CoCo: Design and Evaluation of a Collaborative Application for Detecting and Coding Critical Incidents

Bachelor Thesis

by

Simone Weipert

at the

Universität Konstanz

Human-Computer Interaction Group

Department of Computer and Information Science

First Reviewer: Prof. Dr. Harald Reiterer
Second Reviewer: Dr. Johannes Fuchs

Konstanz, 2021

Abstract

The detection and coding of critical incidents is a vital part of many usability studies performed to evaluate a wide range of products and systems. Current methods still rely on time-consuming and ineffective techniques or methods (i.e. pen and paper) to collect critical incidents. Existing digital systems, which offer potential solutions for an improved collection of critical incidents, often lack key features like collaboration and communication aspects or supportive user features. The application presented in this thesis supports researchers of the HCI group to detect and code critical incidents by incorporating collaboration and communication aspects, and other key features.

This thesis focuses on the design and evaluation of the prototype "CoCo - Collaborative Coding of Critical Incidents". Beginning with the examination of the theoretical background about the detection of critical incidents, the thesis presents the following focus group about critical incident detection with Ph.D. students of the HCI group. It leads to a set of requirements to guide the design process of CoCo. Furthermore, related systems are analyzed regarding the derived requirements. Afterward, the prototype CoCo is described and illustrated. It is a collaborative web application for the remote detection and coding of critical incidents during studies. It offers researchers to observe live streams of studies to detect critical incidents collaboratively. Each researcher can observe the live streams remotely from any device with an internet connection. Team members can communicate throughout the observation. Moreover, CoCo offers the tagging and illustration of detected critical incidents in a timeline that is synced to the live streams. This allows for a logical connection between the video footage and the information collected by the observer. Additionally, CoCo offers researchers to re-watch the live stream after they ended. This enables them to refine the collected data.
Finally, a usability study with four participants within the HCI group was conducted. The goal of the study was to evaluate the overall usability and user experience, as well as the possible improvement from the ineffective methods for critical incident collection. Although users needed some help to get used to the system, they welcomed the collaboration aspect together with the web-based implementation approach. Based on the results of the usability study, potential design improvements and further research directions for the remote detection of critical incidents are described.

Contents

List of Figures

List of Tables

1. Introduction

User Experience (UX) design, good or bad, influences everyone's day-to-day life. It is therefore important to create designs utilizing good UX design. To produce a good UX design, a UX design life-cycle should be employed (see Figure 1.1).

UX evaluation can take place in various forms, one of them being a lab-based evaluation [1]. It involves the observation of participants performing tasks on the evaluated system. The data collected throughout the evaluation is either quantitative or qualitative. Whereby qualitative data is the most meaningful data type, as it assists to identify UX problems and their causes [1]. Various methods can be employed to collect qualitative data, the critical incident collection being one of them. Throughout time, systems were developed to aid researchers to collect qualitative data digitally. However, inefficient methods, like pen and paper, are still frequently applied, especially in the field of critical incident detection. This entails several disadvantages [2]. Firstly, employing the pen and paper method is often stressful for the observer, as they have to watch participants and write down critical incidents simultaneously. Secondly, it is cumbersome to correlate the handwritten notes with other collected data mediums, like video recordings, as written timestamps are often the only correlation created between the data. This lack of context between different kinds of data mediums complicates the post-processing of the data for the researcher.

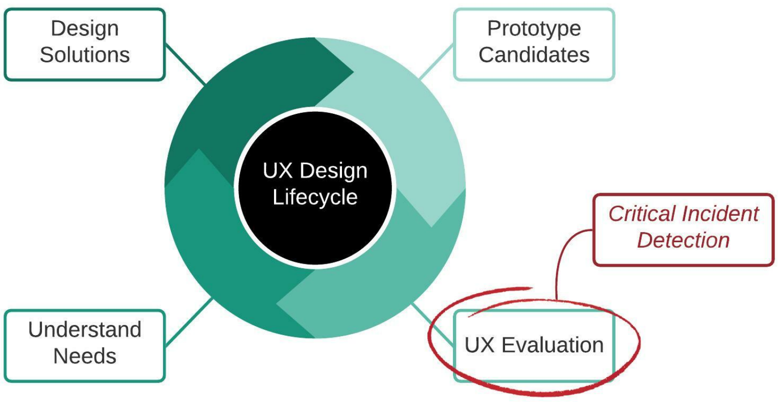


Figure 1.1.: The UX design lifecycle process consists of four main lifecycle activities. UX evaluation is to judge and improve the UX design [1]. Critical incident detection is part of the evaluation activity. *Image taken and slightly adapted from [1].*

This work introduces a collaborative web application for remote critical incident detection and coding. The goal of this application is to ease the detection of critical incident during study sessions and improve the post-processing of the data afterward. Additionally, the prototype facilitates the possibility for remote and effortless teamwork.

The content of the thesis is structured as followed: The next chapter summarizes the foundation of the topic of critical incidents and the detection of them. Afterward, Chapter 3 recapitulates a conducted focus group with Ph.D. students of the HCI group and the resulting requirements for the prototype CoCo. Based on the derived requirements, Chapter 4 analyzes the related work and sets forth a description and analysis of relevant systems. The main focus of the analysis lies on whether the systems fulfill the requirements of the last chapter. Subsequently, Chapter 5 explains the design process for the prototype CoCo. The goal of the design process was to find a design for a web application supporting all relevant requirements listed in Chapter 3. Chapter 6 describes the preparation and conduction of a usability study for the evaluation of CoCo. Afterward, Chapter 7 illustrates the findings of the evaluation. Chapter 8 discusses these results and proposes possible design improvements for the prototype CoCo. Moreover, it inaugurates topics for the future. The last chapter of this work concludes with a summary of important aspects of the previous chapters.

2. Foundations

This chapter summarizes the foundations regarding the detection of critical incidents. The first section provides a widely recognized definition of the term "critical incidents". The next section informs about the critical incident technique, its workflow, and its importance. Discussed in Section 2.3 are multiple possible observer groups that can detect critical incidents. Section 2.4 describes the type of content that needs to be collected during critical incident detection. Section 2.5 explains the importance of time, concerning the perception span of a human, as this heavily influences the quality of critical incident data collected by an observer. The last section describes principle skill sets needed for the successful detection of critical incidents. This chapter was adapted from the preceding seminar [2].

2.1. Definition of Critical Incidents

To understand what an observer needs to be aware of during a study, it is important to define what a critical incident is. Hartson and Pyla define critical incidents as follows:

"A critical incident is an event occurring within usage that indicates a barrier, problem, or difficulty encountered by the user, or simply something the user did not like." [1]

2.2. The Critical Incident Technique

Being the "single most important qualitative data collection technique" [1], the detection of critical incidents is an essential part of the UX evaluation process. The following section clarifies the course of the technique. The critical incident technique, CIT for short, was initiated in World War II for the selection and classification of aircrews [3]. Flanagan [4], alongside other psychologists, continued to develop the technique. At this time it was mainly used to establish critical job requirements [3]. The technique later became an important research method, after the publication of an article in 1954 [4]. The designed CIT was not intended to be a single rigid procedure, but a flexible set of principles that have to be modified to be suitable for specific situations [4]. It contains five essential steps [4] also seen in Figure 2.1:
The first step in the critical incident technique is the determination of the general aims of the studied labor. These general aims should always be briefly stated, simply expressed, and agreed upon by all those involved in the evaluation process. The second step is to make plans and specifications. Third, the observed data has to be collected. This can be performed in multiple forms such as personal or group interviews, questionnaires, and record forms [3]. The collected data then has to be analyzed. This step is the most important and difficult one, consisting of the classification of the critical incidents. The final step of the procedure is to interpret and report the data. Over the years, this method was adapted to suit other fields of interest such as the field of Human-Computer Interaction (HCI) [1]. In this field, the critical incident technique is used for the identification of UX problems and their origin for the improvement of user experience of a system [1].

Figure 2.1.: The five main steps performed within the critical incident technique [4]. *Image taken and adapted from [4].*

2.3. Identifiers of Critical Incidents

In the early use of the Critical Incident Technique, the user reported critical incidents after a performed task [1]. Later, Flanagan collected data with the help of trained observers [1]. However, over time it was discovered that the user or the investigator or a combination of both parties can report these incidents. In the study process introduced by Hartson and Pyla [1] there is a dedicated team member for the collection of qualitative data — including the critical incident information. They can objectively capture indicators of emotional impact through direct observation, whereas self-reporting techniques do not produce objective data [1].

2.4. Content of Critical Incident Data

The level of accuracy and efficiency of the data analysis is dependent on the information contained in the documentation [1]. Capturing detailed information about critical incident data can be difficult because this type of data can be perishable (i.e. if not captured immediately and precisely, the data will be lost) and subtle (i.e. critical incidents might be easily overlooked events, e.g. a head-shaking) [1]. Consequently, it is favorable to know what information needs to be included during the documentation. Hartson and Pyla [1] conducted a list of information that should be included in critical incident data:

- What was the task or activity the user performed at that time?
- Were any items involved?
- What was the users' intention that prompted the critical incident?
- What did the user expect the system to do?
- What happened instead?
- Any information about the psychological condition of the user.
- Could the user recover? How did the user recover?
- Comments of the investigator in regard to possible solutions or other useful thought.

2.5. The Investigators Perception Time Span

Time is a crucial part for the capture of critical incident data. It is generally not possible for an observer to perceive a critical incident the moment it occurs. Figure 2.2 displays the course of the observers' awareness when they perceive an incident. When the incident is recognized, time is needed to decide if the incident is a critical one [1]. As seen in Figure 2.2, the optimal time to report on a critical incident is at peak understanding. The peak of understanding being the moment a critical incident transitions from being a vague notion into a firm abstract concept. A report of critical incidents after the peak of understanding is expected to be less detailed and accurate, due to the limitations of human memory [1].

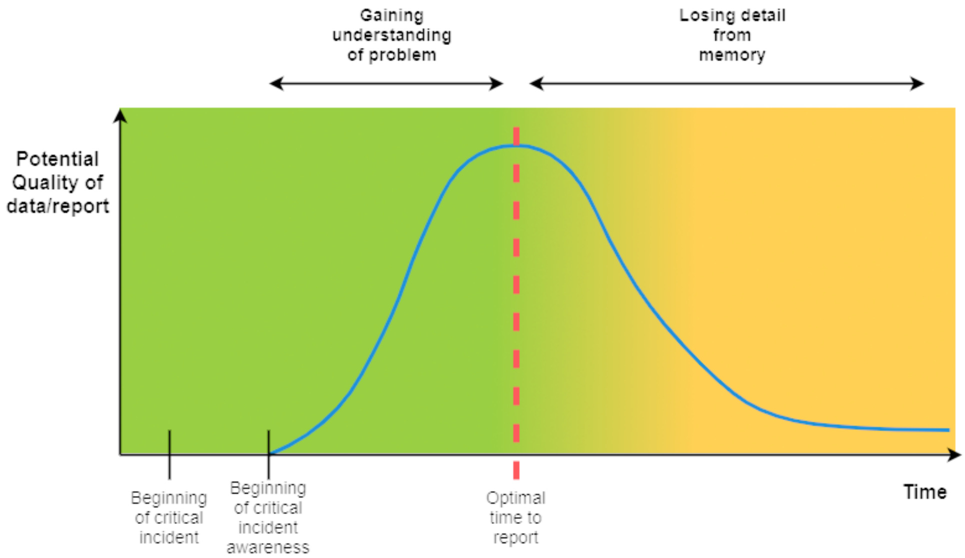


Figure 2.2.: The recognition of critical incidents begins after the incident happened and should be recorded when the observer reaches peak understanding [1]. *Image taken and slightly adapted from [1].*

2.6. Observation and Note-Taking skills

Diverse skills are useful for the collection of qualitative data. These skills are also frequently used in everyday life and thus offer a wide range of applications [1]. For instance, note-taking and observation are central skills for critical incident collection. The act of observation is defined as a *"practice of witnessing an ongoing activity with the objective of understanding underlying phenomenon"* [1] and yields inputs for reasoning and assumptions [1]. While observing the participant, the observer is looking after several events like exceptions, surprises, problems, barriers, and more [1]. This ability, however, is not a naturally given skill, and it has to be practiced for it to become a useful resource. The training of this skill is essential for UX practitioners to realize if something meaningful is happening and not let it slip unregistered. Learning by doing is the leading approach to achieving this skill [1].

Note-taking, being *"a practice of efficiently capturing descriptions of observations"* [1], is a skill that is also not naturally gained during an education or profession [5]. It supports the outsourcing of memories to an external source, like pen and paper, and creating content precisely for a future remark, as well as accomplish better decisions, solving problems, and working more efficiently as a group [5]. The act of note-taking includes several mental processes, such as coordinating the observation, the physical typing, severe time pressure, and the fact, that problematic events can occur faster than one is able to write them down [5]. Note-taking depends heavily on the working memory of the brain, which is responsible for short-term storage and mental transformation [5]. The working memory is used within observation and productions; as a result of its limited capacity, a trade-off between observation and production has to be made, which introduces a struggle to find a balance between these two aspects [5]. To reduce the cognitive burden of the production of notes, an observer can use multiple techniques like stenography or shorthand [5]. As a result of this, the language of the notes will be affected on several levels [5]. A great element to consider about notes is their style. Note-taking styles can be classified into two categories: linear and non-linear [5]. A linear style is comparable to traditional written texts, whereas non-linear include graphical representations like sketches or models. Additionally, the medium, on which the notes are taken, must also be considered. Mostly, it is differentiated between pen and paper and writing on a laptop. Both have their advantages and disadvantages. Using a laptop can reduce cognitive resources, but this can also be accomplished if the note-taker is exceptional at shorthand [5]. Another disadvantage of the laptop is the chance of distraction, which may reduce the quality of the notes [5]. One major problem with pen and paper is the limitation of the writing speed, although it is easier to apply non-linear styles on this medium [5].

A skill favorable for both, observation and note-taking, is abstraction. It is the *"practice of removing detail irrelevant to a given objective."* [1]. The abstraction of an objective allows the generalization from an example and consequently results in a more precise image of what is important without the aberration of irrelevant matter [1].

3. Requirement Analysis

In advance of implementing CoCo, it is crucial to understand and analyze the state-of-the-art critical incident detection of the target user group. These insights into the current process and further information about the users' needs allow a user-specific design and implementation of CoCo. Section 3.1 addresses the execution of a focus group with domain experts. Afterward, Section 3.2 presents the results of the focus group as requirements that serve as guidance in the design process of the CoCo prototype. This chapter has been discussed in the preceding seminar [2], and project report [6] and parts are summarized and updated appropriately for this work.

3.1. Focus Group

A user-centered design is widely perceived as a powerful approach for more effective, useful, and user-friendly Web designs [7, 8]. One core principle of user-centered design is to include the target user group as early in the design process as possible [9]. Specifically, a very popular method to gather information about the target user group, is to perform a focus group [10]. The focus group was conducted as described by Krueger and Casey in "Focus Group Interviewing" [11]. It consisted of four participants and one moderator. All participants were Ph.D. students of the Human-Computer Interaction group of the University of Konstanz. They were invited and received information about the topic so that they were able to prepare themselves. The conversation lasted about one hour, and an audio recording was done for later analysis. Throughout the interview, the participants were asked questions regarding the current process of critical incident detection. The goal of these questions was to collect the participants' opinions and beliefs [9, 10], as well as perceived problems and ideas for improvement.

3.2. Requirements

The collected qualitative data from the focus group [1, 10] led to many suggestions for improving the detection of critical incidents. These ideas could be translated into design requirements for the CoCo prototype. These requirements were organized into four groups: Video Annotation Features, Cooperation Features, Supportive User Features, and Analysis Features. Due to time constraints during the implementation phase of CoCo, it was not feasible to implement all the numerous requirements. An evaluation of the requirements' importance established a line of affordability. All requirements that fall under the line of affordability are considered optional.

3.2.1. Video Annotation Features

Video annotation in various forms is a relevant part of the process to detect critical incidents. A timeline synchronized with the recorded data of a study session is a helpful visualization of detected critical incidents. Additionally, the opportunity to annotate the video stream helps the user to connect recorded data with their notes and thoughts for a critical incident.

R1 **Timeline:** The system needs to have a timeline with jump marks, that represent critical incidents, which jump to the targeted time in the video.

R2 **Video Annotation:** The system needs to support video annotation directly in the video stream.

3.2.2. Cooperation Features

Three more requirements address the need for a remote and collaborative work environment. This includes the observation of study sessions via live streams, a communication channel between team members, and a system that can be used cross-platform.

R5 **Remote Collaboration:** It needs to be possible for more than one user to work on a project or study session concurrently and in different locations via live streams.

R6 **Communication between Users:** The user is able to communicate with other investigators through multiple communication methods, which is the only form of information exchange.

R7 **Dynamic Usage:** The system can be used cross-platform and support multiple User Interfaces to ensure desirable usage with different devices and within the different stages of the study session.

3.2.3. Supportive User Features

Emphasizing streams, cameras, recordings, and eye-tracking data is essential to aid users with additional context for the observed and recorded data. Another option for adding additional context and information is to include categories for critical incidents. One or more categories can be assigned to an incident. These categories can be added previously to study sessions with an optionally assigned hotkey. To further assist users in detecting incidents, note-taking is enhanced by the offer of several tools. The following list summarizes these requirements.

R9 **Categories:** The user can define categories for critical incidents before the study session with suitable hotkeys. A complete list of categories is shown during the study session.

R10 **Note-Taking:** The system needs to provide several tools to support note-taking. The tools include editable, previously defined text segments, tagging of critical incidents during the live stream and the possibility to write notes about a tagged critical incident.

R11 **Real Time Data:** The user can supply additional real time data, like screen recordings, to a project or study session.

3.2.4. Analysis Features

Along with observation via live streams, refining the detected critical incidents while playing back recorded data is a likewise crucial feature. Therefore, every user can revisit the recorded data to edit their findings. Building on these refinements, the quality of collected data can be ensured by obtaining intercoder reliability [1] and supporting statistics.

R12 Play Afterward: When the live stream ended and the study session is over, it will become available as video on demand to be watched again.

R13 Statistics: Users can calculate the intercoder reliability after the study session and export all collected data for further analysis tasks.

3.2.5. Requirements Under the Line of Affordability

The following requirements fall under the introduced line of affordability for multiple reasons. They could either be substituted with other requirements that are included in the design, or they are not essential to the functionality of the system.

R3 Voice Input: Users can create voice recordings in order to make notes about critical incidents.

R4 Speech to Text: The system provides a speech to text converter for transcribing what the participant says during the study as subtitles within the video.

R8 Configurations: The user can configure user interfaces and is able to save them and use them again on a later notice.

R14 Text of Previous Sessions: The user can access previously written text or notes from past study sessions or groups within the project.

[1] It is a number to measure the agreement between different coders about their codings of the same data [12].

4. Related Work

Systems related to detecting critical incidents do not have to be inherently designed for this purpose. Consequently, systems included in this work originated from areas like psychological or sociological research, general qualitative data analysis, video annotation, or even video editing. This chapter summarizes various existing systems and whether these systems meet the introduced requirements. The topic of this chapter has been examined by the preceding seminar [2] and has been summarized and appropriately updated for this work.

4.1. BORIS — Behavioral Observation Research Interactive Software

BORIS [13] is a freely available, open-source, and multi-platform program and therefore has a clear advantage over other commercial software (Mangold [14], observer XT [15]). It offers the possibility to digitally review previously recorded videos or live observations. The user can define an ethogram[1], listing various behaviors they want to detect when they create a new project. The main window of BORIS, as seen in Figure 4.1, features a toolbar with well-known video controls like play or pause. It additionally displays the defined ethogram and a list of possible subjects. Additionally, it displays recorded data when a previously recorded session is analyzed. Within a live study, a timer is shown instead of a video. The right side of the main view presents an event list displaying marked events. This event-based data can be exported and visualized when needed.

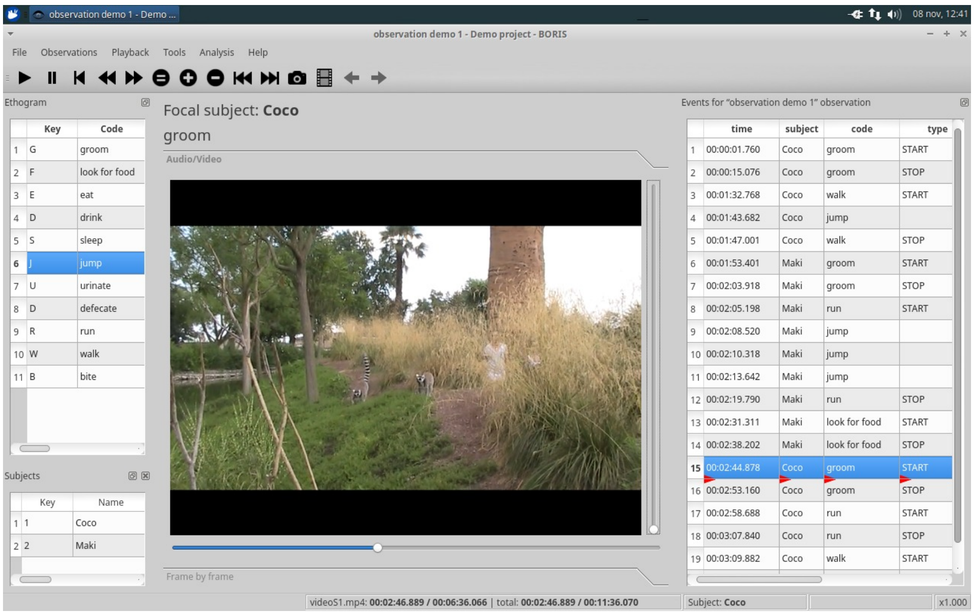


Figure 4.1.: The Main window of BORIS for the analysis of previously recorded footage. *Image taken from [17].*

[1]An ethogram is a catalog of action patterns, or behaviors, specific to a species [16]

When analyzing the system based on the requirements, these results became apparent:

No Timeline in any form is included in the system and therefore does not meet **R1 Timeline**. The only way to annotate a video in the system is by collecting new behaviors seen in the video material, which means that **R2 Video Annotation** is also not supported by BORIS. Hence, BORIS supports none of the requirements of the subgroup Video Annotation Features.

It is possible to perform live observations, but this is only possible without any video. The user has to use the system on site of the observation. A live observation, or any other observation, is only available locally on one device, so working in a team would mean sharing the device and be in the same location. Teamwork is achievable by sharing the recorded data only if it is not a live observation. BORIS is therefore not able to meet requirement **R5 Remote Collaboration**. Consequently, the system does not need a communication channel, which means requirement **R6 Communication between Users** can not be supported either. Since the contributors of BORIS provide the system for several operating systems and an Android app, it meets **R7 Dynamic Usage**.

One of the first steps in a project is to define an ethogram. A behavior listed in the ethogram consists of a type, hotkey, code, and description. The main view of BORIS also includes a complete overview of the ethogram. Selecting the entry on the list or pressing the hotkey tags a behavior. After that, the user can add textual comments to the tagged behavior. To support the detection of behaviors, the system allows users to include supplementing data like audio files, or other video material. To conclude, the system does support requirement **R9 Categories** and **R11 Real Time Data**, but not **R10 Note-Taking**.

Because live observations in BORIS do not include video material, they are not available afterward. Requirement **R13 Statistics** is met because BORIS can compute the inter-rater reliability of a project. BORIS therefore does support **R13 Statistics**, but not **R12 Play Afterward**.

Concerning the requirements under the line of affordability, BORIS does not support any of them. **R3 Voice Input** is not supported, as the system can not add voice inputs as notes for a collected event. Text is the sole format to comment on an event. A speech-to-text conversion for what is said in the video material must take place separately. Subtitles are only displayed if they are part of the video files. Due to this, BORIS does not satisfy **R4 Speech to Text**. The user interfaces provided by the system are minimally configurable (e.g. resizing the video view in the observation). However, a user can not save the preferred interface changes. Due to this, BORIS does not satisfy **R8 Configurations**. With the conclusion of the data analysis, BORIS supports the export of tagged behaviors. Suitable exported files offer the possibility to access previous comments. But this is not sufficient to meet requirement **R14 Text of Previous Sessions**, because the comments would only be accessible outside the system.

4.2. VCode and VData

VCode and VData [18] is open-source software for video annotation. It shares similarities to VACA [19] and other systems (The Observer XT [15], ANVIL [20]). The system is parted into three components, containing the Admin Window, VCode, and VData. The Admin Window facilitates different configurations for VCode. That includes a list of dependent variables to code and secondary data for further analysis. The VCode environment, seen in Figure 4.2, is utilized for the video annotation. It consists of the main window that incorporates three sections. The first section includes all selected videos, with one in focus. Another section displays a timeline which puts found events into view. Besides, graphical illustrations of further data (e.g. audio waves) are shown below the timeline. The last section lists the dependent variables configured in the VCode Admin Window.

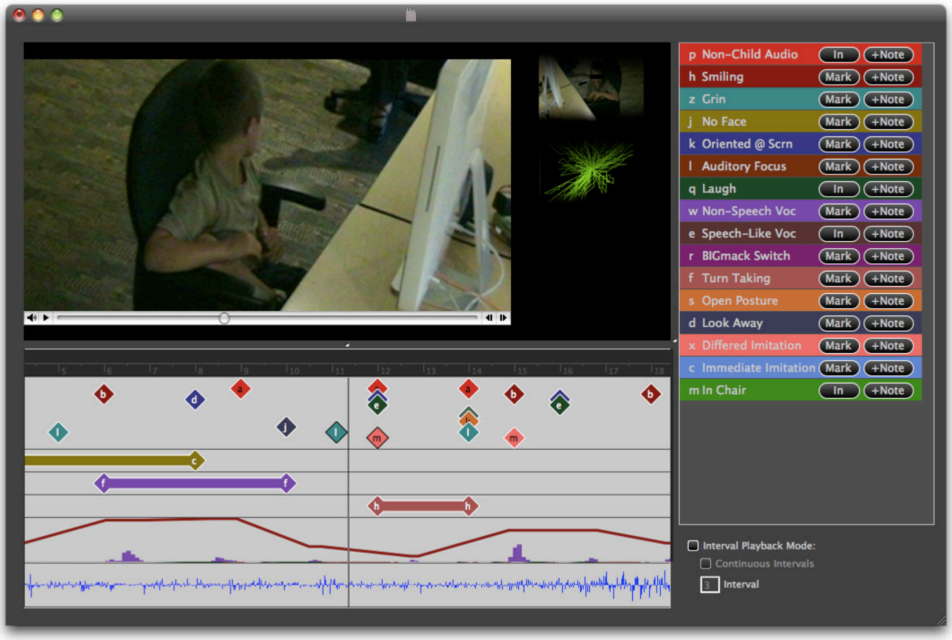


Figure 4.2.: The main component of VCode and VData, the VCode environment. It facilitates video annotation. *Image taken from [18]*

.

The analysis of the system showed the following results:

The system does meet the first requirement **R1 Timeline**. The timeline included in the system is the center of the main view. In VCode and VData, two classes of events exist in the timeline — momentary and ranged. Single diamonds represent momentary events, whereas two connected diamonds present a ranged event. Their position in the timeline can be manipulated by clicking, double-clicking, and dragging.

The system makes no live observations possible. Video material included in the system needs to be recorded beforehand. Additionally, the user needs to use an Apple device with a Mac OS 10.5.x to use VCode and VData. In conclusion, the system does not meet any requirements mentioned in the subgroup Cooperation Features.

R9 Categories is supported, because VCode and VData can define tracks. They consist of a name, hotkey, color, and a checkbox to know if an event of this track is ranged or momentary. The system displays a complete list of tracks in the main window. It also supports requirement **R11 Real Time Data**, as VCode and VData does include real time data, like sensor data. In contrast, the system only partially supports **R10 Note-Taking**. Tagging and commenting events is possible, but only in recorded data. It is also not possible to define text segments.

Requirement **R12 Play Afterward** is not met, because the system is designed for prerecorded observations. Statistics on the quality of the coding agreement between two users can be calculated by VData, if two VCode files are loaded into the system. This implies that data can be exported into VCode files, but not other formats for further analysis. Consequently, the system only partly meets requirement **R13 Statistics**.

VCode and VData does not support any requirements under the line of affordability. It does not comply with **R2 Video Annotation** because it is not able to let the user annotate directly on the video stream. Furthermore, it does not provide a speech-to-text functionality. The system is also only available on the device it is installed on, making teamwork achievable when everyone is in the same location. Hence, the system also does not need a communication channel for team members. The user interfaces of the system are set and are not configurable. The last requirement, **R14 Text of Previous Sessions** is not met, because it can only import previous made observations for the analysis.

4.3. ATLAS.ti

ATLAS.ti [21], like MAXQDA [22], is a widely used commercial software for qualitative data analysis. It consists of a home view to create a new project or to select an already existing one. The main view seen in Figure 4.3 displays the chosen, or newly created, project. It consists of different sections. On the top, it displays a ribbon similar to the Microsoft Word [23] design. The ribbon provides tools grouped by functionality for analysis purposes. Below, a navigation panel gives an overview of every type of data included in the project (e.g. documents, video files, codes) through an expandable tree structure. The main workspace of ATLAS.ti shows all opened files, focusing on one file at a time. The user can select other files by clicking on the according tab that is displayed. Users perform analysis on files by adding code or memos to specific parts.

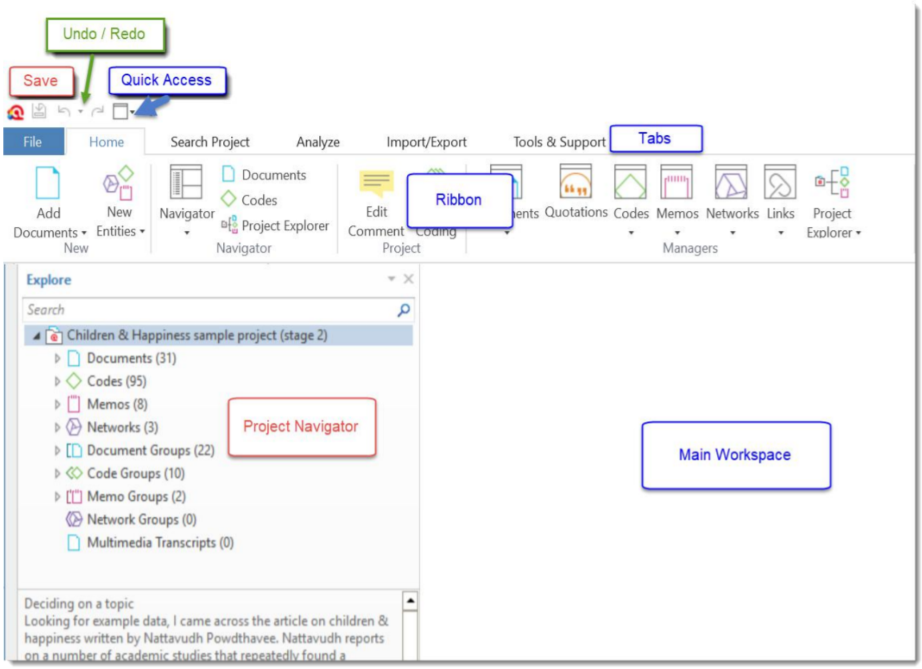


Figure 4.3.: The main view of ATLAS.ti, for analysis purposes. *Image taken from [21]*

.

ATLAS.ti displays video files with a vertical and horizontal timeline, meeting requirement **R1 Timeline**. Both timelines consist of an audio wave, playhead, and preview images. Quotations — marked sequences of the video — can be made by the user. This can be interpreted as tagging an event. A user can play the quotation by clicking on it. The system provides several options to annotate the data, however not video annotation.

A user can use the system cross-platform. ATLAS.ti facilitates teamwork, but only asynchronously and not via live streams. It provides no communication channel. In consequence, the system does meet requirement **R7 Dynamic Usage**, partly meets **R5 Remote Collaboration**, but not **R6 Communication between Users**.

ATLAS.ti includes codes, used to represent different categories. Every code consists of a name and comments about the code. Optionally, a list of codes is displayed while analyzing video material. With this in mind, the system meets requirement **R9 Categories**. The user can add comments to the quotation and can therefore take notes. However, the system can not provide previously made text segments; it does not meet requirement **R10 Note-Taking** entirely. Requirement **R11 Real Time Data** is met by the system through the support of many file formats, hence the ability to load supplementary data into a project.

This system is designed for prerecorded video or audio files, thus not meeting requirement **R12 Play Afterward**. Whereas requirement **R13 Statistics** is partially met, since it can calculate intercoder agreement but can only export codes as an Excel file.

ATLAS.ti supports one requirement under the line of affordability, **R14 Text of Previous Sessions**. This is due to the project structure of the system. All observations take place in one project, and consequently all codes and notes are accessible by the user throughout all observations. It does moreover allow a user to link transcripts with audio or video files, but that does not satisfy requirement **R4 Speech to Text**. Every device has a suitable user interface, but they are only minimally alterable. In contrast, the system does not allow voice input.

4.4. Vosaic Connect

Vosaic Connect [24] is a commercial system for performance discovery — the process of observing, identifying, and aggregating indicators for performance issues. The system is cloud-based and therefore does not need to be downloaded to be used. To use it, the user has to create an account on the Vosaic Connects website. By logging in, the user can be assigned different user roles and can start analyzing videos on the main view, as seen in Figure 4.4. The main view shows the user's video content to analyze on the left-hand side, called the video player. Beneath the video player, a timeline shows depicted moments — a behavior or action the user wants to identify. On the right-hand side, a list shows all the defined moments. This section is called the moment sidebar. It contains two tabs between which the user can choose. One tab displays a list of defined moments, whereas the other presents information about a specific moment.

The analysis of Vosaic Connect revealed the following results:

One of the main parts of the Vosaic Connect main view is a timeline. It consists of vertical rows, each assigned to a user-defined behavior. Marked behaviors show up in the corresponding row of the timeline. A user can play the marked behavior in the video by clicking on it in the timeline. Consequently, the system does meet the first requirement **R1 Timeline**. Nonetheless, a user is not able to mark behaviors through direct video annotation.

Vosaic Connect allows the user to directly collaborate with other team members and share findings remotely on uploaded video material. However, a live stream can only be initially seen by one user. Therefore, Vosaic Connect only partially supports requirement **R5 Remote Collaboration**. Like the previous systems, Vosaic offers users no communication channel to contact team members within the system. Hence, not supporting requirement **R6 Communication between Users**. The system is web-based and available as an Android and iOS app, making it possible to use with every device on hand. Consequently, meeting requirement **R7 Dynamic Usage**.

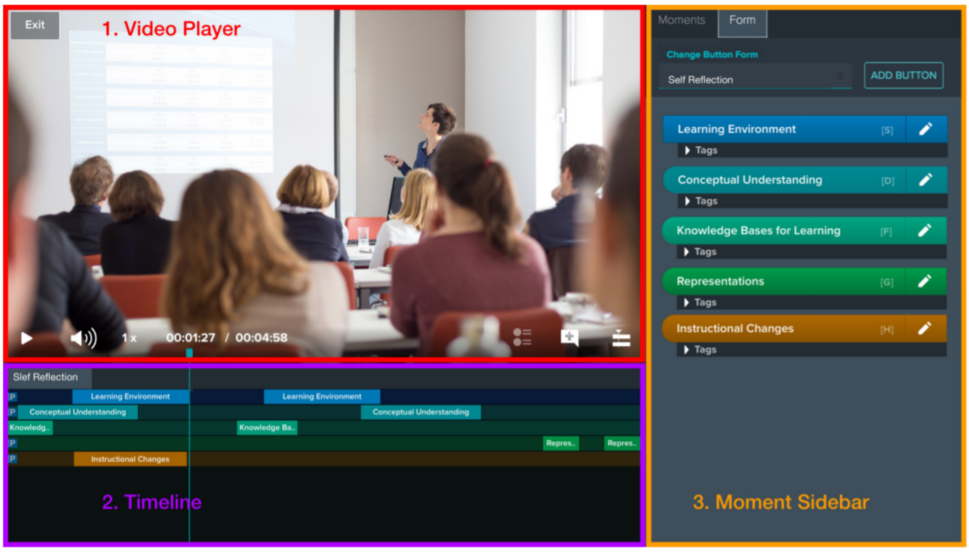


Figure 4.4.: The main view of Vosaic Connect shows a video player, a timeline, and a moment sidebar. *Image taken from [24].*

In Vosaic Connect, a user can define up to 25 moments. They can be considered as categories of a behavior. The system meets **R9 Categories** partially, as the number of moments is limited. It meets the requirement **R10 Note-Taking** partially as well. A feature that the system does not provide is the possibility to include previously defined text segments. Tagging a behavior is done by clicking on a moment in the main view. It will appear as a mark on the timeline. A user can edit its position and length on the timeline. When clicked on a mark, the moment sidebar displays all marked moments. A user can add comments to every mark. Only requirement **R11 Real Time Data** of the system is not met, due to the fact that only one video file can be included.

The system automatically provides a record of an ended live stream. The system can export report files in a .csv or .pdf format. Exported files contain information like timestamps of every moment, the comments added to them, and more. Additionally, Vosaic Connect can calculate the intercoder reliability. In conclusion, the system only fully supports requirement **R12 Play Afterward** and partially meets **R13 Statistics**.

Vosaic Connect does not support any requirements under the line of affordability. It does not allow voice input for comments or subtitles. **R8 Configurations** is not met, since the user interface itself is not configurable. Finally, it does not fulfill **R14 Text of Previous Sessions**, as the system only supports the import of previous defined moments.

4.5. Videostrates

Videostrates [25] is a toolkit for creating collaborative video editing tools in real-time. It is an approach for information strates based video editing. Videostrates are described as "an alternative to the traditional model of applications and documents" [25]. Two key components of the toolkit are vCompositor and vStreamer. vStreamers' main responsibility is to spawn browser instances, catching and streaming the graphic output to all clients. vCompositor is responsible for the playback of composited video, animated SVG elements, animated DOM elements, custom elements, and other transcluded videostrates.

A relevant example for the usage of videostrates is broadcasting live streams and editing them, as seen in Figure 4.5. The same videostrate can be opened with different devices with the possibility to share video data between them. A stream studio videostrate has the possibility for peer-to-peer video streaming and capturing video using modern browser support and WebRTC. The URL of the videostrate can be shared and used from other devices for additional video content, also shown on the laptop. A user can add annotations to a stream by drawing with a pen — this requires a touchscreen.

Figure 4.5.: An example of using videostrates to broadcast live streams to any device. *Image taken from [25].*

An analysis of Videostrates in regard of the requirements of Section 3.2 revealed following results:

An example of Videostrates shows a video editor using a minimalistic editing timeline, that does not support all functionalities stated in requirement **R1 Timeline**. These functionalities must be added from a user independently, thus only partially supporting the requirement. Regarding video annotation, the toolkit enables a user to draw on the video feed using a touchscreen.

Based on the example of broadcasting live streams and editing them, Videostrates demonstrates the possibility of working in remote collaboration with different input devices. Additionally, Videostrates can install packages, like a chat package. Consequently, Videostrates supports the requirements **R5 Remote Collaboration**, **R6 Communication between Users**, and **R7 Dynamic Usage**.

Requirements **R9 Categories** and **R10 Note-Taking** are features that can be added to the functionality of a videostrate, when a user independently programs it. Therefore, these requirements are considered to be only partially supported. Since video substrate contains information unrelated to the video making it possible to add data, like screen recordings of the application, requirement **R11 Real Time Data** is supported.

In Videostrates, vStreamer can record a live stream to a file for later editing purposes. However, a video substrate does not inherently contain the possibility to calculate any statistic values for quality analysis purposes. A user has to implement it themselves first to use it. Hence, it is considered that requirement **R13 Statistics** is only partially supported. **R12 Play Afterward** is met.

Videostrates meets three out of four requirements under the line of affordability. Video substrates are programmable for a user, so they can add new functionality, therefore meeting **R8 Configurations**. Additionally, this potentially facilitates the use of a microphone, fulfilling requirement **R3 Voice Input**. It also contains information unrelated to the video, making it possible to add data, like notes from previous study sessions. It is possible to export video files from a videostrate, therefore it is likely that the system can also export other information, meeting **R14 Text of Previous Sessions**. In contrast, Videostrates demonstrates only a subtitle editor. This is, however, not enough to support **R4 Speech to Text**.

4.6. Summary

As seen in table 4.1, almost every existing system introduced in this chapter only fulfills a few requirements proposed in Chapter 3. BORIS and ATLAS.ti only support four requirements. Vosaic Connect fulfills three requirements, and VCode and VData fulfills just two. In contrast, Videostrates fulfills most requirements, simply four requirements are only partially fulfilled, and one is not supported at all. Therefore, based on the results of the previous analysis, it is not appropriate to consider the existing systems, e.g. BORIS, as a suitable system for researchers of the HCI group. Concerning the requirements, Videostrates would be a suitable candidate. However, Videostrates is a toolkit only provides to implement needed functionalities. It is additionally only integrated into the ecosystem of Webstrates [25]. Since the goal is to integrate the system into an ecosystem for the HCI group of the University of Konstanz, a conjunction of the HCI ecosystem and the Videostrates ecosystem leads to an additional overhead, e.g. running costs for a needed Videostrate server. Consequently, a new system needs to be developed for integration into the HCI ecosystem.

	BORIS (Section 4.1)	**VCode and VData** (Section 4.2)	**ATLAS.ti** (Section 4.3)	**Vosaic Connect** (Section 4.4)	**Videostrates** (Section 4.5)
R1: Timeline	🔴	🟢	🟡	🟢	🟡
R2: Video Annotation	🔴	🔴	🔴	🔴	🟢
R5: Remote Collaboration	🔴	🔴	🟡	🟡	🟢
R6: Communication between Users	🔴	🔴	🔴	🔴	🟢
R7: Dynamic Usage	🟢	🔴	🟢	🟢	🟢
R9: Categories	🟢	🟢	🟢	🟡	🟡
R10: Note-Taking	🔴	🟡	🟡	🟡	🟡
R11: Real Time Data	🟢	🟢	🟢	🔴	🟢
R12: Play Afterward	🔴	🔴	🔴	🟢	🟢
R13: Statistics	🟢	🟡	🟡	🟡	🟡
Line of Affordability					
R3: Voice Input	🔴	🔴	🔴	🔴	🟢
R4: Speech to Text	🔴	🔴	🔴	🔴	🔴
R8: Configurations	🔴	🔴	🔴	🔴	🟢
R14: Text of Previous Sessions	🔴	🔴	🟢	🔴	🟢

Table 4.1.: Overview of all systems in relation to the requirements. A red dot (🔴) indicates, that the requirement is not met. A yellow dot (🟡) indicates that the requirement is met partially. A green dot (🟢) indicates that the requirement is met. *This is an updated version of a table taken from [2].*

5. Prototype: "CoCo — Collaborative Coding of Critical Incidents"

This chapter explains the design process and final implementation of the prototype CoCo. Coco is a collaborative web application for the remote detection and coding of critical incidents. The application has been designed through multiple design iterations, described in Section 5.1. The final implemented design is presented in Section 5.2 and follows the structure of the previously introduced requirements in Chapter 3. The contents of this chapter were discussed in more detail in the preceding project report [6] and have been summarized and updated appropriately.

5.1. Design Process

CoCo is realized as a web application. To design this application, a design process was established. Buxton [26] states that the purpose of design is to establish a trajectory that is later refined in iterative steps in the usability engineering process. To determine the optimal concept, the design process involves exploring and comparing various alternatives of design concepts. In contrast, the usability engineering process improves the design in incremental steps to get the design right. The whole design process is shown in Figure 5.1. First, design concepts were made as paper sketches and evaluated through internal testing. The next step was to refine the best-evaluated concept iteratively. For this purpose, digital sketches were created, using Figma's [27] web-based editing tool.

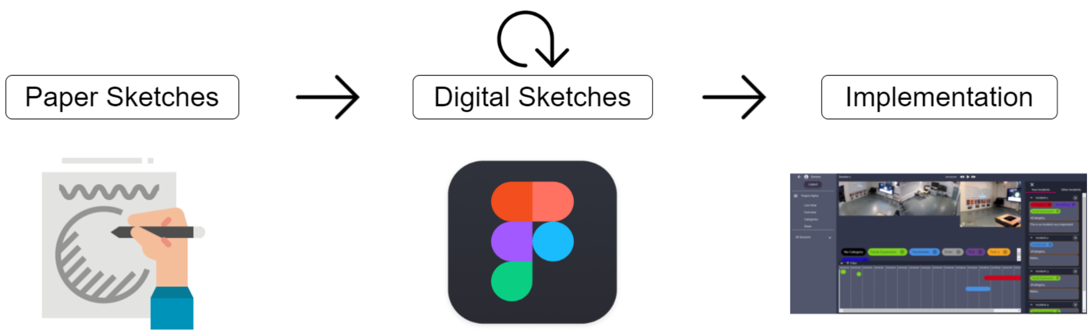


Figure 5.1.: The design process of CoCo. Starting with the exploration of design alternatives with paper sketches. Followed by iterative design refinement with digital sketches and the final implementation. *Icon taken from Flaticon [28], Figma Logo taken from [27].*

The interaction with CoCo starts with the login, as illustrated in the workflow diagram in Figure 5.2. A logged-in user can choose to create a new project or select an existing one. This leads the user to the main view of CoCo. Here, CoCo supports many features and functionalities for the user. If the user wishes to observe a live study session, they start the recording with other team members. During the recording, they can add and edit critical incidents. They can additionally communicate with other team members if necessary, or annotate video streams. If a user is not observing a live session, they can choose to perform different administrative tasks. This includes managing the team by adding or removing team members. Additionally, the user can administer the categories that can be associated with a critical incident. CoCo also supports reviewing previously recorded live study sessions. Meaning, users can re-watch the recorded data and add further information about detected incidents. This is achieved by adding categories, editing their length, and adding or editing notes. Besides, the user is also always free to log out of CoCo anytime. A more detailed description of the design process and the developed sketches can be found in the project report [6].

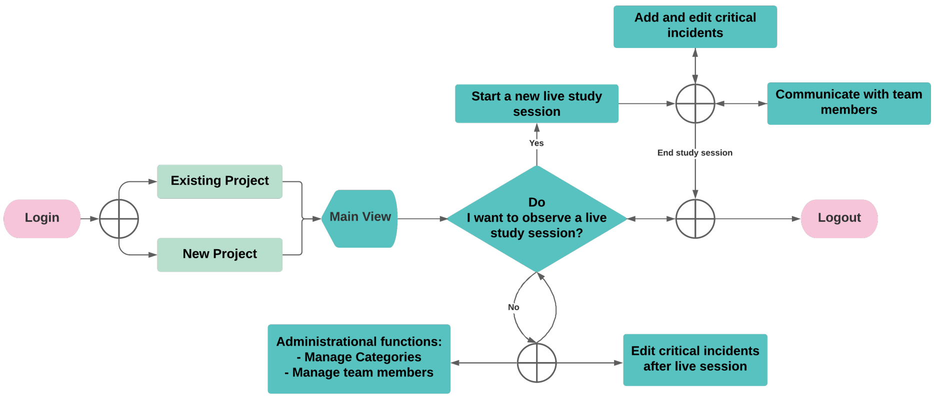


Figure 5.2.: Workflow diagram of the user interacting with the prototype. *This workflow was updated from the preceding project report [6]*

The final implementation of CoCo follows the previously described workflow. It consists of a login and allows the user to choose between an existing project and the creation of a new project. Figure 5.3 illustrates the main view of CoCo. The side menu on the left displays the name of the logged-in user and the selected project. It additionally allows the user to navigate through the different sections of the prototype: *Live View*, *Overview*, *Categories*, *Share*, and *All Sessions*. The following sections present a detailed description of all components and features.

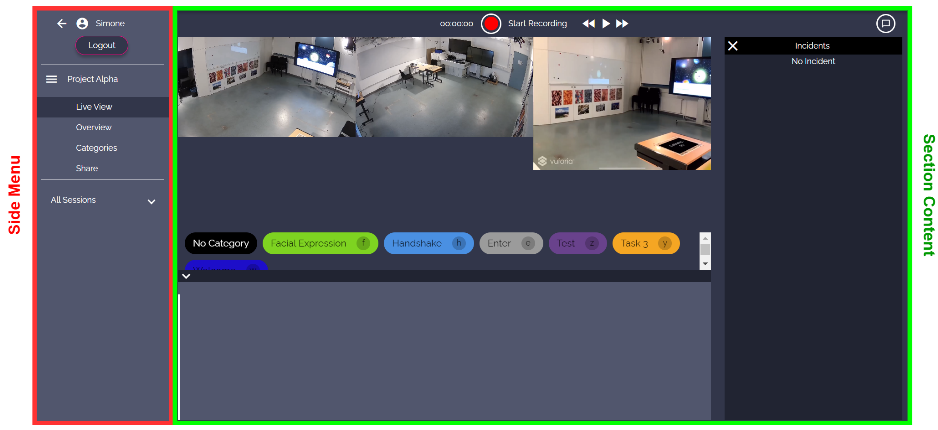


Figure 5.3.: Main view of CoCo. A side menu allows the user to select different sections (e.g. *Live View*, or *Overview*). The sections handle live critical incident detection, basic information about a project, categories, team members, and recorded study sessions.

5.2. Implementation

The structure of the final design descriptions of the implemented prototype CoCo follows the grouping of the requirements named in Section 3.2. Requirements under the line of affordability are not part of the grouping, as no design for them was created. A more detailed description and technical information about the prototype can be found in the project report [6].

5.2.1. R1: Timeline

The final implementation integrates two timelines. Both look similar to common video editing timelines (e.g. Adobe Premiere Pro [29]) and display critical incidents as elements in a spatial linear manner. The elements representing critical incidents belong to one of two types: momentary and ranged. Circles represent momentary elements, whereas ranged elements look like rectangles with rounded edges (shown in Figure 5.4). Every element displays the color of the first associated category. When a critical incident associates no defined category, it is black. Moreover, both timelines in CoCo are interactive. A user can interact with the time indicator through dragging to select a specific point of time for the study session. Likewise, timeline elements for critical incidents have multiple interactive features. For example, with a double click on an element, time jumps to the starting point of the critical incident. These elements are also draggable. A user can pick the element and re-position it through the dragging movement, changing the starting point of the critical incident. Additionally, a user can change the length of critical incident elements through a resize function. Grabbing the right end of the element allows the user to trim or elongate it.

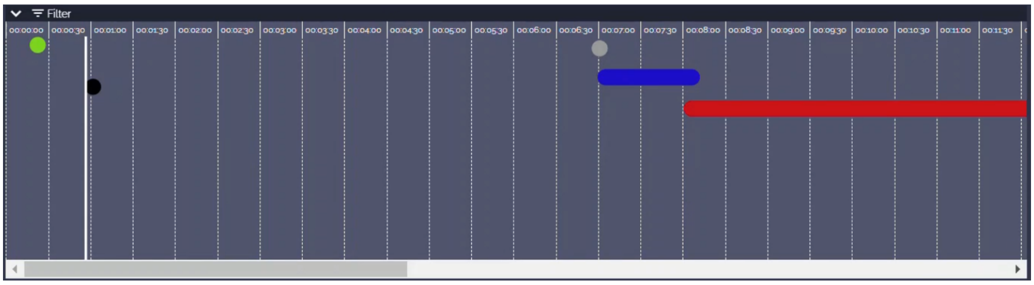


Figure 5.4.: Timelines in CoCo. They display critical incidents as elements. Circles are momentary elements, rectangles are ranged. *Taken from [6].*

Figure 5.5.: The filter function for the timelines of former recorded study sessions. Users can filter after team member and/ or category. *Taken from [6].*

Timelines for previously recorded study sessions additionally include a filter function. A user activates the filter by clicking the filter icon located at the top left of the timeline, as shown in Figure 8.5. The filter function enables critical incident filtering through two options. The first option is to add critical incidents to the timeline from other team members. A drop-down menu lists all team members of the project. A user can select one or more list entries. By clicking the “Apply Filter” button, all critical incidents of the chosen team members are also displayed within the timeline. Additionally to this option, a user can also hide their critical incident element in the timeline. The second filter option is to filter after category. With another drop-menu, a user can select one or more categories. By clicking the “Add Filter” button, all critical incidents are filtered after the selected categories. A combination of both filter options is possible.

5.2.2. R2: Video Annotation

As the user starts the recording of a live study session, CoCo displays every previously selected resource. In case one or more resources are video streams (e.g. network cameras, screen recordings), the user is capable of drawing directly onto the video stream by dragging the mouse over the video stream. The traced path of the mouse is then displayed as red lines on top of the video stream, as seen in Figure 5.6. If the user accesses CoCo on a mobile device (e.g. tablet), they can achieve the same result by drawing over the video stream. Video annotations stay on the video stream permanently.

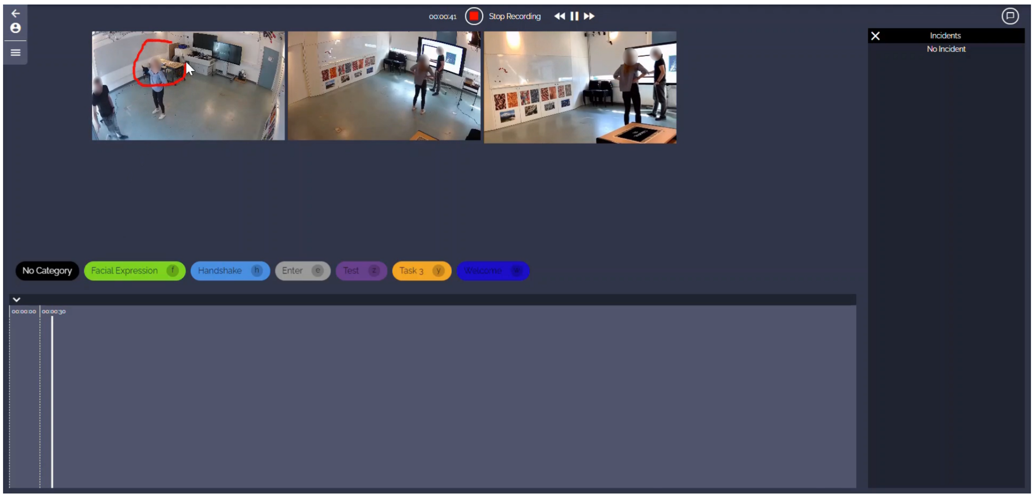


Figure 5.6.: A user can draw on video streams for annotation purposes.

5.2.3. R5: Remote Collaboration

Remote Collaboration contains two aspects: team management and concurrent observation of live streams in different locations.
A user can establish and manage a team in the “Share” section of CoCo, shown in Figure 5.7. Here, the user has an overview of all team members provided by a list view on the left. Each list view entry represents one team member. They are added to the project when a user selects the member through a drop-down menu. The list displays every added team member afterward. In case a team member leaves a project, a user can remove them from the project with the delete button on the right of the corresponding list entry. If an unregistered person needs access to the project, a user can provide an invitation link to them.

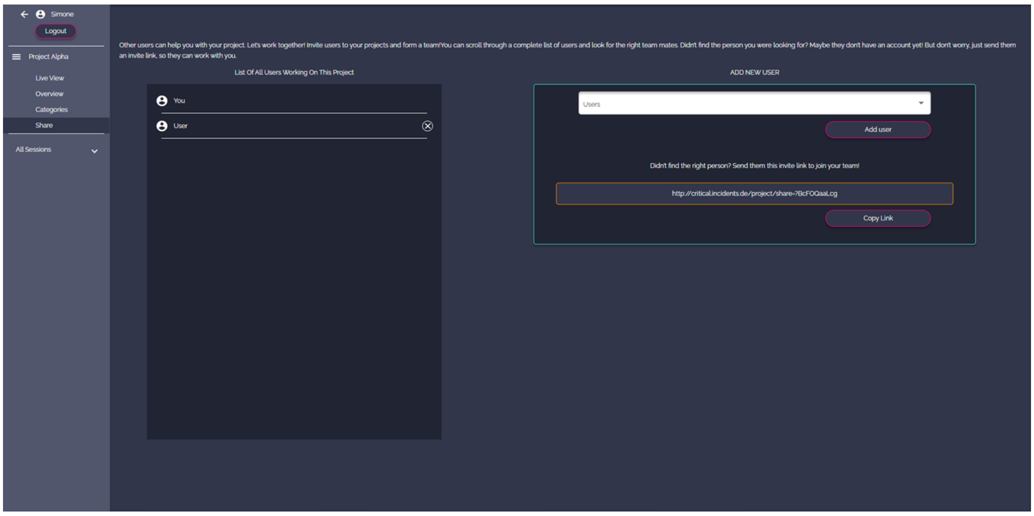


Figure 5.7.: "Share" section of CoCo. It facilitates the team management for projects. *Taken from [6].*

To observe a study session concurrently via live streams, team members can individually go to the "Live Session" section of CoCo. This section consists of six parts, seen in Figure 5.8. One is a video view; it displays all resources selected for a project. Below the video view, a chip list is displaying categories. Section 5.2.6 gives a detailed description of categories and the corresponding chip list. Below the chip list, a timeline displays found critical incidents. Next to it, a container lists detailed information of tagged critical incidents. Another part of the "Live Session" section is the chat on the right. It expands and collapses when a user clicks the chat bubble icon. A toolbar at the top is also a part of the "Live Session" section. It contains the chat bubble icon, basic video controls, and a recording button. A recording of the live streams for all team members starts when a user presses the recording button. While recording the live streams, it is possible to detect critical incidents and communicate with others.

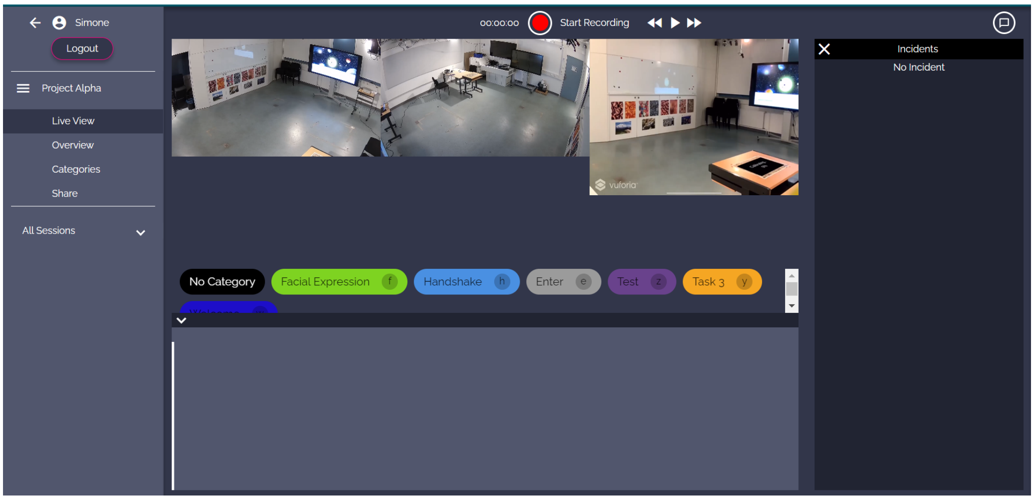


Figure 5.8.: "Live View" section of CoCo. Here, users can concurrently observe and record live streams of study sessions. They can detect and tag incident and chat with other team members. *Taken from [6].*

5.2.4. R6: Communication between Users

A user communicates and exchanges information with others by using the chat function (see Figure 5.9) integrated into the “Live Session” section of CoCo, described in 5.2.3. It utilizes a common design, seen for example in *Overleaf, Online LaTeX Editor* [30]. Messages are sent automatically to all team members present in a live study with a notification. A checkbox indicates whether a message is critical or not critical. A notification for critical messages is more prominent to indicate urgency (seen in Figure 5.10). It displays the sender and the text of the message. To draw attention, the border of the notification blinks pink.

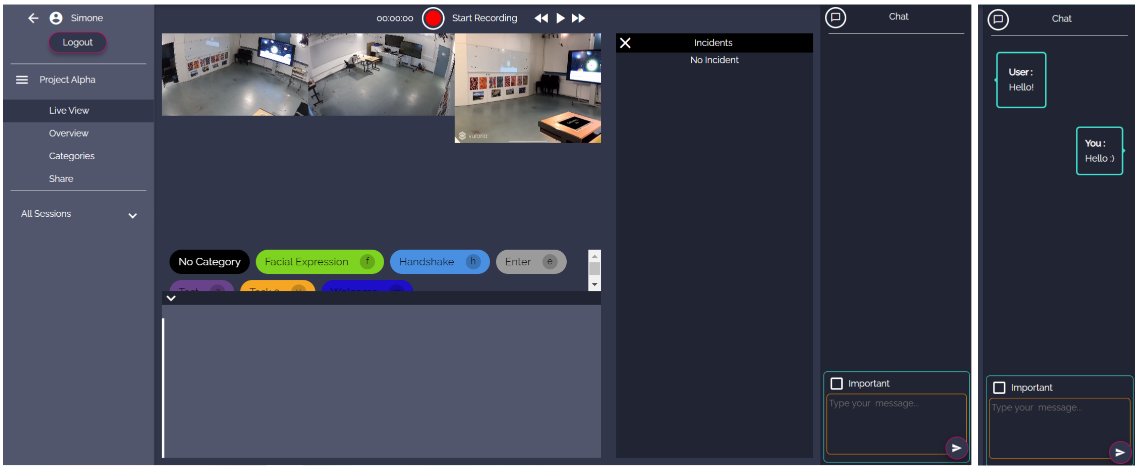


Figure 5.9.: Integrated chat function for a live study session (expanded, on the left) and an example of exchanged messages (right). *Taken from [6].*

Figure 5.10.: Receivers of important messages get a snackbar [31] displayed on the screen as a notification.

5.2.5. R7: Dynamic Usage

CoCo is a cross-platform web application and supports multiple input devices like mouse and keyboard, as well as touchscreens. All user inputs are mapped appropriately depending on the device’s capabilities. This enables users to use the prototype on devices such as desktop PCs, tablets or mobile phones.

5.2.6. R9: Categories

Categories in CoCo are used to easily and intuitively distinguish between critical incidents. Differentiating between critical incidents and their meaning can vary greatly from project to project, as they tackle different topics for evaluation. Therefore, users must be able to define categories fitting for a project. The “Categories” section in CoCo helps to define those categories. It is seen in Figure 5.11 and displays the entire list of categories on the left. A category consists of a color, name, and an associated hotkey. A list entry will show all of those attributes. Additionally, the entry displays a delete button on the right. Next to the list is a form to define new categories. The form provides input fields for the name and hotkey. It also includes a button to trigger a color picker. This allows the user to choose from various colors for the categories. The user can add a newly defined category by pressing the “Add Category” button on the bottom right of the form. Does a user wish to use categories for other projects, they can export and import category files. Furthermore, a chip list displays defined categories, as seen in Figure 5.12 when a user is observing a live study session, as well as revisiting a recorded study session. The chip list displays all categories with hotkeys. It also provides one chip element for critical incidents with no associated category, named “No Category”. A user can interact with chip elements by clicking on them. This triggers the tagging of a critical incident. The timeline automatically displays it, as shown in Section 5.2.1. The timeline updates the length of the critical incident element as long as the user presses the chip element. This accurately displays the length of the critical incident. Alternatively, a user can press a hotkey of a category to tag a critical incident.

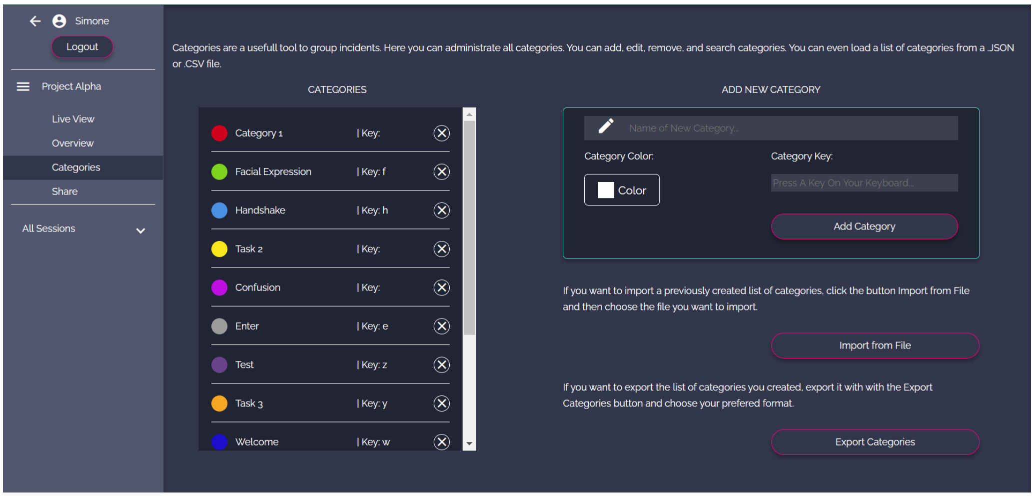


Figure 5.11.: The “Category” section of CoCo for the management of categories within a project. *Taken from [6].*

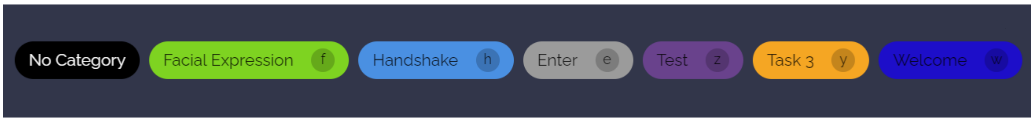


Figure 5.12.: The chip list integrated into CoCo to display categories. *Taken from [6].*

5.2.7. R10: Note-Taking

Note-Taking, includes several functionalities in CoCo. One of them is the tagging of critical incidents. For this, CoCo provides two methods. The first method is to press a chip element. The other is to press an associated hotkey of a category. Section 5.2.6 explains both methods in more detail. If no category can be associated with the critical incident right away, the user can press the according chip or the hotkey “i”.

Another functionality of *note-taking* is the possibility for a user to edit information about a critical incident. CoCo displays containers, as shown in Figure 5.13. They list tagged critical incidents as list entries. Each list entry consists of a category section and a notes section. The category section displays associated categories as small chip elements. The color of them reflects the color of the category. Apart from that, the element also contains the category name and a delete button. Inspired by *Instagram* [32] and *Todoist* [33], a user can add categories through hashtags of category names. The hashtag triggers a list of all defined category names. A user can select the wanted category from the list and filter the list through an autocomplete functionality. It is possible to add an undefined category by entering a new hashtag. This automatically adds a new category named after the hashtag with a random corresponding color to the project. The note section is a simple text area. A user can directly enter their notes to it for further information.

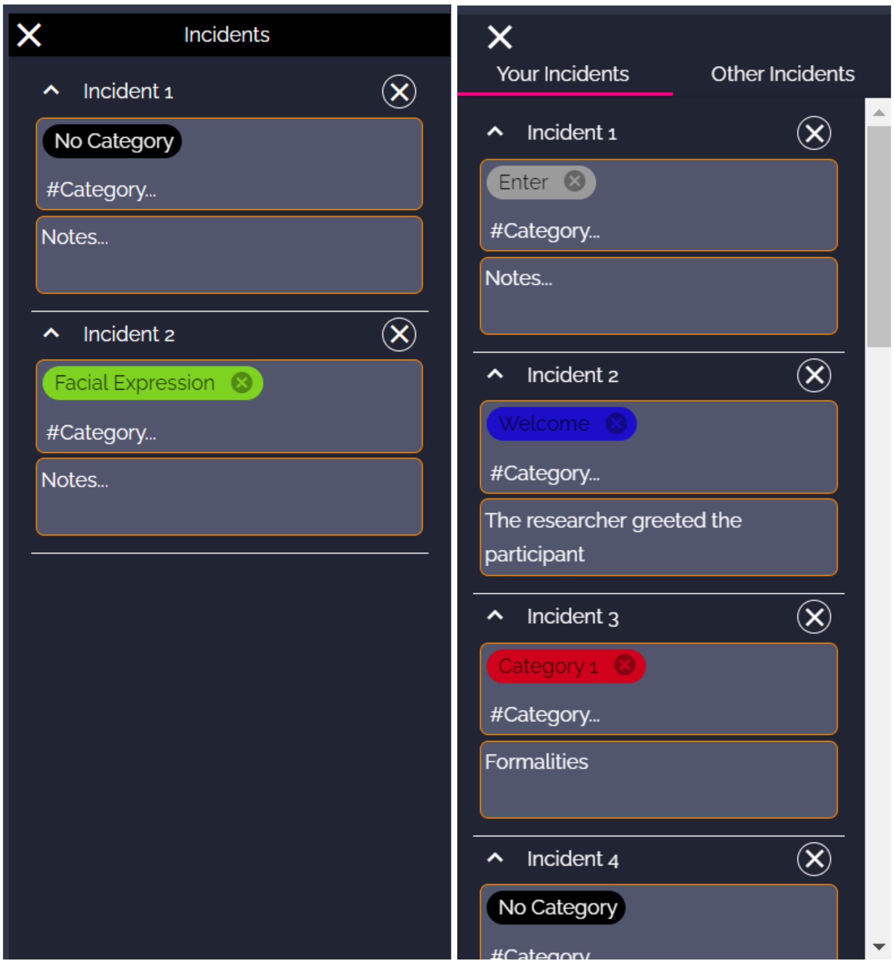


Figure 5.13.: Containers listing all tagged incident, displaying further information and notes. *Taken from [6].*

5.2.8. R11: Real Time Data

To create a new project, the user must select all needed video streams and additional real time data (e.g. screen recordings, eye tracking). The web application allows this by providing a drop-down menu listing all available types of data. By selecting one type, all resources of this type will be displayed automatically on the right side of the screen, as seen in Figure 5.14. There, the user then selects individual resources by clicking on them. Additionally to the resource selection, the user can name the project and include a description of it.

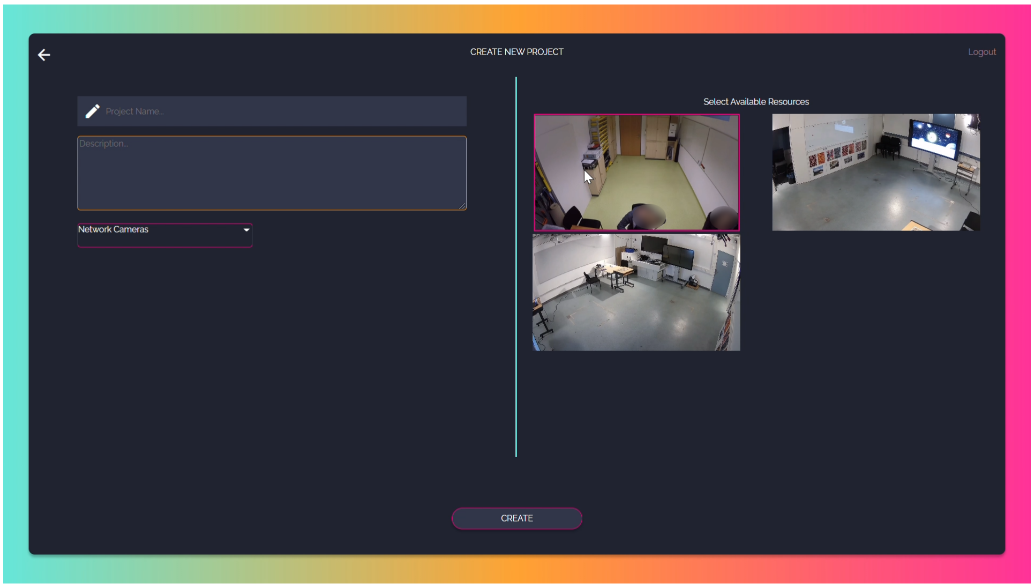


Figure 5.14.: Creation of a new project. Resources with the type “Network Camera” are displayed and selectable.

5.2.9. R12: Play Afterward

The recorded data is available to the team members immediately after the live session ended. A user can revisit the collected data anytime by selecting it under the “All Sessions” list provided through a side menu in CoCo. There, the user can find all recorded live sessions of a project. Selecting a session loads the data into the view, as shown in Figure 5.15. The view is similar to the “Live Session” section of CoCo. Likewise, it contains a video view, chip list, timeline, a container for critical incident information, and a toolbar. Additionally, it includes a filter function. This facilitates the user to filter through the collected data.

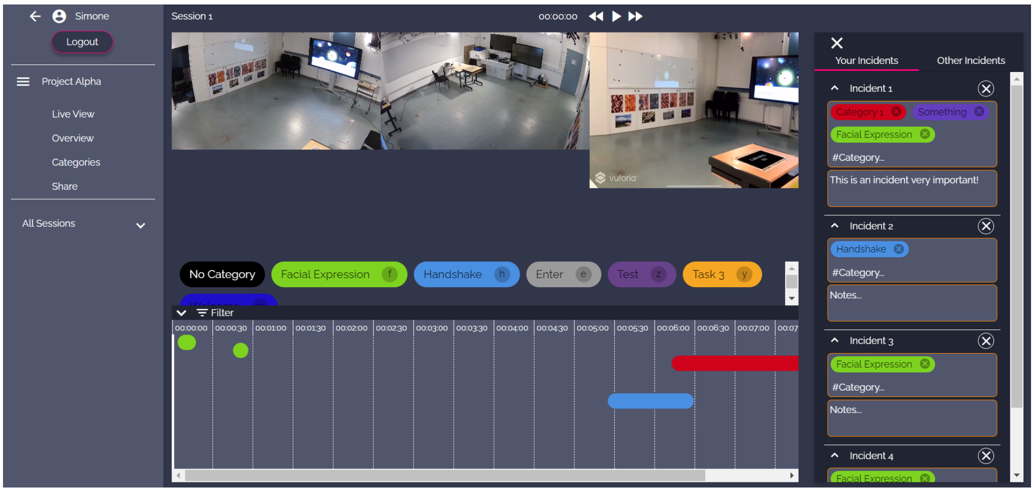


Figure 5.15.: Recorded live sessions can be revisited by users for editing and analysis purposes. *Taken from [6].*

5.2.10. R13: Statistics

The "Overview" section of CoCo, shown in Figure 5.16, provides the user with basic statistics about the project. It displays the description of the project at the top. Below, it presents the intercoder reliability of selected users. A user can calculate the intercoder reliability of an arbitrary group of team members within a project by choosing the wanted team members through the list of team members next to the intercoder reliability. For additional statistic value, the section visualizes the number of incidents per category through a bar chart. At the bottom of the section, a button "Export Data" facilitates the export of the collected data.

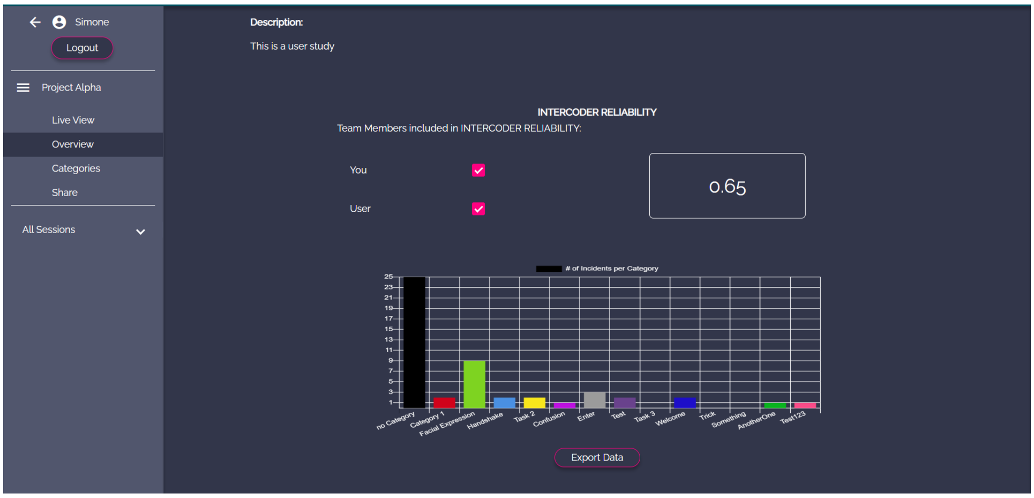


Figure 5.16.: The "Overview" section of CoCo, displaying the intercoder reliability and other statistics of the project to a user. *Taken from [6].*

5.3. Development Framework and Libraries

For the development and implementation of CoCo, Angular 10 [34], a prominent framework for the development of web applications, was used. It was combined with libraries and plugins such as Angular Material [35], socket.IO [36], JSON Server [37], ngx-color [38], chart.js [39], Lodash [40], and FileSave [41]. Through the combination of Express [42] and socket.IO a communication channel between users was established. The addition of interact.js [43] enables easy and intuitive interaction techniques for elements in the timelines (e.g drag and resize).

6. Usability Study

The implemented prototype explained in the previous chapter was evaluated in a usability study. The following sections describe the research questions, the participants of the study, the apparatus, the general procedure of the study, and the analysis methodology used. All quotes in this chapter were translated from German.

6.1. Research Questions

A usability study with four participants was performed to evaluate CoCo. Participants of this study were required to be part of the HCI group of the University of Konstanz, as well as being a Ph.D. student, a domain expert, or both. The usability study was guided by the following research questions:

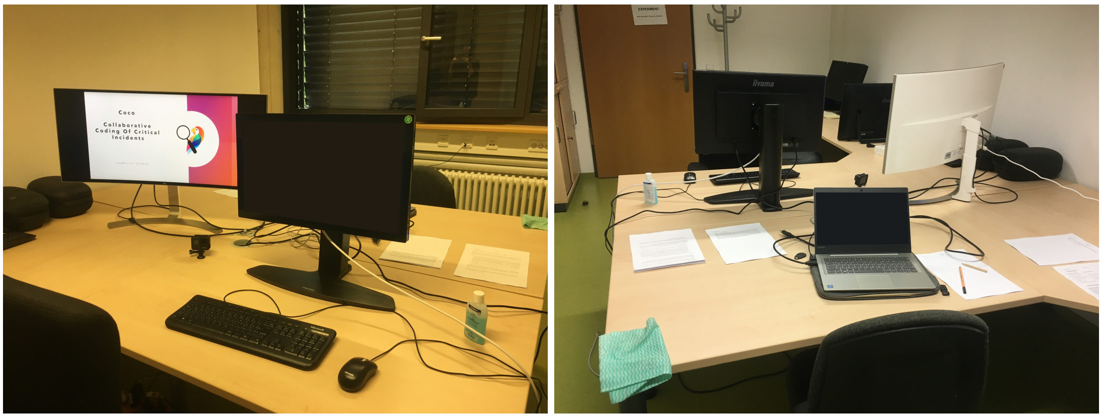
> **RQ1 System Usability:** How well does CoCo perform as a tool for collaborative critical incident detection?
>
> **RQ2 User Experience:** Do users like to use the system?
>
> **RQ3 Improvement:** Does CoCo improve the critical incident detection in the eyes of the user?

6.2. Participants

The study was performed with four participants (everyone identified as male) between 26-35 ($M = 29, SD = 3.67$), with three research assistants and one student assistant of the HCI group of the University of Konstanz. Participants were asked to rate their familiarity with the detection of critical incidents on a binary scale ("I am very familiar with the detection of critical incidents" and "I am not familiar with the detection of critical incidents"). Three of four participants stated that they were very familiar, and one participant was not. An open question invited the participants to comment on possible problems with the critical incident detection process. Two out of four participants chose not to answer this question. One participant stated that there are "a lot of problems", including "many [researchers] lack experience", missing camera perspectives, and the usage of "tools [...] not designed for [critical incident detection]". Another participant reported a problem with the "delimitation of [critical incidents] in longer processes". They were furthermore asked to rate their opinion of the possibility to improve the critical detection process on a scale from 1 ("The process can be improved") to 5 ("The process cannot be improved"). Every participant reported a rating of 2, meaning that they believe the process can be improved. Lastly, the participants were asked to rate their preference for collaboration for the conduction of a study on a scale from 1 ("Not important") to 5 ("Very important"). The participants reported a preference from 2-4 ($M = 3, SD = 0.81$).

6.3. Apparatus

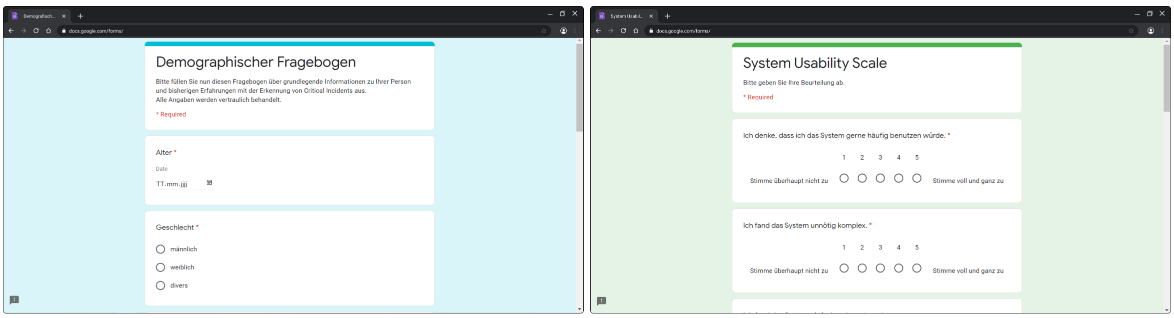
The study was conducted in a laboratory room (see Figure 6.1). It included a display for information about CoCo and enough space on the table to fill out documents. It also provided the participant with an own workspace (i.e. their own computer with the CoCo prototype set up). The screen of the computer was recorded during the study to log all user interactions with CoCo. The study setup also included a microphone on the table, which recorded audio data during the study. Additionally, a camera was placed to record the participants actions.



(a) **(b)**

Figure 6.1.: The room used for conducting the study. (a) The workspace for the participant. Including a computer to fill out questionnaires and performing the study tasks. A display next to the computer shows a introductory presentation. (b) workspace of the study facilitator. Includes space for study documents.

As shown in Figure 6.2, the participants used the internet browser to access CoCo to perform the tasks and also to fill out questionnaires. For this, both questionnaires were implemented with Google Forms [44]. Digitizing the questionnaires was convenient for the evaluation and participants. Participants simply filled out the required questions via mouse clicks or short answers. Submitted questionnaires were immediately available as digital files for evaluation.



(a) **(b)**

Figure 6.2.: Participants filled out different questionnaires directly in the internet browser. The questionnaires where specially created for this. (a) Part of the demographic questionnaire. (b) Part of the System Usability Scale [45].

6.4. Procedure

The participants were first greeted and asked to take a seat at their dedicated workspace and provided with several informational documents. These informed the participant about the general procedure of the study and the consent form (see Appendix A). After the participants filled out the consent form, they were asked to answer an online questionnaire for demographic data. Once the participants completed the questionnaire, they were given an introductory presentation about the concept of CoCo, explaining step-by-step its features and interaction possibilities, as well as the key concept of critical incidents. Next, the participants had to solve seven artificial tasks with CoCo, where the next task was given to the participant when the previous one was completed. The set of tasks were ordered after a possible workflow within CoCo, each task introducing a different feature of the prototype. All participants were encouraged to perform the tasks on their own, but could ask for help at any given time. They were also asked to use the think-aloud technique (i.e. express their thoughts aloud) [1] while completing the tasks. During the tasks, critical incidents were detected and noted to gather qualitative data. Once all tasks have been completed, all participants were asked to fill out another online questionnaire, which was a standard System Usability Scale (SUS) questionnaire [45]. A semi-structured interview was conducted afterward to gather further qualitative data (see Appendix A). Lastly, the participants were offered baked goods as a thank you for participating. In total, a study lasted between 50-60 minutes, with a task completion time of approximately 25 minutes and an interview span of about 16 minutes. No participant aborted any tasks or did not participate in the interview.

6.5. Data Analysis Methodology

The following sections describe the applied analysis methods and techniques for the data analysis. Mainly qualitative data was collected throughout the usability study, therefore the main focus was directed towards this data type. The techniques and analysis process are described in Section 6.5.1. The analysis process of the collected quantitative data is described afterward.

6.5.1. Qualitative Data Analysis Process

The qualitative data of the semi-structured interviews was thematically analyzed with the approach of Braun and Clarke [46]. This approach of analysis was chosen, because it is widely used for data analysis in the field of HCI [47] and it helps to deduce new insights and concepts from the data [48]. The thematic analysis followed a semantic orientation, as it should reflect the content of the interviews closely to find themes and patterns concerning the research questions and prototype. For the thematic analysis, an inductive and semantic approach was chosen, as the interview content should be reflected by the themes. While performing the analysis, the transcripts of the interviews were read twice by the study facilitator. During the two readings, initial notes were taken, followed by initial coding. The codes were made as descriptive of the participants' thoughts and statements as possible. A code was associated to phrases, lines, sentences, or words. At the end, 49 codes were formed from important paragraphs of the interview transcripts. Codes included in the analysis appeared in at least two different interviews. The codes were subsequently used to identify themes derived out of the data. Braun and Clarke [46] define themes as important aspects of the data concerning the given research question. Identifying themes in this work was an iterative process. First, initial themes were formed throughout the codes. Afterward, every initial theme was revised into more specific themes or sub-themes, depending on the content. After revising the themes, thematic headings were given to them. At the end, seven major themes were identified. One theme includes five sub-themes.

During the time participants performed tasks on CoCo, the study facilitator collected qualitative data through critical incident detection. The facilitator firstly recorded the detected critical incidents through pen and paper (i.e. writing down a timestamp and information about the critical incident), capturing data as stated in *The UX Book* [1]. After concluding the study, a project in CoCo was created. Recorded material (e.g. video recording, screen recording and audio recordings) were imported into the project and sorted into study sessions. The facilitator then revisited every study session again and revised the discovered critical incidents to gather more information. After revisiting all sessions of the performed usability study, all critical incidents were exported as a .csv file. These are discussed in more detail in the following section. This work only includes critical incidents that occurred at least twice throughout the study sessions.

6.5.2. Quantitative Data Analysis Process

In addition to mainly qualitative data collection for the usability study, a System Usability Scale (SUS) questionnaire was given to the participants after performing the tasks on CoCo. As the questionnaire was already digitized, the data was immediately available as a .csv file. The results of the SUS questionnaire per item are presented in Table 7.1. The questionnaire consists of ten items, where the odd items are positively phrased and the even items negatively. This means a high score for odd items is good, while low scores are satisfactory for even items. For calculating an overall System Usability Scale score, the formula given by Brooke in "SUS: A 'Quick and Dirty' Usability Scale" [45] was used. Every item of the questionnaire receives a score from 1-5 (chosen by the participant through a 5-point Likert scale). To calculate the SUS score, the score contributions of every item are summed together and multiplied by 2.5. Score contributions for odd items are calculated by subtracting 1 from the score. Contributions for even numbers are calculated by subtracting the score from 5. Applying this formula results in a SUS score for every submitted questionnaire. To determine the overall SUS score, the mean of all SUS scores is calculated. Bangor, Kortum, and Miller [49] introduce an adjective rating system for SUS scores, where systems with a SUS score below 52 are considered "worst imaginable" to "poor", systems with a SUS score from 52 to 73 are considered to have "okay" usability, and system with a score higher than 73 having a "good" to "best imaginable" usability. This rating system was applied to elucidate the achieved SUS score.

7. Results

The following sections describe the results of the qualitative and quantitative data analysis. Results of the thematic analysis and critical incident analysis are grouped after the research questions *system usability*, *user experience* and *improvement*. The results of the System Usability Scale are presented in Section 7.1. The seven major themes presented in the following sections are aggregated into a diagram, as seen in Figure 7.1. It displayed an overview of the themes and sub-themes identified.

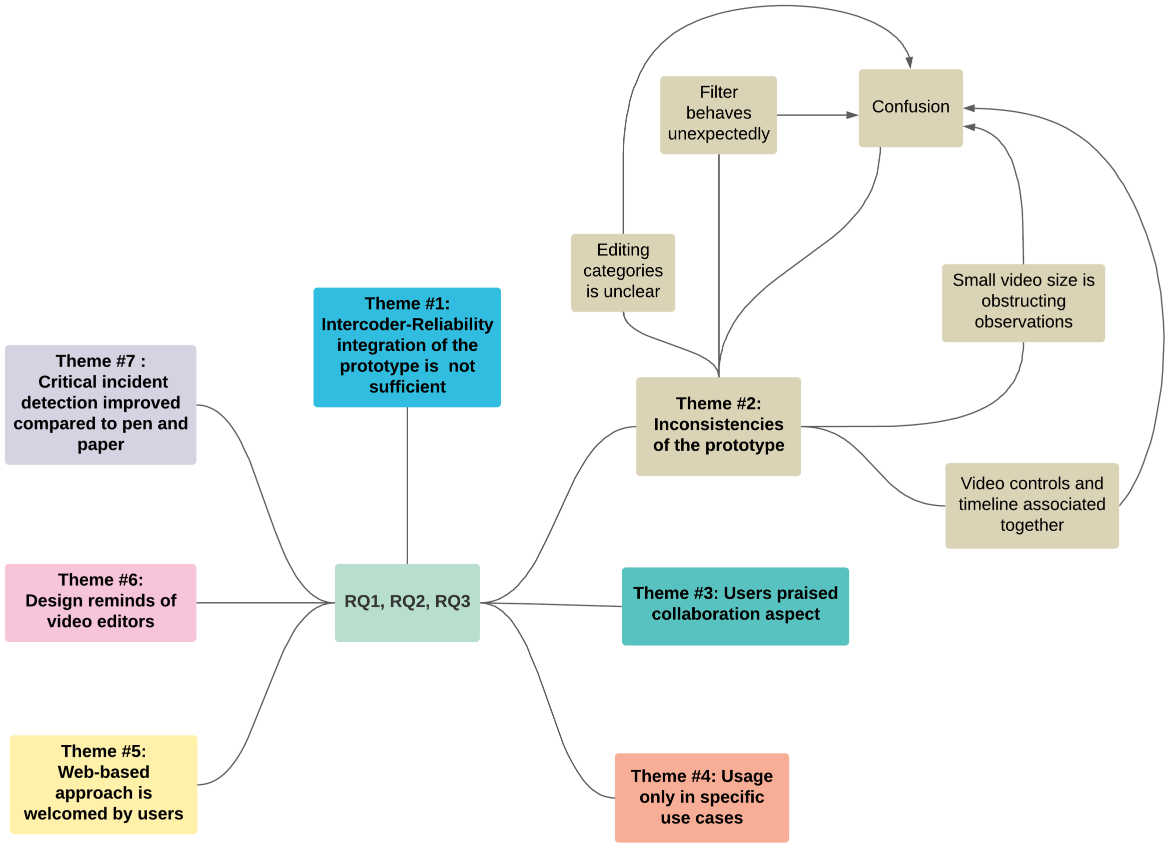


Figure 7.1.: Visual representation of themes identified through thematic analysis.

7.1. RQ1: System Usability

This section presents the results of the qualitative and quantitative data analysis regarding RQ1. At first, the results of the System Usability Scale are illustrated. Thereafter, themes identified by the thematic analysis and relevant to the research question are described in detail.

7.1.1. System Usability Scale

Taking a closer look at the first item of the System Usability Scale (seen in Table 7.1) shows a relatively neutral score on the scale. The reason might be inconsistencies within the system, as the sixth item scored the highest of the negatively worded items. Interestingly, item five was still in close proximity to the mean score given by Bangor, Kortum, and Miller [49]. The fourth item of the questionnaire scored much higher than the mean. Likewise, the tenth item is also higher than the mean. A reason for their scoring might be the introductory tutorial for the prototype at the beginning of the study sessions. Unfortunately, as the prototype implements features that are not obvious to the user (e.g. how to use hotkeys for critical incident collection), a tutorial is necessary. Interestingly, the eighth item scored the lowest on the negatively worded items. Indicating that the prototype was still relatively easy to use, complementing the score of the third item. The reason for that might be that the structure of the prototype follows well-known systems and familiarity aids with its usage. It is potentially also the cause why the ninth item scored comparatively close to the mean score, and the seventh item exceeded it.

Item	CoCo	Mean Score
☺ 1. I think that I would like to use this system frequently.	3.25	3.68
☹ 2. I found the system unnecessarily complex.	2.5	2.34
☺ 3. I thought the system was easy to use.	3.5	3.69
☹ 4. I think that I would need the support of a technical person to be able to use this system.	2.5	1.83
☺ 5. I found the various functions in this system were well integrated.	3.5	3.62
☹ 6. I thought there was too much inconsistency in this system.	3	2.12
☺ 7. I would imagine that most people would learn to use this system very quickly.	3.75	3.82
☹ 8. I found the system very cumbersome to use.	2.25	2.09
☺ 9. I felt very confident using the system.	3.5	3.64
☹ 10. I needed to learn a lot of things before I could get going with this system.	2.5	2.03

Table 7.1.: Results of the SUS questionnaire per item. The scores go from 1 (strongly disagree) to 5 (strongly agree). The column titled with "CoCo" presents the mean scores of the conducted usability study. "Mean Score" presents comparable scores provided by Bangor, Kortum, and Miller [49].

The individual overall SUS scores of the participants were 42.5, 57.5, 67.5, and 80. When computing the overall score for this scale from its individual scores, the prototype reaches a value of 61.875 (see Figure 7.2). Therefore, the prototype is in the range of "okay" usability. However, a score under 70 suggests that the system is a candidate for further improvement and evaluation [49]. As the prototype is designed as an application for everyday use, an overall SUS score under 70 is undesirable and needs to be improved. A possible approach for improvement might be resolving inconsistencies of the prototype and introducing a tutorial within the application for beginners.

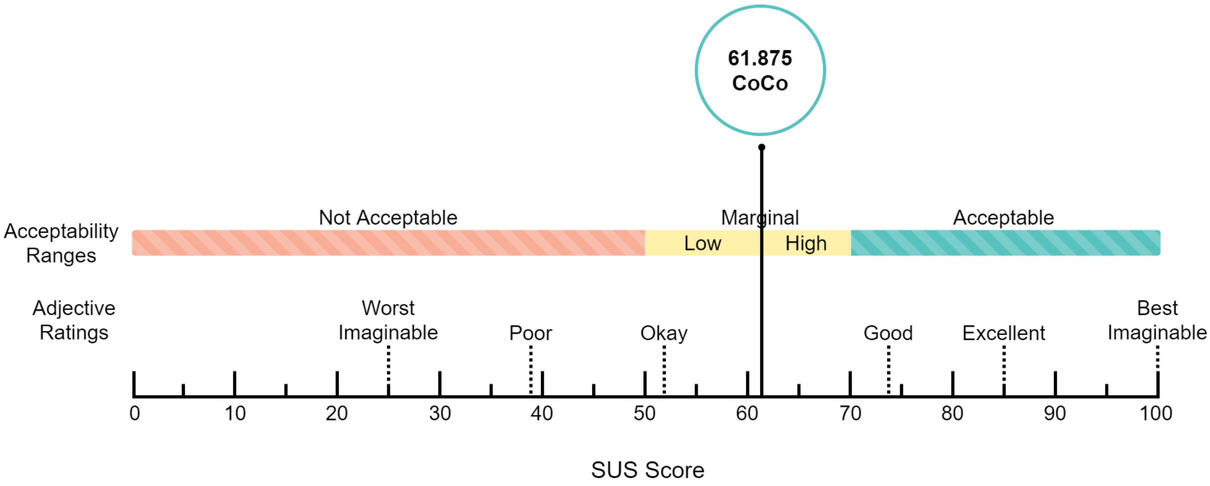


Figure 7.2.: A rating scale developed by Bangor, Kortum, and Miller [49] to interpret the SUS score of a system. *Image taken and adjusted from [49].*

7.1.2. Qualitative Results

The following sections describe the first two identified themes of the thematic analysis. Afterward, relevant critical incidents are listed.

Theme #1: Intercoder Reliability integration of the prototype is not sufficient:

The prototype CoCo allows users to calculate the intercoder reliability according to selected team members of a project. Two participants (P01, P02) articulated concern about this simple structure of the intercoder reliability calculation. They spoke about the nuances of the intercoder reliability and on which factors the result depends:

> *"So it is not just the selection of the people who have coded, but it is also about ehm, which time slots you look at, which session you want to choose. It is not that general, you select a few more things. And that would then have to be supplemented."* (P02)

Thus, the calculation intercoder reliability by the prototype seemed oversimplified to the participants. They feel the need for a more detailed selection option to calculate a more specific intercoder reliability, like a specific time frame or study session.
Participant P01 was not only concerned about the simplification of the intercoder reliability, but also with the content, the user can view from other team members:

> *"[...] you often want to ... depending on what kind of analysis you are doing, you do not want to be able to see what the others have done before you are done. Because otherwise you could be influenced by it. [...] but if I already know what categories there are and so on [...] and I see that and think 'Ah, there is this category' then someone else has already put it in there and then well... "* (P01)

A user can only view their own critical incidents during a live study session in CoCo, but the participants also suggested hiding categories of other team members, when a user adds the category during a live session. The reason for this suggestion is stated in the last part of the previous statement. It was a concern that new categories might influence the user when they see them.

Theme #2: Inconsistencies of the prototype:

After testing the prototype by performing tasks, all participants noticed some inconsistencies, which led to confusion among them. Those inconsistencies are described here.

a. Filter behaves unexpected

   Firstly, testing the filter of the prototype left the participants feeling confused. This confusion mostly arose because of the lack of information to the user, when filter options are selected. One participant expressed very clearly their expectations to see their selected filter options when the filter is not displayed anymore:

   > *"When I select a filter, I would kind of expect to somehow see the selections when I am not in the filter menu anymore."* (P01)

   Another reason for confusion about the filter was its positioning. When the filter menu is expanded, it covers the timeline and prevents the user from seeing the tagged critical incidents:

   > *"When I select a filter, I can no longer see the rest [of the timeline]."* (P03)

   Thus, the participant was confused about what was happening in the timeline during the time they used the filter. Through the covering of the timeline, the filter concealed information about the tagged critical incidents and possible feedback of the timeline, when filter options were selected.
   Selected filter options are applied to the critical incidents in a timeline when the button "Apply Filter" is clicked. In contrast to the participants' expectations, this button click does not cause the filter to collapse to display the timeline again. Users need to toggle the filter icon again to collapse it, which was confusing to all participants. Participant P03, in particular, expressed his confusion about this:

   > *"[...] and when I press 'Apply Filter', the filter does not fold in. This way, it is not visible whether I am in the filter or not."* (P03)

   The lack of feedback resulted in the participant being confused about whether the filter option was actually applied to the timeline or if they were still in the filter at all.

b. Video controls and timeline associated together

   Upon feeling confused about the filter of the prototype, half of the participants (P01, P03) were initially confused and irritated about the placing of the video controls in CoCo. When using the prototype, they wanted the controls to be close to the timeline and not in the toolbar at the top. This is best expressed by one participant:

   > *"... the first thing I noticed was that I thought to myself, 'Why is the play control with pause and stuff up there and not down there?'. [...] I find it too far away from where I interact with the timeline below."* (P03)

A possible reason why participants might expect and want the timeline and video controls in proximity to each other was best expressed by participant P01:

> *"For example, the play/ pause and so on, that is at the top... I would have expected that on the timeline. Ehm, because, so to speak, the timeline also controls that and everything that controls that is together. This is probably also because I am used to it in video editing programs."* (P01)

Not only does this participant associate the timeline with the controls, because it can manipulate the displayed time as well, but also because they were used to this structure from video editing tools. Since the structure and design of CoCo remind many participants of video editors (see Theme #6: Design reminds of video editor:), the participants expected certain functionalities within CoCo that they were already familiar with from other similar-looking systems.

c. Small video size is obstructing the observation

Using the prototype to remotely detect critical incidents requires the users to observe events through live streams. CoCo displays the live streams above the timeline. Are multiple streams included, then they are displayed next to each other and the size of the displayed stream is adjusted respectively. Participants testing the remote observation with CoCo had trouble identifying critical incidents because they had difficulties observing the events through the live streams. Later, they explained this was due to the size of the streams:

> *"One weakness for me in the first view was that the videos were relatively small and [...] [that] was a bit more difficult."* (P04)

The limited size of the streams led to a loss of detail, which makes it more difficult for the user to identify small events that could be considered as a critical incident:

> *"Zoom in the video, that would be cool. Sometimes I could not really see what... whether they have a facial expression, for example."* (P03)

d. Editing categories is unclear

Another inconsistency of the prototype was the category editing within tagged critical incidents. The participants performed tasks where they were asked to edit certain tagged critical incidents and their information about it. When adding a category to a critical incident, a user can choose from a list. This list can be refined by using hashtags. For two participants it was not clear, if they needed to use the hashtag or not:

> *"Exactly, with the categories I had the problem that I was not quite clear, do I have to enter the hashtag or not?"* (P01)

Another participant (P03) had similar problems with the editing of categories within a critical incident. They wanted to add a new category to describe a critical incident and added it with a hashtag, which resulted in the addition of this new category to the whole category list. As no hotkey can be associated to a category when it is newly added with a hashtag, it was not displayed in the chip list above the timeline. The missing chip element for the new category led the participant to believe it was not added at all:

> *"[...] and then when I, for example, ehm, add a category on the right, it is not new in [the chip list]."* (P03)

Hence, they were very confused, as the category was included in the critical incident, but seemingly not in the full list of categories.

e. Confusion

The above described sub-themes caused confusion among the participants. Often, those inconsistencies were one of the first things the participants noticed. This is best described by one participant when asked about their first impression of the prototype:

> *"What was a bit confusing was that I would have expected some things to be somewhere else. [...] Yes, those were the things that struck me at the time."* (P03)

Another participant also stated that the inconsistencies were something they noticed. Not only was it something they noticed very quickly, but something they perceived as a real weakness of the prototype:

> *"The weaknesses are... when I was using it I noticed, I think there were some inconsistencies, but also, mainly, some things that did not work as I expected it."* (P01)

In conclusion, the previously stated inconsistencies caused confusion among participants in various parts of the prototype.

Critical Incident Data

A common barrier or something the participants did not like was the error message in the category section. It is displayed when a user chooses a color, name, or hotkey that is already in use by another category. All participants were first confused by the occurrence of an alert window in the application. When they read the message, they tried to figure out what attribute of the category is already in use. It was difficult for them to find out what it was, especially when two attributes about the category were in usage. All participants (P01 — P04) needed several tries until they figured out what attributes to change to add the new category.

Critical Incident 1:
Error message in the category creation results in confusion due to the lack of precise feedback.

After participants successfully added a new category to the list, another problem arose. Although no error messages were displayed, most participants (P01 — P03) were unsure if the category was added to the list. This was due to the fact, that the category was added to the end of the list. Participants only realized the category was added when they scrolled to the end of the list.

Critical Incident 2:
Participants were confused if the category is added to the list, as the list does not scroll the new entry into view.

When participants wanted to edit categories for a critical incident list entry, they often wanted to add already defined categories. They mainly chose to select them through a list which is shown when a user enters the autocomplete field in the category area. This caused trouble, as the list did not always unfold when it was intended to. Three of four participants (P01 — P03) were then confused as to why the list did not appear and clicked multiple times on the input field, trying to unfold the list. Yet they were unsuccessful and subsequently used the hashtag functionality to search for categories. Interestingly, most participants found this option cumbersome, as they did not want to type the hashtag symbol as a trigger to search for categories. One participant explicitly asked if it was necessary to use the hashtag.

Critical Incident 3:
Autocomplete area for editing categories does not always unfold a full list of categories when clicked.

Critical Incident 4:
Hashtags are more cumbersome for some users for category search.

Three of four participants (P01, P02, P04) tried to delete a critical incident through different approaches than the delete button within a critical incident list entry. All participants tried to delete the critical incident through the corresponding element in the timeline. One participant (P02) tried to use a right-click with the mouse to select a delete option through that. Another participant (P04) tried to use the delete key to delete the selected incident.

Critical Incident 5:
Attempt to use other approaches to delete critical incidents than the delete button in critical incident list entries.

When watching the streams, two participants (P01, P03) instantly tried to pause by hitting the space bar of the keyboard multiple times. After finding out that they could only pause the streams through the video controls in the top bar, they were disappointed and said they would have liked to use the space bar instead of clicking the controls.

Critical Incident 6:
Space bar does not react to play or pause the streams.

When using the filter option in already recorded study sessions, all participants decided to filter by different team members. The critical incidents corresponding to other team members have a white border, which three participants (P02 — P04) did not notice. They thought all incidents displayed belonged to the other team members but not to them. Additionally, when critical incidents from other team members are selected, participants (P01, P03) were confused why the jump feature was not working for the list entries in the critical incident container, as they did not recognize the separate list for other team members' critical incidents. Since they did not select the list, the animation to scroll the selected critical incident in view was hidden.

Critical Incident 7:
Participants believed that their critical incidents also belonged to the other team members' critical incidents after filtering.

Critical Incident 8:
Participants believed the jump function of other team members' critical incidents does not work for the critical incident container, as they did not recognize the separate critical incident list.

7.2. RQ2: User Experience

The following sections describe theme #3 to theme #6 of the thematic analysis. Then, critical incidents regarding this research question are illustrated.

Theme #3: Users praised collaboration aspect:

One significant theme identified through the thematic analysis was that participants highly appreciated the collaboration aspect of the prototype. Three participants (P01, P02, P03) expressed very positive views on it. When asked about the advantages of the prototype, they all mentioned the collaboration as a first prominent benefit, or they presented it positively very early on:

> *"I liked that you also brought in this collaborative aspect."* (P02)

One participant (P01) even said that they liked the chat feature of the application so much, they would also use the application solely for the chat functionality:

> *"So what I think is cool is this chat feature. [...] I could even imagine that I would also use this system and even if you do not do the critical incidents at the same time, because you have this live view and the chat and everything in one. I think that is very practical."* (P01)

The same user later also mentioned that they did not know about systems for this purpose used in collaboration:

> *"[...] but I only know them as single-user and um, that is... multi-user is just great."* (P01)

Another participant (P02) pointed out that the collaboration aspect and chat are the novelty factor of the application, as it is otherwise very similar to other systems for qualitative data collection:

> *"They effectively have the same functionalities. Only not with this collaborative aspect, this chat. [...] they do not have that as far as I know."* (P02)

The participant that never collected critical incidents before also recognized the collaboration aspect as a strength of the prototype.

> *"The strengths are that I have a collaborative tool with which I can somehow evaluate studies together with people."* (P03)

In contrast, one participant (P04) liked the collaboration aspect in general, but was unsure of the usefulness of the chat feature within the prototype:

> *"Ehm, I do not know how helpful this general chat is."* (P04)

They later explained that an integrated chat might be not as important, as there are multiple integration options for communication tools (e.g. integration possibilities for Mattermost [50]).

Overall, the study data suggest that the collaboration and chat feature of the prototype are largely enthusiastically received by users. They can be considered the outstanding features of the prototype.

Theme #4: Usage only in specific use cases:

A theme within the study data is the hesitation of the participants to use the system in their studies. The main reason mentioned by the participants was the variety of study types they perform.
For instance, a participant (P04) was reluctant to use the prototype in their upcoming study, as they did not know what kind of study it would be. They said it would depend on the study setting, and when asked what sort of study setting would be fitting, they mentioned it should not be a long-term study setting:

> *Interviewee: "[...] but depending on how the set-up is, it would not fit in. If this is a long-term study."*
>
> *Interviewer: "Exactly, it is more tailored to these laboratory/usability studies. And would you use it there?"*
>
> *Interviewee: "I would definitely use it there."*

One participant (P01) was hesitant to use the prototype, as their next study would be a lab-based usability study that involves multiple rooms. They explained that the prototype would not be fitting to observe multiple rooms because there is no indication of which streams belong to which room. This would result in confusion about which resource or room is connected to a critical incident. However, they seemed positive to use the prototype in a study setting that would not require multiple rooms:

> *"Let's put it this way, if my next study was not a remote study, I could definitely imagine using the system."* (P01)

Another participant (P03) was not entirely concerned about the study setting of their upcoming study when asked if they would use the prototype. In contrast to other participants, they said that they would use the system when they would see it fit, but only under the circumstance that the inconsistencies of the prototype would be resolved:

> *"Well, yes, I would use it if I needed it for my study. Ehm, but the interface would have to be a bit more polished."* (P03)

Altogether, the study data suggest that users would use the prototype in studies under certain circumstances. These include the resolving of inconsistencies of the prototype and the appropriate study setting. Another possibility would be an expansion of the prototype for multiple room support.

Theme #5: Web-based approach is welcomed by users:

Another theme identified through the thematic analysis was the positive attitude towards the web-based prototype by the participants. Every participant mentioned the web-based approach for CoCo positively. They expressed that they liked this approach because of its accessibility. For instance, one participant (P03) expressed it as a clear strength of the system with the simple reason, that it can be used from everywhere:

> *"So the strengths... it is web-based. That means I can use it everywhere."* (P03)

Another participant (P01), who also found the web-based approach very good, elaborated as to why they thought the increased accessibility is an advantage towards the user experience. For them, it would be annoying to deal with data accessibility when it would only be available locally. But with a web application, data automatically becomes accessible from everywhere with an internet connection. Meaning they would not need to care about how to transfer the required data, as they only need to visit a website:

> *"[...] but the fact that it also runs on the browser is also very good. Because I imagine that would be rather annoying... if I am doing it at university and then I come home and, um, either I have to have the file in the cloud... Or another scenario, I am in the media room conducting the study, go over to my office, that is also annoying. Ehm, so I can just go to the website."* (P01)

For many participants (P01, P03, P04), it seemed that a web-based approach is not only a personal advantage, but it is also advantageous towards the collaboration aspect of the system (see Theme #3):

> *"That is an advantage of your system. That is of course, ehm, across the globe, ehm, it works everywhere."* (P03)

Thus, this shows that the web-based approach for CoCo is a clear advantage for the user, which is welcomed throughout.

Theme #6: Design reminds of video editor:

A further theme identified in the study data was that most participants thought that the structure of the prototype is strongly comparable to a common video editor. Although only parts of the prototype were inspired by video editing tools (i.e. the timelines), participants' (P01, P03) first impression often included the reminiscence of video editors. One participant (P01) in particular mentioned the similarity when asked about the first impression of the prototype:

> *"Ehm, yes, I think the structure is actually quite clear. You know, from video editing programs."* (P01)

Interestingly, this was not seen as a disadvantage of the prototype. Participants liked the structure and similarity to video editors because they already understand it. It additionally meant that they had no issues connecting the timeline and the critical incident elements shown in the timeline with the video material displayed:

> *"As a strength, it is just a totally, um, easily understandable interface with this timeline. And you have this typical video editor interface."* (P04)

In consequence, users easily understand the basic structure of the prototype, as its similarities towards common video editors are helpful for users.

Critical Incident Data

Throughout the study sessions, participants often commented on the wish for audio while detecting or refining critical incidents. They said that audio tracks would help to understand the events in the live streams. However, it is worth mentioning that audio can be used in CoCo, but the resources available for the study session offered no audio tracks.

> **Critical Incident 9:**
> Participants wished for audio tracks in the study sessions to understand the context of the live streams better.

7.3. RQ3: Improvement

This section reports the last theme identified within the thematic analysis.

Theme #7 : Critical incident detection improved compared to pen and paper

The last theme identified through the semi-structured interview was that the application does show improvements to the pen and paper method for critical incident detection and coding.
For example, one participant (P04) solely used the pen and paper method so far to collect critical incident data and found this to be a very ineffective approach:

> *"[...] but I have typically done it with pen and paper and timestamps, which is an extremely ineffective way to do it in comparison, so I would definitely prefer the system."* (P04)

Another participant (P03) never collected critical incidents themselves but knew about the often used pen and paper method. Likewise to the previously mentioned participant, they imagined that the pen and paper method entails disadvantages that are eliminated with the prototype:

> *"Writing that down on paper and then going through the videos again and then... No... because it is linked, the timeline is linked to the video and, um, I can mark things in the timeline at the same time. It is great for that."* (P03)

Interestingly, other participants (P01, P02) did not perceive the application as an improvement for critical incident detection, as they are familiar with other digital applications used for this purpose. For them, the functionality and structure of the prototype was perceived as very similar to other known applications.

> *"But this is, so to speak, not very different from classic video coding, where I have a video image and then I just press some buttons [...], no difference, I think."* (P02)

Importantly, they did not think that the application decreased the capabilities to detect critical incidents.

> *"I would say it is actually neutral in terms of detection. Ehm, because you actually have this standard structure."* (P01)

To conclude, users noticed an improvement for the collection of critical incidents from inefficient methods like pen and paper. However, they do not believe that the prototype shows improvement in comparison to other digital system for qualitative data collection.

8. Discussion

This chapter discusses the results presented in the previous chapter in regard to the introduced research questions. It additionally illustrates possible design improvements in Section 8.4. The chapter lastly introduces opportunities for future work in Section 8.5.

8.1. Usability

System Usability: How well does CoCo perform as a tool for collaborative critical incident detection?

To evaluate the usability of the prototype, two data collection types were carried out. Quantitative data was acquired by a System Usability Scale. For more insights, qualitative data was also collected. This was done by collecting critical incidents during the task completion part of the study and a concluding interview.

The System Usability Scale (SUS) was used to evaluate the overall usability of the prototype. Items of this questionnaire included statements of whether the prototype is easy to use, or if the functionalities are well integrated. The total score of 61.875 of the prototype is in the range between low and high marginal acceptance. However, as this prototype implemented an application for everyday use, this score is not acceptable. It suggests further improvements and evaluation. A possible explanation for a score in this range might be several inconsistencies within the prototype and the need for a tutorial for features of the prototype.

The concluding interviews functioned as the base of the followed thematic analysis, and the collected critical incidents provided more insight into the usability of the prototype. The themes identified by the thematic analysis showed again that there are several inconsistencies in the prototype. Those who could be identified by the analysis include problems with the filter feature of the prototype. It obstructed users so see changes in the timeline and did not give enough feedback to the user about the application of filter options. The participants had trouble editing categories because the autocomplete functionality for the category search behaved unexpectedly at times. Additionally, the participants thought the hashtag as a trigger for the autocomplete search was cumbersome. This may be because they have to select the hashtag symbol before the search, which may seem ineffective to users. The small video size hindered participants from observing the video streams more closely. The inconsistencies caused numerous instances of confusion and frustration. The analysis of the detected critical incidents showed further problems and barriers of the prototype. Another theme of the thematic analysis has handled the concern of participants regarding the intercoder reliability. They were concerned about the too few selection possibilities for the calculation.

Overall, the prototype still has some inconsistencies and problems that have a negative impact on usability. Users are likely to be confused or frustrated by them. Nonetheless, most inconsistencies come down to fixable design modifications, which do not require changing the prototype's philosophy. However, the prototype could benefit greatly from improving upon these issues. Several proposed design improvements for selected issues are described in Section 8.4.

8.2. User Experience

User Experience: Do users like to use the system?

Participants enjoyed certain aspects of the prototype. For one, all participants were excited about the collaboration possibilities. For most participants, those were the outstanding features. Although the possibility to assemble teams for projects was considered valuable to the participants, many mentioned the chat as an outstanding collaboration feature. One participant considered using the prototype solely because of the integrated chat. CoCo being a web-based application was also found to be advantageous by most participants. With the approach, access to the collected study data is not locally restricted and users can access data from anywhere with an internet connection. Participants like this aspect of the prototype because it is more convenient. Despite the favorable opinions towards certain aspects of the prototype, participants would only use it in particular circumstances. It is only considered to be useful for lab-based usability studies, as long-term studies do not provide enough resources. However, even with lab-based usability studies, the study setting has to be carefully considered so that the prototype can be beneficial. As the prototype does not spatially differentiate between resources, it is difficult for users to determine if streams belong to different rooms of the study setting. Therefore, multi-room study settings might be unsuitable.

In the end, participant did not dislike any components of the prototype. However, base functionalities of the prototype were not distinctively recognized, as these are also often offered by other systems. Functionalities and features that are not often provided by systems (e.g. collaboration features or a web-based approach) stand out to the user and leave a positive impression.

8.3. Improvement

Improvement: Does CoCo improve the critical incident detection in the eyes of the user?

Opinions about the improvement of critical incident detection differed. Throughout the interviews, some participants argued that there is no apparent improvement in the detection of critical incidents with the prototype compared to other digital systems for qualitative data collection. This is due to the fact that the prototype offers the same functionalities, complemented by the collaboration possibilities. Other participants stated that the prototype provides many improvements compared to other methods such as pen and paper. Those arguments are not irrelevant because the same participants affirmed that they still use inefficient methods like pen and paper in studies to collect critical incidents. The prototype, therefore, offers improvement to some users, but not all.

The prototype effectively implements the same functionalities for the detection of critical incidents as other related systems. Interestingly, the design process was mainly guided by requirements derived from information and suggestions of members of the HCI group from the University of Konstanz. Therefore, it can be deduced that these functionalities are essentially needed and wanted for the digital detection of critical incidents. This raises the question whether critical incident detection can be further improved solely through the digitization of the process.

8.4. Design Improvements

This section illustrates examples of possible design improvements for critical incidents and inconsistencies introduced in Chapter 7. As some critical incidents indicate software difficulties (e.g. inconsistent unfolding of the category list in the autocomplete field, Critical Incident 3) and not design issues, they are not further discussed in this section. The design improvements are aided by suggestions of participants made throughout the study sessions.

8.4.1. Intercoder Reliability

The results of the thematic analysis show an oversimplification of the intercoder reliability calculation. To overcome this problem, several calculation options can be added to the current design. One participant pointed out additional options that would be appropriate for the calculation of the intercoder reliability.

Suggestion:

> *"Ehm, you need an exemplary time window, so to speak. It can be one session, it can be several. It makes sense if there is more than one session, because then you have different things that can happen. Ehm, it would be cool if I could choose more."* (P02)

Figure 8.1.: Improved settings for the intercoder reliability calculation in the *Overview* section.

A design including those options is shown in Figure 8.1. In the redesign, users can choose team members, sessions, and specific time frames to calculate an accurate intercoder reliability.

8.4.2. Video Controls

When using the system during the study sessions, several participants mentioned confusion about the positioning of the video controls. Due to the resemblance of video editors of the prototype, they expected certain functionalities to be implemented after common video editors. This includes the video controls as well. Therefore, they should be re-positioned in CoCo, as proposed by a participant:

> **Suggestion:**
>
> *"The timeline is where time kind of changes because of that, I will say I would put it under the video. Between the timeline and the video."* (P03)

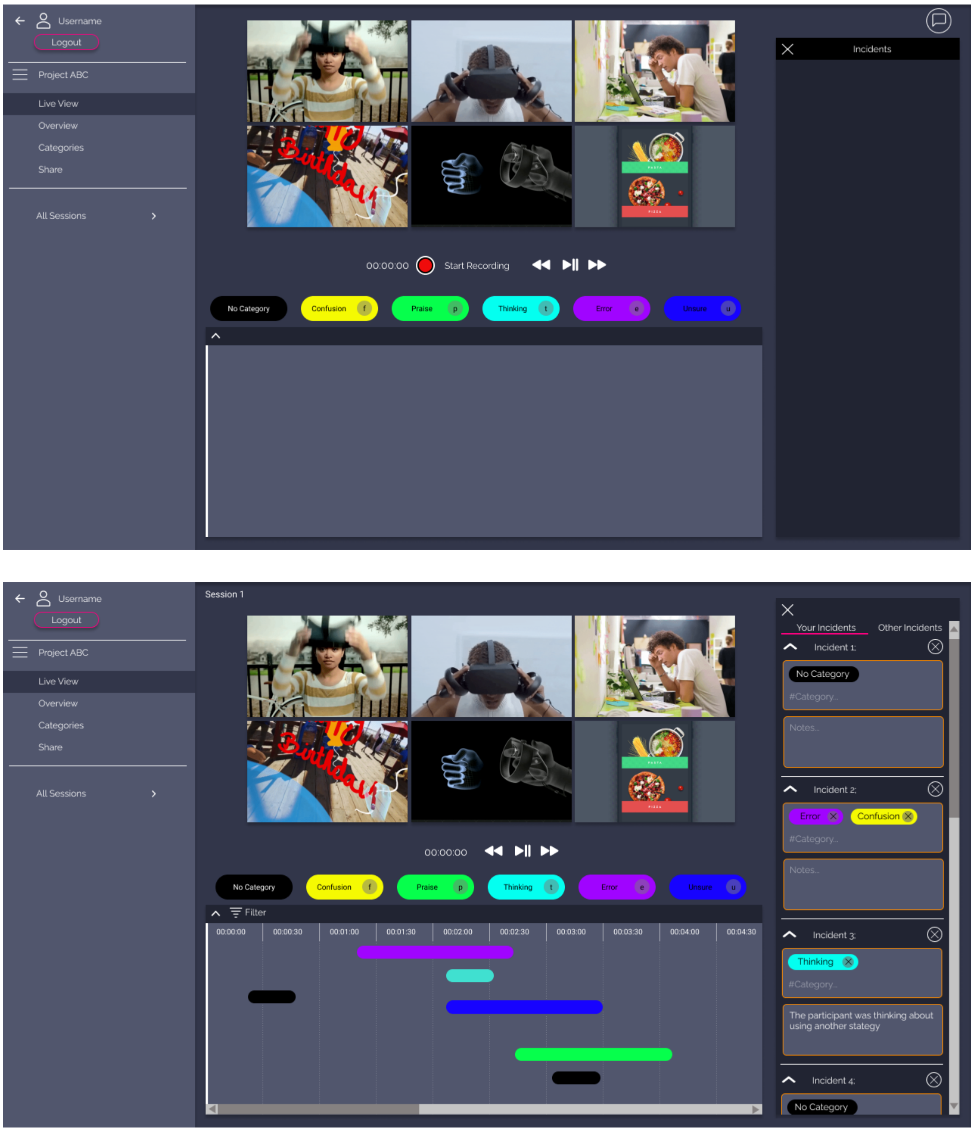


Figure 8.2.: Newly positioned video controls in the *Live View* of the prototype. Controls are now located directly above the timeline.

Figure 8.2 illustrates the re-positioned video controls modelled after the suggestions of the participant. They were not directly placed above the timeline because the chip list with categories is connected to the elements within the timeline. By positioning the video controls under the video streams, they remain in close proximity of the timeline without canceling out the connection between chip list items and timeline elements.

It could potentially be beneficial to integrate further functionalities of common video editors (e.g. space bar to control play and pause).

8.4.3. Categories

Regarding the categories in the prototype, several design decisions could potentially be improved.

Error Message

The first aspect that can reasonably be improved is the error message provided to users when they define new categories in the *Categories* section of CoCo. The error message will appear when users chose either an already utilized category name, color, or hotkey, or a combination thereof. The message is implemented as an alert window containing the text: *"Please choose a key, color, and name that is not already used!"*.

> **Critical Incident 1:**
> Error message in the category creation results in confusion due to the lack of precise feedback.

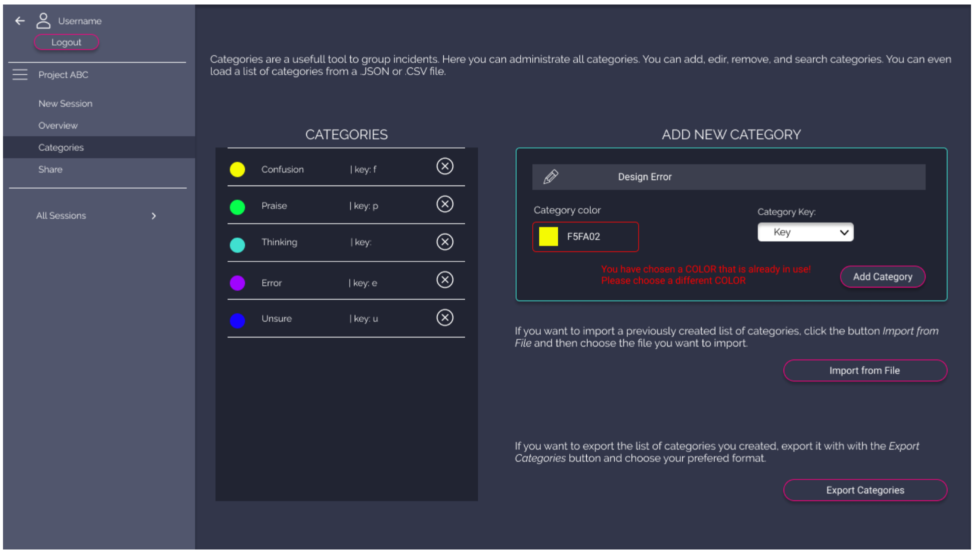


Figure 8.3.: Displaying error messages specified to the attribute that needs to be replaced.

An improved error message could be implemented as shown in Figure 8.3. Here, no alert window is provided to users. Instead, a notable message left to the "Add Category" button will appear. The message will precisely tell users which attribute of the new category is already utilized by other categories and needs to be changed. Additionally, a red border around the input area for the aforementioned attribute further informs users which attribute needs to be changed.

Hashtag for Category Search

Since the participants' opinions towards the hashtag functionality for category search in critical incident list entries were either antipathetic or neutral, this functionality could be withdrawn from the prototype.

Critical Incident 4: Hashtags are more cumbersome for some users for category search.

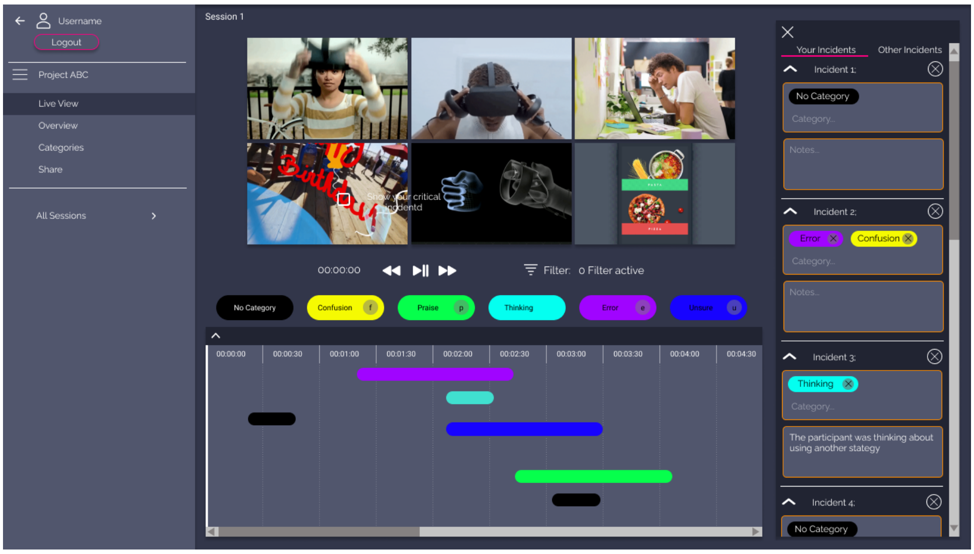


Figure 8.4.: Searching for categories to add to a critical incident list entry without using a hashtag as the autocomplete trigger.

The withdrawal from the hashtag as a trigger for the autocomplete search of categories is illustrated in Figure 8.4. The input field for the search does no longer indicate the need for a hashtag before a category name. Users can immediately type out the category name they want to find. Such a redesign might not only lift the confusion and frustration caused by hashtags, but might also increase the efficiency of the category search, as users do not need to type in an extra character before the search is initiated.

8.4.4. Filter

The evaluation of the prototype identified shortcomings of the filter feature. Design improvements for two shortcomings are presented below.

Filter Position

The thematic analysis of the last chapter identified frustration and confusion among participants caused by the positioning of the filter feature. Participants opposed the filter overlay above the timeline, as it obscures information about the timeline. Some participants also had trouble with the lack of feedback the filter offered about the applied options after the overlay was closed to reveal the results in the timeline. One participant suggested a re-positioning of the filter feature so that it would not obstruct the timeline:

> **Suggestion:**
>
> *"Ah yes, the filter I would make maybe somehow as a separate thing, that it is above [the timeline] [...] and then I see the filter and then I close the filter again."* (P03)

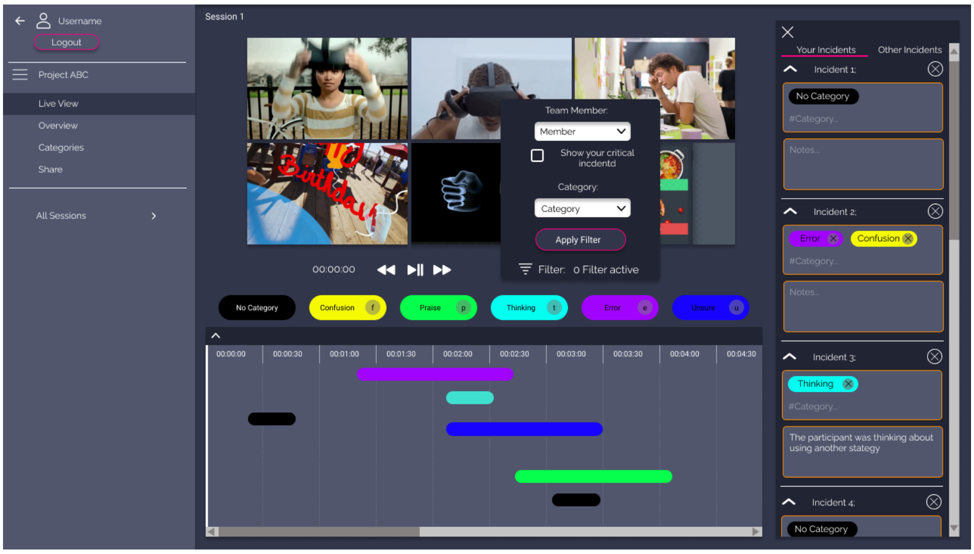


Figure 8.5.: New positioned filter feature. The separation of timeline and filter aims to present more feedback in the timeline when filter options are applied.

A possible design of a re-positioned filter is illustrated in Figure 8.5. The redesign places the filter with the also re-positioned video controls between video streams and chip list. The filter feature would open up as a type of dialogue window over the video streams instead of the timeline. Since users most likely focus their attention on the timeline, it could be considered acceptable to overlay the video streams in this situation. Additionally, it could be beneficial to provide textual feedback for the filter options. The redesign offers this feedback as text, telling users the number of currently applied filter options. The text is permanently shown so that the prototype provides constant feedback for the user.

Applied Filter Options

Results of the applied filter option turned out to be confusing to users at times. This applies especially to filtering critical incidents by other team members. Participants were confused by their own critical incidents still being displayed, and believed the jump functionality was not working for other team members' critical incidents.

> **Critical Incident 7:**
> Participants believed that their critical incidents also belonged to the other team members' critical incidents after filtering.

> **Critical Incident 8:**
> Participants believed the jump function of other team members' critical incidents does not work for the critical incident container, as they did not recognize the separate critical incident list.

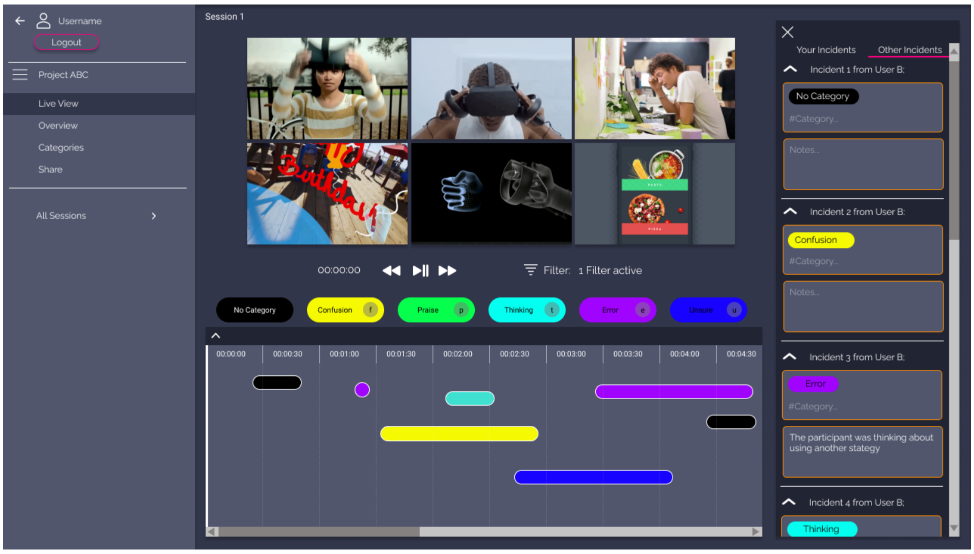


Figure 8.6.: Improved filter display for applied filter options.

To tackle the described problems, Figure 8.6 illustrates an applied filter for another team member. In the redesign, users' own critical incidents are automatically hidden, and only the critical incidents of the selected team member are presented in the timeline. Additionally, CoCo automatically switches to the critical incident list for other team members.

8.5. Future Work

Aside from the presented design improvement possibilities in the previous section, there are multiple opportunities for future work. The following section introduce these opportunities.

8.5.1. Requirements Under The Line of Affordability

CoCo currently implements 10 out of 14 requirements derived from a focus group about critical incidents (see Chapter 3). These ten requirements are considered essential, while the four unimplemented requirements are only considered optional. The implementation could now be enhanced with these four requirements. The resulting application would then address all the preferences and needs of the researchers of the HCI group.

8.5.2. Improved Video Annotation

The current state of the video annotation feature of the prototype CoCo (introduced in Section 5.2.2) only implements basic functionalities. Users can only draw on video streams without deleting or editing the annotations. The annotations are also not time-dependent, as they will stay on the video stream the whole time after the annotation is made. Improving the functionalities of this feature could include: Displaying them only for a certain amount of time (e.g. the length of a critical incident). This would allow for a sophisticated annotation possibility for users. The improvement could also include an editing tool for the annotations, allowing users to correct mistakes or change the annotation to their liking afterward.

8.5.3. Multiple Room Support

CoCo displays defined resources to the user (e.g. video streams, screen recordings, or audio). The application currently makes no spatial distinction between the resources. This could be extended further with multiple room support: Resources could be categorized into multiple rooms and be displayed grouped into rooms. This would allow users to observe different locations used in the study setting more easily.

8.5.4. Detection and Coding in Teams

To additionally relieve stress during the detection of critical incidents, a staggered workflow could be adopted. Since collaboration is an already widely appreciated feature, it could be even further expanded. A staggered workflow could include dividing the project team into subteams for different tasks for the detection and coding process. One subteam would solely be responsible to quickly mark critical incidents. Another subteam would then view marked critical incidents in more detail and edit them (i.e. take notes, edit the length, or add categories) in real time. This way, team members would assist each other throughout the study session. As such, teams have specific tasks and can focus on them rather than doing everything by themselves. However, this approach of teamwork makes it necessary for team members to observe the work of others in real time, making an intercoder reliability calculation between individual team members invalid. Therefore, it should be carefully considered if the potential stress reduction is an advantageous trade off over the intercoder reliability.

8.5.5. System Modularity

A possibility to decrease complexity within the system design could be the separation of components of the system into individual modules. The selection of the modules would take place with the creation of a project, giving the user full control over the structure and functionality of the system. The user would have the opportunity to include only those components and features that they explicitly require. This additionally facilitates the possibility to offer more features to the users without excessively complicating the over all system. It also provides the opportunity to offer valuable features that are generally mutually exclusive.

8.5.6. Aided Critical Incident Detection with Keyword Spotting

Keyword spotting is the identification of specified keywords or phrases in speech [51]. Already used in today's voice assistants like Alexa or Google Assistant [52], it could be an interesting addition to the detection of critical incidents. An integrated keyword spotting functionality for audio during the observation of live streams could potentially assist the researcher with the detection of critical incidents. The identification of keywords or phrases (e.g. "not like" or "bad") could indicate potential critical incidents. This could aid the researcher during the study session and reduce the stress. Phonexia [53] and TensorFlow [54] already offer possibilities to integrate keyword spotting. Using an open-vocabulary approach could facilitate the opportunity to define project-specific keywords [52].

9. Conclusion

This thesis presented the design and evaluation process of the collaborative web application for remote critical incident detection and coding during studies, CoCo.

Through the initial literature research, theoretical knowledge about the detection of critical incidents was acquired. In order to learn more about the current methods and techniques used by the researchers of the HCI group for the detection of critical incidents, a focus group was conducted. Four Ph.D. students of the HCI group took part in the approximately one hour long focus group. It addressed the process of critical incident detection nowadays. The goal of the focus group was to collect information about the participants' opinions, beliefs, and ideas for possible applications. The following analysis of the gathered data resulted in a set of requirements.
Conducting further research prompted the identification of related systems. These include applications for general qualitative data collection up to a toolkit for collaborative video editing. The research revealed that these systems are mostly not specifically designed for critical incident detection. An analysis of the individual systems concerning the derived set of requirements disclosed shortcomings of the systems for the wishes and needs expressed through the requirements. Since many systems do not fulfill all crucial requirements, it was decided to develop an application according to them. For this, an iterative design process facilitated the development of a collaborative application. The resulting application is web-based and supports users to detect critical incidents during studies in collaboration with selected team members. The detection is carried out remotely through the observation of live streams. During live observations, a chat enables the communication between team members. Recorded live streams and collected data are available after a live session to reduce stress. Users can consequently come back to edit critical incidents and refine the gathered data. The editing possibilities consist of adding categories or notes. Categories can be defined beforehand or newly created. Other features of CoCo include the annotation of video streams directly on the stream and the visual representation of critical incidents in a timeline. Several filter options facilitate the specification of displayed data.

For the evaluation of CoCos design implementation, a qualitative usability study was conducted. The study was guided by three research questions aimed to deduce how HCI research members respond to the implemented application. The main focus was on the usability of the prototype, as well as the remaining aspects of the user experience, and a possible improvement of the critical incident detection process.
Participants of the study were all members of the HCI group, three of four being familiar with critical incident detection. The analysis of the collected data shows a need for improvement in the usability of the prototype. Results of the applied System Usability Scale only show marginal usability. In combination with the qualitative results and the interpretation of the individual items of the questionnaire, it can be assumed that inconsistencies of the prototype contribute highly to the limited usability. Participants often described the application as very similar to well-known and used video editors. The resemblance to these systems may have been a factor in the participants' expectation that the functionalities of video editors would also be present in CoCo. Nevertheless, participants also embraced the resemblance as it offered a sense of security and familiarity. They recognized the structure and understood it easily. Further analysis showed that CoCo would only be applicable in the eyes of the participants under certain circumstances. They would only use CoCo in laboratory-based studies that take place in a single room. However, the web-based approach was well received by participants as it implies certain comforts. The leading aspect here is the accessibility of the data across multiple devices. The most prominent feature of CoCo, by far, was the collaborative aspect of the application. Participants were pleased with the opportunity to work in teams. They enthusiastically embraced the integrated chat, describing it as

particularly useful. Concerning the improvement of the critical incident detection process, participants expressed two standpoints. Participants either thought the application offered considerable improvement or none at all. Participants that mentioned no improvement compared the application to other digital systems, while the other participants compared CoCo with pen and paper.

Given the limited usability of the prototype and the amount of discovered inconsistencies and design problems, further design iterations are recommended. That allows for further enhancement of the application by applying not only crucial requirements to the design, but also enhancing the video annotation. Another possible approach for improvement of critical incident detection is the assistance through keyword spotting.

References

[1] Rex Hartson and Pardha Pyla. *The UX Book*. Second. Morgan Kaufmann, 2018 (cit. on pp. 1, 3–7, 30, 31).

[2] Simone Weipert. *Detect Critical Incidents during Studies*. Seminar to the Bachelor's Project. Konstanz, 2020 (cit. on pp. 1, 3, 7, 10, 17).

[3] Norman E Amundson et al. "Fifty years of the critical incident technique: 1954–2004 and beyond". In: *Qualitative Research* (2004). DOI: 10.1177/1468794105056924 (cit. on p. 3).

[4] John Flanagan. "Psychological bulletin. The critical incident technique". In: *Psychological bulletin* 59.4 (1954), pp. 257–72. ISSN: 0033-2909. URL: http://www.ncbi.nlm.nih.gov/pubmed/19586159 (cit. on pp. 3, 4).

[5] Michael C Friedman. "Notes on Note-Taking: Review of Research and Insights for Students and Instructors". In: *Harvard Initiative for Learning and Teaching* (2014), pp. 1–34. URL: http://hilt.harvard.edu/files/hilt/files/notetaking%7B%5C_%7D0.pdf (cit. on pp. 5, 6).

[6] Simone Weipert. *Detect Critical Incidents during Studies*. Bachelor Project Report. Konstanz, 2021 (cit. on pp. 7, 18–20, 22–27).

[7] Mica R. Endsley. *Designing for Situation Awareness: an Approach to User-Centered Design, Second Edition*. CRC Press, 2016 (cit. on p. 7).

[8] Ji-Ye Mao et al. "The State of User-Centered Design Practice". In: *Commun. ACM* 48.3 (Mar. 2005), pp. 105–109. ISSN: 0001-0782. DOI: 10.1145/1047671.1047677. URL: https://doi.org/10.1145/1047671.1047677 (cit. on p. 7).

[9] Kate Crane and Brian Still. *Fundamentals of User-Centered Design: A Practical Approach*. CRC Press (cit. on p. 7).

[10] Mei-Ying Chang and Li-Ling Hsu. "[Qualitative research: an introduction to focus group methodology and its application]". In: *Hu li za zhi The journal of nursing* 53.2 (Apr. 2006), pp. 67–72. ISSN: 0047-262X. URL: http://europepmc.org/abstract/MED/16602049 (cit. on p. 7).

[11] Richard A. Krueger and Mary Anne Casey. "Focus Group Interviewing". In: *Handbook of Practical Program Evaluation: Fourth Edition* (2015), pp. 506–534. DOI: 10.1002/9781119171386.ch20 (cit. on p. 7).

[12] Cliodhna O'Connor and Helene Joffe. "Intercoder Reliability in Qualitative Research: Debates and Practical Guidelines". In: *International Journal of Qualitative Methods* 19 (Jan. 2020), p. 160940691989922. DOI: 10.1177/1609406919899220. URL: https://doi.org/10.1177/1609406919899220 (cit. on p. 9).

[13] Olivier Friard and Marco Gamba. "BORIS: a free, versatile open-source event-logging software for video/audio coding and live observations". In: *Methods in Ecology and Evolution* 7.11 (2016), pp. 1325–1330. ISSN: 2041210X. DOI: 10.1111/2041-210X.12584 (cit. on p. 10).

[14] Mangold(2015). *Software for Observational Research and Data Analysis*. URL: https://www.mangold-international.com/ (cit. on p. 10).

[15] *Behavioral coding - Event logging software: The Observer XT*. URL: https://www.noldus.com/observer-xt (cit. on pp. 10, 11).

[16] Emily Lescak. "Ethograms". In: *Encyclopedia of Evolutionary Psychological Science*. Ed. by Todd K. Shackelford and Viviana A. Weekes-Shackelford. Cham: Springer International Publishing, 2017, pp. 1–2. ISBN: 978-3-319-16999-6. DOI: 10.1007/978-3-319-16999-6_2743-1. URL: https://doi.org/10.1007/978-3-319-16999-6_2743-1 (cit. on p. 10).

[17] Olivier Friard and Marco Gamba. *Behavioral Observation Research Interactive Software (BORIS) user guide.* URL: https://boris.readthedocs.io/en/latest/ (cit. on p. 10).

[18] Joey Hagedorn, Joshua Hailpern, and Karrie G. Karahalios. "VCode and VData: Illustrating a new framework for supporting the video annotation workflow". In: *Proceedings of the Workshop on Advanced Visual Interfaces AVI* (2008), pp. 317–321. DOI: 10.1145/1385569.1385622 (cit. on pp. 11, 12).

[19] Brandon Burr. "VACA: A tool for qualitative video analysis". In: *Conference on Human Factors in Computing Systems - Proceedings* (2006), pp. 622–627. DOI: 10.1145/1125451.1125580 (cit. on p. 11).

[20] Michael Kipp. *The Video Annotation Research Tool.* URL: http://anvil-software.org/ (cit. on p. 11).

[21] ATLAS.ti. *The Qualitative Data Analysis & Research Software.* URL: http://downloads.atlasti.com/docs/manual/atlasti_v8_manual_en.pdf (cit. on p. 13).

[22] MAXQDA. *Die #1 Software für Qualitative & Mixed-Methods-Forschung.* URL: https://www.maxqda.de/ (cit. on p. 13).

[23] *Official Home Page.* URL: https://www.microsoft.com/de-de (cit. on p. 13).

[24] Vosaic Connect. *Video Coaching, Online Teaching, Video Analysis.* URL: https://d1nr41ij4wjmd1.cloudfront.net/Vosaic-Connect-Manual.pdf?mtime=20180410180532&focal=none (cit. on pp. 14, 15).

[25] Clemens N. Klokmose et al. "Videostrates: Collaborative, distributed and programmable video manipulation". In: *UIST 2019 - Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology* (2019), pp. 233–247. DOI: 10.1145/3332165.3347912 (cit. on pp. 15–17).

[26] William Buxton. *Sketching user experiences: getting the design right and the right design.* Elsevier/Morgan Kaufmann, 2007 (cit. on p. 18).

[27] Figma. *Figma.* https://www.figma.com. Online; accessed 29 January 2021 (cit. on p. 18).

[28] Nhor Phai. *Free Vector Icons and Stickers - Thousands of resources to download.* https://www.flaticon.com/free-icon/drawing_2722201?term=sketching&page=1&position=27&page=1&position=27&related_id=2722201&origin=search. Online; accessed 17 July 2021 (cit. on p. 18).

[29] Adobe Inc. *Adobe Premiere Pro.* https://www.adobe.com/. Online; accessed 29 January 2021 (cit. on p. 20).

[30] John Hammersley and John Lees-Miller. *Overleaf, Online LaTeX Editor.* https://www.overleaf.com. Online; accessed 18 September 2021. 2012 (cit. on p. 23).

[31] Google. https://material.io/components/snackbars. Online; accessed 4th September 2021 (cit. on p. 23).

[32] *Instagram.* https://www.instagram.com/. Online; accessed 19 July 2021 (cit. on p. 25).

[33] Doist. *Todoist.* https://todoist.com/. Online; accessed 29 January 2021 (cit. on p. 25).

[34] Google. *Angular.* https://angular.io/. Online; accessed 29 January 2021 (cit. on p. 27).

[35] Google. *Angular Material.* https://material.angular.io/. Online; accessed 29 January 2021 (cit. on p. 27).

[36] Damien Arrachequesne. *socket.IO.* https://socket.io/. Online; accessed 29 January 2021. Feb. 2021 (cit. on p. 27).

[37] typicode. *JSON server.* https://www.npmjs.com/package/json-server. Online; accessed 29 January 2021 (cit. on p. 27).

[38] Scttcper. *Angular Color.* https://github.com/scttcper/ngx-color. Online; accessed 29 January 2021 (cit. on p. 27).

[39] Nick Downie. *Chart.js.* https://www.chartjs.org/. Online; accessed 29 January 2021 (cit. on p. 27).

[40] *Lodash.* https://lodash.com/. Online; accessed 29 January 2021 (cit. on p. 27).

[41] Eli Grey. *file-saver.* https://www.npmjs.com/package/file-saver. Online; accessed 29 January 2021 (cit. on p. 27).

[42] Strongloop. *Node.js web application framework.* https://expressjs.com/. Online; accessed 29 January 2021 (cit. on p. 27).

[43] Taye Adeyemi. *JavaScript drag and drop, resizing, and multi-touch gestures for modern browsers (and also IE9).* https://interactjs.io/. Online; accessed 14 July 2021 (cit. on p. 27).

[44] *Google Forms: Free Online Surveys for Personal Use.* https://www.google.com/forms/about/. Online; accessed 19 July 2021 (cit. on p. 29).

[45] John Brooke. "SUS: A 'Quick and Dirty' Usability Scale". In: *Usability Evaluation In Industry.* CRC Press, June 1996, pp. 207–212. DOI: 10.1201/9781498710411-35. URL: https://doi.org/10.1201/9781498710411-35 (cit. on pp. 29–31).

[46] Virginia Braun and Victoria Clarke. "Using thematic analysis in psychology". In: *Qualitative Research in Psychology* 3.2 (Jan. 2006), pp. 77–101. DOI: 10.1191/1478088706qp063oa. URL: https://doi.org/10.1191/1478088706qp063oa (cit. on p. 30).

[47] Ann Blandford, Dominic Furniss, and Stephann Makri. "Qualitative HCI Research: Going Behind the Scenes". In: *Synthesis Lectures on Human-Centered Informatics* 9.1 (Apr. 2016), pp. 1–115. DOI: 10.2200/s00706ed1v01y201602hci034. URL: https://doi.org/10.2200/s00706ed1v01y201602hci034 (cit. on p. 30).

[48] LaiYee H. *How to do thematic analysis.* https://delvetool.com/blog/thematicanalysis. Online; accessed 06 September 2021. July 2021 (cit. on p. 30).

[49] Aaron Bangor, Philip T. Kortum, and James T. Miller. "An Empirical Evaluation of the System Usability Scale". In: *International Journal of Human-Computer Interaction* 24.6 (July 2008), pp. 574–594. DOI: 10.1080/10447310802205776. URL: https://doi.org/10.1080/10447310802205776 (cit. on pp. 31, 33, 34).

[50] *Mattermost: Open-source, high-trust, developer-centric collaboration platform.* https://mattermost.com/. Online; accessed 27 August 2021. July 2021 (cit. on p. 39).

[51] Halima Bahi and Nadia Benati. "A New Keyword Spotting Approach". en. In: (2009). DOI: 10.13140/RG.2.1.4581.8722. URL: http://rgdoi.net/10.13140/RG.2.1.4581.8722 (cit. on p. 51).

[52] Théodore Bluche, Maël Primet, and Thibault Gisselbrecht. "Small-Footprint Open-Vocabulary Keyword Spotting with Quantized LSTM Networks". In: (2020), pp. 1–13. arXiv: 2002.10851. URL: http://arxiv.org/abs/2002.10851 (cit. on p. 51).

[53] Phonexia s.r.o. *Keyword Spotting - Speech Recognition Technology: PHONEXIA.* https://www.phonexia.com/en/product/keyword-spotting/. Online; accessed 30 August 2021. Apr. 2021 (cit. on p. 51).

[54] Google Brain Team. *Simple audio recognition: Recognizing keywords : TensorFlow Core.* https://www.tensorflow.org/tutorials/audio/simple_audio. Online; accessed 30 August 2021 (cit. on p. 51).

A. Appendix

A.1. Declaration of Independent Work

Ich versichere hiermit, dass ich die anliegende Bachelorarbeit mit dem Thema

CoCo: Design and Evaluation of a Collaborative Application for Detecting and Coding Critical Incidents

selbständig verfasst und keine anderen Hilfsmittel und Quellen als die angegebenen benutzt habe.

Die Stellen, die anderen Werken (einschließlich des Internets und anderer elektronischer Text- und Datensammlungen) dem Wortlaut oder dem Sinn nach entnommen sind, habe ich in jedem einzelnen Fall durch Angabe der Quelle bzw. der Sekundärliteratur als Entlehnung kenntlich gemacht.

Die Arbeit wird nach Abschluss des Prüfungsverfahrens der Bibliothek der Universität Konstanz übergeben und katalogisiert. Damit ist sie durch Einsicht und Ausleihe öffentlich zugänglich. Die erfassten beschreibenden Daten wie z. B. Autor, Titel usw. stehen öffentlich zur Verfügung und können durch Dritte (z. B. Suchmaschinenanbieter oder Datenbankbetreiber) weiterverwendet werden.

Als Urheber/in der anliegenden Arbeit stimme ich diesem Verfahren zu.
Eine aktuelle Immatrikulationsbescheinigung habe ich beigefügt.

Konstanz, den 20. September 2021

Simone Weipert

A.2. Welcome Letter

Teilnehmer ID: *Willkommen*

Herzlich willkommen!
Vielen Dank, dass Sie an der heutigen Studie teilnehmen! Wir danken Ihnen sehr für Ihr Interesse und Ihre Bereitschaft, mit Ihrer Mitarbeit unsere Studie zu unterstützen.

Ziele
Im Rahmen der Stduie evaluieren wir "CoCo - Collaborative Coding of Critical Incidents". CoCo ist eine Web-Applikation zur kollaborativen und ortsunabhängigen Erkennung von Critical Incidents. Die Nützlichkeit und die Benutzererfahrung von CoCo wird getestet, ebenso wie der kollaborative Aspekt des Systems und der verbesserte Prozess der Erkennung von Critical Incidents.

Ablauf

1. Sie werden einen Fragebogen über grundlegende Informationen zu Ihrer Person und bisherigen Erfahrungen mit der Erkennung von Critical Incidents ausfüllen.
2. Anschließend werden Sie eine Reihe von Aufgaben nacheinander ausführen. Die Anweisungen der Aufgaben sind auf eigenen Dokumenten festgehalten. Nach jeder Aufgabe erhalten Sie das nächste Dokument mit der nächsten Aufgabe. Bei jeder Aufgabe bitten wir Sie darum die "Thinking-Aloud-Technik" anwenden. Das bedeutet, dass Sie Ihre Gedanken und Gefühle laut aussprechen, während Sie die Aufgabe lösen.
   Sollten Sie dabei Hilfe benötigen oder sonstige Fragen haben, dürfen Sie gerne fragen.
3. Nachdem Sie alle der Ihnen gestellten Aufgaben durchgeführt haben, werden Sie gebeten, einen Fragebogen auszufüllen. Bitte füllen Sie den Fragebogen nach bestem Wissen und Gewissen aus.
4. Nach Abschluss der Studie wird ein kurzes Interview mit Ihnen geführt, um mehr über Ihre persönlichen Meinungen und Vorstellungen zu erfahren.

Um möglichst umfassende Erkenntnisse zu erhalten, werden Video-, Audio- und Bildschirmaufzeichnungen der Studie angefertigt. Zusätzlich werden Ihre Mausbewegungen und Tastatureingaben aufgezeichnet. Für diese Aufzeichnung ist Ihr Einverständnis erforderlich. Im Gegenzug werden alle Aufnahmen pseudonymisiert und nur zu Auswertungszwecken verwendet. In diesem Zusammenhang ist eine Einverständniserklärung zu diesem Schreiben beigelegt.

An dieser Stelle möchten wir darauf hinweisen, dass wir nicht Sie oder Ihre Leistung bewerten, sondern ausschließlich an der Nützlichkeit und Benutzererfahrung von CoCo interessiert sind.

Vielen Dank für Ihre Unterstützung!

1

A.3. Consent Form

Teilnehmer ID: *Einverständniserklärung*

Einverständniserklärung
Usability-Studie des Prototyps "CoCo - Collaborative Coding of Critical Incidents".

Informationen zur Studienleitung

Name: Simone Weipert
Institution: Arbeitsgruppe Mensch-Computer-Interaktion, Fachbereich Informatik und Informationswissenschaft, Universität Konstanz

Zweck und Dauer dieser Studie
Das Ziel dieser Studie ist es, die Nützlichkeit und die Benutzererfahrung von CoCo zu bewerten. Außerdem soll der kollaborative Aspekt des Prototyps und der verbesserte Prozess der Erkennung Critical Incidents bewertet werden.
Die Dauer der Studie beträgt insgesamt ca. 1 Stunde. Die Studie wird nicht entlohnt.

Freiwilligkeit der Teilnahme und Abbruchsrecht
Die Teilnahme an der Studie ist freiwillig. Sie können jederzeit und ohne Angabe von Gründen die Teilnahme an dieser Studie beenden, ohne dass Ihnen daraus Nachteile entstehen. Sie können zu jeder Zeit eine Pause machen.

Erklärung
Über das Ziel, den Inhalt und die Dauer der Studie wurde ich aufgeklärt. Im Zuge der Studie werden durch Fragebögen personenbezogene Daten erhoben. Des weiteren wird die Studie per Video aufgezeichnet, Audioaufnahmen werden gemacht und Mausbewegungen, Tastatureingaben sowie Bildschirmaufnahmen werden gesammelt. Ich bin darüber informiert, dass personenbezogene Daten vertraulich behandelt werden und nicht an Dritte weitergegeben werden. Alle Daten werden pseudonymisiert ausgewertet. Die Ergebnisse der Auswertung können in nachfolgenden Publikationen veröffentlicht werden. Absolute Diskretion ist dabei gewährleistet. Zu keinem Zeitpunkt werden Rückschlüsse auf Ihre Person möglich sein.
Ich verstehe, dass meine Teilnahme freiwillig ist und dass es mir freisteht, jederzeit und ohne Angabe von Gründen von der Studie zurückzutreten.

Optional: *(Bei Zustimmung bitte ankreuzen)*

☐ Ich bin damit einverstanden, dass meine Video-, Audio und Bildschirmaufnahmen zusätzlich zu internen Präsentationszwecken genutzt werden können.

Hiermit erkläre ich mich mit den in diesem Dokument genannten Punkten und den angekreuzten optionalen Punkten einverstanden:

Konstanz,

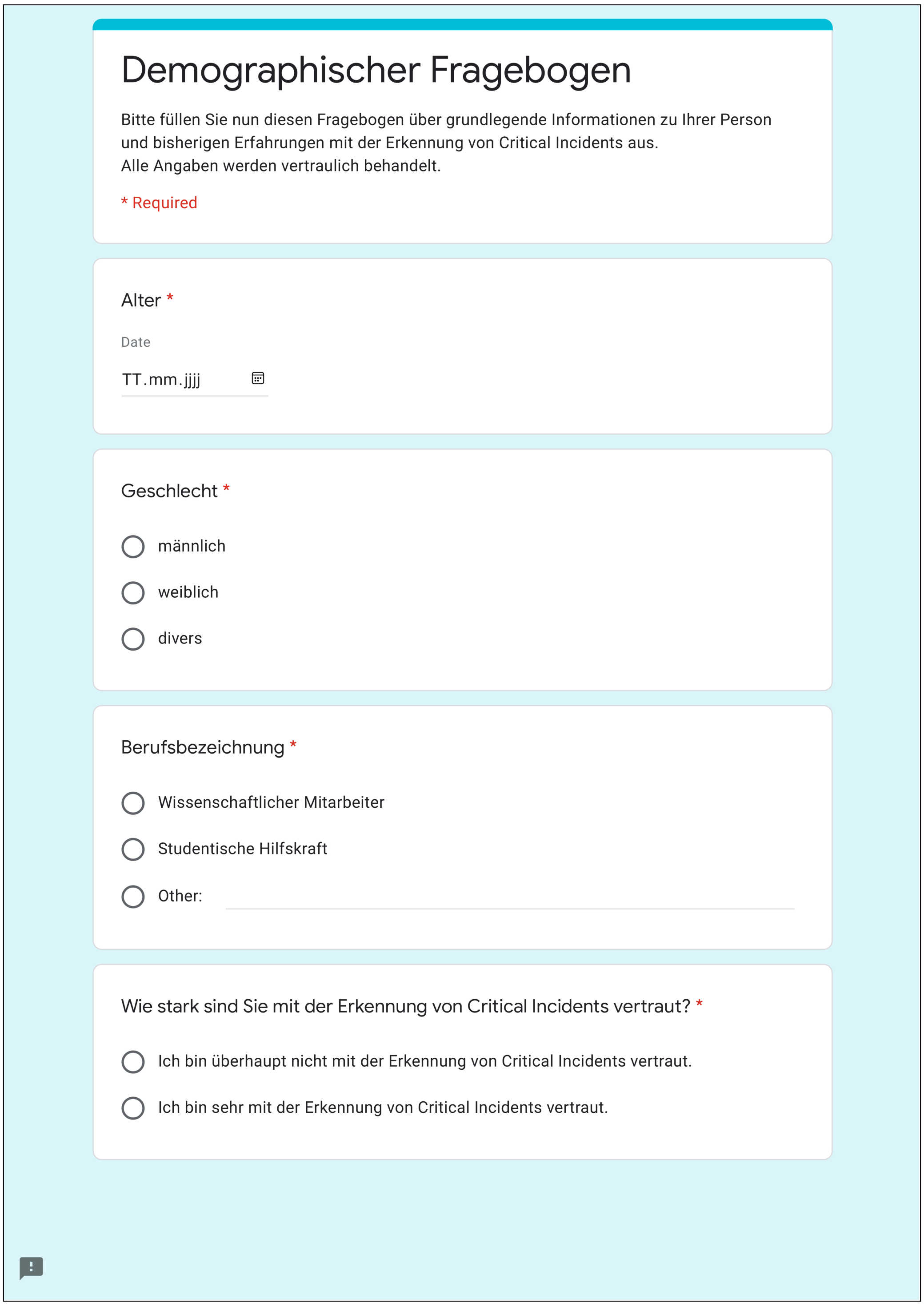
Ort, Datum	Name	Unterschrift

Hiermit verpflichtet sich die Studienleitung, die Video- und Audioaufzeichnung sowie sämtliche sonstigen gewonnenen Daten lediglich zu Auswertungszwecken im Rahmen dieser Untersuchung zu verwenden:

Konstanz,	Simone Weipert	
Ort, Datum	Name	Unterschrift

1

A.4. Demographic Questionnaire



Demographischer Fragebogen

Bitte füllen Sie nun diesen Fragebogen über grundlegende Informationen zu Ihrer Person und bisherigen Erfahrungen mit der Erkennung von Critical Incidents aus.
Alle Angaben werden vertraulich behandelt.

* Required

Alter *

Date

TT.mm.jjjj

Geschlecht *

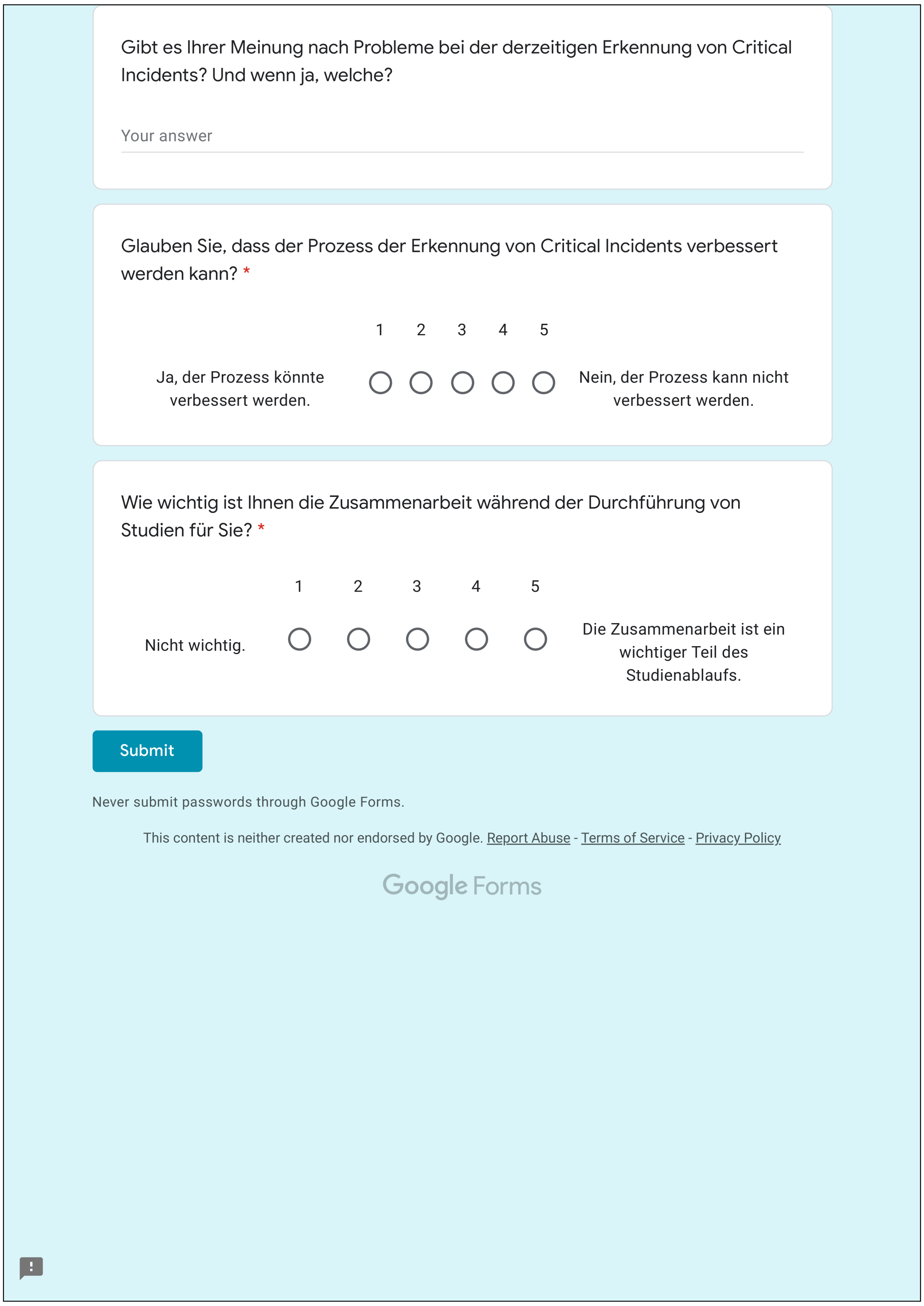
- männlich
- weiblich
- divers

Berufsbezeichnung *

- Wissenschaftlicher Mitarbeiter
- Studentische Hilfskraft
- Other: ____

Wie stark sind Sie mit der Erkennung von Critical Incidents vertraut? *

- Ich bin überhaupt nicht mit der Erkennung von Critical Incidents vertraut.
- Ich bin sehr mit der Erkennung von Critical Incidents vertraut.



Gibt es Ihrer Meinung nach Probleme bei der derzeitigen Erkennung von Critical Incidents? Und wenn ja, welche?

Your answer

Glauben Sie, dass der Prozess der Erkennung von Critical Incidents verbessert werden kann? *

	1	2	3	4	5	
Ja, der Prozess könnte verbessert werden.	○	○	○	○	○	Nein, der Prozess kann nicht verbessert werden.

Wie wichtig ist Ihnen die Zusammenarbeit während der Durchführung von Studien für Sie? *

	1	2	3	4	5	
Nicht wichtig.	○	○	○	○	○	Die Zusammenarbeit ist ein wichtiger Teil des Studienablaufs.

Submit

Never submit passwords through Google Forms.

This content is neither created nor endorsed by Google. Report Abuse - Terms of Service - Privacy Policy

Google Forms

A.5. Explanation of Study Prototype

Usability Studie "CoCo" — Intro

Qualitative Daten

Critical Incidents

Ein Critical Incident ist nach Definition ein Ereignis, das während des Gebrauchs eines Systems auftaucht.

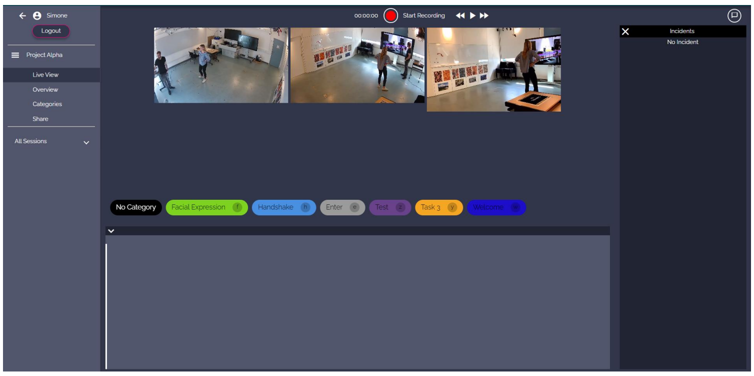
Was für ein Ereignis geschehen ist, ist sehr unterschiedlich. Das einzige, was jeder Critical Incident gemein hat, ist, dass alle wichtig für die Datensammlung sind.

Beispiel:

- Teilnehmer*in der Studie ist verwirrt von dem System.

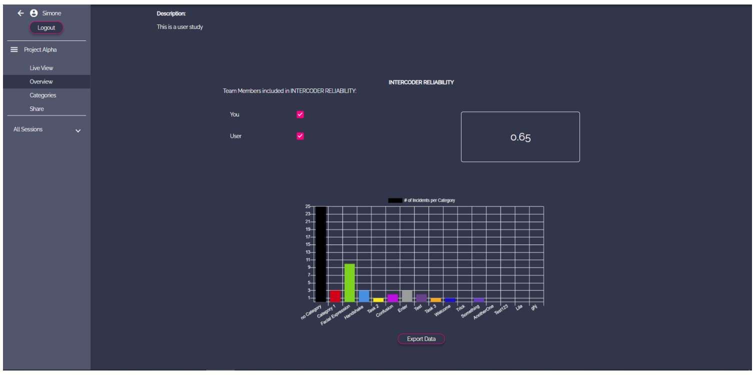
Usability Studie "CoCo" — Aufbau

Erkenne und markiere Critical Incidents durch Live Streams in der Live View mit anderen Teammitgliedern



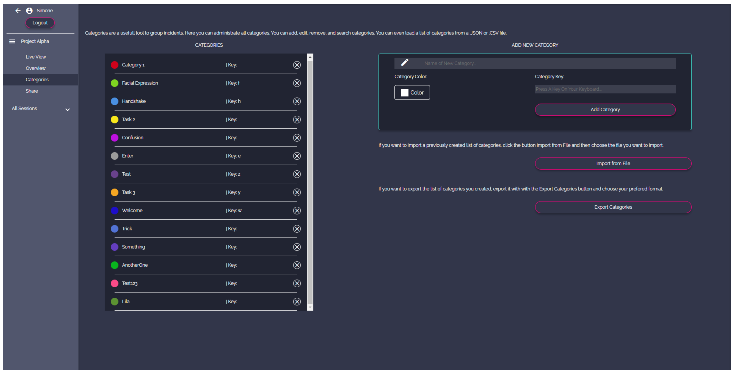
Usability Studie "CoCo" Aufbau

Sieh dir eine Übersicht zu deinem Projekt an



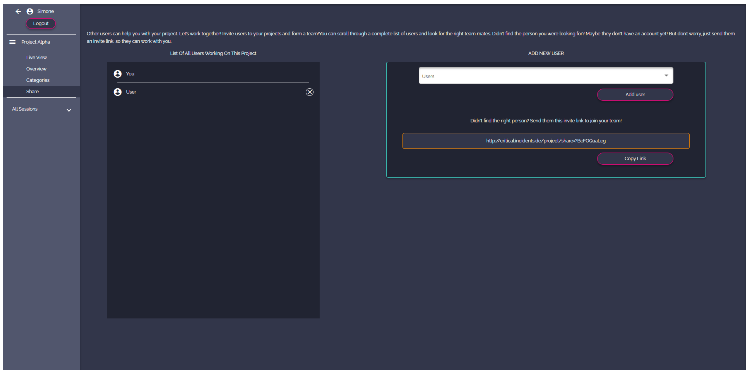
Usability Studie "CoCo" Aufbau

Verwalte die Kategorien des Projekt in „Categories"



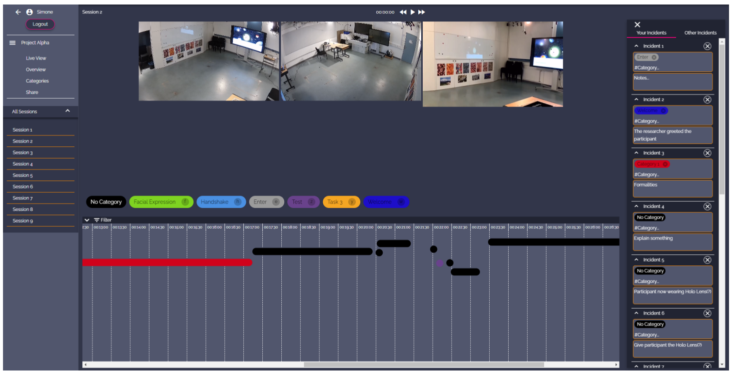
Usability Studie "CoCo" Aufbau

Verwalte das Team in „Share":
- Füge Teammitglieder hinzu
- Entferne Teammitglieder
- Verschicke Invite-Links



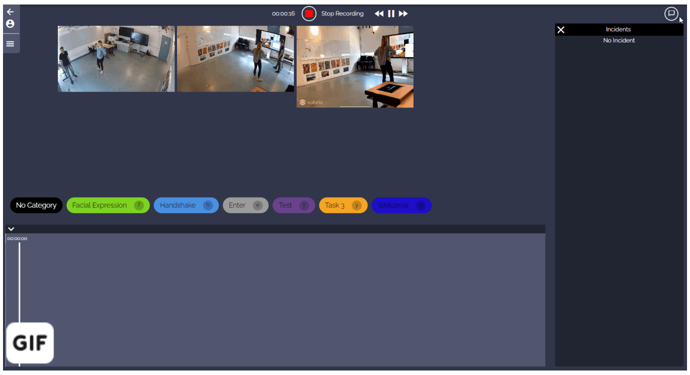
Usability Studie "CoCo" Aufbau

Unter "All Sessions" kannst du dir aufgezeichnete Live Streams nochmal ansehen und Critical Incidents bearbeiten



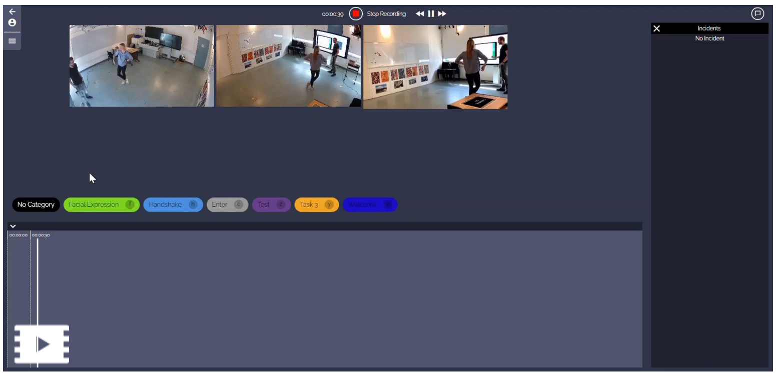
Usability Studie "CoCo" — Kommunikationsmöglichkeiten

Schreibe anderen Teammitgliedern eine Nachricht!



Usability Studie "CoCo" — Interaktionsmöglichkeiten

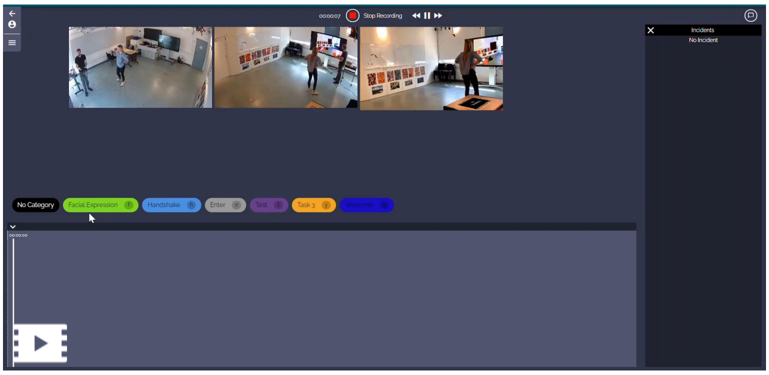
Annotiere ein Live Stream, indem du darauf zeichnest



Usability Studie "CoCo" Interaktionsmöglichkeiten

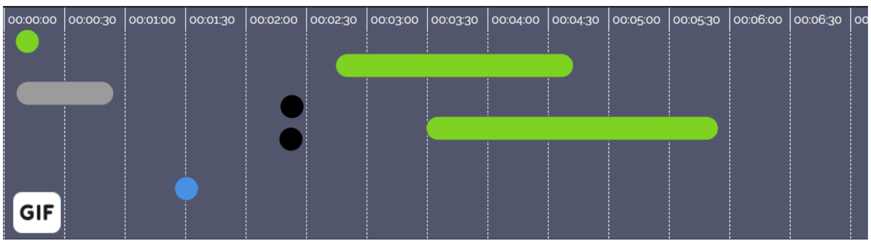
- Markiere einen Critical Incident in Echtzeit durch die Auswahl eines Chipelements
- Halte das Chipelement gedrückt um die Länge des Incidents anzupassen

Alternative:
- via Hotkey



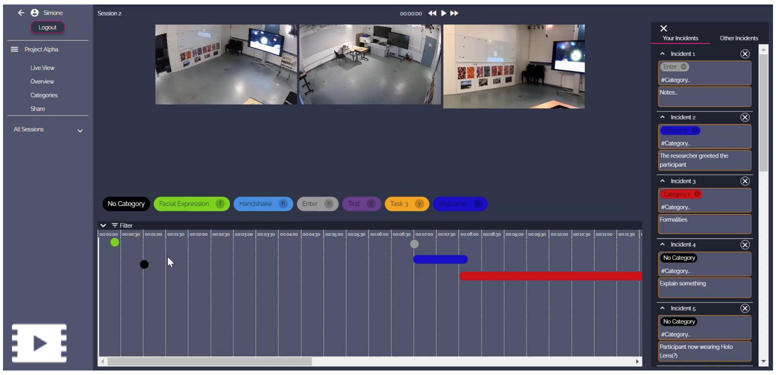
Usability Studie "CoCo" Interaktionsmöglichkeiten

- Interagiere mit Critical Incidents durch die Timeline:
- Durch Drag kann ein Critical Incident bewegt werden.
  Rechts-/Linksbewegung: Veränderung des Start und Endzeitpunkts des Incidents.
- Die Resize-Funktion verändert die Länge des Critical Incidents.



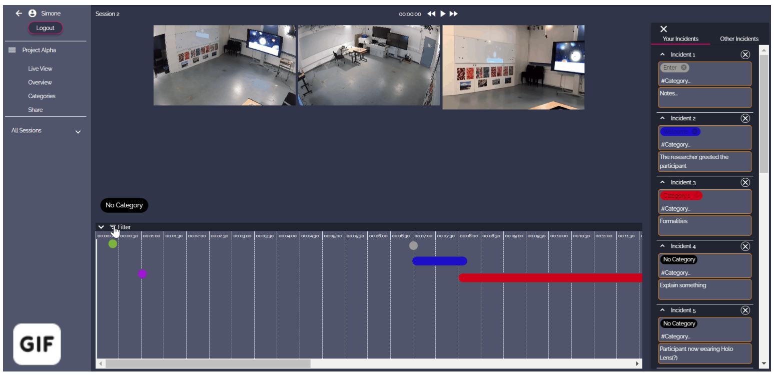
Usability Studie "CoCo" Interaktionsmöglichkeiten

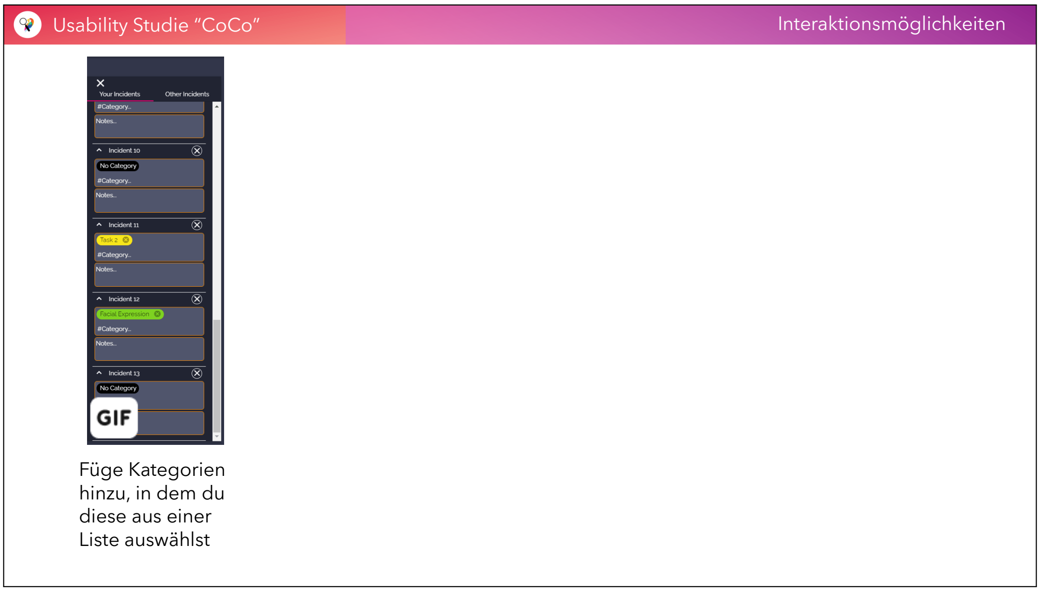
Klicke ein Critical Incident an, um es nochmal anzuschauen

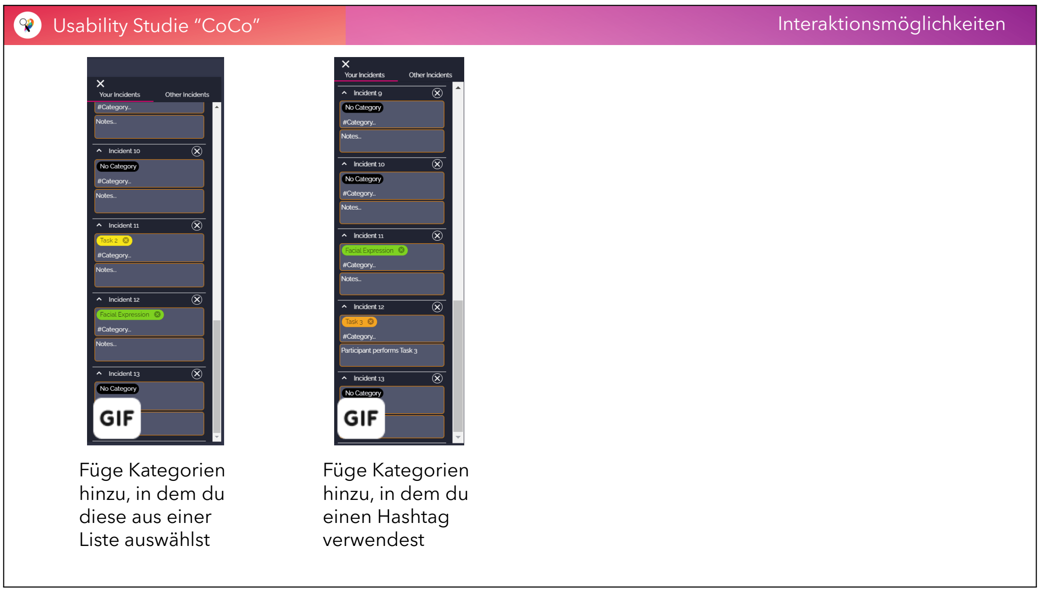


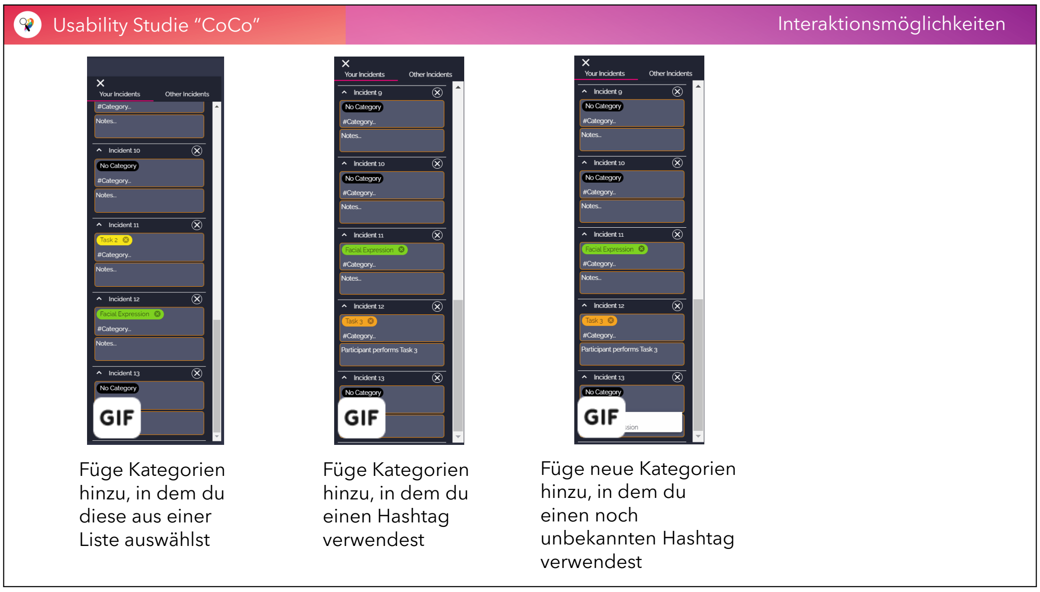
Usability Studie "CoCo" Interaktionsmöglichkeiten

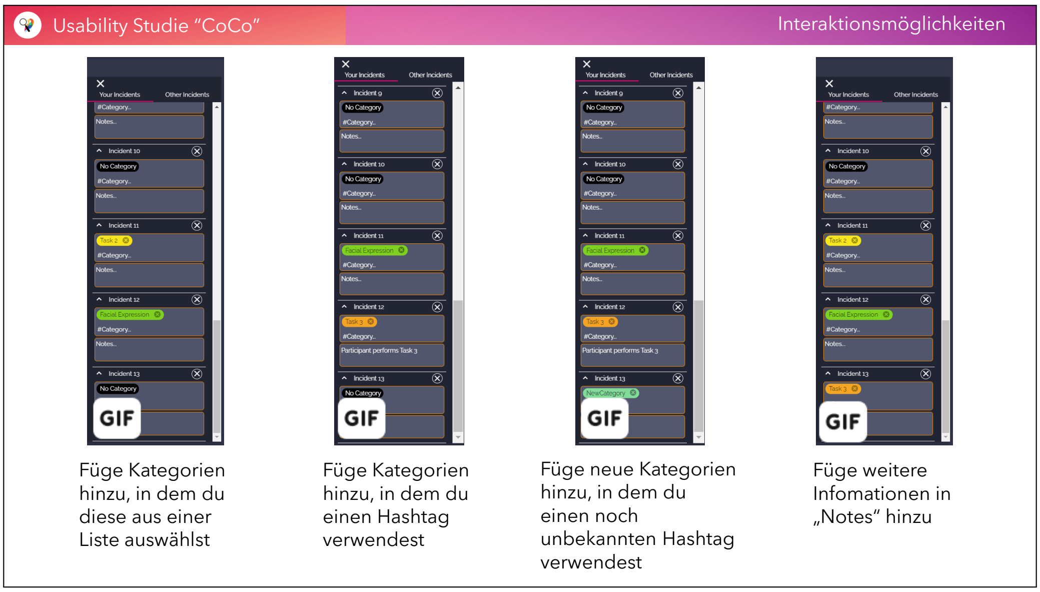
Klicke das Filter Symbol, um Critical Incidents nach Users und/oder nach Kategorien zu filtern




Usability Studie "CoCo"
Interaktionsmöglichkeiten
GIF
Füge Kategorien hinzu, in dem du diese aus einer Liste auswählst


Usability Studie "CoCo"
Interaktionsmöglichkeiten
GIF
GIF
Füge Kategorien hinzu, in dem du diese aus einer Liste auswählst
Füge Kategorien hinzu, in dem du einen Hashtag verwendest


Usability Studie "CoCo"
Interaktionsmöglichkeiten
GIF
GIF
GIF
Füge Kategorien hinzu, in dem du diese aus einer Liste auswählst
Füge Kategorien hinzu, in dem du einen Hashtag verwendest
Füge neue Kategorien hinzu, in dem du einen noch unbekannten Hashtag verwendest


Usability Studie "CoCo"
Interaktionsmöglichkeiten
GIF
GIF
GIF
GIF
Füge Kategorien hinzu, in dem du diese aus einer Liste auswählst
Füge Kategorien hinzu, in dem du einen Hashtag verwendest
Füge neue Kategorien hinzu, in dem du einen noch unbekannten Hashtag verwendest
Füge weitere Infomationen in „Notes" hinzu

A.6. Tasks

Teilnehmer ID:

Task 1 Team Members

Hallo und herzlich willkommen!
Sie sind seit einiger Zeit Teammitglied in einer HCI-Forschungsgruppe. Sie und Ihre anderen Teammitglieder haben das Projekt "SPATIAL" gestartet. Es läuft sehr gut und Sie werden sogar später am Tag eine Studiensitzung mit einigen anderen Teammitgliedern durchführen!

Aber bevor Sie mit der Studiensitzung beginnen können, müssen noch einige andere Arbeiten erledigt werden.

Bitte wählen Sie aus allen Projekten, an denen Sie bereits beteiligt sind, Ihr Projekt "SPATIAL" aus. Nachdem Sie das richtige Projekt ausgewählt haben, möchten Sie Ihren guten Freund und Kollegen **Bobby Singer** zu diesem Projekt hinzufügen. Wenn Sie Bobby unter all den anderen Benutzern nicht finden können, können Sie einfach eine Einladung für ihn kopieren.

Teilnehmer ID:

Task 2 Categories

Während der letzten Studiensitzungen, die Sie für dieses Projekt durchgeführt haben, ist Ihnen eine mögliche neue Kategorie für einige der gefundenen Critical Incidents aufgefallen. Leider hat noch niemand diese Kategorie zu diesem Projekt hinzugefügt.
Bitte fügen Sie die Kategorie **"Hesitation"** hinzu. Die Kategorie soll mit dem Hotkey **"h"** versehen werden. Wenn dieser Hotkey bereits von einer anderen Kategorie verwendet wird, wählen Sie einen Hotkey, der noch verfügbar ist. Sie sind außerdem der Meinung, dass Gelb die beste Farbe ist, um die Kategorie damit zu kennzeichnen.

Teilnehmer ID:

Task 3 Previous Study Sessions

Sie möchten eine frühere Studiensitzung bearbeiten, bevor Sie die Studiensitzung von heute durchführen. Dazu wählen Sie die letzte Sitzung aus, die Sie und andere Teammitglieder durchgeführt haben.

Sie erinnern sich, dass Sie sehr schnell sein mussten, um alle Critical Incidents in dieser Sitzung zu markieren.
Deshalb muss die Länge einiger Critical Incidents, die Ihnen in der Zeitleiste angezeigt werden, bearbeitet werden.
Sehen Sie sich die ersten beiden Critical Incidents erneut an und bearbeiten Sie die Länge des Critical Incidents in der Zeitleiste so, dass sie der tatsächlichen Länge des aufgetretenen Critical Incidents entspricht.

Teilnehmer ID:

Task 4 Notes and Categories

Ein Ziel von Ihnen ist es, für jeden Critical Incident, den Sie in dieser Sitzung gefunden haben, mindestens eine Kategorie zu finden.
Dazu wählen Sie zwei Critical Incidents aus, die keine zugehörige Kategorie haben. Beobachten Sie die kritischen Vorfälle erneut und entscheiden Sie dann, welche Kategorie am besten passt, um den Critical Incident zu beschreiben. Wenn es keinen passenden Vorfall gibt, denken Sie an eine Kategorie, die passt, und fügen Sie sie dem Critical Incident hinzu.

Notieren Sie die Nummern der von Ihnen ausgewählten Critical Incidents und die Kategorie, die Sie ihnen hinzugefügt haben:

Incident Number: ____________ Added Category: ____________________________

Incident Number: ____________ Added Category: ____________________________

Teilnehmer ID:

Task 5 Filter

Sie hatten eine private Unterhaltung mit Ihrem Teamkollegen Fred über diese Sitzung. Fred ist der Meinung, dass es fünf Critical Incidents mit der Kategorie **" Facial Expression"** gab, Sie aber denken, dass es nur vier waren.
Deshalb wollen Sie Freds erkannte Critical Incidents überprüfen und nach der Kategorie " Facial Expression" filtern.

Wie viele Critical Incidents mit der Kategorie "Facial Expression" hat Fred wirklich?

Freds Critical Incidents: ____________

Teilnehmer ID:

Task 6 Communication

Es ist Zeit für Ihre heutige Studiensitzung!
Bitte wählen Sie den richtigen Bereich von CoCo aus, mit dem Sie eine Live-Studiensitzung durchführen können.
Bevor Sie mit der Aufzeichnung der Lernsitzung beginnen, müssen Sie sicherstellen, dass Ihr Teamkollege **Fred** ebenfalls bereit ist.
Schreiben Sie eine Nachricht und fragen Sie, ob Fred bereit ist, mit der Studiensitzung zu beginnen.
Warten Sie auf seine Antwort.

Teilnehmer ID:

Task 7 Detect Critical Incidents

Alles ist jetzt eingerichtet und die Studiensitzung ist bereit für die Aufzeichnung.
Beobachten Sie in den nächsten 5 Minuten, was während der Studiensitzung passiert.
Wenn Sie einen Critical Incident feststellen, fügen Sie ihn zu CoCo hinzu.
Beginnen Sie mit der Aufzeichnung der Sitzung.

Wenn die 5 Minuten vorbei sind, beenden Sie bitte die Aufzeichnung der Sitzung.

A.7. System Usability Scale

System Usability Scale

Bitte geben Sie Ihre Beurteilung ab.

* Required

Ich denke, dass ich das System gerne häufig benutzen würde. *

Stimme überhaupt nicht zu 1 2 3 4 5 Stimme voll und ganz zu

Ich fand das System unnötig komplex. *

Stimme überhaupt nicht zu 1 2 3 4 5 Stimme voll und ganz zu

Ich fand das System einfach zu benutzen. *

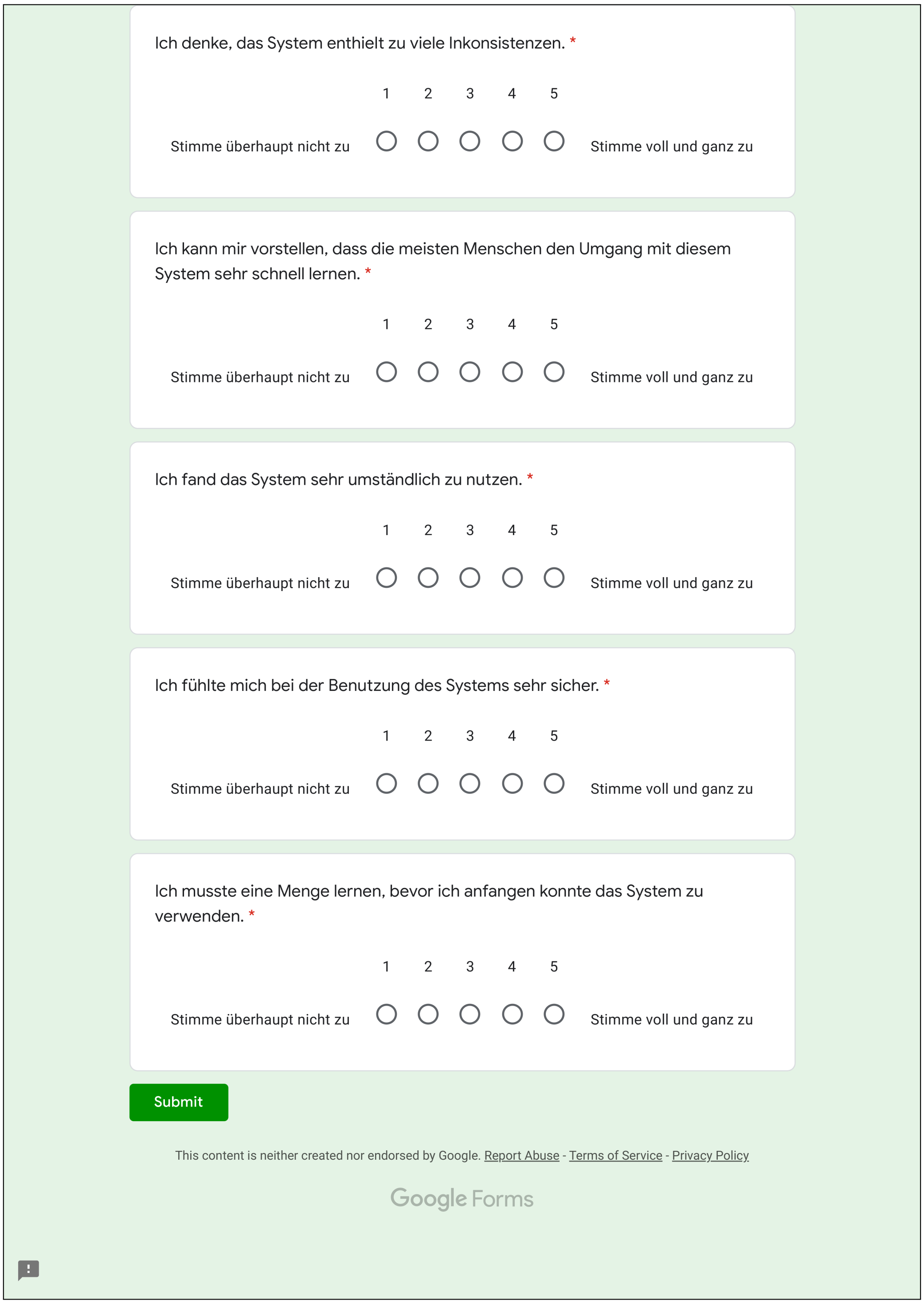
Stimme überhaupt nicht zu 1 2 3 4 5 Stimme voll und ganz zu

Ich glaube, ich würde die Hilfe einer technisch versierten Person benötigen, um das System benutzen zu können. *

Stimme überhaupt nicht zu 1 2 3 4 5 Stimme voll und ganz zu

Ich fand, die verschiedenen Funktionen in diesem System waren gut integriert. *

Stimme überhaupt nicht zu 1 2 3 4 5 Stimme voll und ganz zu



Ich denke, das System enthielt zu viele Inkonsistenzen. *

	1	2	3	4	5	
Stimme überhaupt nicht zu	○	○	○	○	○	Stimme voll und ganz zu

Ich kann mir vorstellen, dass die meisten Menschen den Umgang mit diesem System sehr schnell lernen. *

	1	2	3	4	5	
Stimme überhaupt nicht zu	○	○	○	○	○	Stimme voll und ganz zu

Ich fand das System sehr umständlich zu nutzen. *

	1	2	3	4	5	
Stimme überhaupt nicht zu	○	○	○	○	○	Stimme voll und ganz zu

Ich fühlte mich bei der Benutzung des Systems sehr sicher. *

	1	2	3	4	5	
Stimme überhaupt nicht zu	○	○	○	○	○	Stimme voll und ganz zu

Ich musste eine Menge lernen, bevor ich anfangen konnte das System zu verwenden. *

	1	2	3	4	5	
Stimme überhaupt nicht zu	○	○	○	○	○	Stimme voll und ganz zu

Submit

This content is neither created nor endorsed by Google. Report Abuse - Terms of Service - Privacy Policy

Google Forms

A.8. Questions of the Concluding Interview

Teilnehmer ID:

Beobachtungen und Interview

Task 1

Teilnehmer ID:

Beobachtungen und Interview

Task 2

Teilnehmer ID:

Beobachtungen
und
Interview

Task 3

Teilnehmer ID:

Beobachtungen
und
Interview

Task 4

Teilnehmer ID:

Beobachtungen und Interview

Task 5

Teilnehmer ID:

Beobachtungen und Interview

Task 6

Teilnehmer ID:

Beobachtungen
und
Interview

Task 7

Teilnehmer ID:

Beobachtungen und Interview

Interview "CoCo - Collaborative Coding of Critical Incidents"

1. **Frage:**
   Was war dein erster Eindruck von CoCo??

Teilnehmer ID:

Beobachtungen und Interview

2. **Frage:**
   Was sind die Stärken von CoCo? Was sind die Schwächen von CoCo?

Teilnehmer ID: *Beobachtungen und Interview*

3. **Frage:**
   Hat dir mein Benutzen von CoCo irgendeine grundlgegende Funktion gefehlt?

Teilnehmer ID:

Beobachtungen
und
Interview

4. **Frage:**
   Sind dir ähnliche Systeme bekannt oder hast du schon mal ähliche Systeme benutzt?
   (Ja: Wie heißt das System?)
   (Ja:Haben die Systeme Vor- oder Nachteile im Vergleich zu CoCo?

Teilnehmer ID:

Beobachtungen
und
Interview

5. **Frage:**
   Könntest du dir vorstellen, CoCo für die Erkennung von Critical Incidents in deiner nächsten Nutzerstudie zu verwenden? Warum (nicht)?

Teilnehmer ID: *Beobachtungen und Interview*

6. **Frage:**
   Glaubst du, dass das System die Fähigkeit, Critical Incidents zu erkennen, verbessert hat? Warum oder warum nicht?

Teilnehmer ID: *Beobachtungen und Interview*

7. **Frage:**
   CoCo soll eine Kollaboration, also eine Zusammenarbeit im Team ermöglichen. Was hat für dich bezüglich der Kollaborationsfunktion von CoCo gut funktioniert und was würdest du gerne verbessern oder anders machen?

Teilnehmer ID:

Beobachtungen und Interview

8. **Frage:**
   CoCo soll helfen, dass von überall mit Internetzugriff als Forscher an Studien teilnehmen kann und Critical Incidetns erkennen kann. Was hat für Sie/dich bezüglich dieses Features von CoCo gut funktioniert und was würden Sie/würdest Sie/du gerne verbessern oder anders machen?

Teilnehmer ID: *Beobachtungen und Interview*

9. **Frage:**
   Hast du weitere Anmerkungen?

B. Enclosure

The attached USB device includes all relevant digital files for this work:

1. A digital copy of this work.
2. A video illustrating the features of CoCo.
3. The Seminar to the Bachelor's Project (file name: bachelor_seminar_simone_weipert.pdf)
4. The Bachelor Project Report (file name: project_report_simone_weipert.pdf)

Additionally, the source code of the implemented prototype CoCo can be found on GitLab with the following link: `https://gitlab.inf.uni-konstanz.de/ag-hci/student-projects/bsc-weipert`