detail in this chapter.



Filter by Task Status
Active and passive

HOME > WUV2024 >

Active	Name	Priority	Messages	Terms	People	Tags	Created	Modified
	WUV2024 An UX workshop held in Dresden	≡	1			ongoing X	2.9.24	2.9.24 12:59
	Present thesis	≡				● school public ● CTU	2.9.24	2.9.24 12:59
	Visit Dresden						2.9.24	2.9.24 12:56
	Buy presents						2.9.24	2.9.24 12:53
	Chocolate for Brother		2				2.9.24	2.9.24 12:52
	Ornaments for Dad		1				2.9.24	2.9.24 12:53
	Eat local food						2.9.24	2.9.24 12:53
	Sleep zzzzzzzzzzzzzzzzzzzz	≡	1				2.9.24	2.9.24 12:53
	Prepare presentation	≡	3	6 days left	+		2.9.24	5.9.24 12:56
	Practice speaking		16				2.9.24	5.9.24 12:55
	Prepare screenshots	≡	2	5 days left			2.9.24	5.9.24 12:56
	Re-learn Power Point		1			painful	2.9.24	5.9.24 12:55

Figure 5.1: Screenshot with problem elements highlighted; 1. Filter, 2. Breadcrumb navigation, 3. Column filter, 4. Tag ribbons, 5. Focus icon

5.1 Initial impression

In order to improve the user's initial impression, changes should be made regarding the problematic elements found in testing. The five problematic elements were addressed using different approaches, each according to the problems they had.

5.1.1 Filter

The first element to be changed was the filter functionality, numbered 1. in Figure 5.1. Originally, the filter was represented by two lines of text, the first line detailing the chosen column to sort by, and the second the category of sorting. This representation was not appropriate, with users searching for a typical filtering icon (a funnel-like icon) or a way to control filtering through the column headers; therefore, two versions of the filter design could be created.

The first version would resemble the old filter design, with the added funnel icon the users expected. The element should invite interaction, either through the text color or its background, as users often ignored the element due to its neutral color. Initially, a fully highlighted yellow background version was created. This version was deemed too salient and was later replaced by a combination of both previous designs. The final design uses both the highlighted funnel icon and the text explaining the current filtering settings. The versions can be seen in Figure 5.2.

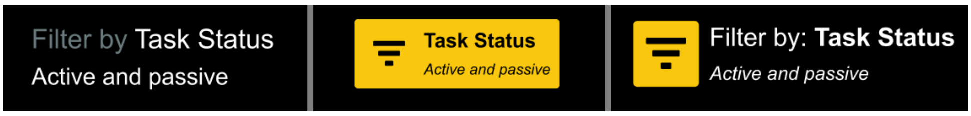


Figure 5.2: Three filter versions; original (left), fully highlighted (center), final version (right)

The second variant of the filter could instead replace the filter functionality with a header-based filter, with the users being able to click on a header to sort along the column. This version would require implementing new functionality and should therefore be considered in situations where the first version fails.

5.1.2 Breadcrumb navigation

Similarly to the list filter, breadcrumb navigation, numbered 2. in Figure 5.1, was never mentioned during testing, although its function was clear to all participants. Once prompted to talk about the differences between the original and the swapped design, the users overwhelmingly preferred the newer version, due to expecting the navigation to be located above the main work area (citing Windows Explorer as the closest reference).

In order to make the navigation more noticeable, a colored background was added, connecting the tabs in the navigation with the header bar. The contrast between the text and the background increased, and the text should be more readable. The selected tab is also highlighted in the new version. The comparison between both versions is shown in Figure 5.3.

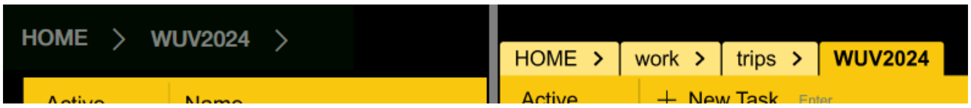


Figure 5.3: The swapped variant (left) and updated version (right) of breadcrumb navigation

5.1.3 Column filter

The third changed element was the column filter, numbered 3. in Figure 5.1. The column filter was previously represented by a three-column icon, located in the table header. Although some users correctly identified its function, most struggled, which is why an alternative was created. This alternative design appends itself to the last column in the header in order to simulate the look of a new column. The header of the new column contains a plus sign, which is a more recognizable symbol for adding compared to the previous icon. The comparison between both versions can be seen in Figure 5.4.

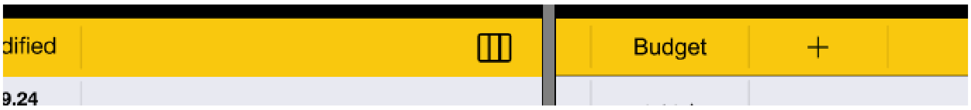


Figure 5.4: The original icon (left) and the updated version (right) for adding columns

A functionality for removing columns should also be created, either by adding the option to remove a column by clicking on its header, or by keeping the old icon to retain the functionality to manage columns while also having a simpler method for adding new columns.

5.1.4 Tag ribbons

The tag ribbons, numbered 4. in Figure 5.1, were added to improve the readability of the list. However, they were unsuccessful, as the users couldn't find the connection between their color and the tag colors in the screenshots. Due to this issue, the tag ribbons were removed from the newer designs, with only the colored circles remaining as the color indicator of a tag.

5.1.5 Focus button

The focus button, numbered 5. in Figure 5.1, selects a task, places it on top of the list view, and hides other tasks in the hierarchy. Initially, the participants struggled to confidently guess the function of the button. However, after analyzing the usage of the button during interaction testing in Chapter 6, no changes were made. The function of the button became clear once the users were able to interact with the interface, which means that even though they initially struggled to identify its function, the interaction revealed the function of the button properly. This is similar to the "Kayak" problem described by Krug [6], where users encounter a problem but recover quickly and without help, meaning the button design in CommonTongue is sufficient.

5.2 Aesthetics

After collecting aesthetic feedback from the tests, the new design was further updated with smaller changes. The main elements to be updated were the hierarchy button, the new task button, and the inconsistently colored elements in the new designs.

The hierarchy button was originally represented by three lines headed with dots indicating a hierarchy or a list. This version was replaced during the initial redesign with an arrow version, which was created with the goal of uniting the icon's style with the rest of the interface. This resulted in making an icon that is too similar to the priority icon, while also making it less readable.

An alternative simplistic icon was tested; however, the icon wasn't popular with users. A cross between both initial designs was selected as the final version. This icon is consistent with the other icons, while also utilizing the list-like quality of the first icon. The evolution from the original icon to the latest can be seen in Figure 5.5.

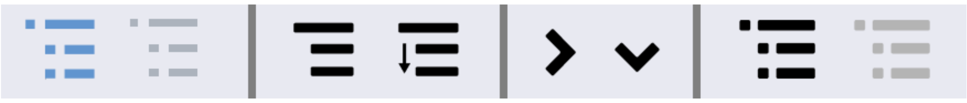


Figure 5.5: The evolution of the hierarchy icon from original (first) through the experimental versions (second and third) to the final version (last)

The new task button was originally part of the header of the table. This meant that it was clearly visible to all users who opened the application, since the font on the text was bigger than the rest of the interface and had a plus icon next to it, hinting at the function of the button. The original assumption was that the location of the button was obstructing headers for the name and activity columns, which led to the creation of an alternative design in order to allow the labeling of all columns. During user testing, participants were split between both designs, with some favoring the older button and others favoring the newer design. Due to the split decision, an updated version of the older design was created to include the shortcut to create a new task (Figure 5.6).

The final design of the new task button should be selected based on the filtering functionality implementation. The column-based filter requires labeling all columns in order to program the functionality of sorting along a column successfully. If the filter remains similar to the original filter, the *new task* button can be placed inside the header.

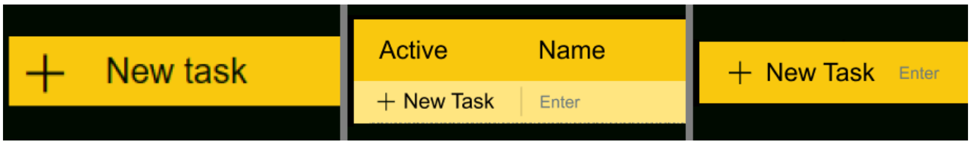


Figure 5.6: The evolution of the new task button

Several interface elements were recolored: the activity button now matched the header of the table, and the separators on the left side were updated to be darker and more defined to better emphasize the hierarchy of the tasks. There were also small changes to the notification color, which was changed to a brighter orange. The size of the message count text was increased to make it more readable, as the original size was inadequate according to the users. A comparison between these changes can be seen in Figure 5.7.

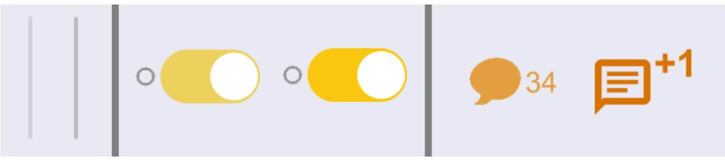


Figure 5.7: Recoloring changes comparison between old and new; separators (left), active button (center) and notification color and font size (right)

5.3 Other changes

Two factors that inhibited participants' ability to correctly identify the function of certain columns were their names and their content. The names of columns are an important identifier and can deter users from finding the information they need if used incorrectly. Using unfamiliar technical terms instead of more familiar names can negatively impact the system performance (Krug [6], Chapter 1), which is why the twelve data columns present in the *List* view were analyzed with three columns selected to be updated. The selected columns were *Term*, *When* and *Budget*.

The *Term* column, which contains a clock icon and a number of days until the task deadline, was renamed to *Deadline*. The replacement of the word was necessary – the word *Term* seemed confusing to the users, and they couldn't identify the function of the column without any supporting data. Most of the participants started calling the data in the column "deadlines", which is what influenced the new name. The difference between the old and new *Term* column can be seen in Figure 5.8.

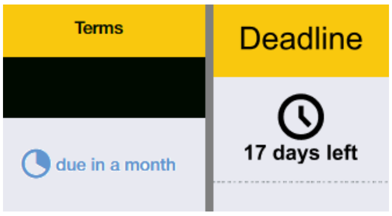
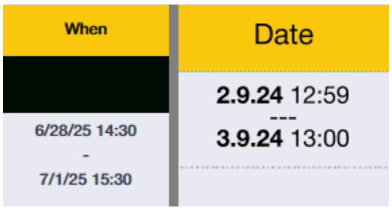


Figure 5.8: Comparison between the old (left) and new (right) columns

The *When* column contains two dates, separated by dashes, showcasing the time period when the planned task takes place. The older version of the interface uses slashes to separate the numbers in the date, while the newer version uses dots. The dots were chosen to unite the format of the column with the *Terms* tab, discussed in Section 7.4. The comparison is shown in Figure 5.9. In the newer versions, the dates are also highlighted in bold to differentiate them from the times. This should improve the readability of the column by making the times and dates different, since the user can focus on the bold text to search for dates and the plain text to search for times (using the pop-out effect [3]). The column was also renamed to *Date*.

The original name made sense to the users when data was visible – they saw *When* the task takes place. The problem arose when they viewed the name without the context of the data. When tasked with finding which of the column names relate to time, the participants were looking for familiar keywords they associate with time. The words mentioned during the testing were terms such as calendar, period, timeline, time, date, deadline, etc. The word *When* is not typically used in interfaces in this context, which means the users didn't immediately recognize its function.



When

[redacted]

6/28/25 14:30
-
7/1/25 15:30

Date

2.9.24 12:59

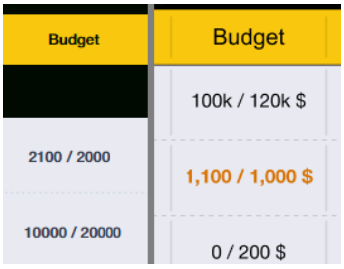
3.9.24 13:00

Figure 5.9: Comparison between the old (left) and new (right) columns

The last updated column was the *Budget* column, which is used to showcase the total expenditures compared to the available budget. Originally, the column visualized this relationship by displaying the two numbers separated by a slash. This visualization can be hard to read for participants – the interface doesn't show currency, and doesn't separate thousands, meaning large numbers can be difficult to read at a glance. A way to quickly gauge whether the task is over budget was also missing.

The new version of the *Budget* column properly separates thousands, divides the numbers into groups of three, and makes them easier to read and remember. This supports what is known as chunking. Chunking and its role in the memorization process was initially described in the famous psychological study *Magic number seven, Plus or Minus Two* by G. A. Miller [9]. The idea of memory chunks was further explored by other researchers, for example in the publication by Nelson Cowan [10] in which Cowan explores the limits of chunking in more detail and sets the limit as 4±1 chunks, which is well within the scope of the *Budget* column visualization. The *Budget* column also shows the chosen currency in the column, and if the budget is spent, the text turns orange and becomes bold. Both versions can be seen in Figure 5.10.

A method was also devised to visualize the budget with a limited horizontal space in scenarios where adaptive column width isn't possible. Each of the values should always contain a fixed number of characters, rounding to thousands and millions when appropriate. This ensures that



Budget
2100 / 2000
10000 / 20000

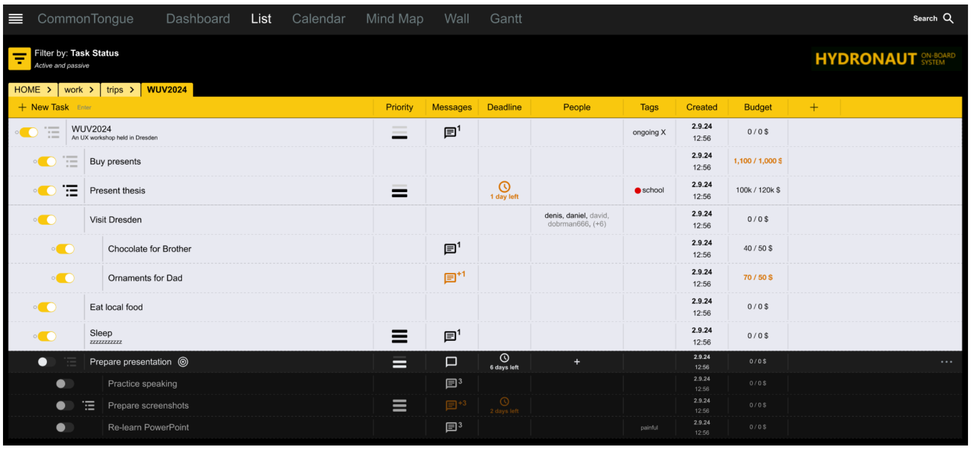
Budget
100k / 120k $
1,100 / 1,000 $
0 / 200 $

Figure 5.10: Comparison between the old (left) and new (right) budget columns

the data always fit the column. The data displayed in the budget column should be used as an estimate of the actual budget, which means rounding doesn't impact the readability, as the user can display precise values of the budget using task detail.

5.4 Resulting design

An updated version of the tested designs was created with the goal of eliminating the remaining issues and inconsistencies not addressed in the new *List* view design. The updated design is visible in Figure 5.11.



Filter by: **Task Status**
Active and passive

HOME > work > trips > **WUV2024**

+ New Task Enter	Priority	Messages	Deadline	People	Tags	Created	Budget	+
WUV2024 An UX workshop held in Dresden	≡	1			ongoing X	2.9.24 12:56	0 / 0 $	
Buy presents						2.9.24 12:56	1,100 / 1,000 $	
Present thesis	≡		1 day left		● school	2.9.24 12:56	100k / 120k $	
Visit Dresden				denis, daniel, david, dobrman666, (+6)		2.9.24 12:56	0 / 0 $	
Chocolate for Brother		1				2.9.24 12:56	40 / 50 $	
Ornaments for Dad		+1				2.9.24 12:56	70 / 50 $	
Eat local food						2.9.24 12:56	0 / 0 $	
Sleep zzzzzzzzzzz	≡	1				2.9.24 12:56	0 / 0 $	
Prepare presentation	≡		6 days left	+		2.9.24 12:56	0 / 0 $	···
Practice speaking		3				2.9.24 12:56	0 / 0 $	
Prepare screenshots	≡	+3	2 days left			2.9.24 12:56	0 / 0 $	
Re-learn PowerPoint		3			painful	2.9.24 12:56	0 / 0 $	

Figure 5.11: The updated version of the Chapter 3 design

The goal of the updated design was to make smaller changes, which objectively improve on the original, or fix mistakes created during development. With the positive feedback received during the testing, the new interface should be a significant improvement with regard to usability and readability.

In addition to updating the design, it will be necessary to create an updated version of the introductory scenario in CommonTongue. The updated version of the scenario should showcase more of the program functionality, and should contain a deadline, budget, calendar date, and priority, to better showcase the possibilities of the system.

Chapter 6

Interaction testing

Evaluating the interface through screenshots provided a method of measuring initial impressions; however, users didn't interact with the system itself. Interaction with a system is a complex process, influenced by many factors imperceptible in an uninteractive screenshot. To fully evaluate the system and properly redesign its main view, additional usability tests focusing on interaction with the system had to be conducted.

The ideal method to test both the user interaction and the design in Chapter 3 would be to integrate the design into a live version of the website or to use a prototype created in Figma. Regretfully, neither of the two approaches was viable – the implementation was not possible due to unforeseen security circumstances, which made accessing the source code impossible, and the Figma prototype would not allow sufficient data manipulation options. Instead, the current live version of CommonTongue was used for the testing, which is the same design discussed in the *Interface analysis* section of the thesis (Section 2.1).

The goal of the test is to find usability problems in the main workflow and observe how users interact with the task detail, the primary method of adding and modifying data in the task list. The test should analyze the task detail window and lead to the creation of a new design of the interface in Chapter 7. Before creating the test, all tabs of the task detail were analyzed, and their function will be explained in the following section.

6.1 Task detail

The task detail is an additional interface used to modify and view data. Clicking on a task opens the task detail window, which covers the right half of the screen and points to the currently selected task with an indication arrow. An example of a task detail is shown in Figure 2.4. There are six different tabs in the task detail, each representing a category of data to modify. The tabs are in order: *Overview*, *Messages*, *Terms*, *Budget*, *Timesheet*, and *Biomeasures*.

The *Overview* (Figure 6.1) contains general information about the task and is used to set the task priority, description, and tags. The date of creation is visible in the top section of the tab.

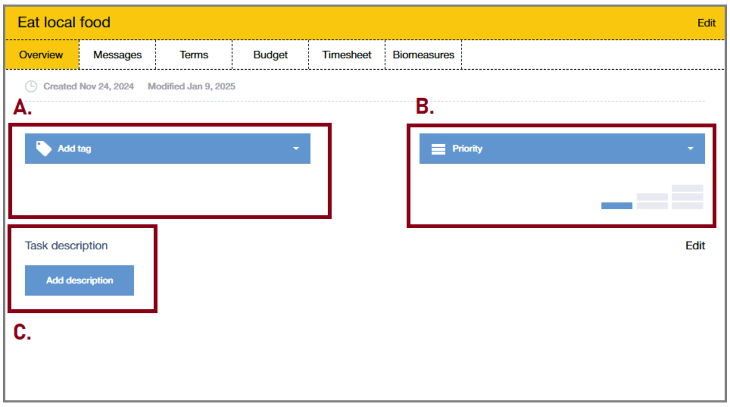
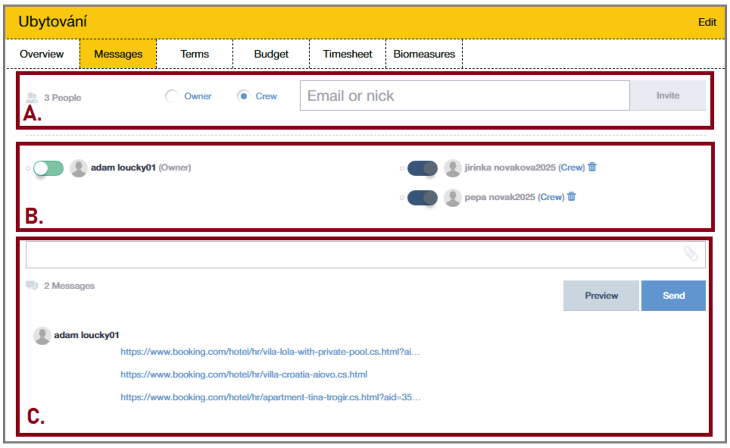


Figure 6.1: Overview tab, A. tags, B. priority settings, C. task description

The *Messages* tab (Figure 6.2) is used to communicate with other users within the task. The tab can be used to add new users to the task, manage their activity, and also write messages into the task chat. The chat offers the option to send images and uses markdown language to format messages.



adam loucky01

https://www.booking.com/hotel/hr/vila-lola-with-private-pool.cs.html?ai...

https://www.booking.com/hotel/hr/villa-croatia-aiovo.cs.html

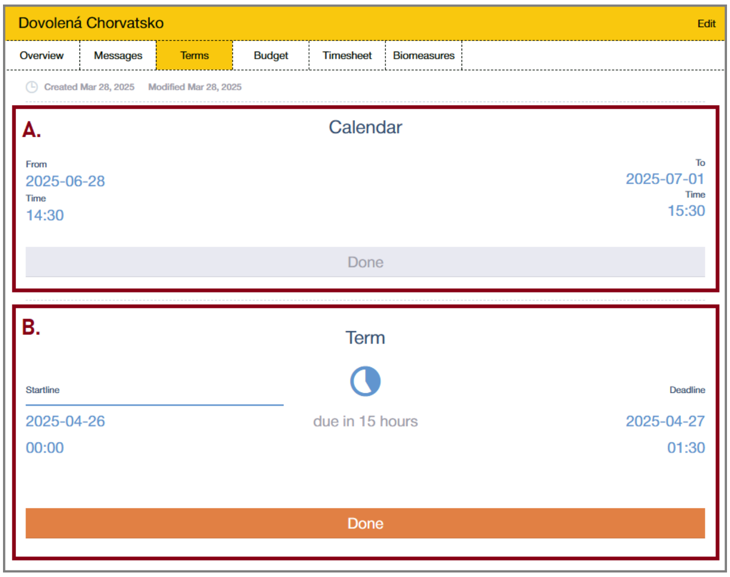
https://www.booking.com/hotel/hr/apartment-tina-trogir.cs.html?aid=35...

Figure 6.2: Messages tab, A. inviting, B. activity managing, C. chat

The *Terms* tab (Figure 6.3) is used to set the calendar date and the task term. The calendar date should be set to the time the event takes place, and the term should be the period during which the event is prepared or the task is done. The interface uses a calendar pop-up window to set the dates and a scrolling selection to select times.

The *Budget* tab (Figure 6.4) contains two sections, the first is used to set and visualize the total budget, and the second is used to add spendings to a list. Each of the entries in the spending list has a cost and a mandatory description. The sum of all expenditures in the list is deducted from the total budget and visualized using a bar.

The *Timesheet* tab (Figure 7.8) is similar to the Budget tab; however, it visualizes time budgeting instead of monetary budgeting. The time can be set to man-days, hours and minutes, or hours and minutes only.



Created Mar 28, 2025 Modified Mar 28, 2025

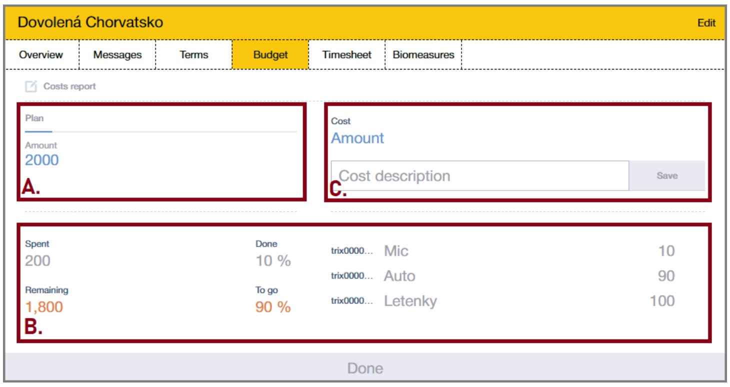
Calendar

From 2025-06-28
Time 14:30
To 2025-07-01
Time 15:30

Term

Startline 2025-04-26 00:00
due in 15 hours
Deadline 2025-04-27 01:30

Figure 6.3: Terms tab, A. calendar section, B. terms section



Costs report

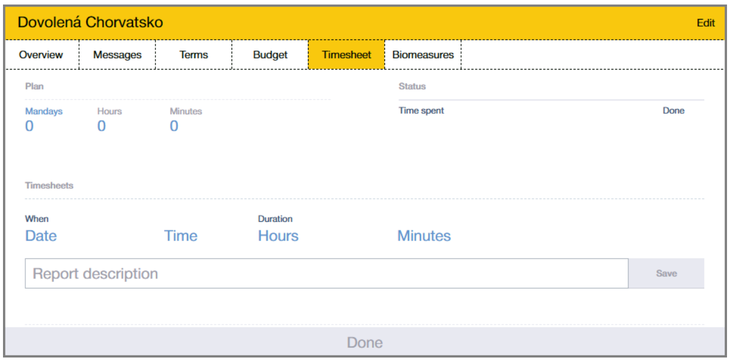
Plan
Amount 2000

Cost
Amount
Cost description

Spent 200
Remaining 1,800
Done 10 %
To go 90 %

trix0000...	Mic	10
trix0000...	Auto	90
trix0000...	Letenky	100

Figure 6.4: Budget tab, A. total budget, B. adding spendings, C. expenditure list



Plan
Mandays 0
Hours 0
Minutes 0

Status
Time spent
Done

Timesheets
When Date Time
Duration Hours Minutes
Report description

Figure 6.5: Timesheet tab in the task detail

Biomeasures are (Figure 6.6) a feature used during Hydronaut missions. The tab shows medical information from sensors placed on the crew during a mission. This tab will not be modified as it is currently being developed.

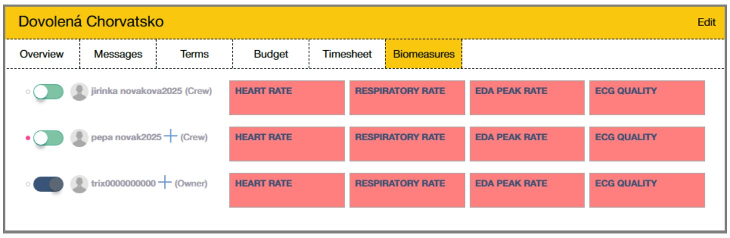


Figure 6.6: Biomeasures tab in the task detail

When comparing the frequency of use for each of the tabs, the *Overview*, *Messages*, and *Terms* should be the most used. These tabs contain basic organizing functionalities and thus will be the most frequently interacted with when creating basic tasks. *Budget*, *Timesheet* are specialized tabs, which are used when planning a detailed project with a focus on budgeting resources. These tabs will be used less frequently. The frequently used tabs will be closely observed when used repeatedly, while the less frequently used tabs will be closely observed on first impression.

6.2 Preparation

Due to the interaction-based nature of the tests, the tests were performed in person, either in a controlled location or at the participant's home, with the goal of testing the software in a familiar environment. The main reason for testing in a familiar environment is to keep the interaction with the program as similar to the natural interaction habits of the participant as possible (Kuniavsky [5], Chapter 11). The duration of a single test was approximately one and a half hours, and the test itself was recorded on video.

The creation of the second test was similar to the preparation of the initial impression tests – the four-step process mentioned in Section 4.1 was used.

6.2.1 Recruiting

A group of four participants was selected for the test. During selection, the questionnaire from the previous Recruiting section (in Section 4.1) was used, with the same selection preferences, since the target user base remains the same in both tests. The participants are all semi-skilled to skilled with computers and work on the computer at least once a week. Participants were asked to bring their own devices if possible.

The program is hosted on a live website, and thus could be launched on the users' computer without preparation, which allowed participants to use their own devices to interact with the program. Using the participant's device for the test allows them to be more comfortable and simulates their natural workspace better than an external facility could (which should be the goal when preparing the test layout according to Kuniavsky's book [5]). Three of the participants used their own devices, while one used a provided computer instead.

6.2.2 Feature selection

The main feature to assess during the test is the task detail (described in Section 6.1). The goal of the test should be to evaluate the current task detail interface, identify usability issues in the main application workflow, and see how quickly users learn to work with the program.

The user should interact with the main task detail tabs during the test, ideally multiple times, in order to capture both the initial impressions and the repeated interaction. The testing should not focus on precisely completing a given task, but should instead focus on observing the user behavior when encountering new features and doing repetitive tasks (such as adding ten new tasks into the task list and naming them).

The test should also be performed on a new account on the website to evaluate the efficiency of the tutorial system. Letting the user log into the website also allows for testing the registration steps. This is important because all users must register before using the program; therefore, registration is the actual first impression. The results of observing the registration are mentioned in Subsection 6.3.1.

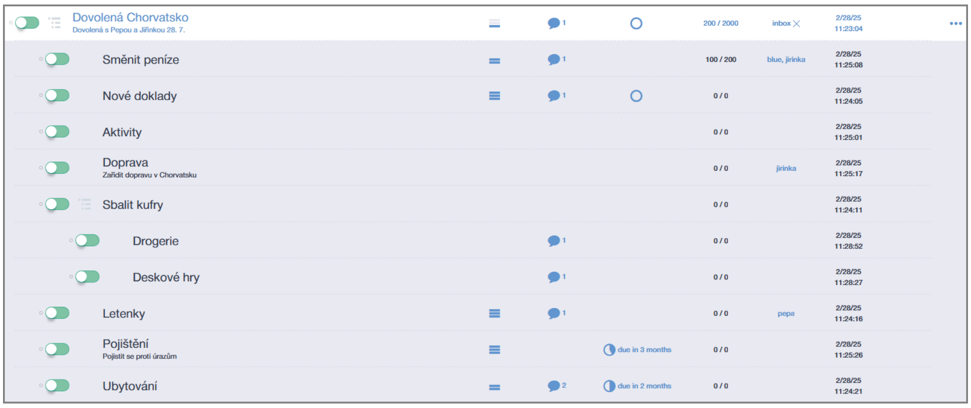
6.2.3 Task creation

In order to ensure a longer interaction with the system, a test scenario was created. The goal of the created scenario is to provide a set of data that would motivate the user to interact with the selected features of the system. The created scenario took inspiration from the user responses to scenarios in the initial impressions test in Chapter 4. Users responded favorably to scenarios with relatively smaller-scale events, such as vacation planning or daily organizing; therefore, a similar scenario was created.

The scenario described planning a vacation with two friends. The group wanted to fly to Croatia, reserve a car and an apartment, and use the free time to go to the beach and go sightseeing. Before leaving, they need to exchange money, check their documents, buy insurance, and pack their suitcases. They divided their responsibilities and wanted to assign each of the tasks to someone. The sub-tasks also had additional data, such as budget constraints, dates, and task priorities.

To create a description of the scenario, a sample hierarchy was created in the system, representing a possible method of entering the scenario data into the program (Figure 6.7). The hierarchy contained a single task called *Croatia holiday*, with sub-tasks describing the activities mentioned in the scenario. During the test, participants should create new tasks, put them in a hierarchy, browse the task detail, add priority to them, update the task terms, invite their friends, and message them using the system. They should also work with the budget and add their spendings to the list.

During testing, the users received a text transcript of the scenario along with the email addresses of the invited friends. An email was also provided, which they could use to log into a new account when necessary.



Dovolená Chorvatsko Dovolená s Pepou a Jiřinkou 28. 7.	1	○	200 / 2000	inbox ✕	2/28/25 11:23:04
Směnit peníze	1		100 / 200	blue, jirinka	2/28/25 11:25:08
Nové doklady	1	○	0 / 0		2/28/25 11:24:05
Aktivity			0 / 0		2/28/25 11:25:01
Doprava Zařidit dopravu v Chorvatsku			0 / 0	jirinka	2/28/25 11:25:17
Sbalit kufry			0 / 0		2/28/25 11:24:11
Drogerie	1		0 / 0		2/28/25 11:28:52
Deskové hry	1		0 / 0		2/28/25 11:28:27
Letenky	1		0 / 0	pepa	2/28/25 11:24:16
Pojištění Pojistit se proti úrazům		due in 3 months	0 / 0		2/28/25 11:25:26
Ubytování	2	due in 2 months	0 / 0		2/28/25 11:24:21

Figure 6.7: Sample scenario hierarchy that users will recreate

6.2.4 Script writing

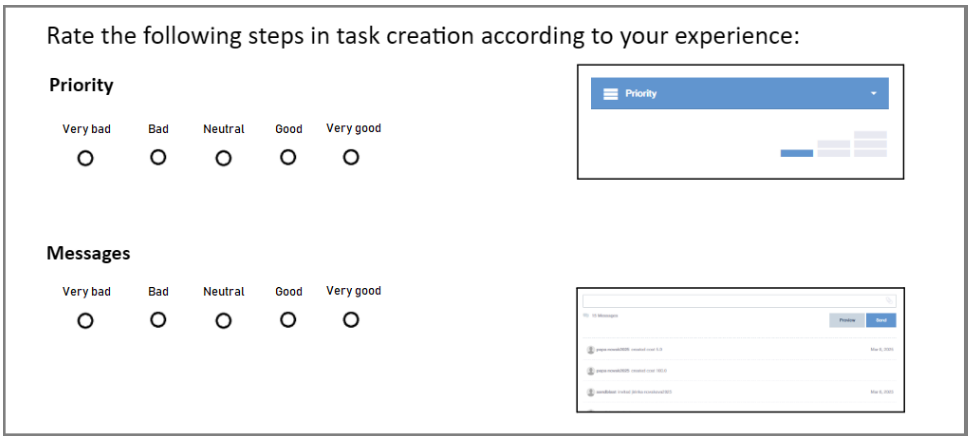
A semi-structured script with four sections was created. The goal of the script was to provide a fixed structure to the test while also being flexible and having the possibility of focusing on a particular section in more detail. The script structure was inspired by the script structure described in Observing the user experience [5] and contained an introduction section, a task section, a survey section, and a reaction section.

The first section introduced the user to the system and explained the basic functionality of tasks: The activity, hierarchies, and column data. They were also introduced to the holiday scenario and explained the initial test steps, such as creating an account on the website and completing the given task. The users were assured that there are multiple ways to structure data in the program and that they should complete the task in a way that is most comfortable and practical to them.

Similarly to the initial impression testing, participants were asked to speak out loud and explain their thought processes when working with the system. During the tests, they were probed for information; however, the participants were not interrupted when working on a difficult task requiring focus, to avoid disturbing them. This was important due to the complex nature of the task, with users parsing through long paragraphs of text while using an unfamiliar interface, in order to not overwork the participant.

The scenario was presented as a formatted text, with tasks highlighted in bold, and additional data underlined. This was done to help the users parse the long paragraph, due to previous difficulties during a practice test. A pen and paper were also provided to all participants in the event that they wanted to make notes or mark finished parts of the tasks.

After finishing work with the program, users were asked to complete two surveys, the first being a standard System Usability Scale (SUS) test [11], and the second a Likert scale questionnaire [12], with task detail sections shown as items in the survey (example of a question in the survey is shown in Figure 6.8). The purpose of the SUS test was to grasp the overall efficiency and usability of the system using a standardized metric, while the Likert scale survey served as a method of grasping subjective participant opinions on the different sections of the interface. The users rated the individual elements according to their experience working with them from *Very bad* (1) to *Very good* (5).



Rate the following steps in task creation according to your experience:

Priority

Very bad	Bad	Neutral	Good	Very good
○	○	○	○	○

Messages

Very bad	Bad	Neutral	Good	Very good
○	○	○	○	○

Figure 6.8: Two example questions from the Likert questionnaire

A pilot test was conducted before starting the official testing in order to correct any errors in the script or task assignment. After the pilot test concluded, a section was added to the test structure – the reaction section.

The reaction section of the test followed after the survey. During the survey section, the test supervisor updated the data in the participant's system, in order to simulate responses from the other people in the scenario. The user received messages from their friends, the data of some tasks was updated, and tasks were marked as inactive. The participant was then additionally tasked to react to these changes. The reaction section served primarily as a method of measuring the success of system notifications, which can signal many different changes in the system, and could be unclear.

After the test was completed, the participants were interviewed about their experience using the program. They were asked to provide two positives and two negatives of the system, what they found unclear, and what they would like to add to make the system better.

6.3 Results

Feedback was divided into six categories according to the questionnaire: Priority setting, Messages, Invite, Terms, Budget, and List controls. Each of the participants provided a rating for each of the selected segments of the task detail.

The scores for each of the selected task detail segments were based on the participant's experience with the system on a scale of *Very bad experience* (1) to *Very good experience* (5), and the averaged scores for each of the segments were:

- **Priority** - 4.25 (Good experience)
- **Messages** - 3.75 (Good experience)
- **Invite** - 4 (Good experience)
- **Terms** - 2.75 (Neutral experience)
- **Budget** - 1.75 (Bad experience)
- **List controls** - 3.5 (Neutral to Good experience)

In the list of results, the clear outliers are *Terms*, *Budget*, and *Priority*, with user scores for *Terms* and *Budget* being negative on average and *Priority* being rated favorably.

The System Usability Score test was conducted along with the Likert test. The average SUS score was 50.6, which is a low score and signals poor usability of the system. Users often mentioned that the system felt cumbersome to use and that they often felt confused and lost when working with the system.

In addition, the interface was examined for other usability issues during the testing. These issues are mentioned in Subsection 6.3.1, and relate to the registration and initial impressions of the program.

6.3.1 Usability

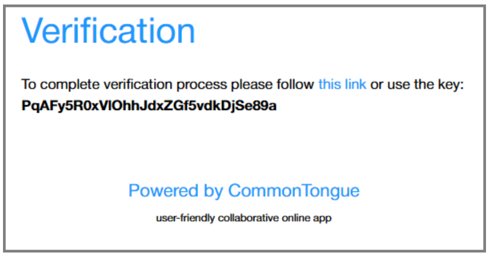
Users were observed performing repeated tasks and the results show a significant improvement in efficiency during the course of the test. The participants created tasks in hierarchies and used the task detail to add data to them. After receiving notifications regarding updates to the data, the users could identify which tasks were changed and could respond to these changes successfully. The main workflow of the program was intuitive; however, users struggled in the initial parts of the test, during registration into the system.

The registration screen was a significant obstacle for the users, who found it hard to engage with the program in the initial parts of the tests. They were instructed to create a new account, which required an email address and a secure password. The password requirements were not specified beforehand, which meant that users often had to update the password multiple times. The registration also contains a CAPTCHA test which is a security test required to verify the password. The combination of the password resetting, the CAPTCHA test, and the lack of password specification meant that users often had repeat the registration up to three times, which took them around 6 minutes total and frustrated them. This could deter a significant number of potential users, as it creates a very negative first impression: *"I feel like the program is making fun of me. Do*

I really have to fill this in every time? How was i supposed to know the password had to have a special character in it!".

The registration should display the password requirements at the same time as the user is creating the password, either in a small pop-up window under the text box or anywhere else in the interface, where the user can see the requirements before entering the password into the text box. This is an external standard that should be followed [4], common in systems with multiple password requirements such as password length, capital letter requirement, symbol requirement, etc.

After registering, users had to enter a key from their email to verify the account. The key is a 30 character long combination of uppercase and lowercase letters and numbers, requiring the user to copy and paste the key due to its complexity. There is an option to verify using a link in the email, but the link is not salient enough compared to the key, and the users never noticed it, as they were prompted to input a key by the interface. The link should be the primary way to confirm the registration – most of the users used their phone to verify the system, which makes copying the key into the computer difficult. The link to verify the account should be displayed in a more prominent way in the email, for example as a button labeled *Verify the account*. The current design of the verification email can be seen in Figure 6.9.



Verification

To complete verification process please follow this link or use the key:
PqAFy5R0xVlOhhJdxZGf5vdkDjSe89a

Powered by CommonTongue

user-friendly collaborative online app

Figure 6.9: Email format received when verifying an account

After examining the registration process, the remaining parts of the test were performed, focusing on the task detail.

6.3.2 Task detail

The goal of the task detail should be to provide an intuitive interface to work with the task and its data. Users should be able to determine the functionality of each of the tabs and should be able to work with the interface without any previous experience. Participants were observed when they first opened the task detail and the results of their work were compiled into a list of observations, and will be discussed for each of the task detail tabs in the following segments.

Overview

The overview was a positively rated tab, and most users understood its function. The users interacted with all three features in the tab: tags, task priority, and task description. The new design of the Overview tab is presented in Subsection 7.2.

Tags (visible in Figure 6.10) are a feature to group and label task categories. During testing, two of the users used tags to assign responsibilities in the vacation scenario, which is not a valid workflow in CommonTongue. The assigned tags are saved locally, which means other users of the system don't see them. The term *Own tags* was not sufficient to properly communicate the function of the tags, therefore users expect the behavior to be similar to other program functionalities like Trello labels or Jira labels. The tags should either be marked appropriately or reworked into a global functionality.

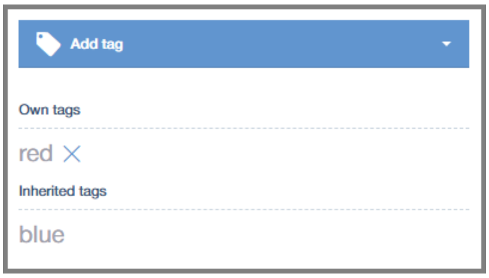


Figure 6.10: Tag menu in the Overview, marked as A. in Figure 6.1

Task priority is a feature used to mark the importance of a task, with 1 being the default priority and 3 the maximum, as can be seen in Figure 6.11. Priority is a well-understood feature, with users marking tasks according to their importance without error. The participants attributed this to the fact that they saw immediate feedback in the interface, with the default priority column changing when a different priority is set. The users also wanted to set a priority of one and have the priority displayed in the interface. This was not possible because the priority of one is deemed default and is not displayed.

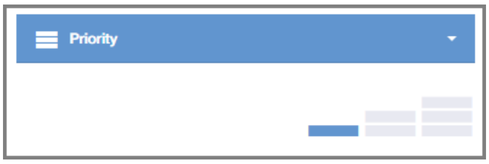
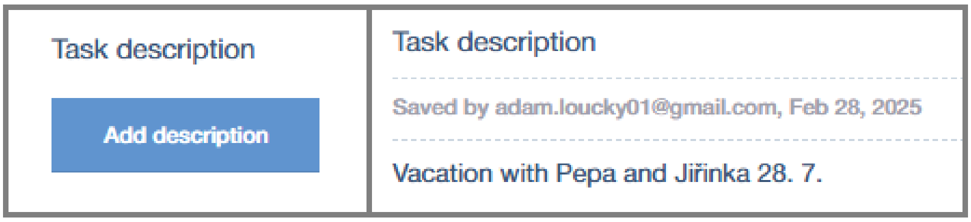


Figure 6.11: Priority menu in the Overview, marked as B. in Figure 6.1

The task description allows adding additional information about the task and displaying the information in the main interface, under the task name in the task list. The interface element used to add a description to a task is visualized by a button (visible in Figure 6.12), which led the users to believe that it was part of the tag interface located above the button. The button design is inconsistent with other elements used to enter text in the interface, which are always a variant of a text box. When the description is filled in, the text is displayed under the label, along with a date and the name of the user who wrote the description. The date and name are displayed above the description, which seemed unusual to users and made one user question if they typed the description correctly at first glance: *"What is that gray text, did I type that?"*.


Task description

Add description

Task description

Saved by adam.loucky01@gmail.com, Feb 28, 2025

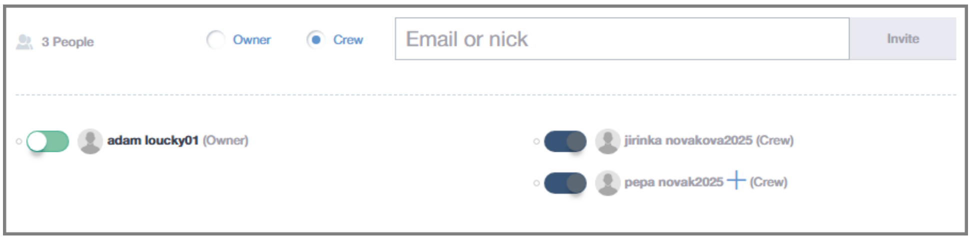
Vacation with Pepa and Jiřinka 28. 7.

Figure 6.12: Empty (left) and filled (right) task description in the Overview, marked as C. in Figure 6.1

Messages

The message tab is the main communication area of the program and had mixed user ratings. The tab is split into two sections, the people management section and the chat. The new design of the Messages tab is presented in Subsection 7.3.

The people section (Figure 6.13) contains an invitation text box, with a toggle between *Crew* and *Owner*, which sets the role of the newly invited user. Below the invitation are two columns, one for active and one for inactive users in the task. The participants invited two people during the test, but never assigned the *Owner* role for any of the invited users, marking them as *Crew* by default. This meant that the invited users couldn't properly update the task and cooperate with the participant in organizing the vacation. The participants were confused by the role terminology: *"I thought I would lose privileges if I toggled someone as an Owner."*. Some also didn't notice the toggle: *"Oh so I have to set their role before inviting them."*. The layout was also difficult to read and users often didn't understand why people are sorted into different columns.


3 People

○ Owner ● Crew

Email or nick

Invite

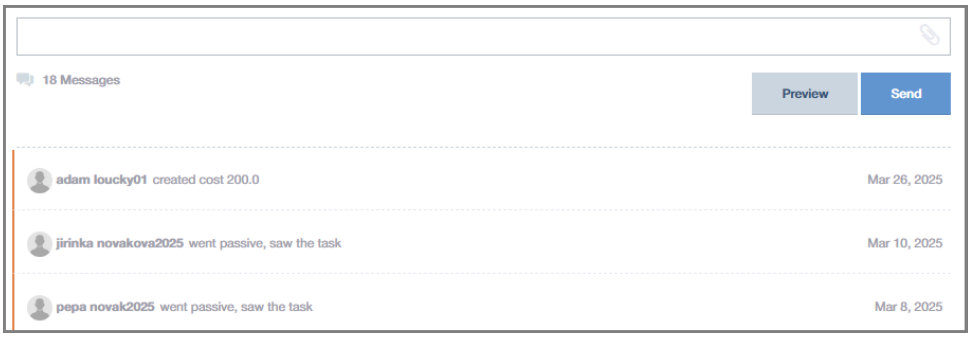
adam loucky01 (Owner)

jirinka novakova2025 (Crew)

pepa novak2025 + (Crew)

Figure 6.13: Columns of active and inactive users in the people section, marked as A. and B. in Figure 6.2

The chat had a familiar design that users recognized from other messaging applications such as Messenger, Slack, and WhatsApp, which meant that users expected the behavior they are used to from these messaging systems. This led to two problems. The first issue was the order of the messages, which is reversed compared to other similar messaging applications. The top-to-bottom order (visible in Figure 6.14) is used in email communication and in journaling, but because the design of the interface is similar to a traditional messaging application, the reverse order was unnatural to the users. The second issue the users mentioned was the lack of messaging features, such as filtering system messages, replying to messages, and tagging users. Users expected more in-depth communication options.



adam loucky01 created cost 200.0 Mar 26, 2025

jirinka novakova2025 went passive, saw the task Mar 10, 2025

pepa novak2025 went passive, saw the task Mar 8, 2025

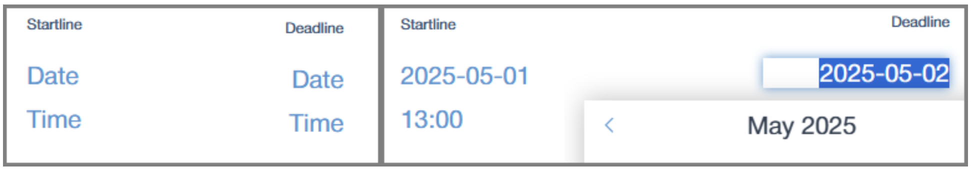
Figure 6.14: Top-to-bottom message order in the chat section, marked as C. in Figure 6.2

Although the ability to filter system messages is present in the system, the user needs to set it in the settings, instead of the chat itself, which makes it difficult to turn on and off.

Terms

The terms tab is used to set dates for the task, and was rated neutrally from a user perspective, although the experiences the users had during testing were mostly negative. The new design of the Terms tab is presented in Subsection 7.4.

A major factor in user frustration was a feature of the system that automatically fills both the *From* and the *To* dates when clicking on one of the text boxes. This meant that users who wanted to set a single deadline for their task automatically created a startline. Creating a task without a startline is possible, but requires the user to delete it manually after it is filled in. Automatic filling (shown in Figure 6.15) caused confusion and great frustration among users: *"Oh my God why does it keep filling in the date, i hate that. I want a deadline, I don't care about the start."*, and should be removed from the interface. Instead of automatically filling the text box with a date, users can press the *Today* button present in the calendar menu to achieve the same effect on demand, which should prevent situations where users fill in data unintentionally, while also keeping the option of adding the current date possible.



Startline 2025-05-01 13:00

Deadline 2025-05-02

Figure 6.15: Before (left) and after (right) clicking on a date in terms section, marked as B. in Figure 6.3

The layout of the tab was also an obstacle for the users, as they often forgot to save the changes they made to the terms, and had other difficulties working with the interface, such as identifying the time periods at a glance. The *Startline* and *Deadline* sections are aligned to opposite sides of the window, making the time period hard to read for the users: *"Why are the dates so far away from each other? Its hard to look at both at the same time."*.

The date format of the terms was hard to read according to the users. Users who preferred to type the dates as text had difficulty entering the date, as they needed to type the numbers strictly in the format of "year-month-day": *"I don't want to scroll through ten months of dates to set up an event for next year when I know the date. I would prefer to type it in."*. The keyboard users would have preferred a parser that would allow them to type the dates without the need to follow a strict text format.

Budget

The budget was the interface with the lowest rating in the system, with an average score of 1.75 (Bad experience). The new design of the Budget tab is presented in Subsection 7.5.

The interaction with the tab was difficult for the users, with one of them choosing to avoid using the interface altogether. The main problem of the budget tab is inconsistent labeling, which is visible in Figure 6.16. The total budget text box (left) is labeled *Amount*, which is a term that users did not recognize as a total amount of money. The left and right segments are not labeled, which means that users often thought that the total budget should be entered into the cost text box on the right.

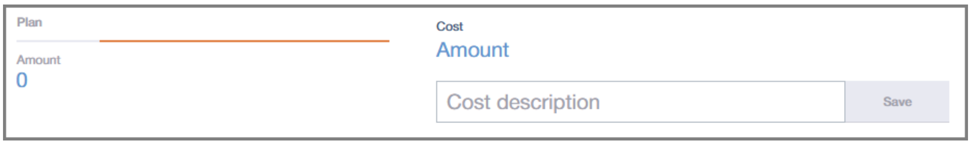
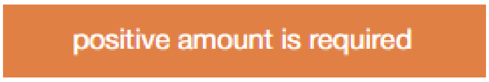


Figure 6.16: Input fields for the budget tab, marked as A. and B. in Figure 6.4

The text box to enter the cost of an expenditure (top right in Figure 6.16), labeled *Cost* contains the word *Amount* when empty. This is inconsistent with the left text box in the tab, which contains a zero to guide the user to fill in a number value. The word *Amount* is also repeated twice in the interface in two elements with different functionality. This confused users because the two text boxes seemed related to each other.

The *Cost description* text box (bottom right in Figure 6.16) is formatted differently from other text boxes in the tab. The combination of the previous error and the words *Cost description* in the empty description box resulted in users filling in the cost of the expenditure in this text box, intended for the description of the expenditure. The users often mistook the word *Amount* above the text box as a header and filled in the incorrect information into the *Cost description* box.

Filling in the interface incorrectly results in an error message; however, the error messages did not help the user understand the problem. The error message for attempting to add an expenditure without filling in its cost produces the message: *Positive amount is required.* (Figure 6.17). This further confused users who mistook the *Cost description* text box for the *Cost* text box: *"What? Two hundred is a positive amount!"*, said one user after they filled in *200* into the description text box. The error message for a missing description is similarly unclear: *"Name is required."*. The interface refers to the text attached to the cost as *Cost description*, and the error message refers to it as *Name*. This is also inconsistent and doesn't help the users identify the error.



positive amount is required

Figure 6.17: Error displayed when the Cost field is empty or negative

The interface also lacks a feature to select the currency, making cost organization difficult. The lack of currency indication in the interface means that the currency has to be communicated by other methods. Implementing this feature in the system could prevent costly mistakes caused by a currency-related misunderstanding.

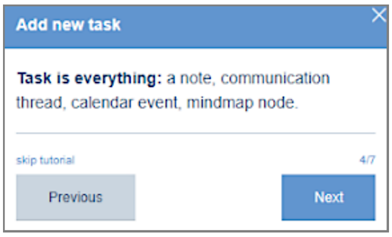
Task list

As users worked with the task list during the test, they rated it neutrally, with a semi-positive score of 3.5 (Neutral to Good experience). The users had some difficulties using the task list, namely with using the filter, working with windows, and the introductory tutorial.

When working with the task list, users often complained that the default sorting order was based on the most recently updated task. The users created a long list of tasks and as they continued working, the tasks changed order in the list frequently. This resulted in users searching the long list of tasks multiple times and getting frustrated with the changing interface. As other sorting orders don't currently work on the live website, the development team should work on fixing this feature to allow users to work with the task list properly.

A major point of frustration was the interaction with the task detail window and the task menu. Users who wanted to close the window expected to close it by clicking outside the window. This is a standard quality of life feature present in all other systems mentioned in the research chapter; however, the feature is not present in CommonTongue. The only way to close the window is by clicking on the close button on the right side of the window. All test participants assumed that the feature would exist in the system, and they attempted to close the window by clicking outside of it multiple times during the test: *"I think this happens in every program nowadays, I just assume that it closes a window if I click away. Having to click on the X is extremely slow."*. Adding the ability to close the window easier should improve the user experience; however, users could accidentally exit without saving their changes. The method of preventing exiting without saving will be addressed in the Saving subsection (Subsection 6.3.2).

The introductory tutorial is made up of 7 steps, each represented by a small window appearing and highlighting an area in the system. The tutorial is constructed with the assumption that the new CommonTongue account has introductory data filled in the system (which can be seen in Figure 6.7). This is true for the variant of CommonTongue called *Speak*; however, *Hydronaut* accounts do not contain data when created. This meant that the tutorial did not work correctly during testing, with the pop-up messages being displayed in the incorrect order. The tutorial text was also deemed unclear by the participants: *"Okay... i don't know what they mean by that, but i will probably get it later."* (Figure 6.18). The tutorial should be rewritten to correctly introduce users to the system according to the *Speak* scenario, or a new introductory scenario according to the specifications in Section 4.2.1. A document with the rewritten tutorial sections is in the appendix of the thesis (titled Onboarding).


Add new task

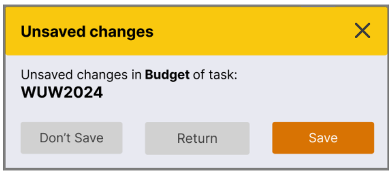
Task is everything: a note, communication thread, calendar event, mindmap node.

Figure 6.18: A snippet of the program tutorial

Saving

During testing, in multiple tabs, users often had problems saving their changes. The users would update data in a tab, for example, the total budget, and then continue without saving their work, resulting in the changes being discarded when closing the task detail. When asked about their reasoning, one user replied: *"I didn't really notice a save button, i thought it saved automatically."*. This is due to inconsistency in the placement and behavior of the confirmation buttons. In the Budget tab, the button is at the bottom of the screen, in Terms, it is under each of the sections, and in the Overview the changes are saved automatically. This inconsistency in the design leads to users not saving their work correctly, while also creating incorrect assumptions about the functionalities of the program. Inconsistencies can be fixed by making the rules of the confirmation buttons consistent across the interface. This would mean unifying the look and behavior of the confirmation buttons and placing them consistently across the interface.

Notifying the user of unsaved changes can be another feature that would benefit the usability of the interface. Users who would close a task detail with unsaved changes would be notified by a notification window (example can be seen in Figure 6.19).

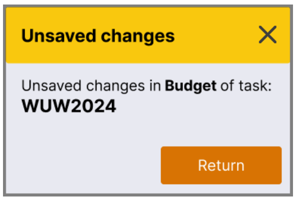


Unsaved changes

Unsaved changes in **Budget** of task:
WUW2024

Figure 6.19: A window for notifying the user of unsaved changes

This window could appear in the bottom right in a similar way to the copy-paste clipboard and should allow the user to discard the changes, return to the changed tab, or save the changes. A reduced version of the window could also be created (example in Figure 6.20), with the only option of returning to the window with unsaved changes, which would work as an *"Undo"* button. This version should be less intrusive to the user and should be used in situations where users abandon changes often.



Unsaved changes

Unsaved changes in **Budget** of task:
WUW2024

Figure 6.20: A simple Undo window for notifying the user of unsaved changes

6.4 Conclusion

The interaction testing resulted in a new understanding of the main program workflow and revealed design flaws present in the task detail interface. Users had mixed experiences with the program and became frustrated with the interface on several occasions. The main workflow was intuitive; however, the interface of the task detail tabs was unclear to users and resulted in a poor initial experience with the program. New task detail designs will be created and described in Chapter 7.

Chapter 7

Task detail redesign

After collecting data on how users perceive the task detail interface, new designs were created, with the goal of updating the interface based on user feedback discussed in Section 6.3.2. The new *List* view design, along with the new task detail designs, should improve the introduction to the program and the task detail interface, which was the most problematic part of the interaction testing according to the participants.

A new version of each tab was created, with the main goal of improving the worst rated tabs (*Budget* and *Terms*), and also unifying the task detail designs with the new *List* view design presented in Chapter 3.

The new designs were also compared to the old design by the four users who participated in the task detail tests, on a scale of the new interface being *Significantly worse* (1) to *Significantly better* (5). The users were shown screenshots with a side-by-side comparison of both versions, which will be shown in the corresponding sections of this chapter. After discussing the two versions, the users provided a rating for each of the tabs they used during the testing, and their ratings were averaged, creating a score for each of the new designs. The scores for each tested tab are listed below.

- **Overview** - 4 (Better)
- **Messages** - 4.5 (Better to Significantly better)
- **Terms** - 4.75 (Significantly better)
- **Budget** - 4.75 (Significantly better)

7.1 Task detail window

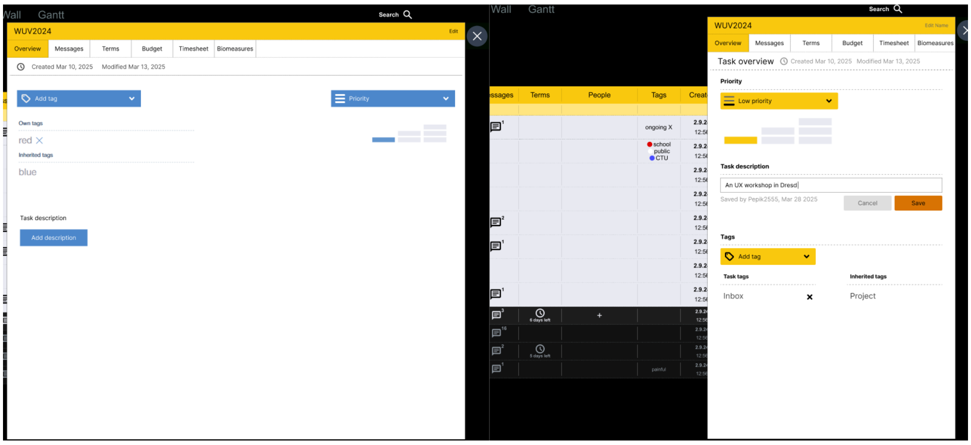
The task detail tabs have multiple different layout schemes, which often led to incorrect associations by participants during the testing. The users associated elements that were visually aligned or similar, leading to confusion and frustration. In order to group elements with common functionality, a single-column design was chosen for the new task detail tabs. The single-column design should improve readability by segmenting the interface into headed sections.

The layout of the old interface resulted in a large amount of unutilized space, leaving the bottom half of the task detail window empty. The single-column design allowed for a slimmer layout, which should also improve the spatial efficiency of the window and prevent unnecessary whitespace in the interface.

The new single-column design was preferred by the participants: *"The window seems nicer, it feels more compact. There is so much empty space in the old design now that I look at it"*.

7.2 Overview

The *Overview* tab was updated to a single-column design, resulting in a smaller window overall. The tab is now structured into sections, each of which contains one of the features of the previous tab. The sections are labeled with a header in order to fix the previous labeling inconsistencies. Each of the segments was also updated. The comparison between both designs can be seen in Figure 7.1.



An UX workshop in Dresd

Saved by Pepik2555, Mar 28 2025

Figure 7.1: Comparison between the old Overview tab (left) and the new tab (right)

The priority section button originally contained the word *Priority* to label the segment. This is no longer necessary, due to the new label, and instead the button now displays the current priority level. This should help users understand the scale better, as some of them expressed confusion regarding the default priority level.

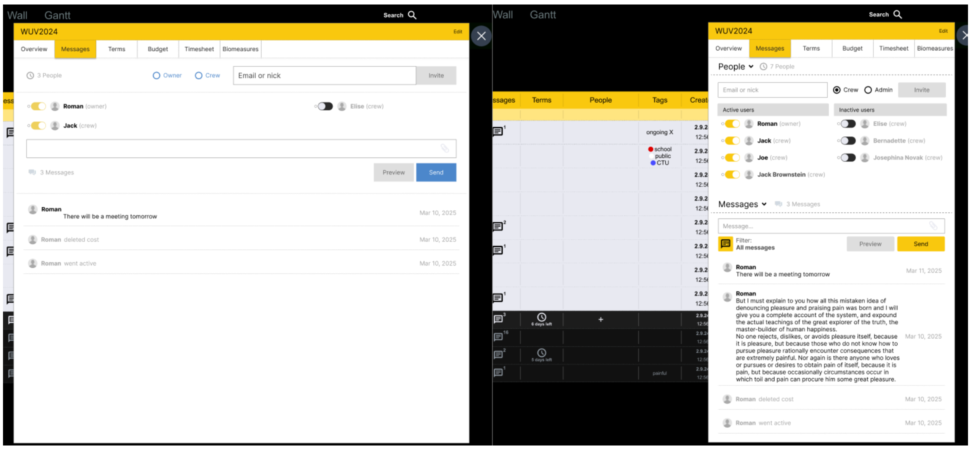
The task description section, which originally used a button to add a description, was updated to a regular text box. The original way to display the task description also included information about the most recent date of modification, which is now displayed under the text box, instead of above it, which should fix the user confusion regarding the order of text in the section.

The tags were previously displayed in a single column and grouped into two groups *Own tags* and *Inherited tags*. This was changed in the new design, and each of the tag groups now has a separate column. This change should make it easier for users to work with tags as it places them in different areas in the interface, separating them perceptually according to the common region principle [7]. The meaning of the *Own tags* column wasn't clear to users, as they thought the task the tags were global. The name was changed to *Task tags*, which should better reflect the new functionality of global tags. In situations where the tags remain personal instead of global, the column should be renamed to *Personal tags* instead. The word *Personal* should reveal the locality of the tags more clearly than the phrase *Own* tags.

The updated *Overview* tab received a score of 4 from users, which meant that the interface was improved. The evaluators credited the improvement to the window design change: *"I like the new design more, but the functionality hasn't changed that much, so i think this should be a four on the scale."*.

7.3 Messages

The *Messages* tab was updated, similarly to the other tabs, to a reduced single-column design. The people and chat sections are now separated with large collapsible headers, in order to divide the tab interface into distinct sections with different functions. Both the old and new design can be seen in the comparison in Figure 7.2.



Roman
There will be a meeting tomorrow
Mar 10, 2025

Roman deleted cost
Mar 10, 2025

Roman went active
Mar 10, 2025

Roman
There will be a meeting tomorrow
Mar 11, 2025

Roman
But I must explain to you how all this mistaken idea of denouncing pleasure and praising pain was born and I will give you a complete account of the system, and expound the actual teachings of the great explorer of the truth, the master-builder of human happiness.
No one rejects, dislikes, or avoids pleasure itself, because it is pleasure, but because those who do not know how to pursue pleasure rationally encounter consequences that are extremely painful. Nor again is there anyone who loves or pursues or desires to obtain pain of itself, because it is pain, but because occasionally circumstances occur in which toil and pain can procure him some great pleasure.
Mar 10, 2025

Roman deleted cost
Mar 10, 2025

Roman went active
Mar 10, 2025

Figure 7.2: Comparison between the old Messages tab (left) and the new tab (right)

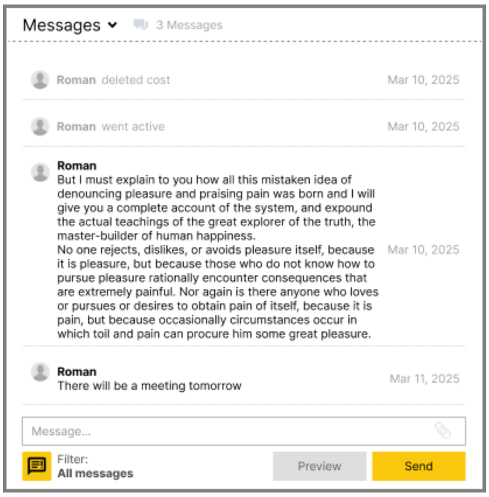
The invitation functionality was mostly unchanged; however, the order of the elements in the segment changed. The previous order of Role-Email-Invite was changed to Email-Role-Invite. Testing revealed that users don't change the default role of the invited person. The updated order of elements should improve the user experience, as the role selection is now located between the text box and the confirmation. This should reinforce the intended workflow of entering an email, setting the role of the person and then inviting them. The name of the administrator role was also changed from *Owner* to *Admin*. This was done to prevent confusion as mentioned in Subsection 6.3.2.

Headers were added to the active and inactive user columns. This change was prompted by three of the four users being confused by the people section and should improve its visual clarity by clearly labeling the sections. Each of the columns should also have a fixed height and have a separate scroll bar to search the column.

The ability to filter system messages was often mentioned during testing. The system messages in a task can often outnumber the user messages, which interrupts the communication between users. It is also important to be able to check them in situations where a change occurred in the task. For these situations, a filtering button was added below the chat text box, which can be used as a toggle between displaying system messages, user messages, or both (which is currently the default). The task list message column should also be updated to show only the number of user messages. Including system messages makes the message counter less informative, as the amount of system messages in the task often exceeds the number of user messages.

The order of messages in the chat was not changed in the current design, although it is recommended to change it to a regular bottom-to-top order seen in messaging applications like Slack or Messenger. Reversing the current messaging order would be more natural for the main CommonTongue user base who uses the chat to message other people in group projects, rather than to journal information into the chat (similarly to comments in Jira, Notion, or Trello). The reversed chat design can be seen in Figure 7.3.

The current design would also be hard to translate to a mobile layout due to the location of the chat text box. The text box would be located in the middle of the phone screen, and thus would be hard to access when compared to the regular bottom-to-top message order.



Roman deleted cost Mar 10, 2025

Roman went active Mar 10, 2025

Roman
But I must explain to you how all this mistaken idea of denouncing pleasure and praising pain was born and I will give you a complete account of the system, and expound the actual teachings of the great explorer of the truth, the master-builder of human happiness.
No one rejects, dislikes, or avoids pleasure itself, because it is pleasure, but because those who do not know how to pursue pleasure rationally encounter consequences that are extremely painful. Nor again is there anyone who loves or pursues or desires to obtain pain of itself, because it is pain, but because occasionally circumstances occur in which toil and pain can procure him some great pleasure. Mar 10, 2025

Roman
There will be a meeting tomorrow Mar 11, 2025

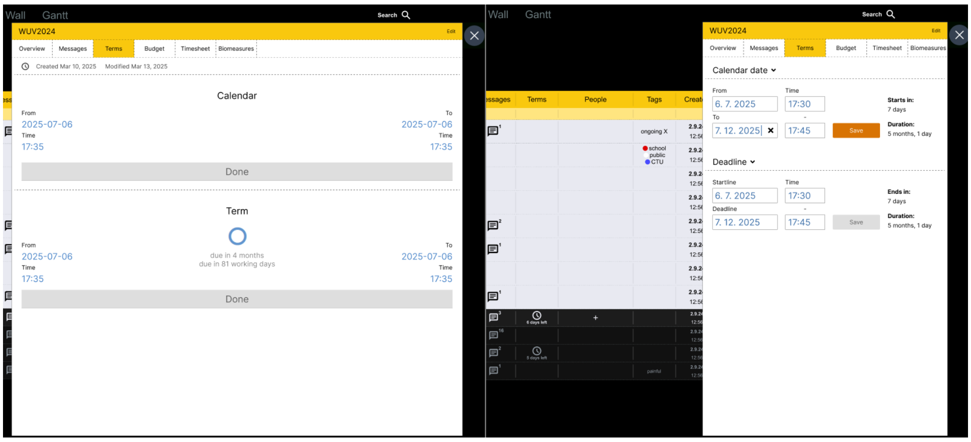
Filter:
All messages

Figure 7.3: Messages section with a reverse message order

The new design for the *Messages* tab received an average score of 4.5 which is a very positive score and means a decent improvement. The evaluators appreciated the ability to filter messages, noting their frustration with the original chat. The People section was a slight improvement according to the users: *"The new design is better, I like how compact it is. I don't see that much change at the top but it looks better overall."*.

7.4 Terms

The *Terms* tab was originally a negatively rated tab and needed to be changed more drastically than previous tabs. The two sections of the interface were preserved and collapsible headers were added to each of the sections. Similarly to the previous tabs, the width of the window was changed to a slimmer design. The comparison of both designs is shown in Figure 7.4.



WUV2024

Created Mar 10, 2025 Modified Mar 13, 2025

Calendar

From 2025-07-06 Time 17:35

To 2025-07-06 Time 17:35

Term

due in 4 months
due in 81 working days

From 2025-07-06 Time 17:35

To 2025-07-06 Time 17:35

WUV2024

Calendar date

From 6. 7. 2025 Time 17:30

To 7. 12. 2025 Time 17:45

Starts in: 7 days

Duration: 5 months, 1 day

Deadline

Startline 6. 7. 2025 Time 17:30

Deadline 7. 12. 2025 Time 17:45

Ends in: 7 days

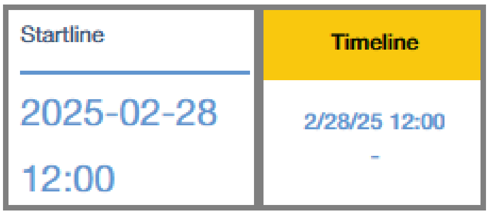
Duration: 5 months, 1 day

Figure 7.4: Comparison between the old Terms tab (left) and the new tab (right)

In addition to the previously mentioned automatic date filling being a problem for the users, the users also struggled with the tab layout and deletion of dates and times. An *X* symbol was added to the currently edited text box, which can be used to delete the date easier than in the previous version of the interface. The text boxes were also redesigned to an internally consistent design, a labeled square outline text box. The layout was also changed to visualize the *From* and *To* dates closer together to associate the time text boxes according to the proximity principle [7].

The additional information in the original *Terms* section, which visualized remaining time until a deadline, was also implemented in the new version. The information is displayed on the right side of each section and shows when the time period starts or when it ends, based on the period start date.

The updated design uses the Day-Month-Year format separated by dots to suit the mostly European user base. The ideal date format depends on the regions where the system will be used and the user base of the system. The previous ISO 8601 format (Year-Month-Day separated by dashes) should be selected if the system is used worldwide, as it is considered a universal format; however, it can be unintuitive for regular users who are not used to these formats. The user's date format could be set in the profile settings, which could solve the problem. The selected format should also be used consistently across the interface, which is currently not the case (as can be seen in Figure 7.5).



Startline

2025-02-28

12:00

Timeline

2/28/25 12:00

-

Figure 7.5: Comparison between formats in the Terms tab (left) and the Terms column (right)

The program should parse the user input according to the chosen date format and accept different inputs such as *"2000 12 3"*, to make it easier to input dates for keyboard users. This was repeatedly expressed by users during the testing, as it was difficult for them to type the date in the precise format present in the interface.

The updated *Terms* tab received an average updated score of 4.75, which means that the interface was significantly improved. The participants noted that the interface was more readable and simple and that they could check the time periods easily. The ability to clear a text box was also a positive according to the participants. The date format had mixed responses, as some users preferred the leading zeros in the date to align the dates in a column, while others preferred the reduced design.

7.5 Budget

The *Budget* tab was originally the lowest rated tab during user tests, mainly due to inconsistent labeling in the interface and lack of features. The layout of the tab was updated to a single column layout to prevent incorrect associations of labels and elements, and each of the sections was also updated to a more consistent design, with labeled and aligned text boxes. The label contents were also rewritten to better describe each of the elements, and the design of the input text boxes in the tab was unified with the other tabs.

A text box to set the currency was added to the budget planning section, which now contains the total budget (titled *Budget*), currency settings (titled *Currency*), and a visualization of the remaining funds (titled *Plan*).

The spending list was updated to be more consistent – the text boxes were labeled and the text of the number input fields is aligned to the right, which is a standard for displaying numbers, as it ensures the digits are aligned to make comparing their sizes easier. A header was also added to the list of expenditures, to label the columns, and to show the function of the list even when it's empty. The comparison between the empty interfaces is shown in Figure 7.6.

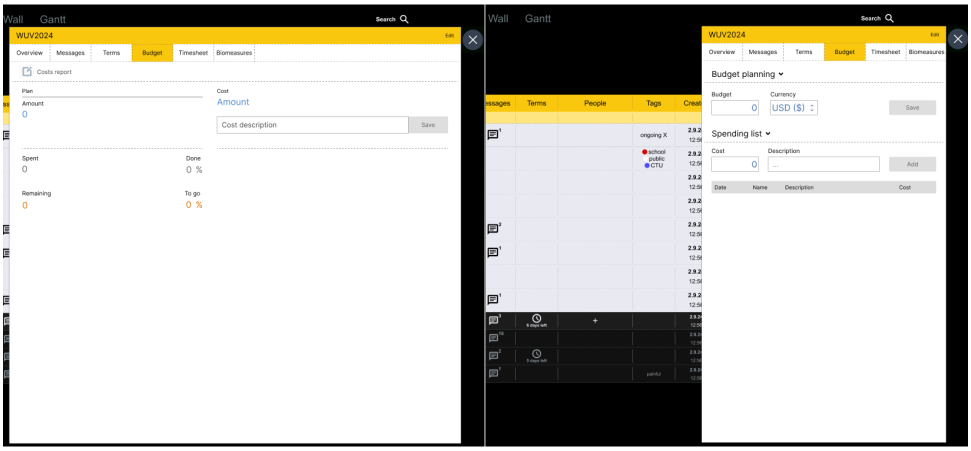
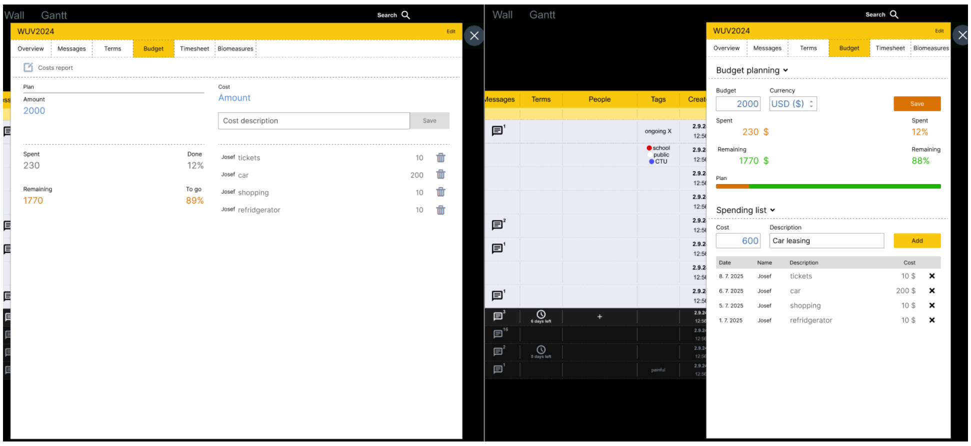


Figure 7.6: Comparison between the empty old Budget tab (left) and the new tab (right)

The confirmation button, originally located at the bottom of the window, was moved to the budget planning section in order to clearly link it to the data it is connected to according to the proximity principle [7]. The text on the confirmation button was also rewritten to *Save* instead of the original *Done*, as the word *Save* better describes the function of the button and uses familiar terminology used in other systems.

After the user fills in the total budget and adds items to the spending list, a visualization of the spent and remaining amounts is displayed in the budget planning section. The orange text displays spent finances, and the green text displays remaining finances. These colors were originally gray and orange, but the coloring was not intuitive for users, as most expected the amount spent to be colored orange. The updated budget planning can be seen in Figure 7.7.



Date	Name	Description	Cost
8. 7. 2025	Josef	tickets	10 $
6. 7. 2025	Josef	car	200 $
5. 7. 2025	Josef	shopping	10 $
1. 7. 2025	Josef	refridgerator	10 $

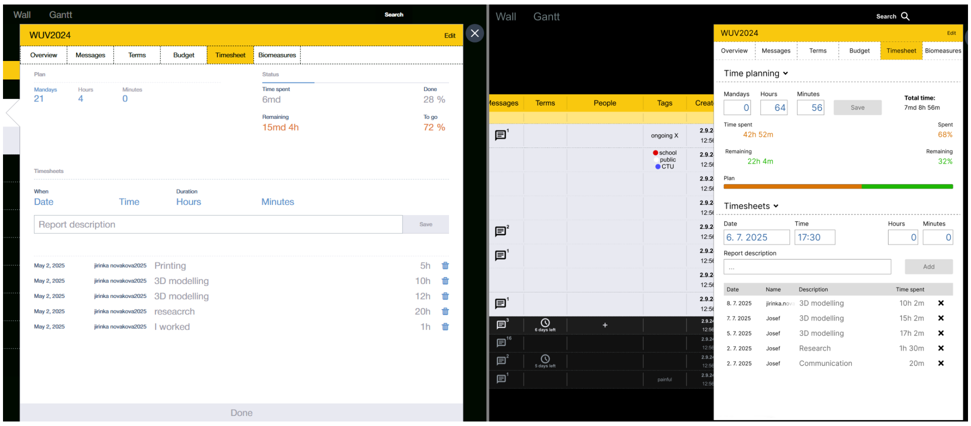
Figure 7.7: Comparison between the full old Budget tab (left) and the new tab (right)

The new *Budget* tab, which scored an average score of 4.75 by the participants, was deemed a significant improvement over the old version. The participants noted that the elements are clearly labeled and the interface is more intuitive than the previous design. The users preferred the new visualization bar, with some users noticing the functionality for the first time in the new interface. The spending list was also positively rated: *"I like the new spending list way more, everything seems more organized."*.

7.6 Timesheet

The *Timesheet* tab is not a feature selected for the testing, and therefore wasn't used by the participants during the interaction testing. Due to its similarity to the *Budget* tab, the design process was similar for both tabs.

The interface was simplified and its text boxes unified in design. Missing labels were added for hours and minutes in the bottom section of the tab to better describe the interface elements and prevent user errors. The general page layout was copied from the *Budget* tab redesign, with a planning section and a list section. The comparison between both versions of the *Timesheet* tab can be seen in Figure 7.8.



Plan
Mandays 21 Hours 4 Minutes 0

Status
Time spent 6md Done 28 %
Remaining 15md 4h To go 72 %

Timesheets
When: Date Time Duration: Hours Minutes
Report description

May 2, 2025	jirinka novakova2025	Printing	5h
May 2, 2025	jirinka novakova2025	3D modelling	10h
May 2, 2025	jirinka novakova2025	3D modelling	12h
May 2, 2025	jirinka novakova2025	reseacrch	20h
May 2, 2025	jirinka novakova2025	I worked	1h

Time planning
Mandays 0 Hours 64 Minutes 56
Total time: 7md 8h 56m
Time spent 42h 52m Spent 68%
Remaining 22h 4m Remaining 32%
Plan

Timesheets
Date 6. 7. 2025 Time 17:30 Hours 0 Minutes 0
Report description ...

Date	Name	Description	Time spent
8. 7. 2025	jirinka.nova	3D modelling	10h 2m
7. 7. 2025	Josef	3D modelling	15h 2m
5. 7. 2025	Josef	3D modelling	17h 2m
2. 7. 2025	Josef	Research	1h 30m
2. 7. 2025	Josef	Communication	20m

Figure 7.8: Comparison between the old Timesheet tab (left) and the new tab (right)

The timesheet functionality of the tab was analyzed after the user tests concluded. When adding an entry to the timesheet list, the original interface requires entering a date, time, a time amount in hours and minutes, and a description. This is different from contemporary timesheet applications such as Clockify, Toggl Track, or TimeCamp, which require either a time period and calculate the time amount automatically or require only the time amount. The time entered into the system is not displayed in the list and doesn't influence any other functionalities of the system, making the input field unutilized. The time text box was not removed from the new design, although the developers of the application should consider this inconsistency and address it by removing the time text box or adding functionality to it.

An issue was found with respect to the time conversion in the time planning section of the interface. The original *Timesheet* tab interface automatically converts hours to man-days and minutes to hours. However, this feature doesn't correctly calculate the total time. In a situation where the time inputted into the system is, for example, 1 man day, 25 hours, and 61 minutes, the program has difficulties correctly converting the information into a correct total sum of time. The conversion should happen only after the user saves the data, which would allow the system to calculate the total time correctly. Instead, the system converts the values immediately after entering them into the text box, overwriting the contents of other text boxes, and miscalculating the total time.

The comparison between both versions of the tab was shown to participants of the comparison testing; although they weren't asked to score the interface, but rather to describe the functionality of the new interface briefly. Due to the similarities between the new *Budget* and *Timesheet* tabs, the users could confidently guess the functionality of the elements of the tab.

7.7 Conclusion

The test results led to the creation of new designs for each of the task detail tabs, which were tested during the interaction tests. The tabs were confirmed to be an improvement over the original designs using comparison tests, with users rating the new interfaces favorably compared to the older versions. The participants evaluated the worst-rated tabs (*Terms* and *Budget*) as significantly improved, which was the goal of the tab redesign.

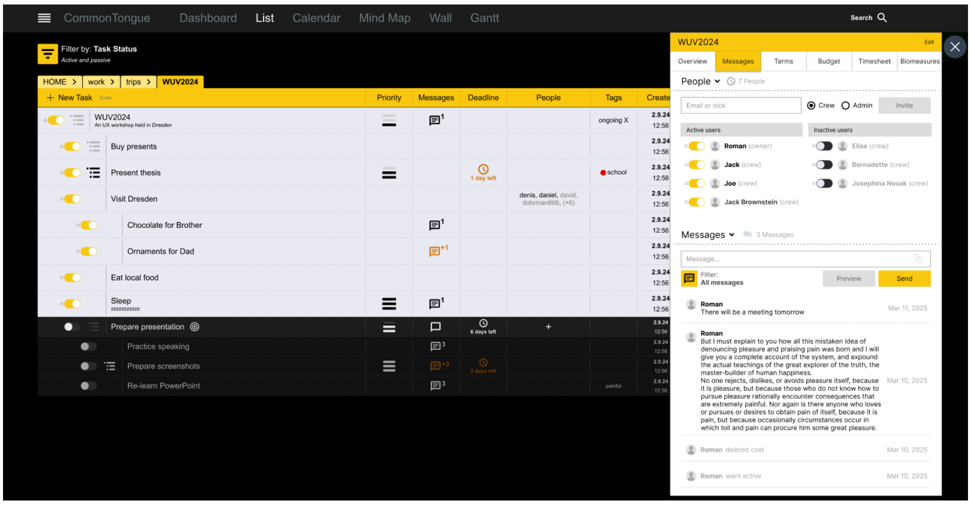
The principles used when designing the new *List* view design (presented in Chapter 3), along with newly created standards for the design of confirmation buttons, text boxes, and the task detail layout, should serve as a guide for the future development of all CommonTongue interfaces. A shortened interface development guide will be provided in a separate segment of the thesis titled *Guide*.

Chapter 8

Conclusion

During the course of the thesis, the *CommonTongue* program was tested with three rounds of usability tests. The program was analyzed and compared to other state-of-the-art project planning applications, and its interface was described in the research chapter. The *List* view and the *Task detail* were chosen as the most important parts of the system and redesigned.

The main interface for interacting with data in the system, the *List* view, was redesigned based on the instructions from the development team, together with consultations with beginner and expert users, and was proven to be an improvement over the old design during the *First impression testing*. The results of the test were also used to further update the design of the *List* view. The resulting interface is visible in Figure 8.1.



	Priority	Messages	Deadline	People	Tags	Create
WUV2024 An UX workshop held in Dresden		1			ongoing X	2.9.24 12:56
Buy presents						2.9.24 12:56
Present thesis			1 day left		● school	2.9.24 12:56
Visit Dresden				denis, daniel, david, dobrman666, (+6)		2.9.24 12:56
Chocolate for Brother		1				2.9.24 12:56
Ornaments for Dad		+1				2.9.24 12:56
Eat local food						2.9.24 12:56
Sleep zzzzzzzzzz		1				2.9.24 12:56
Prepare presentation			6 days left	+		2.9.24 12:56
Practice speaking		3				2.9.24 12:56
Prepare screenshots		+3	2 days left			2.9.24 12:56
Re-learn PowerPoint		3			painful	2.9.24 12:56

● Crew ○ Admin

Active users	Inactive users
Roman (owner)	Elise (crew)
Jack (crew)	Bernadette (crew)
Joe (crew)	Josephina Novak (crew)
Jack Brownstein (crew)	

Filter:
All messages

Roman
There will be a meeting tomorrow — Mar 11, 2025

Roman
But I must explain to you how all this mistaken idea of denouncing pleasure and praising pain was born and I will give you a complete account of the system, and expound the actual teachings of the great explorer of the truth, the master-builder of human happiness.
No one rejects, dislikes, or avoids pleasure itself, because it is pleasure, but because those who do not know how to pursue pleasure rationally encounter consequences that are extremely painful. Nor again is there anyone who loves or pursues or desires to obtain pain of itself, because it is pain, but because occasionally circumstances occur in which toil and pain can procure him some great pleasure. — Mar 10, 2025

Roman deleted cost — Mar 10, 2025

Roman went active — Mar 10, 2025

Figure 8.1: The final List view design with an opened Task detail window

The *Task detail* interface was analyzed and redesigned during *Interaction testing*, and the new designs were evaluated using *Comparison tests*. The *Task detail* designs were considered a major improvement over the older designs, being rated as *Better* to *Significantly better* on the Likert scale used during *Comparison testing*.

The designs for the *List* view and the *Task detail* interfaces were created using the User-Centered Design methodology, and multiple iterations of the changes made in the interface were showcased. The design changes made during the development of the thesis used design concepts such as Nielsen heuristics and Gestalt principles. The main focus of the design was to create a readable and intuitive interface that would satisfy the needs of the *CommonTongue* user base.

The goal of the thesis was achieved and a design guide was created to assist future development of the program.

Bibliography

[1] https://hydronaut.commontongue.com/

[2] https://hydronaut.eu/en

[3] C. Ware, Visual Thinking: for Design. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2008.

[4] J. Nielsen, “10 Usability Heuristics for User Interface Design,” Nielsen Norman Group, 1994. [Online]. Available: https://www.nngroup.com/articles/ten-usability-heuristics/

[5] E. Goodman, M. Kuniavsky, and A. Moed, Observing the User Experience: A Practitioner’s Guide to User Research, 2nd ed. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2012.

[6] S. Krug, Don’t Make Me Think: A Common Sense Approach to the Web, 2nd ed. USA: New Riders Publishing, 2005.

[7] D. Todorovic, “Gestalt principles,” Scholarpedia, 2008. [Online]. Available: http://www.scholarpedia.org/article/Gestalt_principles

[8] “Latin Square Designs,” in The Concise Encyclopedia of Statistics, New York, NY, USA: Springer, 2008. [Online]. Available: https://doi.org/10.1007/978-0-387-32833-1_223

[9] G. A. Miller, "The magical number seven, plus or minus two: Some limits on our capacity for processing information," Psychol. Rev., vol. 63, no. 2, pp. 81–97, 1956. doi: 10.1037/h0043158.

[10] N. Cowan, "The magical number 4 in short-term memory: a reconsideration of mental storage capacity," Behav. Brain Sci., vol. 24, no. 1, pp. 87–114, Feb. 2001; discussion pp. 114–128. doi: 10.1017/s0140525x01003922. PMID: 11515286.

[11] J. Brooke, "SUS: A quick and dirty usability scale," 1996.

[12] R. Likert, "A technique for the measurement of attitudes," Arch. Psychol., vol. 22, no. 140, pp. 1–55, 1932.

Guide

This segment will be used to briefly describe the design process and link to the appropriate chapters in order. Only final versions of changes will be described; for a detailed description, please read the appropriate chapter. The interface versions created during the design process are in the *Figma designs* folder in the additional content appended to this thesis.

List design

The first interface that was updated was the *List* view in Chapter 3.

- **Table header** 3.1 - swapped with breadcrumb navigation
- **Horizontal spacing** 3.2 - added separators to table
- **Inactive tasks** 3.3 - make task background, text and icons darker to reduce contrast; make inactive tasks vertically smaller with smaller icons and text
- **Recoloring** 3.4 - recolor text and icons to black; recolor interface to three colors (yellow+black+white)
- **Replace icons** 3.4 - rounded black icons, terms has clock icon, messages shows number of messages diagonally

The initial design was focused on creating a consistent interface, while also minimizing the amount of colors used where necessary.

List update

After testing the new design, some changes were added in Chapter 5 to improve the *List* view design.

- **Filter** 5.1.1 - updated design, added icon, suggested fixing functionality
- **Breadcrumb navigation** 5.1.2- new design, added highlighting to currently selected task
- **Column filter** 5.1.3 - added plus icon at last column, suggested functionality

- **Hierarchy icon** 5.2 - remade in new design, recolored, round
- **New Task button** - added keyword *Enter* next to it
- **Renaming** 5.3 - *Term* column renamed to *Deadline*; *When* column renamed to *Date*
- **Budget column** 5.3 - added formatting of text; suggested functionality for fitting data in slim column; added currency

The updated design was based on user feedback and therefore focused more on interface details. Its goal was to fix the remaining problems and inconsistencies in the interface, such as poor color choices made in previous versions, and reviewing features which were not updated in the new design.

Task detail

The task detail was updated in Chapter 7 after a third round of testing. The task detail contains six tabs, five of which were modified. All tabs were redesigned to a slimmer single-column design, which better utilizes screen space, and has a simpler structure compared to the previous designs. The main principles for designing the new tabs were:

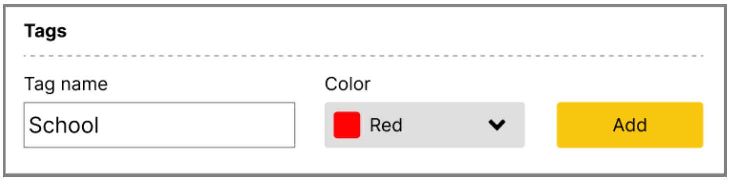
- **Consistency** - all of the interface elements need to look consistent across all tabs
- **Segmentation** - the layout is separated into sections which have different functionalities
- **Clarity** - the interface should be clearly labeled and accessible to users who never saw/used the interface before
- **Alignment** - the elements in the interface are aligned when possible
- **Preventing errors** - the interface should clearly signal unsaved changes in a consistent way to prevent accidents

---

Overview 7.2 tab had minor functionality changes added.

- **Priority** - *Priority* text in the selection shows the currently selected priority
- **Task description** - looks like a text box, consistent with other input fields; shows saving data below the text in gray
- **Tags** - tags are shown in two different columns; updated icon to remove a tag; adding a color to a tag can be added (Figure 8.2)

Messages 7.3 tab was divided into two segments, the people section and the chat section.



Tags

Tag name

School

Color

🟥 Red

Figure 8.2: A concept for tag functionality with colors included

- **Invitation** - changed order of elements to email-role-invite; renamed Owner to Admin
- **People** - people columns labeled
- **Chat** - added filter toggle between system and user messages; removed chat name offset

---

Terms 7.4 tab was heavily updated, due to the original negative reception during testing.

- **Layout** - changed the horizontal time periods to vertical, grouped closer together
- **Automatic filling** - suggested removing automatic filling of both dates on click
- **Removing dates** - added a button to remove the currently edited date
- **Format** - suggested to unify the format with the when and timeline column; updated to European format DD.MM.YYYY (YYYY-MM-DD also possible if used internationally)
- **Information** - information about the time period added to each segment

---

Budget 7.5 tab was also heavily updated, due to the original negative reception during testing.

- **Layout** - split the layout into two segments, planning and spending list
- **Currency** - added a selection for currency; added currency to list of spendings
- **Visualization** - updated visualization colors and bar was made bigger
- **List** - unified form design; date of adding was added; icon for removing items replaced by cross

---

Timesheet 7.6 tab was updated to be similar to the budget tab

- **Layout** - split the layout into two segments, planning and spending list
- **Formatting** - suggested updating the formatting functionality to properly count the total time

- **Visualization** - updated visualization colors and bar was made bigger
- **List** - unified form design; suggested reconsidering timesheet input fields (the reason for time period + time amount); icon for removing items replaced by cross

Additional content

All materials used to complete the thesis are added as an attachment to the thesis. The structure of the folder can be seen in the following diagram:

```
Masters thesis
├── Images
│   ├── Comparison testing
│   ├── Figma designs
│   ├── Initial impression testing
│   └── Thesis
└── Testing content
    ├── Onboarding.pdf
    ├── Scripts.pdf
    ├── Surveys.pdf
    └── Test results.pdf
```

The **Images** folder contains images used in the thesis and during testing.

The **Figma designs** subfolder contains screenshots of the Figma workspace and captures the design process behind the new designs created in this thesis.

The **Testing scripts** folder contains the scripts and surveys used during the user testing. The scripts are written in Czech.