BRNO UNIVERSITY OF TECHNOLOGY
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FACULTY OF INFORMATION TECHNOLOGY
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

TRANSCRIPTION AND ANNOTATION COMPONENTS FOR WEB EDITOR IN REACT

REACT KOMPONENTY PRO WEBOVÝ EDITOR TITULKOVÁNÍ A ANOTACÍ AUDIA

BACHELOR'S THESIS
BAKALÁŘSKÁ PRÁCE

AUTHOR — JAKUB DUGOVIČ
AUTOR PRÁCE

SUPERVISOR — Ing. IGOR SZŐKE, Ph.D.
VEDOUCÍ PRÁCE

BRNO 2024

BRNO UNIVERSITY OF TECHNOLOGY FACULTY OF INFORMATION TECHNOLOGY

Bachelor's Thesis Assignment

Institut:	Department of Computer Graphics and Multimedia (DCGM)	155761
Student:	**Dugovič Jakub**	
Programme:	Information Technology	
Title:	**Transcription and annotation components for web editor in React**	
Category:	Web applications	
Academic year:	2023/24	

Assignment:

1. Learn about the React framework and the existing audio transcription and annotation platform. Next, study the available transcription tools. Focus on UX/UI.
2. Identify the appropriate components for implementation and UX improvement.
3. Design and implement selected components. During implementation, evaluate their UX regularly. Also integrate you solution into an existing API regularly.
4. Test these components on the appropriate user group. Verify the ease of use of the interface.
5. Discuss the achieved goals and suggest directions for further development.
6. Create an A2 poster and a 30-second video presenting the results of your work. Publish the tool as open source on GitHub.

Literature:
- Tidwell et al.: Designing Interfaces: Patterns for Effective Interaction Design, O'Reilly, 2020
- Steve Krug: Don't Make Me Think, Revisited: A Common Sense Approach to Web Usability, ISBN: 978-0321965516
- Steve Krug: Rocket Surgery Made Easy: The Do-It-Yourself Guide to Finding and Fixing Usability, ISBN: 978-0321657299
- Joel Marsh: UX for Beginners: A Crash Course in 100 Short Lessons, O'Reilly 2016
- Next, according to the supervisor suggestions.

Requirements for the semestral defence:
Items 1-3 of the assignment.

Detailed formal requirements can be found at https://www.fit.vut.cz/study/theses/

Supervisor:	**Szőke Igor, Ing., Ph.D.**
Head of Department:	Černocký Jan, prof. Dr. Ing.
Beginning of work:	1.11.2023
Submission deadline:	9.5.2024
Approval date:	9.11.2023

Abstract

This thesis aims to implement modular user interface for audio transcription and annotation. It expands upon existing work in order to enable and improve working with hours-long conversation recordings. The solution is implemented in TypeScript using React and additional libraries from the React ecosystem. Applying principles from the studied literature, avoiding issues identified during the research a similar platform, and verifying the interface throughout the development using qualitative testing, the interface strives to achieve high degree of good user experience.

Abstrakt

Cieľom tejto práci je implementovať modulárne užívateľské rozhranie na prepis zvukových nahrávok a ich anotáciu. Rozširuje dotrajšiu prácu s cieľom umožniť a zjednodušiť prácu s hodiny dlhými nahrávkami rozhovorov. Riešenie je implementované v TypeScripte pomocou Reactu a ďalších knižníc z reactového ekosystému. Aplikujúc princípy naštudované z literatúry, vyhýbajúc sa chybám identifikovaným počas prieskumu obdobnej platformy a overujúc užívateľské rozhranie počas vývoja pomocou kvalitatívneho testovania, vyvýjané rozhranie sa usiluje dosiahnuť vysokú mieru dobrej užívateľskej skúsenosti.

Keywords

React, Redux, web app, audio player, audio transcripion, phonetic transcription, Wavesurfer, graphical user interface, GUI, UI, usability, user experience, UX, experience desgin, XD, web design, frontend, qualitative testing, interface testing, SUS, long audio files, long recordings, entity grouping, metadata, entity tagging

Kľúčové slová

React, Redux, webová aplikácia, audio prehrávač, prepis audia, fonetický prepis, Wavesurfer, grafické užívateľské rozhranie, GUI, UI, použiteľnosť, užívateľská skúsenosť, UX, dizajn užívateľskej skúsenosti, XD, webový dizajn, frontend, kvalitatívne testovanie, testovanie užívateľských rozhraní, SUS, dlhé zvukové súbory, dlhé nahrávky, zoskupovanie entít, metadáta, značkovanie entít

Reference

DUGOVIČ, Jakub. *Transcription and Annotation Components for Web Editor in React.* Brno, 2024. Bachelor's thesis. Brno University of Technology, Faculty of Information Technology. Supervisor Ing. Igor Szőke, Ph.D.

Transcription and Annotation Components for Web Editor in React

Declaration

I hereby declare that this Bachelor’s thesis was prepared as an original work by the author under the supervision of Mr. Igor Szőke. The supplementary information was provided by Mr. Jozef Žižka. I have listed all the literary sources, publications and other sources, which were used during the preparation of this thesis.

. . . .
Jakub Dugovič
May 8, 2024

Acknowledgements

I would like to express my heartfelt gratitude to the thesis supervisor for all the consultations, insightful advice, and honest feedback that enabled me to progress and improve the outcome. My thanks also go to Mr. Žižka, who helped me connect the app to the API and made changes to the backend, as well as to all the participants who volunteered to join the testing sessions.

Contents

Chapter 1

Introduction

This thesis deals with the topic of audio recording transcription and tagging. Audio transcript editing and annotating have plenty of practical applications. Among the many uses, for example, one rising to prominence is annotating data for machine learning model training.

My work is related to project JARIN which uses machine learning to preserve, document, and present Czech dialects. It endeavors to explore, identify, and implement components and features for a graphical editor that would enable and simplify work with long audio recordings and their transcripts.

At the beginning, this thesis explores various principles to achieve good user experience. The theoretical research describes user subconscious behavior. Then it synthesises the core aspects needed for a good and understandable graphical interface that users will navigate and control with ease. Practical research explores an existing interface with some shared functionality.

The initial draft was based on the Bachelor's thesis of Jan Plhal[7] that deals with a similar issue. He created a transcript editor for air traffic control communication. Chosen functionality was adopted and tweaked to fit the app's use cases. Based on the acquired knowledge, new components expanding functionality were proposed as well. They aim to improve user orientation and enable bulk tagging with groups. Alongside, unified interaction design should improve many small details and the overall feel of the product.

The practical part of this thesis aims to create a user-friendly interface that would be easy to learn, adapt, and use for the target user base. Implementation strives to create a modular and easily expandable project. Implementation technologies were chosen and described. The interface is broken down into several main areas. They are further described down to subcomponent purpose, structure, and interactions. Adopted components are implemented in a way that complements the work's use case.

Testing was an essential part of the development. In parallel with the implementation, there were four rounds of testing. They resulted in various modifications, improvements, and new feature requests. Outcomes from testing were incorporated into the implementation and resulted in, for example, a new component that simplifies work with accented characters that denote certain phonemes.

Finally, the work was assessed and created interface was analyzed. Possible expansions in the future were discussed too.

Chapter 2

Theoretical and Practical Research

User experience (UX) is the overall impression and satisfaction one has when interacting with a product, service, or system. Good UX is an emergent phenomenon. There is no simple list of things to tick that will lead to good UX, rather it is a result of a meticulous process of drafting, prototyping, testing, evaluation and all over again many times. However, there are several aspects to consider to achieve the desired results.

2.1 Approach to Design

This section discusses how users tend to use web pages and, with this in mind, how to provide information and understandably structure the contents. Further, there will be additional remarks on accessibility and color. Ideas in this section mostly come from the book *Don't Make Me Think* [4] further complemented by specific topics from *Universal Principles of Design* [5].

When Average Joe Opens a Website

Seldom do users come to a website or a web app without a goal in mind. They want to find some information, perform a task, or buy something. Purpose of a website should be identifiable at a glance. Rather than reading the contents line-by-line or the menu items one after the other, it is more efficient to just scan the whole page and skip ahead to the relevant section.

In general, a site is scanned from top to bottom left-to-right[1]. There is a tendency to click on the first relevant link or button as opposed to finding the most relevant one. In Figure 2.1, heatmap visualization is created from data obtained by eye-tracking that illustrates this principle.

Once a process that leads to the desired results is discovered, users stick with it even though there might be a better alternative. Moreover, *Universal Principles of Design* states: *„There are three types of problems where satisfiability needs to be taken into account: very complex problems, time-limited problems, and problems where exceeding sufficient solution leads to diminishing returns.“* [5, page 210].

After the system is in use for a while, so-called *desire lines* emerge, e.i. sequence of steps to achieve some goal. The steps may or may not follow what the designer intended; how-

[1]This is influenced by reading direction and might be different for right-to-left written languages such as arabic or hebrew.

ever, incorporating the unexpected procedures into the workflow is preferred to enforcing the originally intended method.

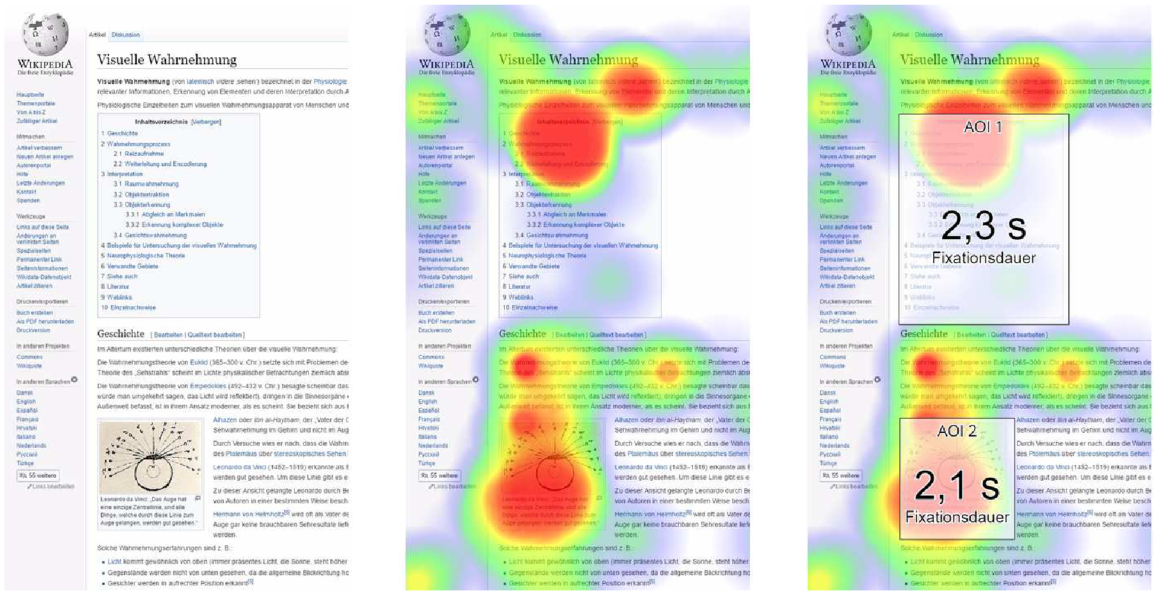


Figure 2.1: Example of how websites are scanned in the first few seconds. There is a screenshot (left), heatmap generated from eye-tracking data from three participants (center), and two areas of interest – the contents and an image – with duration. The original German description was cropped off from this image. Attribution: EYEVIDO, Copyrighted free use, via Wikimedia Commons.

Understandable Design Is Usable

In his book [4, page 29], Steve Krug lists these ways to convey information:

> - *Take advantage of conventions.*
> - *Create effective visual hierarchies.*
> - *Break pages into clearly defined areas.*
> - *Make it obvious what's clickable.*
> - *Eliminate distractions.*
> - *Format content to support scanning.*

Conventions & Consistency

Conventions provide something familiar and thus self-evident. It's important to follow design conventions that have evolved in the past decades and, if applicable, employ fitting physical world metaphors (e.g. shopping cart). This can be best demonstrated on shopping site top bars (Figure 2.2). Upholding conventions helps create clear and navigable interfaces. Conventions emerge when functional design gets adopted, or by convergent evolution when multiple approaches end up with very similar results after a series of smaller improvements.

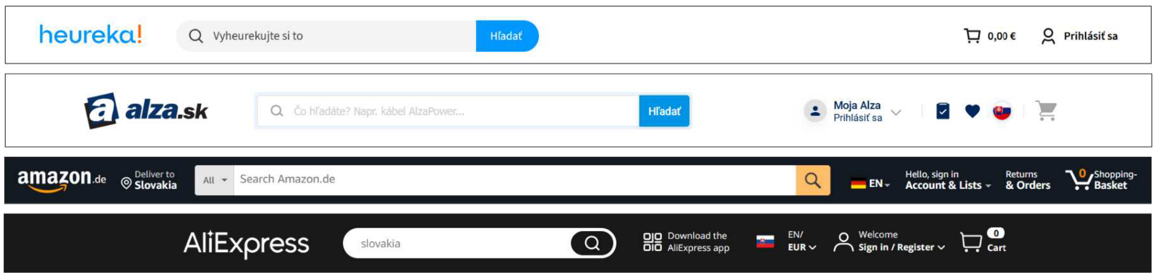


Figure 2.2: Example of website conventions, specifically shopping site top bars. There is a logo on the left, a search bar in the middle, and a login/account and shopping cart on the right.

Consistency helps make the components usable and there are several types of consistency. Consistency in esthetic features and style, which are associated with a brand for example. Functional consistency, consistency with other components of the same interface, and consistency with similar components in other systems.

Additionally, clickable elements should be easily discernable and this can be achieved through a consistent combination of visual features (e.g. underline, color, font, hover effects).

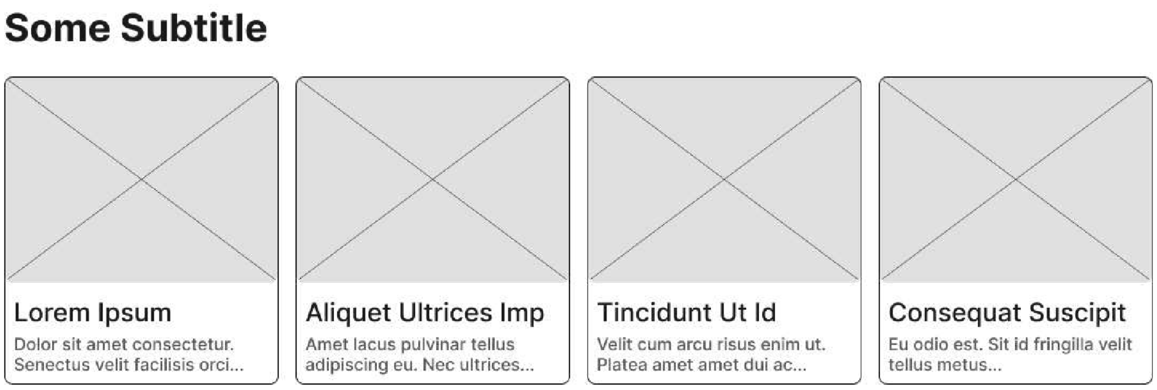
Hierarchy & White Space

Parsing of the site happens subconsciously from the first moment one looks at it. The main blocks or parts should be clearly visually bounded and their purpose should be immediately obvious. *Visual cues* help define the boundaries of elements and show their relationship visually. They include, for example, spacing around and between objects, dividing lines, size, and color. There are several relationship types to consider:

- *Similar* things should look the same or very similar in the whole interface, for example all buttons on a page or all the links in a text.
- *Relatedness* is visually represented by closeness. Elements that are closer together are perceived as related. Visual grouping based on relatedness decreases perceived visual complexity and improves clarity because the elements (e.g. playback control buttons) fall into one „box“ when scanning.
- *Importance* can be shown with size, color, more spacing, higher contrast, or being closer to the top of the page. Careful choices are required because if everything is important nothing is important – if an article was written in title case, the title would lose its importance.
- Being *part of something* is visually represented by nesting. For example, an item is part of a list, the list is part of a navigation menu, the menu is located in the top bar, which is one of the main parts of the whole page.

Moreover, *Principles of Web Design* [6] states that space between elements is almost as important as the elements themselves. An appropriate amount of white space gives enough „breathing room“ to the elements. On one hand, using more white space can make us perceive elements as separate; on the other hand, less white space signals that elements belong to the same group. Multiple levels of spacing are deliberately used to create a hierarchy of related and less related parts of an interface. Also called the *figure-ground relationship*, the designer needs to strike the right balance between the element and the white space around it. Larger empty space around an object signifies its importance.

Insufficient use of these principles forces users to think about what belongs where and thus induces a higher cognitive load. These principles are further illustrated by examples in Figure 2.3 and Figure 2.5.



Some Subtitle
Lorem Ipsum
Dolor sit amet consectetur. Senectus velit facilisis orci...
Aliquet Ultrices Imp
Amet lacus pulvinar tellus adipiscing eu. Nec ultrices...
Tincidunt Ut Id
Velit cum arcu risus enim ut. Platea amet amet dui ac...
Consequat Suscipit
Eu odio est. Sit id fringilla velit tellus metus...

Figure 2.3: Illustration of principles that create hierarchy. The cards are nested under the subtitle and there is a larger whitespace around the whole section than between the elements within the section. The cards show the same type of content and the spacing between them reflects it (the cards could be, for example, a list of articles, recipes, accommodations, products, movies, or places to visit). The importance of the subtitle is shown – it is bold and the font size is increased. The elements of a card are nested inside – visually shown by the border and small margin around it. In the card, the image is important – it is large, and the heading is important too – it is larger than the normal text and has higher contrast with the background. On the other hand, the text under the card heading is of regular size and gray color in order not to attract too much attention; it is the least important part of the card.

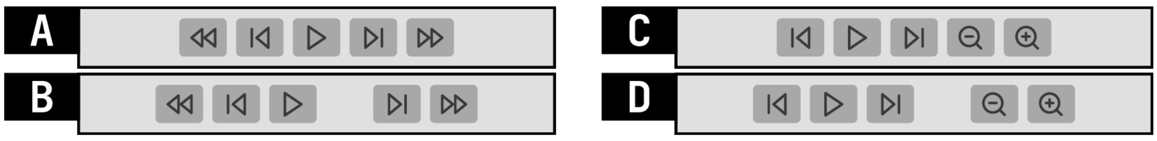


Figure 2.4: In `A` and `C`, buttons are perceived as a group of 5, all are spaced apart equally and therefore perceived as equally related. In `B` and `D`, there is one larger space creating two groups of buttons. The buttons within a group are more related than the buttons in different groups. The arrangement of buttons in `A` is clear, and the playback controls are related to each other equally. In `C`, the arrangement is the same; however, it is flawed since it does not correctly visually represent the relatedness of the buttons; zoom controls are not playback controls. A better arrangement for this case is illustrated in `D`. Analogically, in `B`, the two-group arrangement is not fitting for the set of equally related buttons.

Accessibility Matters

Applying the principles mentioned in previous sections adds a lot of accessibility by itself. The design will be understandable and controllable. Improvements to accessibility are generally beneficial for all users (e.g. better readability). Accessible design should also be perceivable and forgiving to users with diverse needs and abilities.

Guidance on Web Accessibility and the ADA[2] lists these accessibility issues:

> - *Poor color contrast.*
> - *Use of color alone to give information.*
> - *Lack of text alternatives ("alt text") on images.*
> - *No captions on videos.*
> - *Inaccessible online forms.*
> - *Mouse-only navigation (lack of keyboard navigation).*

Some of these apply to every website, and some are more specific. In general, some websites require a higher degree of accessibility (e.g. government websites, news websites) than others (e.g. games and specialized software). The interface developed in this thesis falls more on the specialized software side, and not all points in the list are achievable.

Nevertheless, there is some „low-hanging fruit" that helps a significant percentage of the potential user base. For example, there might be people with impaired vision or old monitors with incorrect color representation and/or bad viewing angles. Among the numerous small improvements belong:

- **Sufficient text size.** GOV.UK Design System[3], which is an inspiration in this aspect for this thesis, uses the progression 14, 16, 19, 24, 27, 36, 48, 80 [px].
- **Good color contrast.** The Web Content Accessibility Guidelines recommend a minimal contrast ratio of 4.5:1 for text and 3:1 for large text and UI components.[4]
- **Large enough clickable targets.** The Web Content Accessibility Guidelines[5] recommend at least 44 by 44 CSS pixels; however, there are some exceptions (e.g. inline links).
- **Sans-serif fonts.** In general, sans-serif fonts are more readable than serif ones.

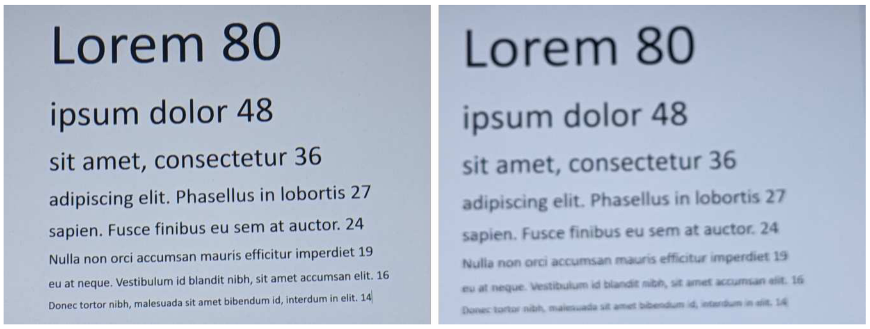

Lorem 80
ipsum dolor 48
sit amet, consectetur 36
adipiscing elit. Phasellus in lobortis 27
sapien. Fusce finibus eu sem at auctor. 24
Nulla non orci accumsan mauris efficitur imperdiet 19
eu at neque. Vestibulum id blandit nibh, sit amet accumsan elit. 16
Donec tortor nibh, malesuada sit amet bibendum id, interdum in elit. 14

Figure 2.5: Approximate illustration how I see with (left) and without (right) dioptric glasses, distance from the monitor is roughly 40cm.

[2] https://www.ada.gov/resources/web-guidance/, quoted 2024-01-09

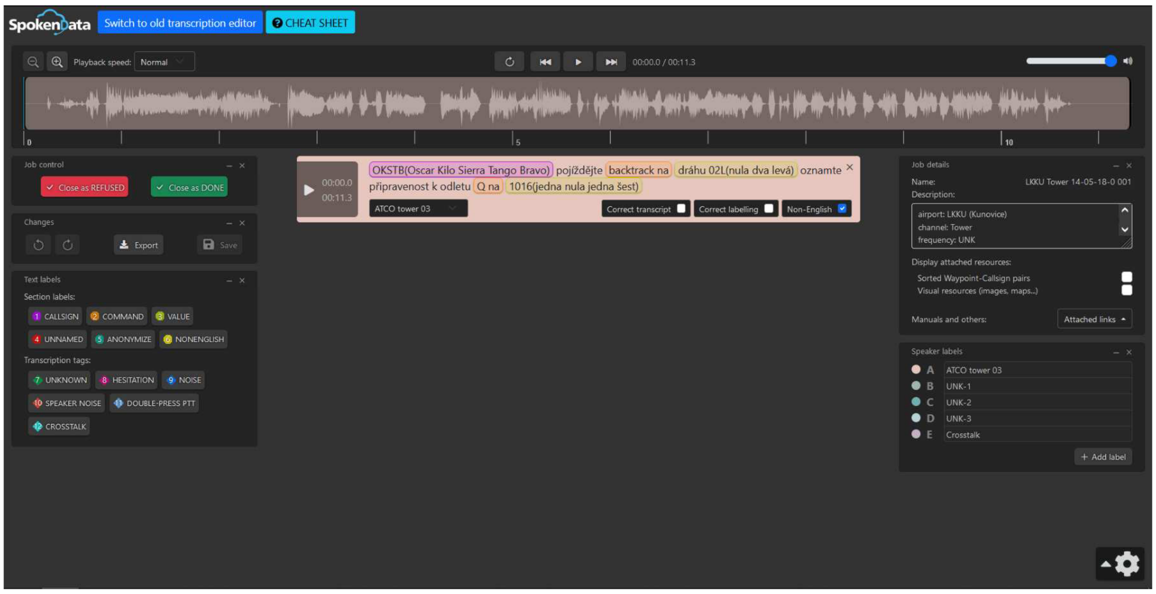
[3] https://design-system.service.gov.uk/

[4] For text recommendations see https://www.w3.org/WAI/WCAG21/Understanding/contrast-minimum, for non-text recommendations see https://www.w3.org/WAI/WCAG21/Understanding/non-text-contrast

[5] https://www.w3.org/WAI/WCAG21/Understanding/target-size.html

2.2 Analysis of the SpokenData Interface

This section focuses on the assesment of an existing interface with a similar purpose as the developed interface. Analysis of some annotation services, namely Prodigy and Deepsy, was already conducted by Plhal [7] and some findings were implemented in his SpokenData[6] editor. The interface focuses on checking, labeling, and editing communication between a pilot and an air traffic control officer. A screenshot of the interface is in Figure 2.6.



00:00.0
00:11.3

OKSTB(Oscar Kilo Sierra Tango Bravo) pojíždějte backtrack na dráhu 02L(nula dva levá) oznamte připravenost k odletu Q na 1016(jedna nula jedna šest)

ATCO tower 03

Correct transcript ☐ Correct labelling ☐ Non-English ☑

Name: LKKU Tower 14-05-18-0 001
Description:
airport: LKKU (Kunovice)
channel: Tower
frequency: UNK

Display attached resources:
Sorted Waypoint-Callsign pairs ☐
Visual resources (images, maps...) ☐

Manuals and others: Attached links

A	ATCO tower 03
B	UNK-1
C	UNK-2
D	UNK-3
E	Crosstalk

Figure 2.6: Screenshot from the SpokenData editor created by Plhal [7]. The interface contains a waveform, a transcript segment, cards with actions, labels, file information, and speakers.

Initially, I had to learn how the interface works, as I had never edited or annotated an audio transcript before. During the learning process, I came across multiple difficulties that need to be addressed. Some of them are on the level of short mental back and forth that adds unwanted cognitive load, but some are serious issues and solutions to which I only discovered after reading Plhal's thesis [7]. The most important points are:

- **Segment addition.** I came across a recording where I needed to add a segment. After several minutes I could not find a solution and I could not finish reviewing the transcript. Later I found out that the only way to add new segments is to drag on the waveform visualization.

- **Editing segment text.** To edit the text or to add labels, the user needs to click on the text. This took me only a couple of seconds; however, those seconds were a bit confused because the text does not provide any visual cues that it is clickable. Even on hover, there is no interaction. Also, when selected, the only way to identify that the text is selected is by the blinking cursor. Not only is this insufficient feedback for the user but also is it inconsistent with the rest of the interface.

- **Changing segment start or end.** The side handles of segment visualizations on the waveform are black and blend into the dark gray background. This makes it easy

[6] https://www.spokendata.com/atc

to overlook that they are draggable. Since it is the only way to change the start and end times of a segment, it is a design flaw.

- **The whole page scrolls.** The cards/boxes/areas are draggable and can be arranged by the user. In some arrangements, the contents of the page become higher than the viewport height and a scrollbar appears (one which scrolls the whole page). When scrolling, parts of the interface leave the view, e.g. the playback controlls or part of the waveform, which caused repeated scrolling up and down throughout the editing process.

Additionally, having spent hours editing recordings in the editor, there were some inconveniences and unusual design choices. On one hand, they were not limiting user's ability to do the work; on the other hand, I would describe them as non-ergonomic or ugly. They are not substantial on their own, but together decrease user experience by a noticable amount.

- **Placement of the settings button.** Bottom right is a very odd place to put a settings button. By convention, settings appear somewhere in the top bar or the menu.
- **Top bar „visual salad“.** The two buttons next to the logo are disturbing. They are bright and saturated, and thus signal a lot of importance, which they should not have. Both have different styling as well.
- **Controls above the waveform.** Playback controls are one of the most used actions in the whole app. Placing them below the waveform, instead of above, would reduce mouse travel from the segments when editing and annotating.
- **Confusing labeling.** For example, the first time I saw „Close as REFUSED“, I spent a good 30 seconds wondering what it meant. Will I renounce the editing of the transcript and someone else will be able to edit it? Does it mean that I will label the recording as unusable? What does it mean to refuse a transcript? Those were some of the questions I was asking myself.

To conclude, the interface does its job and there are only a few flaws. There are also things it does well, for example, the overall default layout. By improving many small things, the user experience could be improved further.

Chapter 3

Initial Draft

While the research was being conducted the early work on initial drafts started. This chapter captures the early proposals including planned functionality, general user interface layout, and some notable wireframes. A new interface should provide an easy workflow to transcribe, edit, and annotate long audio recordings.

3.1 Proposed Functionality

Most of the functionality from the original SpokenData[1] editor and the editor developed by Jan Plhal as his Bachelor's thesis [7] was kept. New components and changes were chosen based on user feedback and after consultation with the thesis supervisor.

Adopted Funcionality

A selection of features that were considered also fitting for the use case of this thesis was chosen:

- Waveform view with segment visualization and the ability to zoom.
- Audio playback, skipping, and playback speed and volume settings.
- Text segment visualization with the ability to edit start and end times, speakers, and segment labels.
- Text editing, the ability to label one or more words and to insert special labels.
- Segment playback, creation, and merger of neighboring segments.
- Workflow actions – close as done, close as refused, save, . . .
- View recording details, manage speakers, view a list of labels, playback settings.
- Undo and redo changes.
- Execute actions using keyboard shortcuts.

[1] https://www.spokendata.com/atc

Added Features

New functionality from additional requirements:

- **Minimap view.** It will provide a constant overview of the whole recording during the whole process alowing the users to orient themselves easily and jump in between different parts of the long audio file.
- **Segment grouping** and **group labels** will enable segment tagging in bulk. All segments in a chosen portions of a recording would be associated with a set of selected labels.
- Unified interactions, behavior, and visual design. It will nudge users in the right direction. All editable or clickable components will provide visual feedback.
- Any new requirements that arise during development and testing.

3.2 User Interface

The general placement of the main interface features/components was established during the early interface planning and has not changed since. It has proven understandable and functional in the draft, implementation, and testing phases.

Design Inspirations

The overall design draws inspiration from Material Design[2] and Carbon Design System[3]. Various design elements and visual features were kept from Plhal's interface [7].

Figure 3.1: Carbon Design System grayscale color palette used in the interface. It consists of a wide range of shades that are still distinguishable from one another.

Interface Composition

The user interface consists of five main areas. They are *top bar*, *playback*, *sidebar*, *transcript*, and *groups*. Figure 3.2 is a simplified wireframe that shows the placement of each of the areas. Some of the most important components are marked as well. Each one of the main areas is tied to a core feature.

[2]https://material.io/
[3]https://carbondesignsystem.com/

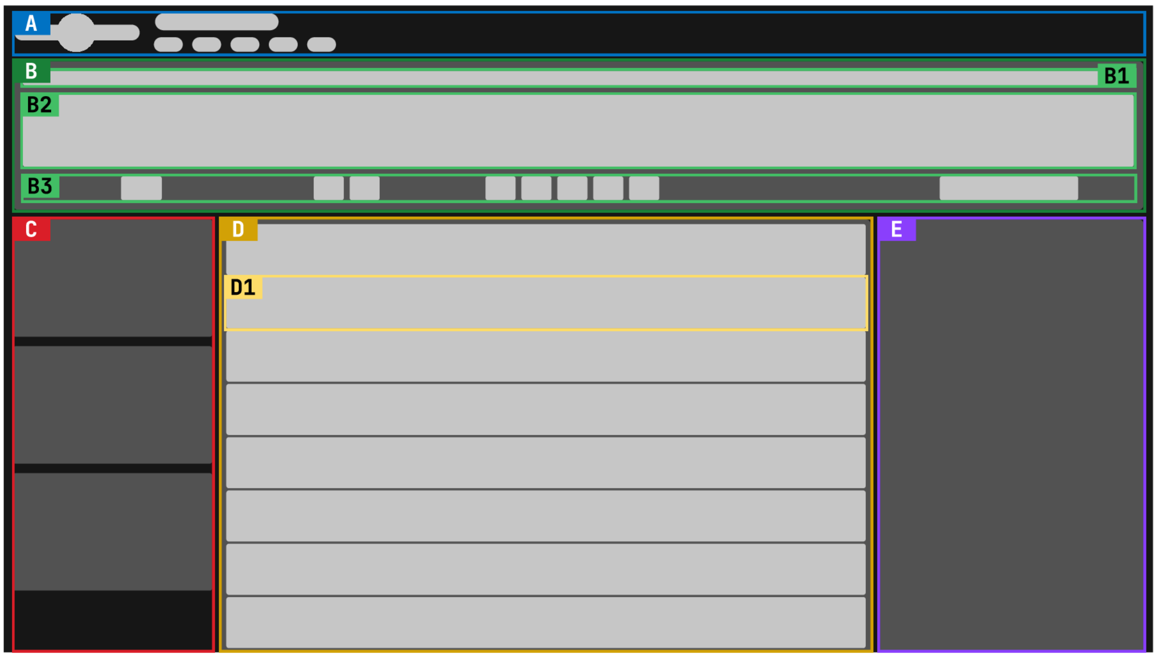


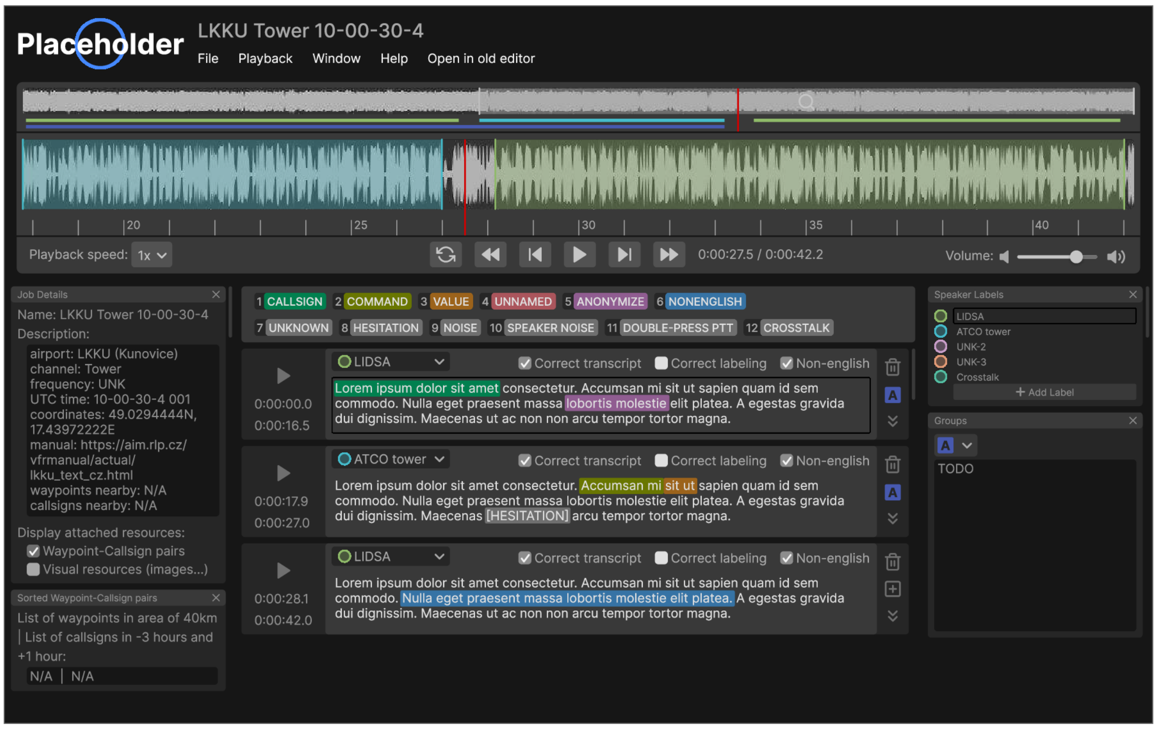
Figure 3.2: A simplified wireframe showing all five main areas of the interface. Additionally, some significant components are highlighted. The area `A` is *top bar*. `B` is the *playback* area containing a *minimap* in `B1`, the *waveform* in `B2`, and the *controls* in `B3`. In `C`, there are auxiliary components. Area `D` marks the *transcript* that is comprised of *segments*, one of which is highlighted in `D1`. Lastly, there are *groups* in `E`.

The main areas of the interface:

- **Top bar** contains a logo, a title, and the main menu. Its purpose is to inform a user and to provide buttons for important actions, e.g. save, undo, and redo.
- **Playback** area consists of the minimap, waveform, and controls. Its purpose is to present an audio file and to work with it. Minimap provides a visualization of the whole recording. Waveform shows a subsection of it. This allows the user to focus closely on and work with a section that is just a few seconds long while keeping track of what part of the potentially hours-long recording is playing.
- **Sidebar** contains additional content, such as a speaker list.
- **Transcript** is a list of segments. A segment is a word or a continuous squence of words (like a short phrase, a sentence, or a few sentences) from the recording. It also shows which speaker said it.
- **Groups** area consists of groups and a way to create new groups. Each group assigns some metadata tags to a continuous non-empty subset of the segments.

3.3 Wireframe

During the initial phase of designing, wireframes were created using Figma [4]. Wireframes provide a quick visualization without interactivity. After meticulous design considerations and evaluations as well as consultations with the supervisor, there were some iterations and improvements shown in figures 3.3, 3.4, and 3.5. The focus was mainly on the groups and the whole segment area.



LKKU Tower 10-00-30-4

Playback speed: 1x

0:00:27.5 / 0:00:42.2

Volume:

Name: LKKU Tower 10-00-30-4
Description:
airport: LKKU (Kunovice)
channel: Tower
frequency: UNK
UTC time: 10-00-30-4 001
coordinates: 49.0294444N, 17.43972222E
manual: https://aim.rlp.cz/vfrmanual/actual/lkku_text_cz.html
waypoints nearby: N/A
callsigns nearby: N/A

Display attached resources:
☑ Waypoint-Callsign pairs
☐ Visual resources (images...)

List of waypoints in area of 40km | List of callsigns in -3 hours and +1 hour:
N/A | N/A

LIDSA — ☑ Correct transcript ☐ Correct labeling ☑ Non-english
0:00:00.0
0:00:16.5
Lorem ipsum dolor sit amet consectetur. Accumsan mi sit ut sapien quam id sem commodo. Nulla eget praesent massa lobortis molestie elit platea. A egestas gravida dui dignissim. Maecenas ut ac non non arcu tempor tortor magna.

ATCO tower — ☑ Correct transcript ☐ Correct labeling ☑ Non-english
0:00:17.9
0:00:27.0
Lorem ipsum dolor sit amet consectetur. Accumsan mi sit ut sapien quam id sem commodo. Nulla eget praesent massa lobortis molestie elit platea. A egestas gravida dui dignissim. Maecenas [HESITATION] arcu tempor tortor magna.

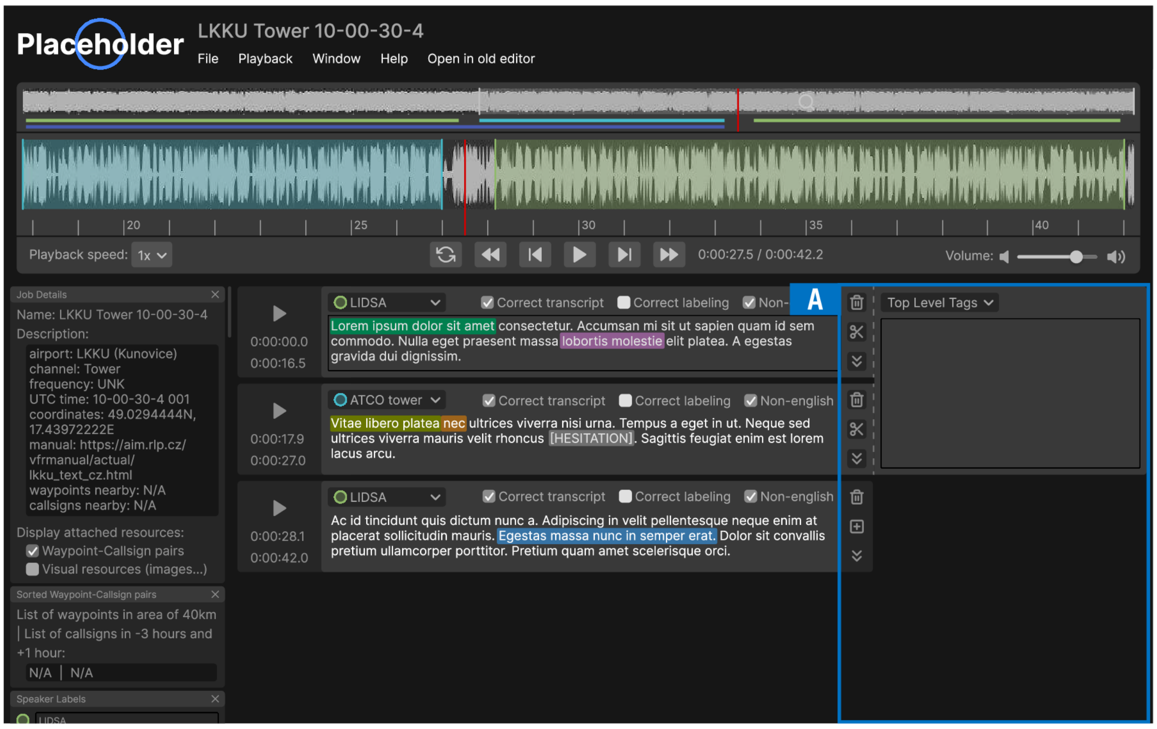
LIDSA — ☑ Correct transcript ☐ Correct labeling ☑ Non-english
0:00:28.1
0:00:42.0
Lorem ipsum dolor sit amet consectetur. Accumsan mi sit ut sapien quam id sem commodo. Nulla eget praesent massa lobortis molestie elit platea. A egestas gravida dui dignissim. Maecenas ut ac non non arcu tempor tortor magna.

Speaker Labels:
- LIDSA
- ATCO tower
- UNK-2
- UNK-3
- Crosstalk

Groups: A
TODO

Figure 3.3: Initial wireframe showing the first proposal of the interface.

[4]https://www.figma.com/



Name: LKKU Tower 10-00-30-4
Description:
airport: LKKU (Kunovice)
channel: Tower
frequency: UNK
UTC time: 10-00-30-4 001
coordinates: 49.0294444N, 17.43972222E
manual: https://aim.rlp.cz/vfrmanual/actual/lkku_text_cz.html
waypoints nearby: N/A
callsigns nearby: N/A

Display attached resources:
✓ Waypoint-Callsign pairs
Visual resources (images...)

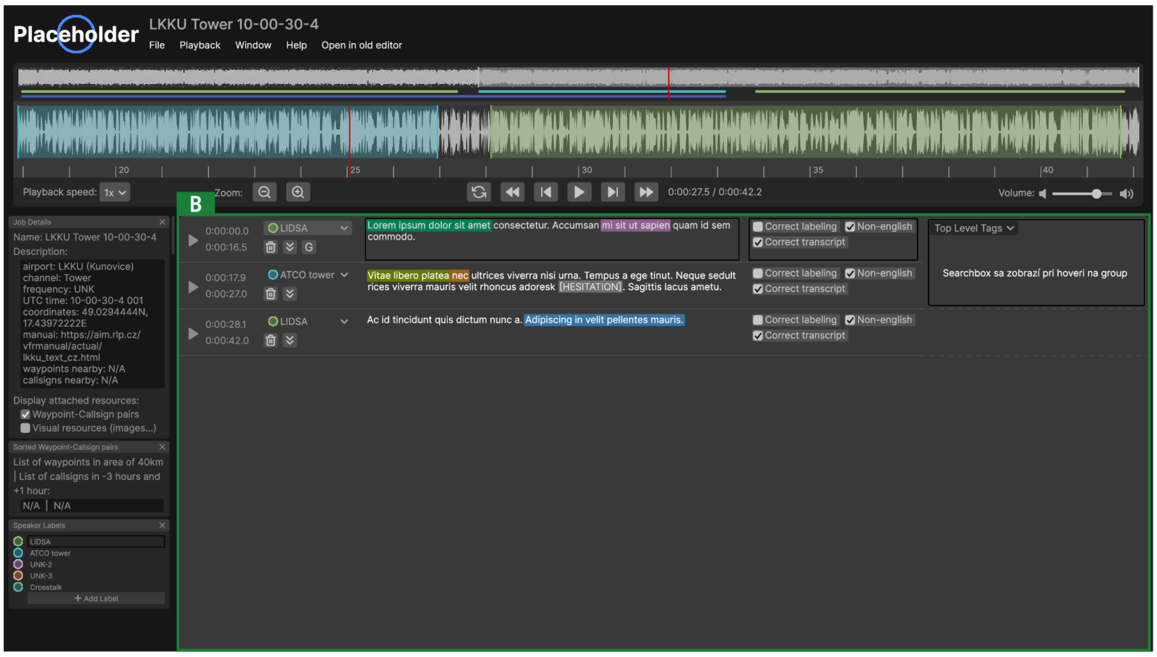
List of waypoints in area of 40km | List of callsigns in -3 hours and +1 hour:
N/A | N/A

0:00:00.0 – 0:00:16.5 — LIDSA — ✓ Correct transcript, Correct labeling, ✓ Non-
Lorem ipsum dolor sit amet consectetur. Accumsan mi sit ut sapien quam id sem commodo. Nulla eget praesent massa lobortis molestie elit platea. A egestas gravida dui dignissim.

0:00:17.9 – 0:00:27.0 — ATCO tower — ✓ Correct transcript, Correct labeling, ✓ Non-english
Vitae libero platea nec ultrices viverra nisi urna. Tempus a eget in ut. Neque sed ultrices viverra mauris velit rhoncus [HESITATION]. Sagittis feugiat enim est lorem lacus arcu.

0:00:28.1 – 0:00:42.0 — LIDSA — ✓ Correct transcript, Correct labeling, ✓ Non-english
Ac id tincidunt quis dictum nunc a. Adipiscing in velit pellentesque neque enim at placerat sollicitudin mauris. Egestas massa nunc in semper erat. Dolor sit convallis pretium ullamcorper porttitor. Pretium quam amet scelerisque orci.

Figure 3.4: The second iteration of the interface design. The main difference is the complete redesign of the groups based on clarified requirements.



Name: LKKU Tower 10-00-30-4
Description:
airport: LKKU (Kunovice)
channel: Tower
frequency: UNK
UTC time: 10-00-30-4 001
coordinates: 49.0294444N, 17.43972222E
manual: https://aim.rlp.cz/vfrmanual/actual/lkku_text_cz.html
waypoints nearby: N/A
callsigns nearby: N/A

Display attached resources:
✓ Waypoint-Callsign pairs
Visual resources (images...)

List of waypoints in area of 40km | List of callsigns in -3 hours and +1 hour:
N/A | N/A

0:00:00.0 – 0:00:16.5 — LIDSA — Correct labeling, ✓ Non-english, ✓ Correct transcript
Lorem ipsum dolor sit amet consectetur. Accumsan mi sit ut sapien quam id sem commodo.

0:00:17.9 – 0:00:27.0 — ATCO tower — Correct labeling, ✓ Non-english, ✓ Correct transcript
Vitae libero platea nec ultrices viverra nisi urna. Tempus a ege tinut. Neque sedult rices viverra mauris velit rhoncus adoresk [HESITATION]. Sagittis lacus ametu.

0:00:28.1 – 0:00:42.0 — LIDSA — Correct labeling, ✓ Non-english, ✓ Correct transcript
Ac id tincidunt quis dictum nunc a. Adipiscing in velit pellentes mauris.

Searchbox sa zobrazí pri hoveri na group

Figure 3.5: The third iteration of the design focuses on tighter segment layout and simplification of visual components.

Chapter 4

Implementation

Various aspects of implementation are described in this chapter. Chosen libraries are detailed as well as their part in the whole system. Then, there is a breakdown of the final interface and some information about the project's structure.

In parallel with implementation, there was testing. One could say the implementation was carried out in cycles, where a feature was developed, a testing round was held, and its outcomes determined the direction of development in the next cycle.

The aim of the implementation is to create a modular and easily expandable app that satisfies the needs of its users and provides a good user experience. All that using the latest approaches, best practices in the field, and future-proof libraries.

4.1 Selection of Libraries

The application was implemented in React because it was one of the requirements. Additionally, it is the most widely used JavaScript framework, it has a large and active user base and numerous libraries.

Approach To Styling

There are three main approaches to styling:

- custom styles
- unstyled components
- component libraries

Custom styles require the developer to do all the work. All interactions, basic functionality, accessibility, and so on. On the other hand, component libraries come with pre-styled set of components with specific designs and interactions. It may be hard to mimic the style and behavior when creating a custom component that is missing; or to modify the behavior to fit project needs.

Unstyled components have become a popular option in recent years and the de facto industry standard for larger projects. They provide the base skeleton and basic functionality, and ensure accessibility, but leave the final look to the developer. This enables the creation of a consistent look even when using multiple libraries, and provides unified behavior for similar interactions.

To conclude, unstyled components were chosen. They provide the most shared features and reduce the work for the developer while maintaining the flexibility in final design and looks. These unstyled libraries were considered:

- headless UI[1]
- Radix[2]
- Reach UI[3]

Ultimately, Reach UI was chosen because of several reasons. It contains only basic building blocks and thus the library is compact. Components can be included separately, which further reduces loading time and data transfers. The library is from the developers behind well-established React libraries such as `react-router` and `remix`.

4.1.1 Core Libraries

- **Redux**[4] with **Redux Toolkit** – application state management.
- **styled components**[5] – custom styling for unstyled components.
- **wavesurfer.js**[6] with plugins – audio playback and visualization.
- **audiowaveform**[7] – audio file preprocessing for visualization
- **Axios**[8] – communication with the API
- **Carbon Colors**[9] – color palette
- **Material Icons**[10] – icons
- **uuid**[11] – unique entity identifier generation

Redux

Redux is a library for complex state management. Redux Toolkit is also used. As the landing page of its website puts it, Redux Toolkit is: *„The official, opinionated, batteries-included toolset for efficient Redux development.“*[12] Simplified syntax, built-in best practices, and a wide selection of utility functions are some of the ways it improves Redux development. It is also the recommended way to use Redux from the Redux documentation[13].

[1] https://headlessui.com/
[2] https://www.radix-ui.com/
[3] https://reach.tech/
[4] https://redux.js.org/
[5] https://styled-components.com/
[6] https://wavesurfer.xyz/
[7] https://github.com/bbc/audiowaveform
[8] https://axios-http.com/
[9] https://carbondesignsystem.com/
[10] https://mui.com/material-ui/material-icons/
[11] https://github.com/uuidjs/uuid
[12] https://redux-toolkit.js.org/, quoted 2024-01-10
[13] https://redux.js.org/introduction/getting-started

In Redux, the data is saved into an immutable store. The store is divided into custom *slices* for easier implementation of selectors and reducers; for example playback slice, segment slice, and settings slice. Selectors return data saved in the store, while reducers create a new and updated state that replaces the old state. Keeping the history of the store snapshots allows for `undo` and `redo` actions.

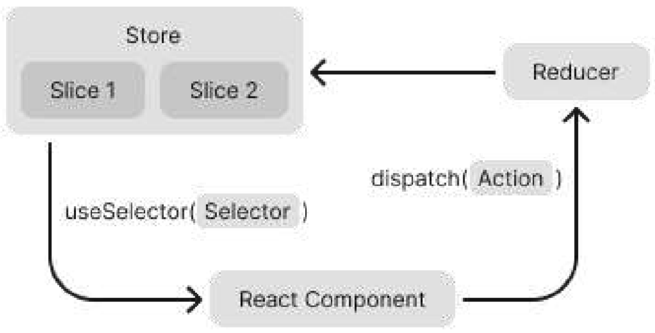


Figure 4.1: Simplified illustration of the Redux workflow and interaction with React components. Data is saved in Redux store, the store is subdivided into slices. React component retrieves data from the store by calling useSelector with the corresponding selector, which is a function that returns data from the store. Based on some trigger, the component dispatches an action and a reducer creates a copy of the data and modifies it. Finally, the data in the Redux store are replaced with the new data, since the store itself is immutable.

Styled-components

Styled-components is a React library that employs the CSS-in-JS approach to styling. It uses tagged template literals to enable writing CSS directly in JavaScript/TypeScript files. The styling of a component can be placed in the same file as the component logic. In the case of more complex components, styling and logic can be located in separate files – the styles get imported.

Associating styles directly with the components enhances the maintainability and readability of the code. This increased modularity also simplifies the process of replacing, rewriting, or extending a component. Additionally, the library enables the use of custom properties and themes.

Wavesurfer.js

Wavesurfer.js is a JavaScript library focusing on audio visualization with several plugins fitting for this project. React wrapper for components from this library is custom, the implementation from Plhal [7], who also used this library, could not be used since it expects only short audio files and extensive modifications would have to be made.

There are several plugins for this library that were used:

- *Regions plugin* implements region and mark visualization on the waveform canvas. The regions are draggable and scalable, and other actions can be triggered on click.
- *Timeline plugin* visualizes time ruler. The timeline aids users with orientation, it shows where the playback cursor is within the audio file.

- *Minimap plugin* also helps with navigation within the waveform visualization, especially when working with long audio files. The plugins can be combined as well, enabling regions on the minimap for example.

Figure 4.2 shows demonstration illustrations of these plugins.

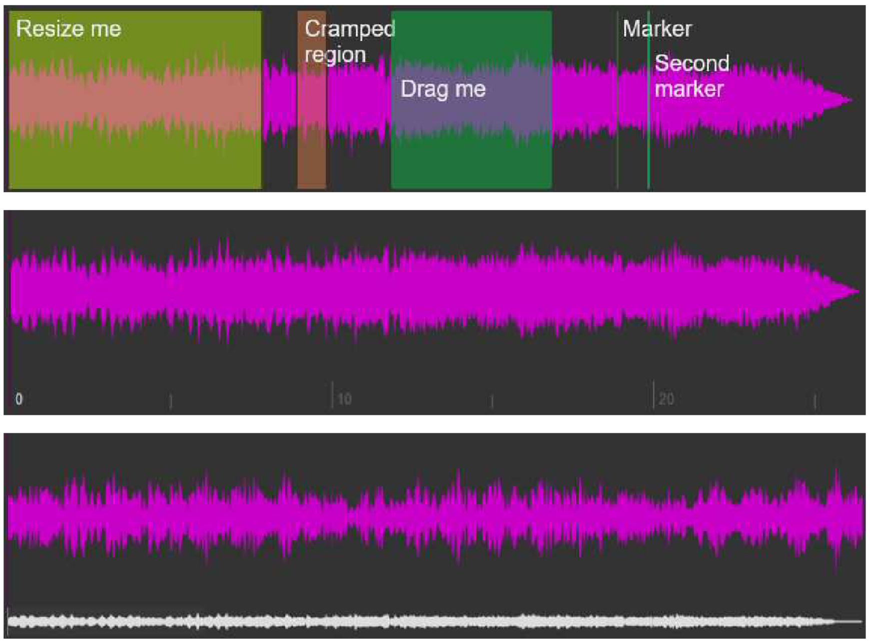


Figure 4.2: Demonstration example of the wavesurfer.js library plugins from the documentation [15]. From top to bottom: *Regions plugin*, *Ruler plugin*, and *Minimap plugin*.

Audiowaveform

Audiowaveform is a C++ application that utilizes quantization to generate data for audio recording visualization. It also supports bit reduction. It is not directly part of the interface; however, it is part of the workflow with audio files. They need to be preprocessed (especially the long ones) to ensure fast loading and a pleasant user experience. If this preprocessing was not done, the visualization would be calculated in a browser and therefore would be substantially slower – several minutes of waiting in case of hour-long files.

Axios

Axios is a JavaScript library for HTTP requests. It builds on the well-supported and robust native `XMLHTTPRequest` and improves it in a way that the developer experience is closer to the native `fetch`; however, less verbose and with superior error handling.

4.1.2 Interoperability of the Libraries

One might ask how does it all work together. Let's illustrate with a general example:

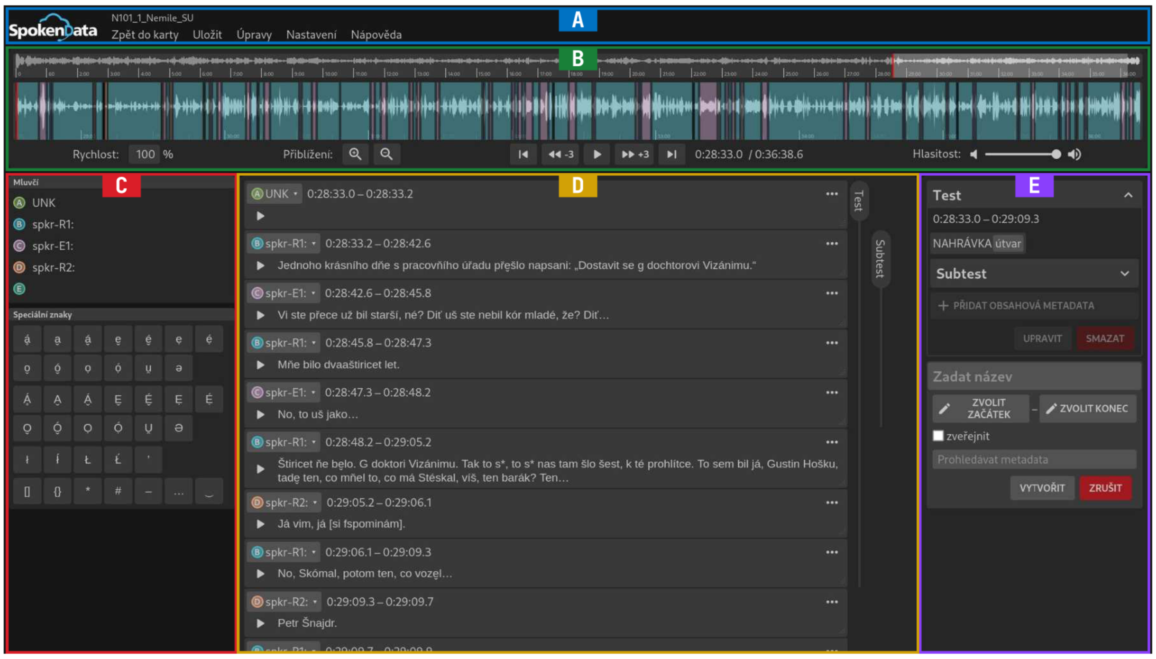
Even before a user opens the interface, a recording is preprocessed by **Audiowaveform**. The whole interface is implemented in **React**. Initially, a request to the API is sent

[15] https://wavesurfer.xyz/examples/, quoted 2024-05-04

using **Axios**, in fact, several requests, loading a job, a transcript including groups, and the preprocessed data for visualization. Job and transcript data are saved into **Redux** store that is implemented using **Redux Toolkit** syntax. During the initial load, each segment and group are assigned an id generated with a utility from **uuid**. The visualization data and audio URL are passed to the **wavesurfer.js**, which creates the waveform as well as the minimap. Some of the components use underlying base components from **Reach UI**. All interface components have CSS applied using **styled components** and colors from **Carbon Colors**. Finally, all icons are provided by **Material Icons** library.

4.2 Breakdown of the Interface

This section describes all components of the interface in greater detail. The implementation follows the general layout introduced in Section 3.2. The final version of the app can be seen in Figure 4.3, where the main areas are highlighted.



UNK 0:28:33.0 – 0:28:33.2

spkr-R1: 0:28:33.2 – 0:28:42.6
Jednoho krásního dňe s pracovňího úřadu přešlo napsani: „Dostavit se g dochtorovi Vizánimu.“

spkr-E1: 0:28:42.6 – 0:28:45.8
Vi ste přece už bil starší, né? Diť uš ste nebil kór mladé, že? Diť…

spkr-R1: 0:28:45.8 – 0:28:47.3
Mňe bilo dvaaštiricet let.

spkr-E1: 0:28:47.3 – 0:28:48.2
No, to uš jako…

spkr-R1: 0:28:48.2 – 0:29:05.2
Štiricet ňe belo. G doktori Vizánimu. Tak to s*, to s* nas tam šlo šest, k té prohlítce. To sem bil já, Gustin Hošku, tadę ten, co mňel to, co má Stéskal, víš, ten barák? Ten…

spkr-R2: 0:29:05.2 – 0:29:06.1
Já vim, já [si fspominám].

spkr-R1: 0:29:06.1 – 0:29:09.3
No, Skómal, potom ten, co vozęl…

spkr-R2: 0:29:09.3 – 0:29:09.7
Petr Šnajdr.

Figure 4.3: A screenshot of the final version of the interface with the five main areas marked. *Top bar* in A, *player* in B, *sidebar* in C, *transcript* in D, and *groups* in E.

4.2.1 The Top Bar

Containing the logo and the title, it serves a vital information role. Moreover, the main menu provides access to common workspace actions as well as related resources. Figure 4.4 shows the decomposition of the top bar.

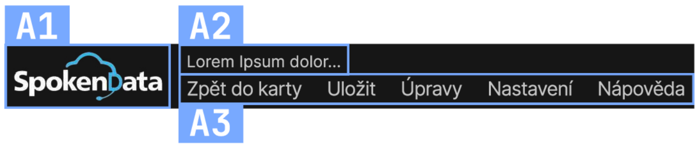


Figure 4.4: The decomposition of the top bar. The *logo* is in `A1`, the *title* in `A2`, and the *main menu* in `A3`.

The Logo and The Title

They help the user orient themselves. As described in Section 2.1, users subconsciously start scanning a website from the upper-left corner. The logo indicates on what site they are and the title informs them which recording is loaded.

The Main Menu

It consists of a list of clickable buttons, each of which initiates an action or opens a submenu. The button labels are in Czech, the translations and expected actions are:

- **Zpět do karty** (Eng. „back to the catalog“) redirects back to the catalog of recordings.
- **Uložit** (Eng. „save“) saves the changes and makes a `PUT` request to the API.
- **Úpravy** (Eng. „edit“) opens a submenu with:
  - **Krok zpět** (Eng. „undo“),
  - **Krok napřed** (Eng. „redo“).
- **Nastavení** (Eng. „settings“) opens a modal window with settings.
- **Nápověda** (Eng. „help“) opens a submenu with:
  - **Klávesové zkratky** (Eng. „keyboard shortcuts“) that open a modal window with the list of available keyboard shortcuts,
  - **Videomanuál** (Eng. „video manual“).

4.2.2 The Player

The player is arguably the most used area in the whole interface. It presents an audio file in a comprehensible way and provides controls to interact with it. The breakdown of the area into components is shown in Figure 4.5.

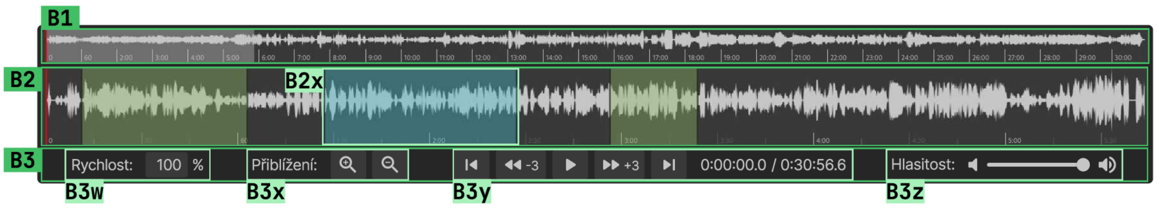


Figure 4.5: Decomposition of the player. The *minimap* is in `B1`. The *waveform* in `B2` and it contains a *region* in `B2x`. *Controls* in `B3` comprise the *speed control* in `B3w`, the *zoom control* in `B3x`, the *playback controls* in `B3y`, and the *volume control* in `B3z`.

Minimap

The minimap shows a whole recording and thus offers an overview. The portion currently displayed in the waveform is highlighted. There is a red cursor that is synchronized with the waveform cursor. It shows the current position in an audio track. The minimap is clickable. A click jumps to the target time; if outside of the waveform view, the view jumps to the waveform cursor.

There is a known bug when zoomed in very close and the recording is rather longer. The highlighted area disappears. Unfortunately, this is an error in the wavesurfer.js library. However, this is not an issue, as, at this level of zoom, the highlighted area would be so thin that it would only appear in the immediate proximity of the cursor. Therefore, the cursor itself suffices for orientation in this case.

Waveform & Regions

The waveform displays a portion of the audio enabling the user to focus on a shorter part in detail. As in minimap, there is a cursor and it also seeks on click. When playing, the cursor stays in view. The waveform is scrollable, on hover a scroll bar appears.

There are marked sections in the waveform called *regions*. A new region gets created on click and drag on the waveform. The drag determines its start and end. The regions are the visual representations of segments from the transcript. When a new region is created a corresponding segment is created as well. On both sides of the region, there are handles that are used to resize the region. Region color is determined by the speaker of the matching segment. A click into a region scrolls the transcript to the associated segment.

Controls

The four controls provide various means of interacting with the recording. Most controls have a text label accompanying them. The controls are:

- *Speed control* modulates the playback pace. It is given in per cent. The range is from 1 to 100, speeding up the playback is not needed in user workflow. Furthermore, the value is loaded from an input. It can be inserted precisely or adjusted with keyboard arrows when the input is focused.
- *Zoom control* comprises two buttons to zoom in and out. Zoom steps are exponential because it feels more natural. The zoom of the visualization is expressed in pixels per second. There is a default level, an upper, and a lower limit.
- *Playback controls* includes a pair of jump buttons, a pair of skip buttons, a play-pause button. Instead of a label, current time and duration are displayed. The jump button seeks to the closest region's start. Skip buttons skip by custom time, three seconds by default.
- A slider serves as the *volume control*.

4.2.3 The Sidebar

The sidebar serves as the place for auxiliary components. Its placement right next to the transcript makes it an ideal location for the list of speakers and the set of special characters.

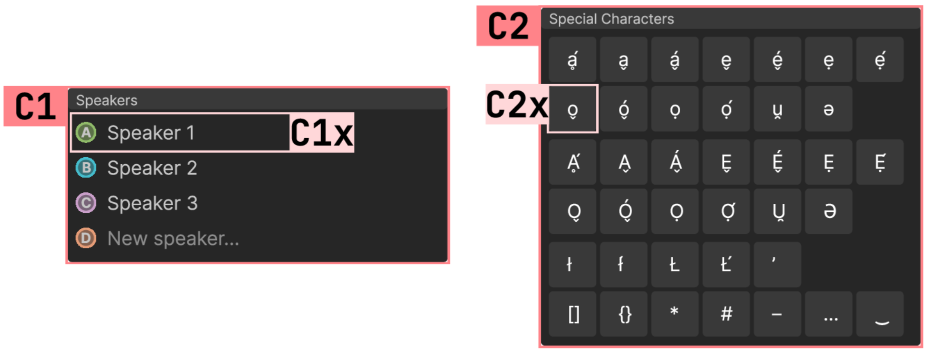


Figure 4.6: Decomposition of the sidebar. There are two cards, the list of *speakers* in `C1` with an individual *speaker item* highlighted in `C1x`, and the set of *special characters* in `C2` with an individual *special character button* in `C2x`.

Speakers

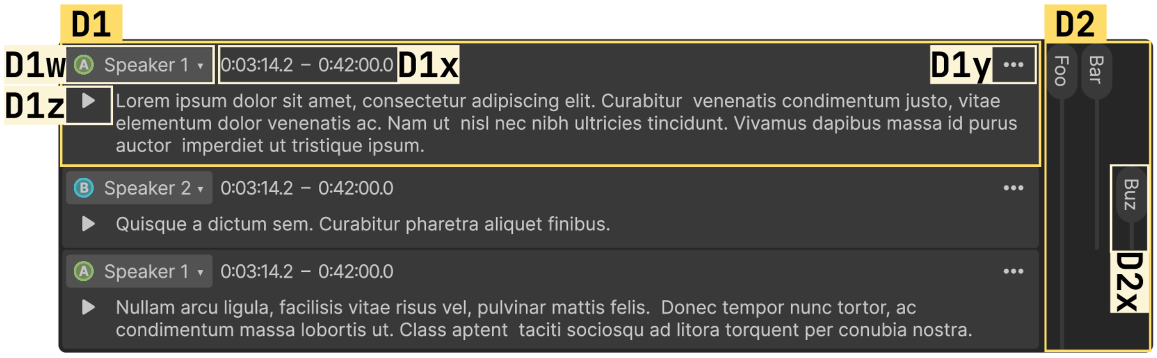
There is a simple list of all the speakers appearing in a recording. Each speaker label can be edited. As the last item on the list, there is an empty speaker. When a label is given, the speaker is created. A speaker gets deleted when its label is removed. Speaker color is assigned automatically.

Special Characters

The set of special characters is fixed. They are displayed as buttons in a grid arrangement. On click, the character gets inserted at the cursor (or replaces a text selection) in the text area of the currently focused segment.

4.2.4 The Transcript

The central part of the interface handles the text form of the recorded interview. It displays the conversation transcript which is split into segments. Its structure is detailed in Figure 4.5.



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Curabitur venenatis condimentum justo, vitae elementum dolor venenatis ac. Nam ut nisl nec nibh ultricies tincidunt. Vivamus dapibus massa id purus auctor imperdiet ut tristique ipsum.

Quisque a dictum sem. Curabitur pharetra aliquet finibus.

Nullam arcu ligula, facilisis vitae risus vel, pulvinar mattis felis. Donec tempor nunc tortor, ac condimentum massa lobortis ut. Class aptent taciti sociosqu ad litora torquent per conubia nostra.

Figure 4.7: Decomposition of the transcript. There is the list of segments, one of which is marked `D1` and *group visualization* that is in `D2` containing a *mark* in `D2x`. A *segment* consists of a *speaker selection* highlighted in `D1w`, a *time range* in `D1x`, a *segment actions* menu in `D1y`, a *segment play button* in `D1z`, and a *segment text*.

Segment

A segment is an organizational unit of transcript containing its individual text fragments. Speaker selection, time range, segment actions, segment play button, and segment text are the components that constitute a segment. More detailed description of the components:

- *Speaker selection* facilitates a way to select from the list of speakers. A segment has exactly one speaker. Click on the speaker selection opens a menu with other available speakers. Speaker change is also reflected in the waveform region, as its color is derived from the associated segment speaker. A click on an empty part of a segment seeks in the audio to the segment's start time.
- *Time range* displays the start time and the end time of a segment. It can be updated by resizing the corresponding region on the waveform.
- A horizontal dots menu hides the *segment actions*, which reduces the visual clutter. It also makes the actions easily expandable. On click, the menu opens, revealing two actions: delete and merge down. Delete removes the segment from the transcript altogether, while merge down prepends the segment's text to the following one and extends its start time.
- *Segment play button* simply plays the segment from the start to its end.
- *Segment text* contains a fragment of transcript of the recording. Special characters can be added with the special character buttons. The text area is resizable.

Group Visualization

Group visualization is located in between the list of segments and the groups which signifies that it is related to both. It aims to visualize group membership of segments with vertical marks. A *mark* consists of a head and a tail. The head contains a group title, or its part when a title is too long, or it is empty if there is no title. A tail extends a header to the desired length if needed.

4.2.5 The Groups

Looking back, there is a way to play a recording and a place to transcribe it, but one core thing is still missing. Users need a way to categorize specific sections of a recording by discussed topic. This is what groups enable. The breakdown of the groups area is presented in Figure 4.8 and all component names are introduced in its caption.

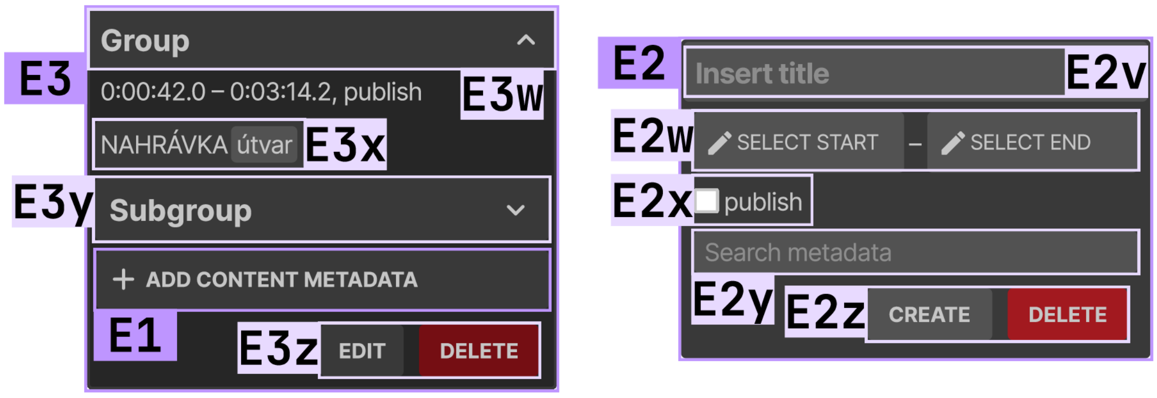


Figure 4.8: Decomposition of the groups area. Three types of components can appear here. An *add content metadata button* is highlighted in `E1`. A *group form* in `E2` consists of a *title input* marked `E2v`, a *start and end segments selection buttons* in `E2w`, a *publish checkbox* in `E2x`, a *metadata seachbox* in `E2y`, and *group form actions* in `E2z`. Its sibling component is the *group* highlighted in `E3`. Groups can be collapsed or expanded. They can be nested too. `E3w` denotes *group header* containing a *group title*. `E3x` highlights a *metadata tag*. `E3y` contains a collapsed *subgroup*. Finally, *group actions* are in `E3z`.

Add Content Metadata Button

The button uses language the future users already know to trigger an action that is new for them. A click shows a group form in its place. When no groups are created yet, an add content metadata button is the only thing displayed in the groups area. There is one top-level button and one button in each group to create its subgroups.

Group

The purpose of a group is to associate a set of metadata tags with a portion of a recording. Metadata are a list of trees of labels that have a common theme. A tag is a root label, or a path in a tree to a nested label, or a combintaion of paths to multiple nested labels. Each group contains one or more tags.

A group presents the tags as well as related information such as a start and an end time that the tags apply to. It also provides several actions, e.i. edit group, delete group, and add a subgroup. A list of nested groups is displayed within the parent group. More than three levels of nesting are rarely expected.

Groups are collapsible, just a group header stays visible when collapsed. This reduces clutter in the list of groups when dozens of groups exist. It also improves orientation when there are more subgroups in a group.

Group Form

A group form provides fields for all items in a group besides subgroups. Description of the parts:

- Similarly to the title in a group, there is the *title input* in a group form.
- *Start and end segments selection buttons* are used to select the start and end time range of a gorup. On click, the selection process is initiated. The segments are selected

from the segment list. A selected segment is outlined and lightened. Also, segments in between the selected start and end segments are lightened. When selecting segments of a subgroup, only segments in the time range of the parent group are displayed to select from.

- A simple but effective *publish checkbox* to mark whether a portion of a recording the group corresponds to should be allowed for publishing or not.
- On focus of *metadata searchbox* input, a menu of metadata labels opens. The list of labels is scrollable and can be narrowed down with a search term. Nested labels are indented. On selection, a tag is added.
- Two buttons comprise *group form actions*. There is a create and a delete button.

Furthermore, group form is used for the creation as well as editing of groups. The only difference is that when editing a group, the existing values are loaded from it and can be changed or removed.

4.3 Project Folder Structure

React does not have a predetermied project structure. According to the documentation: *„React doesn't have opinions on how you put files into folders.“* [16] Many patterns to organize files have emerged.

The one used falls into the family of feature-centric approaches. It aims to be minimalistic, provide high clarity, have deterministic file placement, be easily expandable, and group related files together based on app features.

The root of the project contains the following items:

- `assets` folder that is intended to store static assets such as images.
- `src` folder with the source files for all components.
- `index.html`
- `README.md`
- Several other configuration and auxiliary files.

The `src` folder is further organized into subdirectories. Subdirectory name determines its purpose. There are several types:

- `components`
- `features`
- `redux`
- `style`
- `types`
- `utils`

[16] https://legacy.reactjs.org/docs/faq-structure.html, quoted 2024-05-03

Folders `components`, `style`, `types`, and `utils` contain files implementing app-wide functionality. Redux setup and the store setup are implemented in `redux`. Folder `features` contains subfolders implementing app features based on Section 3.1. There are four feature subfolders, e.i. `grouping`, `player`, `transcript`, and `workspace`.

Apart from these folders, there are also three files. `App.tsx` and `main.tsx` are standard React files serving as entry points for the application. `app.config.ts` stores an API key that is loaded when running the interface locally. The config file is in `.gitignore` to prevent an accidental key leak.

Each component is implemented in a separate file, which makes it easily replaceable. Related files are kept close together. Files are also kept rather short improving the redability of the code.

Feature Folders Structure

The general structure is similar to `src` folder structure and shares most folder types. Each feature folder contains a subset of these directories:

- `components`
- `config`
- `hooks`
- `redux`
- `types`
- `utils`

Directories with names `components`, `types`, and `utils` are the same as `src` folder subdirectories but implement feature-wide functionality. Configuration objects go into `config`, files with functionality extracted to custom hooks belong into `hooks`, and redux slice into `redux`.

Chapter 5

Testing

There comes a point when all the design knowledge is not sufficient. The only way to explore how would a developed interface be used is testing. In general, there are two types of testing, e.i. qualitative and quantitative. Qualitative testing does not require a large number of participants and can discover most bottlenecks and errors. It can also delve deeper into certain areas and explore the reasoning of the participants. These are also the reasons it was chosen.

5.1 Testing Process Design

The testing process is based on the methodology and recommendations from the book *Rocket Surgery Made Easy* by Steve Krug [3] and Chapter 9 from his book *Don't Make Me Think* [4].

The first testing can start with the assumption that there is a lot of undiscovered errors. Most of the severe mistakes are usually discovered by any test participant. In this phase, it is not necessary to test with real future users.

When an error is discovered, the participant can be helped and move on. When the user is confused, they should be given a while to figure things out on their own; however, if they take too long, it indicates that there is something confusing or unclear. After a bug is found, the tester can skip to the next part, it is not necessary to finish every task.

Before testing the participants should be instructed to say their thoughts aloud and to point out any confusion or if something is unclear. The participants should be reminded that they are not tested – the developed interface is. They should not be afraid not to know something – that is just what the testing expects, and what allows the developer to fix the discovered errors before real use of the system.

If there are multiple participants, they are never present while their predecessors execute the tasks. This is done to avoid any possible influence among the participants, as they might be inclined to repeat the points they had heard.

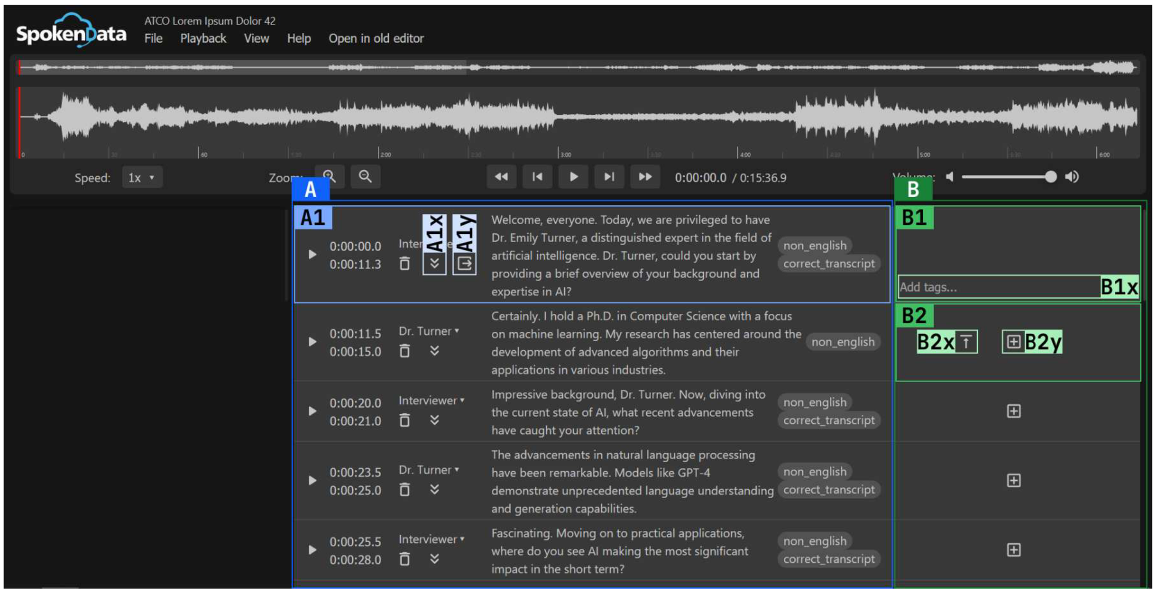
5.2 The Testing Sessions in General

There were four testing sessions overall. All of the sessions took place at the Czech Language Institute of the Czech Academy of Sciences. The testing started during the winter semester. The first session was held roughly two months into the drafting and development. Testing continued and intensified during the summer semester. The last one took place about a week before the thesis deadline.

The composition of the sessions was always the same. After a brief preparation, it started with individual tests where the participants executed predetermined tasks. They were followed by a discussion. All participants and others present at the testing (e.g. thesis supervisor or other employees) took part. Testing tasks and discovered issues as well as feature proposals and possible improvements were discussed. In the days after the session, notes and recordings were analyzed, and the outcomes were evaluated to determine the priorities for the future direction of development.

5.3 The First Round – Whole Interface and Groups

Since this was the first testing round, it focused on the whole interface and the main feature – groups, where metadata are added. A screenshot of the still-in-development interface can be seen in Figure 5.1. Also present were Igor Szőke (the supervisor of this thesis), Jozef Žižka (FIT BUT researcher, SpokenData co-founder), and Hauryliuk Matsvei (a student with the same thesis assignment).



0:00:00.0 / 0:15:36.9

0:00:00.0 0:00:11.3	Interviewer	Welcome, everyone. Today, we are privileged to have Dr. Emily Turner, a distinguished expert in the field of artificial intelligence. Dr. Turner, could you start by providing a brief overview of your background and expertise in AI? — non_english, correct_transcript
0:00:11.5 0:00:15.0	Dr. Turner	Certainly. I hold a Ph.D. in Computer Science with a focus on machine learning. My research has centered around the development of advanced algorithms and their applications in various industries. — non_english
0:00:20.0 0:00:21.0	Interviewer	Impressive background, Dr. Turner. Now, diving into the current state of AI, what recent advancements have caught your attention? — non_english, correct_transcript
0:00:23.5 0:00:25.0	Dr. Turner	The advancements in natural language processing have been remarkable. Models like GPT-4 demonstrate unprecedented language understanding and generation capabilities. — non_english, correct_transcript
0:00:25.5 0:00:28.0	Interviewer	Fascinating. Moving on to practical applications, where do you see AI making the most significant impact in the short term? — non_english, correct_transcript

Figure 5.1: The developed interface as it looked at the time of the first testing session. Some parts were fully functional, some were just mockups, and the development of some had not started yet. Areas or components mentioned during the tests are marked. Area `A` contains segments and area `B` contains groups. `A1` highlights a /textitsegment. Inside the segment, there is `A1x`, a „Merge down“ button, and `A1y`, a „Detach from group“ button. The area `B` consists of a *group* in `B1` and options related to groups for each segment, one of whitch is marked in `B2`. A group contains a way to search and add segments in `B1x`. In `B2`, there are two options – in `B2x`, there is a button to attach the corresponding segment to the group above, and in `B2y` is a button to create a new group.

The Tasks

The testing consisted of three tasks focusing on the groups feature. The tasks increase in difficulty.

1. Create a group from the first four segments.
2. Add these tags to the group: *Praha, knedlíky, řemesla.*
3. You found out that the third and fourth segments did not belong to the group with the first two. They should be in a separate group with tags: *Praha, luční rostliny.*

These tasks were constructed in a way to include all group actions. They test the whole workflow from group creation, through adding tags, to subsequent group modification.

Individual Tests

There were three participants who volunteered to partake in the testing, all women. All the participants had not seen the interface before testing. There was a time limit – roughly 5 minutes per participant. Every participant agreed beforehand to screen and audio being recorded during the session. Additionally, the introduction and instructions were written and read to ensure consistency.

Participant #1

The tester immediately recognized segments and understood that she should find a way to group them. She seemed a bit confused and tried to use a „Merge down“ button, but immediately after that found the button to create a new group. However, then she proceeded to delete the group by clicking the „Detach from the group button“ that appeared. After my nudge, she created the group again and then found a way to add the tags.

She proceeded to add the tags. After typing „Praha“ into the group selection input, there was an empty list because it is in a subcategory that would have to be chosen before, she commented that she did not remember to which category every tag belonged. She also remarked that the tags should be in alphabetical order, which they were not.

The third task cannot be completed without having finished the first one. The tester proceeds to delete the group with the tags again, as she tries to add other segments into the group. She remarks that she was looking for „an arrow to add [a segment to the group]“ – which in fact is in the second segment, right under the group; however, she either overlooked it or did not associate the icon with the function. She played with the system for about another minute and did not find the solution. I showed her how to do it so she could proceed to the third task.

At this point, she starts to explain that she misunderstood the wording of the first task. She thought that she was supposed to merge the first four segments. Explaining further, they use the term „obsahová metadata“ (eng. „content metadata“) instead of „tagy“ (eng. „tags“) and that was what confused her.

Returning back to the third task, she did not understand that she needed to create a different group for the second set of tags. After exploring the interface for another 30 seconds, she did not find a way to complete the task. „I think this will never happen to us,“ she added on the third task. At that point, time was up and the second participant was ready.

Participant #2

In the beginning, the participant did not understand what segments were. After a nudge, she proceeded to successfully create a group; in fact, four groups – one for each of the first four segments. She then said her thoughts aloud and explained that she was stuck, did not know what to do next, and did not know what was meant by „skupina“ (eng. group). Afterward, she discovered how to delete a group. After about another 20 seconds of stagnation, I showed her how to do it (the error had already been discovered) so she could proceed to the next task.

I remark that tags are metadata – this issue was already discovered. The participant clicked the input in the group but proceeded to be confused about how to add the tags, even though the list of categories showed up. The list was partially off the screen and she noted that it was not fully visible. After almost a minute of trial and error, I proceeded to show her so she could continue to the next task.

After reading the task, she did not know what was expected. Then I explained that it was a theoretical scenario, she understood but kept acting confused and could not figure out the next step. Shortly after that; however, the time ran out.

Participant #3

From the beginning, the participant looked uneasy. After reading the first task, she commented that she did not know what she should do. She apologized that the system was totally new to her and that she did not know what to do. Having hovered over one of the buttons, a label appeared to which she reacted that she did not speak any English. She did not seem willing to continue so I ended the session.

Group Discussion

After the testing finished, there was a group discussion about the presented interfaces, expectations, clarifications, change suggestions, and the errors discovered.

Firstly, we identified that one of the major issues was that we used different terminology. They were not familiar with *segment*, *group*, or *tag*, but used *metadata*. The terminology was clarified.

Secondly, they explained that their current workflow consisted of adding text notes to the transcript in Microsoft Word and listening to the audio using Audacity. The proposed interface seemed too complicated and they proposed it be simplified.

Moreover, we discussed expected usage. They do not expect to use a lot of tags for any of the groups, mostly just two. Usually, there are two or three speakers in a recording, the maximum is nine. Also, there is a pre-selection of tags that are highly likely to be used and they should be displayed first when selecting. There needs to be a group-level option to publish or not, but a segment-level option is not needed. The most used action is segment replay, and slowing audio playback is also very common.

Finally, crosstalk was discussed. Various strategies to denote it, or whether to ignore it completely.

Analysis and Outcomes

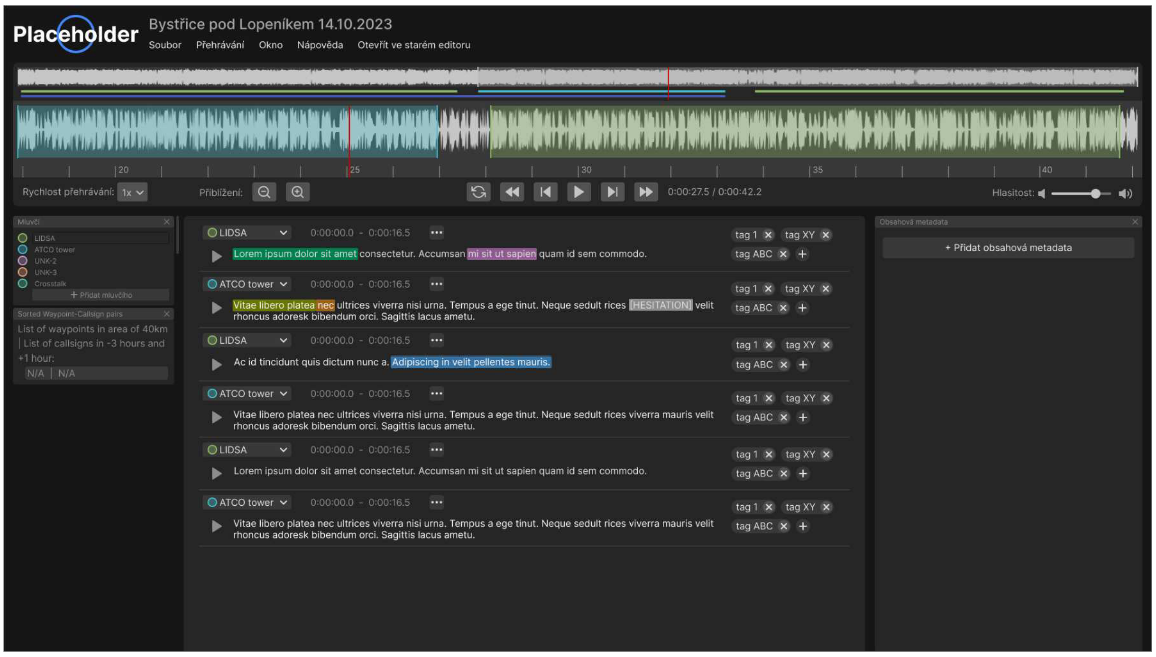
None of the three participants were able to complete all three tasks. Many minor flaws and major deficiencies were discovered. These are the main points that need to be addressed:

- **Grouping.** The proposed solution proved to be too complex, the test participants struggled to understand it and work with it.
- **Language.** A Czech language version is needed, also opening the possibility of adding other language versions.
- **Tree selection.** Whenever using an input to search a tree of options, search through all the layers of the tree instead of selecting category and subcategories.

Overall, this round of testing discovered major flaws and provided valuable experience in this aspect. Groups feature will need to be simplified and reworked.

5.4 The Second Round – Groups and Group Visualization

As suspected and confirmed in the previous round of testing, segment grouping and the addition of metadata tags was the hardest part of the interface to design and implement. The second round of testing focused mainly on user expectations about the whole process of group creation and group visual representation. One of the aims of this round was also to choose the group visualization approach. This round focused more on exploring user expectations and preferences.



List of waypoints in area of 40km | List of callsigns in -3 hours and +1 hour:

LIDSA 0:00:00.0 - 0:00:16.5
Lorem ipsum dolor sit amet consectetur. Accumsan mi sit ut sapien quam id sem commodo.

ATCO tower 0:00:00.0 - 0:00:16.5
Vitae libero platea nec ultrices viverra nisi urna. Tempus a ege tinut. Neque sedult rices [HESITATION] velit rhoncus adoresk bibendum orci. Sagittis lacus ametu.

LIDSA 0:00:00.0 - 0:00:16.5
Ac id tincidunt quis dictum nunc a. Adipiscing in velit pellentes mauris.

ATCO tower 0:00:00.0 - 0:00:16.5
Vitae libero platea nec ultrices viverra nisi urna. Tempus a ege tinut. Neque sedult rices viverra mauris velit rhoncus adoresk bibendum orci. Sagittis lacus ametu.

LIDSA 0:00:00.0 - 0:00:16.5
Lorem ipsum dolor sit amet consectetur. Accumsan mi sit ut sapien quam id sem commodo.

ATCO tower 0:00:00.0 - 0:00:16.5
Vitae libero platea nec ultrices viverra nisi urna. Tempus a ege tinut. Neque sedult rices viverra mauris velit rhoncus adoresk bibendum orci. Sagittis lacus ametu.

Figure 5.2: Updated mockup of the system shown to the participants during the second round of testing. There is a new preview of the groups feature.

The Task

The testing consisted of four questions. They consist of taking an action, imagining the reaction, giving feedback, and choosing a variant of the visualization. List of the questions and relevant context:

1. How do you add metadata?

    - The correct answer is to click the „Přidat obsahová metadata“ button (Eng.: „add content metadata“).

2. What would you expect would happen after the click?
3. What if a modal window with inputs for data, including input to search the metadata, opened? What do you think about this proposal?
4. Which one of the group visualizations do you like the most?
    - Participants were shown the proposed alternatives (see Figure 5.3).

Individual Tests

Five volunteers including men and women agreed to take part in the testing. The testers also covered more age groups than in the previous round. Some of them participated in the first round and others were new and had not seen the interface before. There was no time limit per participant. Recording of this session was deemed unnecessary, and hand-written notes proved sufficient.

Participant #1

The participant immediately recognized the button to add metadata and said that she would click it. She imagined that a metadata tag menu would open. One would be able to choose applicable tags. She also imagined that one would be able to select start and end segments, thus defining a subset of segments the metadata tags would apply to.

She commented on the proposal in the next question, a modal window with input fields appearing, that manually filling the start and end times would be a nuisance and a waste of time. She liked her idea of selecting start and end segments more.

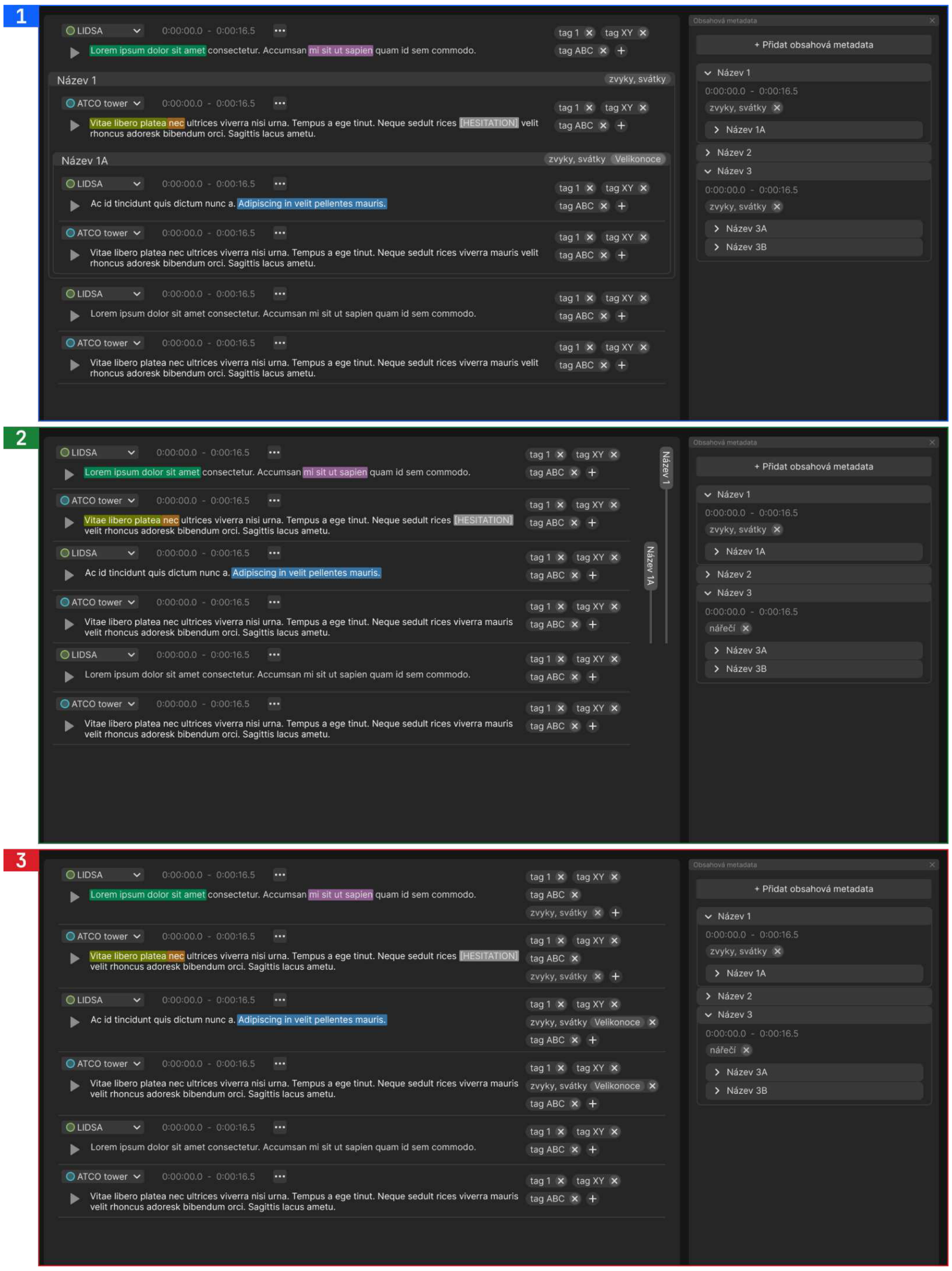
Regarding the group visualization proposals, the participant disliked option 1 and found both options 2 and 3 good.

Additionally, she suggested that filtering segments by tags might be a good idea to implement.

Participant #2

The button to add metadata tags was quickly located. She would expect an input to appear. One would be able to select from a list or search for the tags to apply. She considered a modal window opening viable.

Having explored the visualization proposals, she concluded that option 2 was the best and did not see much difference between the other two options.



Lorem ipsum dolor sit amet consectetur. Accumsan mi sit ut sapien quam id sem commodo.

Vitae libero platea nec ultrices viverra nisi urna. Tempus a ege tinut. Neque sedult rices [HESITATION] velit rhoncus adoresk bibendum orci. Sagittis lacus ametu.

Ac id tincidunt quis dictum nunc a. Adipiscing in velit pellentes mauris.

Vitae libero platea nec ultrices viverra nisi urna. Tempus a ege tinut. Neque sedult rices viverra mauris velit rhoncus adoresk bibendum orci. Sagittis lacus ametu.

Lorem ipsum dolor sit amet consectetur. Accumsan mi sit ut sapien quam id sem commodo.

Vitae libero platea nec ultrices viverra nisi urna. Tempus a ege tinut. Neque sedult rices viverra mauris velit rhoncus adoresk bibendum orci. Sagittis lacus ametu.

Figure 5.3: The three proposed versions of group visualization. In the proposal 1, the groups are represented by visual nesting. Individual segments are placed inside frames that represent the corresponding groups and subgroups. Variant 2 uses vertical markers on the right side of the segments to show group membership. Proposal 3 lists segment tags and group tags together presenting a minimalistic approach.

Participant #3

The participant also immediately recognized the button to add metadata. A searchbox or a menu of metadata tags should appear, and he speculated segments could be selected one by one. He preferred inputting start and end times to selecting segments the way he conceived. The best way to visualize groups was option 2, followed by proposal 1. Option 3 was considered bad.

Participant #4

This participant was the only one who missed the button at first glance. However, she found it after a short search through the interface. She would expect a field to input and search metadata tags. „I would like that a lot,“ was her reaction after the question about the modal window.

She explained that the users would probably know the start and end times before creating a group. Therefore, she suggested that segments should be selected first and a group would be created afterward.

After the possible group visualizations were shown, she expressed that option 2 was the best, in her opinion, and did not comment on the other two.

Participant #5

Similarly to most of the previous participants, the button was spotted without hesitation. Clicking it should open a modal window with a way to select group tags. Since her idea was almost the same as the one in the next question, the question was skipped.

Contrary to others, she liked option 2 the least because the text was rotated. She explained it would be harder to read for her that way. Option 1 was the most clear one, in her opinion, while option 3 was too visually crowded.

Group Discussion

Someone from the tester praised the button text, saying it was easy to understand and find when they wanted to add metadata tags.

Group visualization option 2 was chosen as the one to implement since everyone except the last participant found it the best.

Furthermore, the group modal window was discussed. We concluded that the modal is not ideal. A component with the form to input group data should be implemented instead. It should be located on the right, where the button to add metadata is.

Next, all agreed that selecting the start and end segments for a group is superior to manually typing the start and end times. The segments between the selected start and end segments should automatically be part of the group, and all tags in the group should apply to them. The start time of a group would be the start time of the start segment and, similarly, the end group time would be the end time of the selected group end segment. Additionally, there should be the option to add a group title. The title would be rather short, just a word or a short phrase. Other necessary form fields were discussed.

At the time of this round of testing, in the in-development system, there was still an option to add segment tags. Those were tags that would only apply to a single segment and were not related to group tags. This feature was deemed unnecessary since only group tags are needed.

Analysis and Outcomes

- The **group visualization** proposal to implement was chosen. It is the option 2 from the proposals in Figure 5.3.
- **Group component** was flashed out. The group components will be located on the right side of the interface. There should also be some form, which will include an input for the group title, a way to select start and end segments, a way to search in the list of metadata tags, and a checkbox specifying whether the group could be published.
- **Removal of segment tags.** There would be no use for them in the end.

To conclude, during this round of testing, the group visualization approach was successfully chosen. Contents of a group itself were further specified as well. This information was used to create a working prototype of the grouping feature, which was assessed in the third round of tests.

5.5 The Third Round – Groups and the Playback Area

A working draft of groups had been developed, this round of tests focused primarily on groups, group creation, and group data input. Various other parts of the interface including the minimap and segment actions were explored as well. All areas and parts of the interface mentioned in this subsection are denoted in Figure 5.4 and their names are listed in its caption.

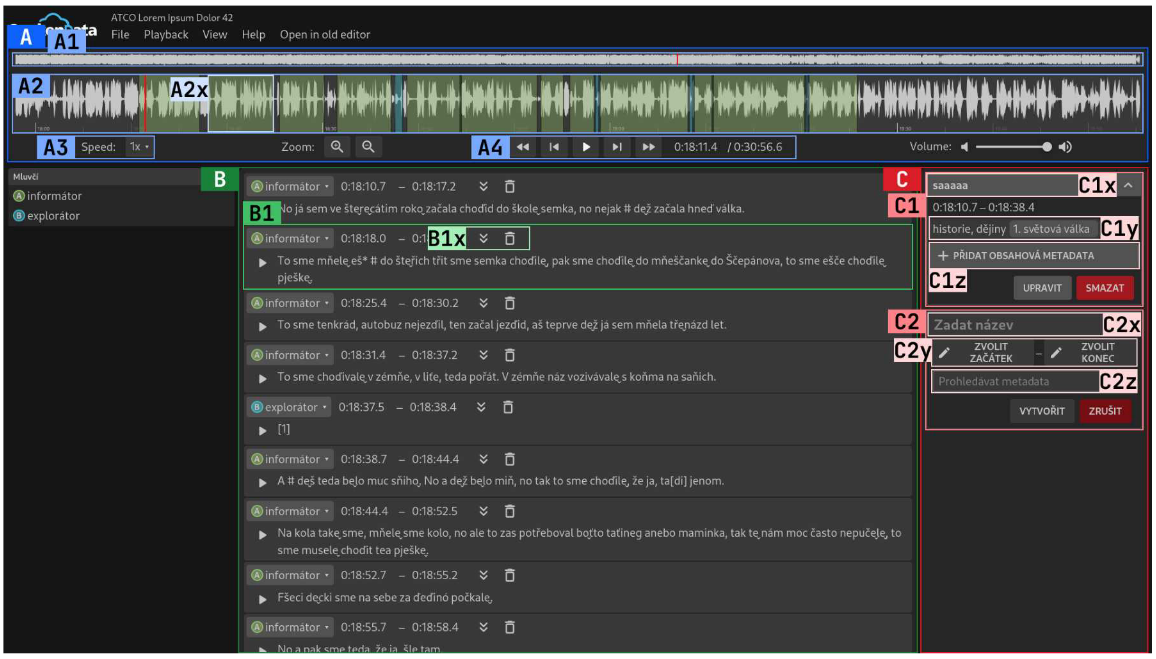
The Task

The task consists of two parts. The first step consists of creating a group and a subgroup. The second one is unrelated to the first task and can be completed independently.

1. Create a group with a subgroup. The group should contain the first several segments and the subgroup some subset of the segments. Both group and subgroup should have some metadata tags added.
2. Play freely with the audio player and describe your thoughts.

Individual Tests

There were four testers who all had taken part in some of the previous rounds of testing. As hand-written notes proved sufficient before, this session was not recorded. Although there was no hard time limit per participant, the session lengths were kept approximately the same.



informátor 0:18:10.7 – 0:18:17.2
No já sem ve štereցátim rokoˌzačala choďid do školeˌsemka, no nejak # dež začala hneď válka.

informátor 0:18:18.0 – 0:1…
To sme mňeleˌeš* # do šteřich třit sme semka choďile, pak sme choďileˌdo mňeščankeˌdo Ščepánova, to sme ešče choďile, pješke,

informátor 0:18:25.4 – 0:18:30.2
To sme tenkrád, autobuz nejezďil, ten začal jezďid, aš teprve dež já sem mňela třenázd let.

informátor 0:18:31.4 – 0:18:37.2
To sme choďivaleˌv zémňe, v liťe, teda pořát. V zémňe náz vozivávaleˌs koňma na saňich.

explorátor 0:18:37.5 – 0:18:38.4
[1]

informátor 0:18:38.7 – 0:18:44.4
A # deš teda belo muc sňiho, No a dež belo miň, no tak to sme choďile, že ja, ta[di] jenom.

informátor 0:18:44.4 – 0:18:52.5
Na kola takeˌsme, mňeleˌsme kolo, no ale to zas potřeboval boťto taťineg anebo maminka, tak teˌnám moc často nepučele, to sme museleˌchoďit tea pješke,

informátor 0:18:52.7 – 0:18:55.2
Fšeci decki sme na sebe za ďeďinó počkale,

informátor 0:18:55.7 – 0:18:58.4
No a pak sme teda, že ja, šle tam

saaaaa
0:18:10.7 – 0:18:38.4
historie, dějiny 1. světová válka

Zadat název
Prohledávat metadata

Figure 5.4: State of the developed interface at the time of the third round of testing. Areas marked in the screenshot were mentioned during the individual tests or the group discussion. There are 3 main areas of interest – A is the *playback*, B is the *segment list*, and C is the *group list*. Each of these areas has some parts marked. Firstly, inside A, there is the *minimap* in A1. A2 is the *waveform*. Waveform contains A2x, which is a *waveform region* or just a *region*. A3 is the *speed control* and A4 are the *playback controls*. Secondly, inside B, there is a *segment* marked B1. Each segment has *segment actions* highlighted in B1x. Thirdly, the area C contains a *group* in C1. A group consists of the *group title* in C1x, the *metadata tags* also called *group tags* in C1y, and the button to add a subgroup in C1z. In C2 there is a *group form*, which is used to create groups. Similarly to the title in a group, there is the *title input* in C2x. *Start and end segments selection* in C2y. C2z marks the *metadata search* box.

Participant #1

The participant started creating a group, choosing the segments, and adding tags without any problems. When he pressed the button to create the group an error message appeared. It said that not all mandatory fields are set. That confused him a bit, because he did not understand why the title would be mandatory. He proceeded to fill it in and created the group.

Then, he tried to create another group and explained that he wanted to create a subgroup first and the parent group after.

Next, Mr. Szőke asked the tester about the „merge down“ button (double arrow in B1x in Figure 5.4). He did not know what it would do. After explanation, he said that that was not what he would expect.

Finally, he proceeded to play with the player and controls. He suggested making the waveform component higher, or that it should be resizable by dragging. He also expected the mouse wheel to scroll the waveform and discovered that there was no way to scroll the waveform with a mouse, which was a bug. When nothing happened after a click on the minimap, he commented that it should seek on the waveform. Other parts of the player worked well in his opinion.

Participant #2

When creating a group, she tried to select group start on the waveform. However, she quickly discovered that she was supposed to select a segment. She proceeded to create a group per segment instead of one group containing multiple segments.

While working with the player, she suggests to highlight the currently playing segment in the transcript. Then, she complained that the speed control, which had options to change speed to 0.5x, . . . , 0.9x, or 1.0x, provides not fine enough steps for her needs. On the other hand, she praised that when the recording plays slower, the voices of the speakers do not deepen.

Additionally, during her current workflow, she often skips back a few seconds to replay a short portion of the audio. She said three seconds would be a good skip length for her.

Participant #3

The tester creates a group with a subgroup. He understands how to select start and end segments and how nested groups work. The only hiccup was when he omitted the title because he thought it was optional.

When testing the playback area, he proposes several changes. The mouse wheel should zoom in and out. He proposed two sets of skip buttons. A shorter skip, about five seconds, and a longer one of about ten seconds. Similarly to the first participant, he noticed that the waveform does not scroll and he proposed to add a slider.

In the end, Mr. Szőke pointed out the „merged down“ button. This participant said he would expect some menu to open.

Participant #4

At first, she tried to drag the button to create a new group, probably by accident. She has trouble selecting start and end segments and she does not understand that groups can be nested. In the end, she successfully creates a group applied to some segments and another group applied to a subset of its segments, however, as two top-level groups and not as a group with subgroup.

When selecting metadata tags, she suggests that on tag deletion, only the most specific subcategory should be removed, e.g. a–a1–a1x would become a–a1 instead of deleting the whole tag. She would also prefer to have more contrast between the parent group and its nested groups.

When interacting with segments, she keeps subconsciously clicking a segment to replay it before realizing there is a button for it. She also suggests the segment actions should be spaced further apart from the speaker and the start-end time range, noting she would put them on the right side of a segment.

Group Discussion

Four main topics were discussed.

When creating a phonological transcript, the users will need a set of **special characters** to transcribe some phonemes. The set of characters if always the same. The characters should be listed somewhere close to the segments and copied on click or inserted into segment text.

Various **skip lengths** were discussed. Three seconds are mentioned most often. The length may also depend on the recording since some speakers talk slower and some talk faster.

Several proposals on how to rework the **playback speed control** to allow for finer adjustments were drafted. Changing the speed by 0.1 is not fine enough.

Group title was discussed, what is its purpose, and whether it is needed at all. In the end, the conclusion was achieved that it should stay, but not as a mandatory field.

Analysis and Outcomes

This round resulted in one missing feature request and several proposals for smaller adjustments.

- **Special characters.** a new component will be added that will display special characters and a way to insert them into the segment text will be conceived.
- **Skip length** will be made adjustable. Three seconds will be the default setting. Adding a second set of skip buttons with editable skip length will be considered.
- **Playback speed control** will be reworked to allow for finer adjustments.
- **Group title** will be made optional.
- **Segment actions** will be moved to the right side of the segment and the „merge down“ icon needs to change.
- **Waveform synchronization.** Minimap and segment click will seek on the waveform.

Overall, the interface has gotten refined. There are fewer big changes and more smaller adjustments and improvements. All of the main features have clear outlines and are functional. There is still potential for improvement in clarity and visual feedback.

5.6 The Fourth Round – Interconnectivity of the System

This round focused on the system as a whole, how well its parts work together, and tried to find any deficiency in this manner. In comparison with the previous rounds, all main and auxiliary components were fully implemented. At the beginning of the group discussion, all participants answered ten questions from a questionnaire, which is detailed in the next section.

The Task

In the limited conditions that the sessions offer, the task tried to imitate the everyday work of future users as closely as possible. There was only one task:

1. As if you were doing your regular work, transcribe and annotate a couple of sentences.

If a feature was not used, when a tester executed the task, for example, the special characters, they were encouraged to try it at the end of their session.

Individual Tests

There were five testeres who volunteered to undertake the task. All of them had seen the interface before. However, participant #5 is participant #3 from the first round who walked away (see Section 5.3) so she had not worked with the app before. There was no time limit per participant.

Participant #1

From the start, the tester worked with the system competently. She created a segment without any problems. While transcribing she used the player and segment play button naturally. She noted that segment click should not only seek but also start the playback. Three segments were created and the correct speakers were selected. She experienced trouble creating and deleting speakers because there was no label and it was not clear how to create a speaker.

No major issues were discovered when she interacted with groups. It took some time for her to discover how to edit a group, but she successfully edited a group title eventually.

A few suggestions were made. The default speaker for a new segment should load from the previous segment. Group start and end segments selection button labels should be tweaked to reflect what is being selected. She would expect that a click on a group visualization marker would highlight the corresponding group in some way.

Participant #2

In the beginning, the participant needed a hint to be able to create a segment. Then she transcribed a few sentences and set the speakers fine. She had trouble understanding the connection between waveform regions and the segments but got there eventually. Additionally, she found overlapping regions confusing but managed to solve it in the end.

She would welcome a finer zoom level. Also, a bug was discovered. When regions are overlapping and the cursor is in the shared portion, the later segment's play button does not start the playback from its start. She mistook the currently playing segment outline for the segment being selected.

A group was created. After initially trying to select the start in the waveform, start and end segments were selected in the transcript. Metadata tags were added without any issues.

Adding and removing speakers went well. When trying to add special characters she joyfully commented: „That was easy.“ She successfully used the undo action as well. She mistook the currently playing segment outline for the segment being selected.

Participant #3

This participant also needed a hint to create a segment. She decreased the playback speed, zoomed in, and transcribed a few sentences. Her expectation was that the jump button would skip to the closest segment's start instead of at the beginning of the recording.

A speaker was added and segment speakers were selected successfully. She also used undo and inserted some special characters. At first, she expected the segment time range to automatically adjust based on the text, but then she realized that it is achieved by resizing the associated waveform region.

Participant #4

He pressed the space bar to start the playback. After he tried to `ctrl+mouse wheel` to zoom in on the waveform, which did not work so he used the zoom buttons instead. Initially, he needs a hit to be able to create a segmetn too; however, proceeds to work with segments fine after. „On resize, when getting closer to the neighboring segment, the resized region should stick to it," he suggested.

At first, there was some confusion about how to add a new speaker. Then, some special characters were inserted and after that he used `ctrl+z` to undo, which worked well.

Participant #5

This is the participant who had walked away during the first round a few months before.

Initially, she also needed a hint to be able to create a segment. She asked if there was another way, adding she would prefer to click some button. A sentence or two was transcribed and she successfully used `ctrl+z` to undo. She complained about the Czech translation of „undo" and „redo", saying she would prefer it was the same as in Word. Afterward, she inserted some special characters without any problems.

Group Discussion

In the beginning, the participants answered a usability questionnaire (for details, results, and analysis see Section 5.7).

The discussion itself was brief, only one topic was discussed. It was agreed that the system satisfied their needs. After some more experience with it, they would be able to use it for their work. A manual would help. Either a document with screenshots or a video tutorial. Also, students sometimes come to their department and it would be useful to give them something that would introduce them to the app. We agreed it would be created during the early summer (not as a part of this thesis) and added later.

Analysis and Outcomes

Overall, the participants were able to work with the interface. After some practice and maybe occasional help from more technical colleagues, they would all be able to use the system for their daily work. Some improvements will be made based on this round:

- A **new speaker placeholder** will be added to nudge users when they want to add a speaker.
- **Group start and end segment selection button labels** will be tweaked to better reflect what action is expected during selecting.
- **One more finer zoom step** will be added.
- **Segment play button** will be fixed so that it starts the playback from the start of a segment even if segments overlap.
- **Currently playing segment highlight** will be modified to distinguish it from a selected segment in group start and end segments selection.
- **Undo and redo translations** will be updated.

Regarding the segment creation process, most of the participants needed a hint. Once they knew a drag on the waveform creates a segment, they used it fine during the rest of their sessions. Reworking it might produce a more intuitive process; however, a change of this size to the workflow should be properly designed and tested. This is not possible at this stage of the development.

Moreover, the users showed their ability to adapt to the implemented process. Once they get used to it, it may be even quicker than, for example, selecting and subsequently clicking a button. A decision was made not to change it and leave it as a possibility to try in the future.

5.7 Final Quesitonnaire

Since the fourth round of tests was the last one, the developed interface underwent further evaluation. As previously mentioned in Subsection 5.6, apart from the usual qualitative test sessions, the participants completed a final questionnaire.

System Usability Scale

System Usability Scale [2] (SUS) is a questionnaire created by John Brooke. It is commonly used to assess the usability of graphical interfaces. For example, some US federal agencies use it[1]. Producing quite reliable results even for a smaller number of respondents, it is ideal for the purpose of this thesis. It was also used by Plhal [7].

The questionnaire, as well as the Slovak translation read to the participants, can be found in Appendix A. It consists of ten questions about a participant's opinions and assessments of the tested interface. The questions are answered with a number from 1 to 5, where 1 is „strongly disagree“ and 5 is „strongly agree“.

The resulting score is a single number between 0 and 100. However, it should not be interpreted as a percetage. The Brooke's article [2] describes how to calculate it from the responses:

> *To calculate the SUS score, first sum the score contributions from each item. Each item's score contribution will range from 0 to 4. For items 1,3,5,7, and 9 the score contribution is the scale position minus 1. For items 2,4,6,8 and 10, the contribution is 5 minus the scale position. Multiply the sum of the scores by 2.5 to obtain the overall value of SU.*

The Results and Analysis

The scores given by respondents were calculated and are graphed in Figure 5.5. The scores of individual participants were averaged to produce the overall result of **64**.

[1] https://digital.gov/2014/08/29/system-usability-scale-improving-products-since-1986/, accessed 2024-05-05

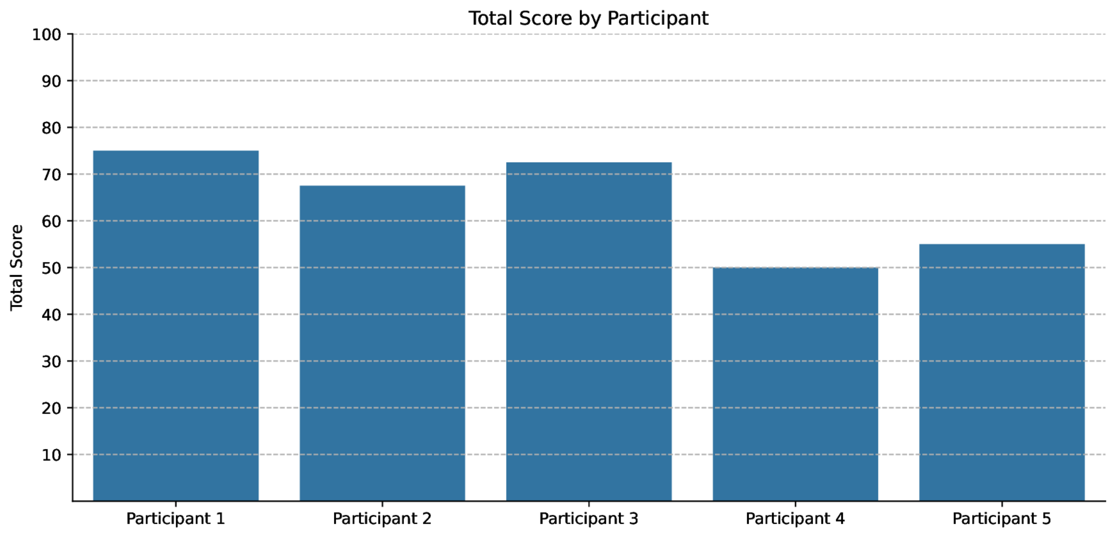


Figure 5.5: Total score by participant from the System Usability Scale Questionnaire. The score can range from 0 to 100.

The interpretation of the result draws from Bangol et al. [1]. A score below 50 is considered not acceptable, a score above 70 is acceptable, and in between is considered marginal. The score the interface received falls into the higher portion of the marginal section. Expressed as adjectives, the score falls somewhere between „OK“ and „good“.

When analyzing the responses in greater depth and looking at them by question (see Figure 5.6), the most interesting one to focus on is the question number 9. It is the only question where the median and average are on the wrong side. The question asks about user confidence while using the system. This may have been caused by the limited time they spent working with it. Further supporting this speculation is the neutral median answer to question number 1 that is asking if the respondents would like to use the system frequently.

Also notable is to contrast the question number 9 with the question number 7, where respondents strongly agreed that most would learn to use the system quickly. In general, people assess the abilities of others more realistically and tend to undermine their own.

On the positive side, apart from the question number 7, the items number 2, 5, and 8 were answered overwhelmingly well. Indicating that the respondents did not consider the system too complex, thought its various parts were well integrated, and disagreed it was cumbersome to use.

The rest of the questions is where the most room for improvement lies.

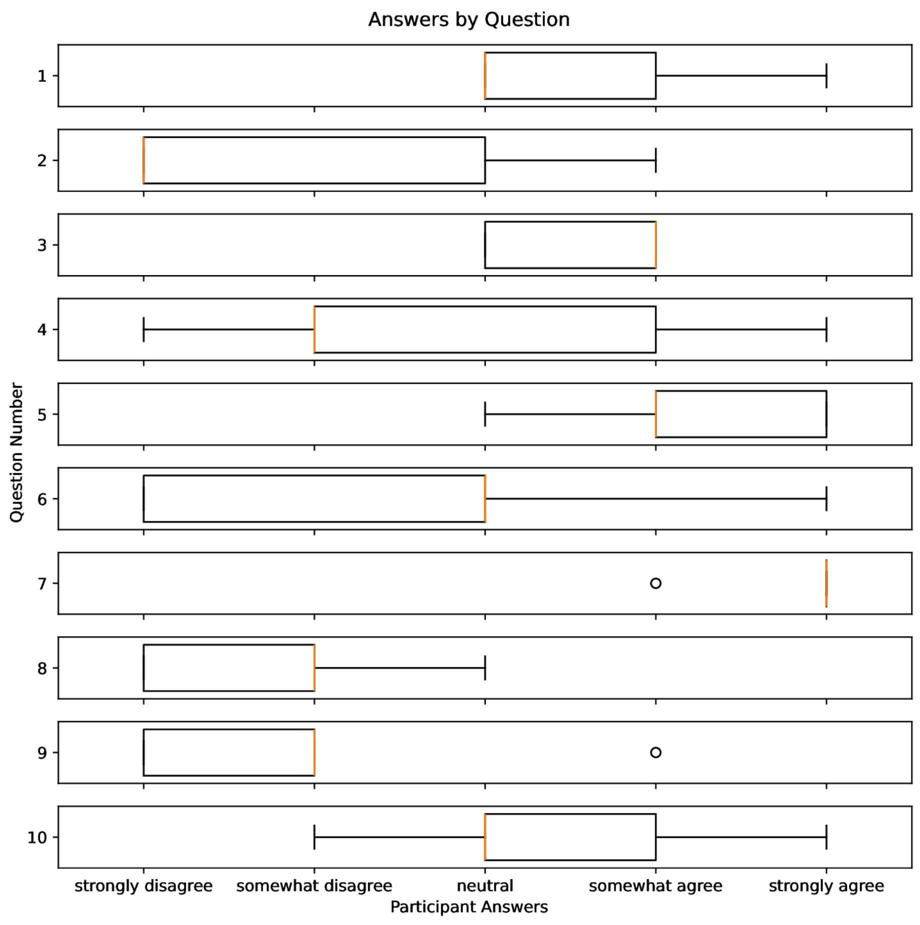


Figure 5.6: The distribution of answers by question. Questions with even numbers are positive, the higher the agreement the better. The odd-number questions are negative, the higher the disagreement the better. Also, the average scores were in order: 3.6, 2, 3.6, 2.8, 4.2, 2.6, 4.8, 1.8, 2, 3.4.

As mentioned in the testing rounds discussions and analyses, users need some time to learn the ins and outs of the system. Some participants left notes on the questionnaire paper commenting that their rating would be better if they had more experience with the app. They would gain more confidence. So the score of 64 can be considered a lower bound for this interface and has the potential to improve with time without any changes to the interface. Improvements and fixes in the future could increase the usability score further.

Chapter 6

Possibilities for the Future

The scope of this project does not allow for implementing everything that arises during the drafting, development, and testing. Functions paramount to the core features were prioritized. There are several offshoots, alternative approaches, or extensions left to try.

More Player Tools

Similarly to common audio and video editing software, more tools could be introduced. For example, a *selection tool* to select on the waveform without creating a segment, *hand tool* to move the waveform instead of scrolling, *cut tool* to split waveform regions, *move tool* to drag the regions, and more.

Implementing such tools with the library currently used for visualization is prohibitively complicated. Therefore, research and comparison of similar libraries would likely be conducted beforehand.

Alternative Segment Creation

Although the currently implemented process works well once the users are familiar with it, there could be a better solution. Utilizing, for example, the aforementioned selection tool, after selecting, there could be a button to create a segment.

Additional Control Buttons

During testing, one of the users wanted to precisely align a segment's start to the previous segment's end. If segments in the transcript were selectable, it would allow for a new set of buttons in playback controls. The buttons would set a selected segment's start or end to the current time.

Speaker Introduction

When working with several speakers, it can be hard to distinguish them from one another. Users need to return to the beginning of a recording where the speakers introduce themselves to identify their voices.

A short fragment of a recording could be associated with each speaker. There could be a way to replay it without seeking to the beginning. A fitting place in the interface for this feature would be somewhere in the speaker list or close to it.

Additional Segment Actions

Two possibilities that were considered during development are *split segment* and *merge segment up*. Splitting would create two segments in a segment's place, the segment text and corresponding regions would split. The merge segment up action would be complementary to the existing merge segment down action.

Improvements to the Group Visualization

This is one of the completely new features. If users find it not contrasting enough after they spend more time with the system, there could be some colors added.

Additionally, there are no click interactions and the visualization only displays groups. A set of fitting click or drag interactions could be explored, designed, and implemented.

Other

Other notable points are:

- light theme,
- horizontal waveform and minimap support,
- transcript search and filtering,
- additional limitations to group creation that would nudge the users in the right direction.

The testing sessions and iterations of implementation brought the app to a well-functioning state. However, some issues might be discovered in the future after the users use the interface for a longer time and gain more experience with it. This is also what may be addressed in the future.

Chapter 7

Conclusion

User-friendly components that simplify and enable long audio transcript creation, editing, and annotation were the goals of this thesis. Such components were successfully implemented and tested as a part of a new interface specifically customized for this use case.

Initially, theoretical research into the fields of interface design and user experience was conducted. Then, a review of an existing interface with similar functionality was executed, further expanding the outlook on the topic.

Having acquired vital knowledge about the conventions and possibilities in the domain, an initial draft was formed. Implementation was heavily influenced by the testing rounds held during all its stages. Employees of the Czech Language Institute of the Czech Academy of Sciences, who are the target users, participated in the testing. New components were based on their feedback, for example, a set of special accented characters used for phonetic transcription of some sounds.

Besides user confirmation that the interface fulfills their needs, it was assessed with the SUS questionnaire [2]. As a part of the final round of testing, the participants answered a set of ten questions related to their experience. It showed that the implemented interface achieved „OK“ to „good“ user experience.

The app is by no means perfect. The users have been switching to a new workflow. Although some showed a great ability to adapt, the testing conditions were limited, and all need more experience with the interface. Some options were intentionally left open because user feedback is needed after the app has been deployed for a longer time. Only then strategic decisions can be made about the interface's future.

Bibliography

[1] Bangor, A., Kortum, P. and Miller, J. Determining what individual SUS scores mean: adding an adjective rating scale. *Journal of Usability Studies*. Bloomingdale, IL: Usability Professionals' Association. may 2009, vol. 4, no. 3, p. 114–123. ISSN 1931-3357.

[2] Brooke, J. SUS: A quick and dirty usability scale. In: *Usability Evaluation in Industry*. CRC Press, June 1995, p. 189–194. ISBN 9780748404605.

[3] Krug, S. *Rocket Surgery Made Easy: The Do-It-Yourself Guide to Finding and Fixing Usability Problems*. Berkeley, CA: New riders, 2010. ISBN 978-0-321-65729-9.

[4] Krug, S. *Don't Make Me Think, Revised: A Common Sense Approach to Web Usability*. 1st ed. San Francisco: New Riders, 2014. ISBN 978-0-321-96551-6.

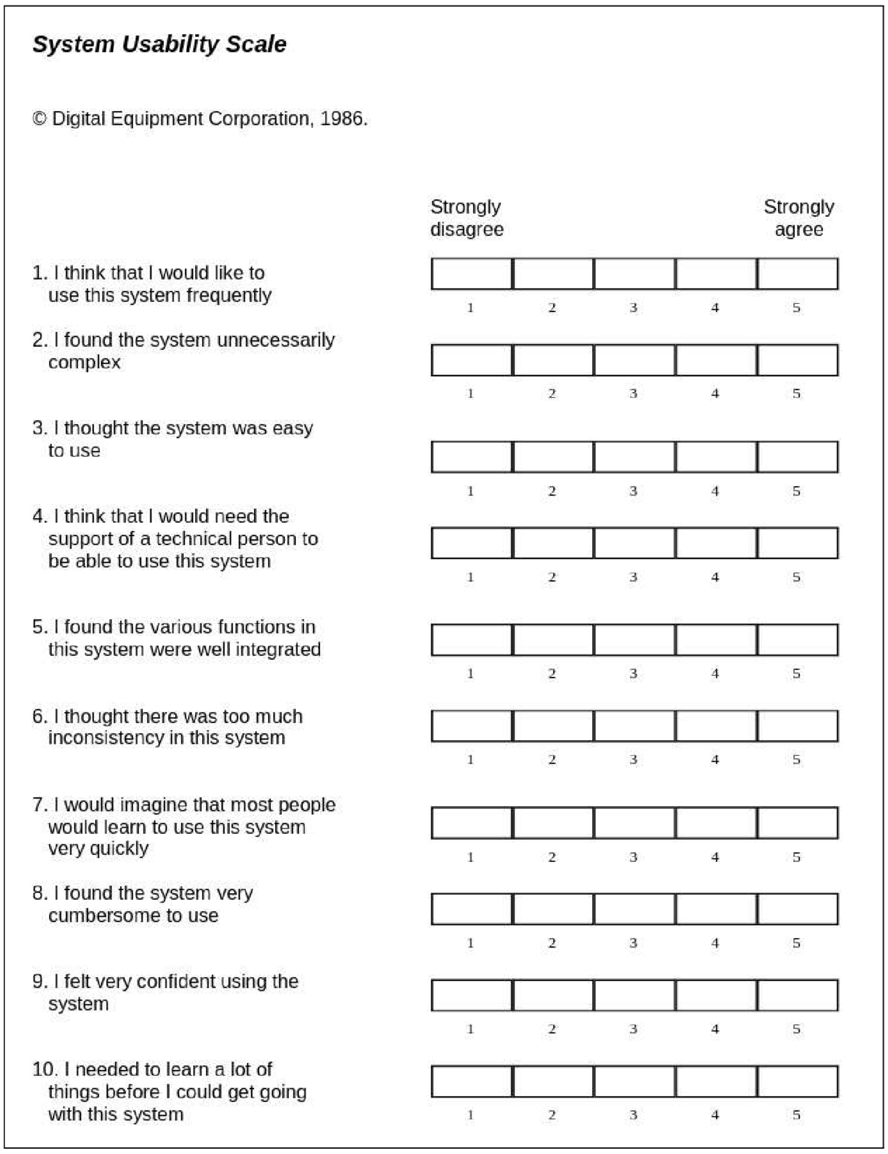
[5] Lidwell, W., Holden, K. and Butler, J. *Univerzální principy designu: 125 způsobů jak zvýšit použitelnost a přitažlivost a ovlivnit vnímání designu*. 1st ed. Brno: Computer Press, 2011. ISBN 978-80-251-3540-2.

[6] Miller, B. D. *Principles of web design*. New York: Allworth Press, 2022. ISBN 978-1-62153-787-8.

[7] Plhal, J. *Web Interface for Human Corrections of Automatic Transcript and Tagging*. Brno, CZ, 2022. Bachelor's thesis. Brno University of Technology, Faculty of Information Technology. Supervisor Szőke, I. Available at: https://www.fit.vut.cz/study/thesis/24414/.

Appendix A

System Usability Quesitonnaire



System Usability Scale

© Digital Equipment Corporation, 1986.

	Strongly disagree				Strongly agree
1. I think that I would like to use this system frequently	1	2	3	4	5
2. I found the system unnecessarily complex	1	2	3	4	5
3. I thought the system was easy to use	1	2	3	4	5
4. I think that I would need the support of a technical person to be able to use this system	1	2	3	4	5
5. I found the various functions in this system were well integrated	1	2	3	4	5
6. I thought there was too much inconsistency in this system	1	2	3	4	5
7. I would imagine that most people would learn to use this system very quickly	1	2	3	4	5
8. I found the system very cumbersome to use	1	2	3	4	5
9. I felt very confident using the system	1	2	3	4	5
10. I needed to learn a lot of things before I could get going with this system	1	2	3	4	5

Figure A.1: SUS questionnaire by John Brooke [2].

The Slovak Translation of the Questions

1. Myslím, že by som tento systém chcel používať často.
2. Považujem tento systém za zbytočne zložitý.
3. Myslím si, že systém sa ľahko používal.
4. Myslím si, že by som potreboval pomoc technicky zdatného človeka alebo odborníka, aby som bol schopný používať tento systém.
5. Myslím si, že rôzne časti systému sú spolu dobre integrované.
6. Myslím si, že v systéme bolo priveľa nekonzistencií.
7. Viem si predstaviť, že väčšina ľudí by sa naučila pracovať s týmto systémom veľmi rýchlo.
8. Myslím si, že systém bol veľmi zložitý na používanie.
9. Cítil som sa, že som používal tento systém sebavedomo.
10. Potreboval som sa naučiť veľa vecí pred tým, ako som mohol začať používať tento systém.