Expanding LogUI: Adding Screen Capturing and a Statistical Analysis Dashboard for Web-Based Experiments

Hugo van Dijk
Delft University of Technology
Delft, Netherlands
l

Claudia Hauff
Delft University of Technology
Delft, Netherlands

ABSTRACT

Logging user interactions allows researchers to study user behavior in web applications. The LogUI framework can be used to log user interactions on virtually any web application. This framework misses two vital features, a screen capture feature and a dashboard for statistical data analysis and visualization for use in web-based experiments. Here we have implemented this dashboard and screen capture feature as an addition to LogUI. We cover tools for screen recording and methods to limit bandwidth usage when sending these recordings between client and server. For the dashboard, we worked in cooperation with interactive information retrieval (IIR) researchers working with the open-source collaborative search system SearchX. Using their input, we determined metrics, visuals, and features to be included in the dashboard. Our results demonstrate that recording a single browser tab as opposed to the entire screen or a browser window is more bandwidth-efficient when sending recordings from client to server. Using the H.264 video format results in much better compression compared to other formats available. A low bitrate setting also results in low bandwidth usage, but this setting should be set with care. The dashboard intends to prevent researchers from having to implement their own scripts for data analysis. Its main focus is comparing aggregations of sessions, but it does allow the analysis of single sessions. Including metrics like dwell time, time between queries, session duration, and event occurrences, as well as visuals like box plots, time series plots, and an event timeline is highly beneficial for IIR researchers using LogUI. Being able to relate logs to their occurrence in a screen recording is also advantageous. In an evaluation session, the researchers were able to complete a series of tasks with little difficulty on average. Some features, like filtering specific sessions, or metrics for the visuals, are not fully comprehensible. However, in its evaluation, five out of five researchers noted that they would use the dashboard in their future research. This shows that the included metrics, visuals, and features have been correctly selected.

KEYWORDS

Logging, screen capture, dashboard, data analysis, data visualization

ACM Reference Format:
Hugo van Dijk and Claudia Hauff. 2021. Expanding LogUI: Adding Screen Capturing and a Statistical Analysis Dashboard for Web-Based Experiments. In *Proceedings of Woodstock '18: ACM Symposium on Neural Gaze Detection (Woodstock '18).* ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

Woodstock '18, June 27, 2021, Woodstock, NY
2021. ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

"Information retrieval (IR) is concerned with the structure, analysis, organization, storage, searching, and dissemination of information." [27]. Since 1950, the primary focus of this field is on text and text documents, like web pages, emails, scholarly papers, books, and news stories [12]. One of the major issues in IR is the *relevance* of a document. "A relevant document contains the information that a person was looking for when she submitted a query to the search engine" [12]. A major subfield of IR is web search using search engines. To improve search engines, they need to be evaluated [12]. This is done by looking at their *effectiveness*, the ability to find the right information, and *efficiency*, the speed at which the information is found [12]. However, solely looking at the IR systems themselves does not show us the full picture. To improve information retrieval on the web, we also need to look at a search engine's users. User measures, like perception and attitude, greatly influence the *effectiveness* and *efficiency* of the search process [12].

The study and evaluation of user interactions with IR systems is called interactive information retrieval (IIR) [10]. Usually, IIR researchers implement their own tools and logging infrastructure in order to perform research in this field [25, 32]. According to [17] there are two types of *observables* in IIR research, namely *direct* and *indirect observables. Direct observables* are user interactions and behaviors, like time spent looking at a document, also known as dwell time. *Indirect observables* are measures that others cannot see, like satisfaction, which "essentially exist within the user's head" Kelly [17].

The LogUI framework-agnostic JavaScript library [22] was implemented to ease the process of logging for web pages. It can be easily integrated with virtually any web application and can log almost any user interaction (i.e., *direct observables*) by exploiting contemporary web technologies. Together with SearchX [24], an open-source collaborative search system, LogUI can be used for efficient IIR research.

LogUI has researchers specifically set up which user interactions they wish to log, like hovers or clicks on a specific element, or query submissions. These logs can then be downloaded for data analysis. However, this is often not sufficient as it might not be clear what a user of that web application saw at those points in time. For example, what did a user do right before or after an event? This helps a researcher understand a user's behavior better. To make this clear, the screen the users see needs to be recorded in an efficient and comprehensible manner so the interactions of a user can be related to their screen. This helps debug applications, but also interpret results. As LogUI uses a client-server architecture, these recordings must not clog up the network. Sending videos uses much more data compared to sending logs, for which the framework was initially designed. So, part of efficient integration of screen capturing is limiting its bandwidth usage. Another feature currently missing from LogUI is a dashboard for data visualization. At the moment, researchers analyzing web applications with LogUI have to implement their own scripts data analysis on the user logs. This is labor-intensive and prevents easy comparison of studies as visuals might look different. Adding a dashboard containing the most important metrics and visuals would make the analysis of logs significantly more efficient. It will also result in a more easy comparison of results obtained with LogUI.

This paper concerns the efficient integration of screen capturing into the LogUI logging process and the design and creation of a dashboard for statistical analysis and visualization of LogUI user logs. These additions can be useful for any researcher examining user behavior on a web application. However, for this study, they are implemented in the context of IIR research with SearchX. The research questions relating to these additions are listed below.

RQ1: How can screen capture be efficiently integrated into the LogUI logging process for use in IIR research?

RQ2: What metrics and visualizations are required for a dashboard with LogUI user logs for IIR researchers, and what features aid in their usability?

To answer RQ1 we examine existing tools for screen capturing, and what measures to take to limit the bandwidth usage between client and server. For RQ2, requirements for the dashboard were elicited by surveying and interviewing 5 IIR researchers. After the dashboard was created, it was evaluated with these same researchers.

2 BACKGROUND

This section covers already existing literary works and documentation which is used for the foundation of our research. Firstly, we describe how user studies are performed in SearchX and how LogUI assists in these studies. Then, we cover existing tools for screen capturing, followed by literature on dashboard design for data scientists.

2.1 Using LogUI with SearchX

SearchX is an open-source search system that supports both single-user search as well as collaborative search. The latter is multiple users searching for and making sense of information together. SearchX is split up into a front-end (figure 3) and a back-end. A variety of document retrieval components are supported, like the Bing search API or Elastic search. It enables research across both desktop and mobile platforms for online experiments, like with crowd workers [24]. A typical user study in IIR consists of stages like registration, pre-experiment questions, search tasks [23] and post-experiment questions. These questions can be used to capture *indirect observables*, and the search tasks for *direct observables*. Initially, to log these experiments each researcher had to implement their own logging strategy and add that to SearchX. Its developers are working on removing all logging code and using LogUI for this process instead [16]. A benefit of using LogUI for the logging process is that it is not coupled with SearchX and can thus be developed in isolation. LogUI allows easy logging of the user interactions during an experiment. When logging with LogUI, the


```
{"eventType": "browserEvent",
"eventDetails":
    {"clientX": 1424, "clientY": 1176, "screenX":
        1424, "screenY": 1279, "pageX": 1424,
    "pageY": 1176,  "pageHadFocus": true, "type":
        "cursorTracking", "trackingType": "
        positionUpdate"},
"sessionID":
    "09f44511-87a0-4bad-a3c4-9d05aff587f3",
"timestamps":
    {"eventTimestamp": "2021-04-24T23:08:12.829Z",
    "sinceSessionStartMillis": 1465,
    "sinceLogUILoadMillis": 1465},
"applicationSpecificData":
    {"userId": "5f09068244f84c18faaa74bc",
    "groupId": "6084a4dbae062cf467fe5275",
    "variant": "tl"},
"applicationID":
    "e386d221-1b45-481d-bd3a-ef9d2679e60b",
"flightID":
    "15349d74-1d13-46d8-acd7-972a82f309db"}
```


Figure 1: Example of a LogUI log. This is a log containing information about the cursor position on the screen. Comes from a SearchX user study [26]

logging sessions are performed within so-called flights. Experiment sessions in a flight all include the same conditions. This means we can, for example, compare the logs of two flights testing different interfaces for the same application. When configuring LogUI for a web application, a researcher has to add all interactions they wish to track. The user interaction logs are a JSON object with a variety of fields (figure 1). A log contains information about the type of event logged, details regarding that event, the session, flight, and application it belongs to, and relevant timestamps. There are also two optional fields: *applicationSpecificData*, that stores additional information about the application with each captured event, and *metadata*, which allows data from different locations, like an element's value, to be included in the log as well. These logs are sent from the client to the server using the WebSocket API [6] [22], where they are stored in a MongoDB database and can be downloaded per flight from the LogUI server.

2.2 Tools for screen capturing

To answer RQ1, we first need to determine what tools there are for screen recording. To get a stream of a user's screen, the Screen Capture API [4] exists. This API's sole method *MediaDevices.getDisplayMedia()* is compatible with most modern desktop browsers (figure 2). Unfortunately, mobile browsers are not supported. The stream generated by this API needs to be recorded to be able to save it. The MediaRecorder API [28] can be used for recording video streams. The RecordRTC library [18] and the MediaStreamRecorder.js library [20] utilize this API and allow you to send the recording to the server. These recordings can be done in custom intervals. This means that the screen is recorded for a set duration, after which this recording can be sent to the server while the following interval

Figure 2: Browser compatibility of the method *MediaDevices.getDisplayMedia()* from Screen Capture API. Taken from [4].

is recorded. These intervals can be cached and bandwidth usage can be divided over time, instead of sending one file of the entire recording. For our use case, the two libraries can be used interchangeably, as they both allow for the same recording settings, like bits per second and file type (section 3.1). RecordRTC has more recent updates than MediaStreamRecorder.js [19, 20].

2.3 Dashboard design for data analysis

The dashboard addition to LogUI is used for analytical purposes. Such dashboards require their displayed information to be accompanied by more context than other types of dashboards [13]. Data scientists benefit from more elaborate visuals to examine complex data and relationships and patterns between elements [13]. This means they are also more willing to put in the time to understand the dashboard's features. An interactive dashboard is highly beneficial, as its users can examine data in more detail, by zooming in or filtering, for example.

Hartwig and Dearing [15] state that "the more one knows about the data, the more effectively data can be used to develop, test, and refine theory". Exploratory data analysis is a method to maximize your knowledge about data [15]. For examining distributions of a single variable, the stem and leaf display and the box and whisker plot, or box plot, can be used. Both these techniques were created by Tukey [34] and can be used to show information about the data, like skewness, outliers, gaps, and peaks. There are a variety of other visualization techniques useful for exploratory data analysis, like a correlation matrix and a Sankey Diagram [31]. The former shows the relationship between variables and the latter shows the flow between elements, like between logging events. For events specifically, time series plots and timelines are of interest [8].

3 THE SCREEN CAPTURE FEATURE

This section covers the choices made for efficient integration of screen recording (RQ1).

Using the information transcribed in section 2.2, we decided to implement the screen capture feature using the Screen Capture API and the RecordRTC library. The latter was chosen as it is most recently updated compared to the MediaStreamRecording.js library.

To limit the screen capture feature's bandwidth usage, we consider compression methods and screen recording settings. Section 3.1 covers these subjects. It compares the file size of 40-second

Codecs	Size (MB)
VP9	11.8
VP8	16.0
H.264	5.30

Table 1: Impact of codecs settings on data size for screen recording. (Chromium 90.0.4430.212, Ubuntu 20.04; 40 second video, 1920x1080p, recorded in 5-second intervals.)

recordings, recorded in 5-second intervals, with different settings. 40 seconds is enough to show the effect of the settings. They are recorded on a screen with 1920x1080 resolution, the most widely used resolution [5]. Section 3.2 explains how screen capturing was implemented and how it is used.

3.1 Limiting bandwidth usage in screen recording

The WebSocket API allows us to send raw data from the client to the server. This prevents us from having to encode the data into string format, which always results in some sort of overhead [7].

The RecordRTC library allows for a variety of recording settings. There are two settings of interest here that impact the file size of the resulting video. First of all, you can set the data type and codec. The latter is responsible for compressing video data for distribution. From the three data types that RecordRTC supports [18], our tests show that both Firefox and Chrome[1] only support the WebM file extension. To the human eye, there appears to be little dissimilarity in video quality between the available WebM codecs, but their compression efficiency does differ (Table 1). It is evident that H.264 results in much better compression. The resulting file size is only 45% as big as the file with the VP9 codec, and compared to VP8 it is only 33% in size. Unfortunately, Mozilla Firefox doesn't seem to support H.264 or VP9, but Chrome and Chrome-based browsers do[1]. Other browsers were not tested.

The bitsPerSecond setting, or bitrate, sets the number of bits that are recorded per second. To see the impact of this setting on file size, three different RecordRTC recorders were initialized, all recording the same video stream simultaneously. We tested 0.1 (the minimum the library allows), 1.0, and 5.0 Mbps (table 2). As the videos are compressed during their recording, the bitrate setting does not directly relate to the file size. For example, $5.0/8 * 40 = 24MB$ and not 12.9MB. A low bitrate setting does negatively impact both the video resolution and the frame rate. Low frame rate means an infrequent update of frames, so movement in a video becomes unclear. So, recordings might become unusable if the bitrate is set too low. The required bitrate also depends on the movement in the recorded scene, as a static scene often results in smaller file size, like explained further on. Unless a researcher has experimented with the bitrate setting for screen recordings of their purposes, we recommend not setting this value specifically. When the value is not set, the MediaRecorder API's documentation states the value will default to 2.5 Mbps [28]. However, some tests showed that that was not the case (table 2). We assume in this case the bitrate is not constant throughout the recording and changes according to the

[1]Tested with Firefox 78.10.1esr and Chromium 90.0.4430.212

bitrate (Mbps)	Size (MB)
Not set	8.08
5.0	6.47
1.0	2.76
0.1	0.44

Table 2: Impact of bitsPerSecond settings on data size for screen recording. When this is not set, the bitrate varies throughout the recording. (Chromium 90.0.4430.212, Ubuntu 20.04; 40 second video, 1920x1080p, WebM, H.264, recorded in 5-second intervals.)

activity on the screen to always result in good video quality and frame rate.

On all browsers, the user is prompted to either accept or reject when a web page requests to capture their screen [4]. For Mozilla Firefox and Google Chrome and Chrome-based browsers, the user can choose what area they want to share. This can either be the entire screen, a specific window, or a single tab within the browser[2]. Figure 3 shows the difference between the captured areas. There is some difference in resolution, as, for example, for the window option the operating system's taskbar is not recorded. The selection here can significantly affect the resulting video file size as shown in table 3. For Google Chrome or Chrome-based browsers, the difference between a single tab recording and the entire screen can be over twentyfold. According to user15175033 and Kaiido [35], this is due to the way these browsers handle screen sharing. When the contrast between frames is small, the frame is duplicated. The difference in resolution also has some influence, but that does not explain the large disparity in table 3. On a static scene, recording the window or tab results in much smaller file size, as compression has a significant effect due to similarity between frames. Recording in Chrome is much more bandwidth efficient than using Firefox. When there is much activity on the scene, the file sizes are closer together, and the single tab recording is not always the most efficient anymore. Here, Chrome and Firefox perform rather similarly. During a web search session, a user spends some amount of time reading documents (dwell time), so there will be some inactivity in the recorded scene. This means that selecting the single tab recording will result in the smallest file size.

3.2 Implementation and usage of the screen capture feature

The screen capture feature is integrated as an optional addition to the LogUI logging process. It can be added to a web application alongside the regular LogUI configuration [21, 22, 38]. Some browsers, like Mozilla Firefox, only allow screen recording when it is initiated by a user gesture handler, like a button click. So, after the web application has loaded (figure 4.1), the *Logui.startRecording()* function should be attached to a button press. As SearchX has functionality for introductory pop-ups before an experiment session, the function can be attached to the *OK* button of the final pop-up (4.2) [38]. A researcher can also opt to only start logging user interactions after this point as well [38]. Clicking that button (4.3) then

[2]Chrome and Chrome-based browsers only

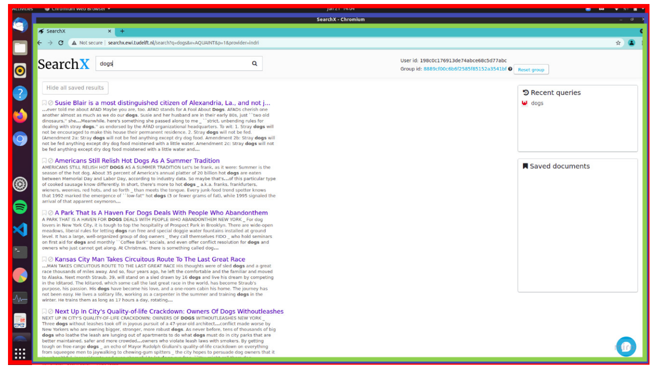


Figure 3: Captured areas for the three options: Entire Screen (Red), Window (Blue), and Tab (Green). Displayed on the SearchX front-end.

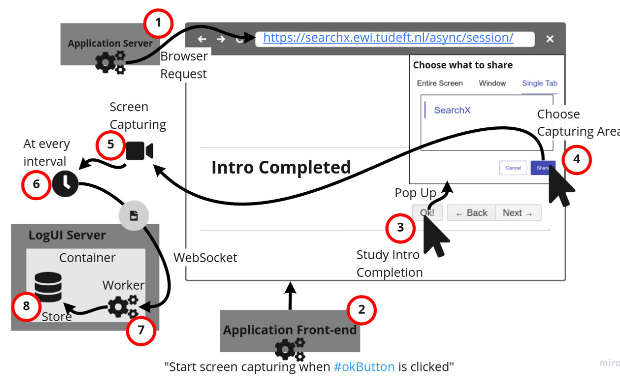


Figure 4: Architecture diagram of the LogUI screen capture feature. Refer to section 3.2 for a detailed explanation.

results in a browser-specific prompt, asking the user to select what area of the screen to share (4.4). After this selection, the screen recording is started (4.5), using the best performing MIME type available for the browser [36]. Every interaction logged after screen recording has started contains the additional timestamp *sinceScreenCaptureStartTimestamp*, which shows the milliseconds passed since the start of the recording, alongside the default timestamps (figure 1). This is used to relate each of the logs to the moment it occurs in the screen recording, described in section 4.2. At every interval, which is set to 5 seconds, the recording over that interval is cached. When the cache of the LogUI client grows too large, its contents are sent to the server (4.6). If network issues arise, five reconnection attempts are done, after which the cache is cleared, meaning both the recordings and logs are lost. When the server receives part of a recording for a session that already has a stored recording, this part is appended to the recording in the database (4.7 & 4.8). The maximum size for a MongoDB document is 16 megabytes [1] and recordings easily outgrow this size. We use the GridFS specification [2] to store the video files, which splits files up into smaller divisions to insert in the database. On retrieval, the divisions are combined again. This prevents us from having to insert each recorded interval separately. A .zip file with the recordings of all sessions in a flight can be downloaded from the LogUI framework.

Captured area	Resolution	Size (MB)			
		Firefox		Chrome and Chrome-based browsers	
		Static scene	Moving scene	Static scene	Moving scene
Entire Screen	1920x1080	4.02	13.5	0.589	13.7
Window	1848x1016	1.84	12.0	0.0483	9.06
Tab	1848x944	-	-	0.0299	10.4

Table 3: Impact of selected area to be captured and scene activity on video file size for Mozilla Firefox and Google Chrome and Chrome-based browsers. (Tested with Firefox 78.10.1esr and Chromium 90.0.4430.212, Ubuntu 20.04; 40 second video)

4 THE DASHBOARD

The dashboard was designed in collaboration with five IIR researchers. These researchers all use SearchX frequently for their studies, and four out of five had used LogUI before. To answer RQ2, firstly, these researchers were sent a survey of which the answers were used to create some initial dashboard requirements and designs. These designs were evaluated with each researcher in an interview, and the outcome of these interviews was used to revise the requirements after which the dashboard was created. Finally, the dashboard was evaluated, again with five IIR researchers.

The survey consisted of 8 open-ended questions (Appendix A). These questions covered the metrics and visuals the researchers typically use as well as how they wish to use the dashboard and the screen recordings (RQ2). On average, the answers were a few sentences long but were enough to understand the necessities for the dashboard. The survey made it clear that are differences between the wishes of each researcher. Accumulating the answers, most metrics looked at are interactions with elements on a webpage. Some more elaborate metrics are session duration, average time between queries, dwell time, and total clicks. Typically used visuals for the analysis of logs also differ between researchers. One only uses tables, others use time series plots and one looks at box plots for exploratory analysis. These visuals should be downloadable, so they can easily be included in a paper. Most researchers will use the screen capture feature for checking 'odd' events in a session after looking at the dashboard statistics. Being able to relate logs to the point in time in that session's screen recording is required.

In combination with the background literature from section 2.3, the answers to this survey were used to elicit some requirements. Some initial designs were made (see figure 9 in appendix B) which were then evaluated in a one-on-one session with each of the researchers. These sessions lasted about 45 to 60 minutes. Their purpose was to verify our understanding of the survey answers, as well as provide the researchers with some visual examples to aid them in explaining their wishes. The requirements were extended using the results of these sessions. The most important outcome of these interviews was that the primary focus of the dashboard should be the analysis of entire flights over specific sessions. Another interesting outcome is that the researchers are indifferent regarding the aesthetics of the dashboard. As long as it is comprehensive and their desired features and metrics are included, they are content. Sections 4.1 and 4.2 describe the metrics and features included in the dashboard respectively, and how they are implemented. Section 4.3 covers the evaluation of the finished dashboard.

The logs used for the figures in this section and the evaluation session were created with LogUI during a SearchX user study [26].

4.1 Included metrics in the dashboard

This subsection explains how the required metrics are included in the dashboard. Anyone can add additional metrics [37].

Session Duration. Session duration is calculated by taking the difference between the first and last log of a session.

Dwell Time. Dwell time is the time looked at a document. For the dashboard to calculate the dwell time, researchers need to make sure to include a mutation observer when initializing LogUI for SearchX [38]. This makes sure a custom message is logged when a document is opened and closed, of which the timestamps are used to calculate both the total and average dwell time.

Time Between Queries. This metric also requires a specific logging event that can be added through LogUI's configuration object [21, 38]. Both the total and average time between queries are calculated.

Event Occurrence Counts. For each type of event, their occurrence count is included in the dashboard. These events can be included through LogUI's configuration object [21]. If a researcher wants to track an event that cannot be added through the configuration object, custom event logs can be added just like for dwell time. To get the number of clicks, such a custom log is required [38].

4.2 Dashboard features

There are five main features on the LogUI dashboard. This section explains these features and how they work [3].

Group Selection Menu. The *Group Selection Menu* (figure 5A) is a drop-down menu containing groups the sessions are divided in, as well as an 'All' option. When a group is selected, only the sessions within that group are included in the *Aggregated Statistics Table* and the *Graph View*.

Aggregated Statistics Table. The *Aggregated Statistics* Table (5B) shows the average of each metric of the selected group.

Graph View. The *Graph View* (5C) contains three types of graph options to visualize the data of all sessions within the selected group. Firstly, it has a box plot option (figure 6a), which shows a box plot for each metric to show the distribution of the values. Each box plot is accompanied by all values to show any gaps just like

[3] Same text included in my GitHub repository [37]

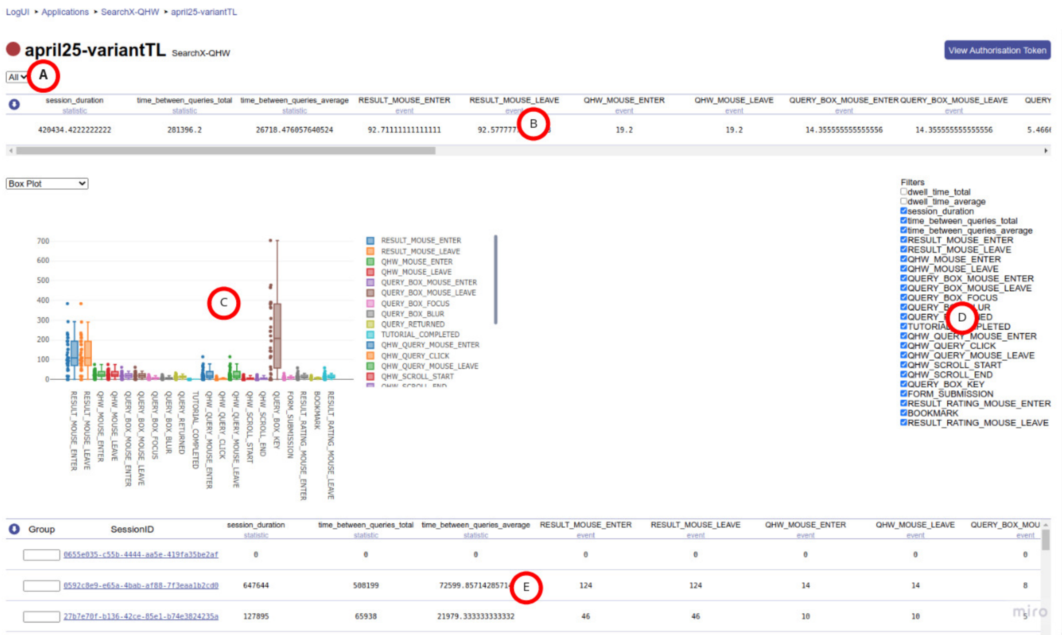


Figure 5: LogUI Dashboard. The Group Selection Menu (A) allows the user to select which group to aggregate results over. The Aggregated Statistics Table (B) shows the aggregated values for each metric in the group. The Graph View (C) shows visuals for the aggregated values in the group. The Filters (D) allows the exclusion of metrics from the dashboard. The Statistics Table (E) show the value for each metric for each session. Clicking on a session ID takes you to The Session Dashboard (figure 7). Refer to section 4.2 for a detailed explanation.

a stem and leaf display does (section 2.3). The second option is a time series plot (figure 6b), which shows the accumulation of event occurrences over time in milliseconds. The last graph option is an event timeline, shown in figure 6c. This graph plots all events on a timeline and can be used to take a deeper look at what events typically follow each other or what session performed an event first. All visuals are interactive in the sense that they allow for zooming and panning, filtering shown metrics, downloading a .png file of the visuals, and show more information when you hover over a data point. They were all created using the Plotly.js library [3].

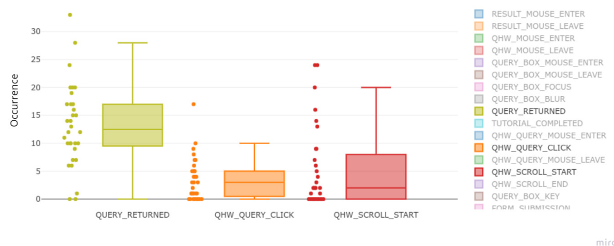
Filters. *Filters* (5D) are in place so IIR researchers can customize the dashboard to their liking. By unchecking a metric, it disappears from the tables in the dashboard.

Statistics Table. The *Statistics Table* (5E) shows values for the metrics per session. The input box at the front of each row can be used to divide the sessions into groups. By inserting a group ID in this box, that ID is added to the *Group Selection Menu*. When you want to take a closer look at a single session, the *Session Dashboard* exists. This can be accessed by clicking on the ID of the session of interest.

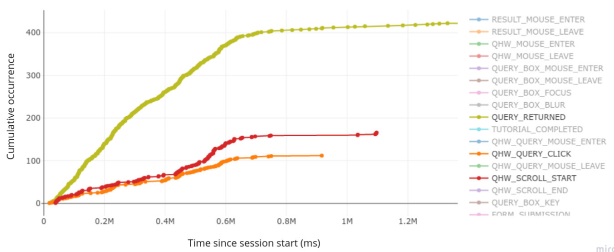
Session Dashboard. The *Session Dashboard* can be used to analyze one specific session. It includes a table containing the values for all metrics within the corresponding flight (figure 7A), visuals (7B), the screen recording of that session (7C), and a table containing all the logs from that session (7D). The available graph options are a Time Series Plot and an Event Timeline. When clicking on one of the logs, the screen recording shows the four seconds surrounding that event, so researchers can see what the user saw at that point in time.

4.3 Evaluation of the dashboard

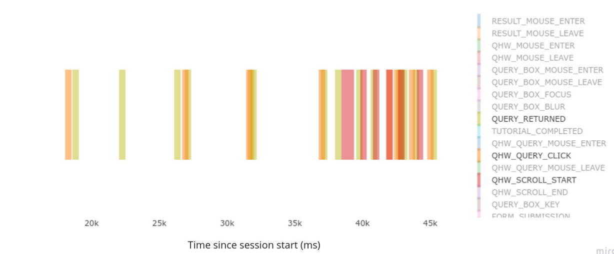
To verify whether we have correctly answered RQ2, we evaluate the finished dashboard. The dashboard was evaluated with the five IIR researchers, four of which being the same as for the survey and one-on-one interview. Firstly, a pilot session was done with a single researcher who worked on the development of the initial LogUI framework. This pilot was used to identify any potential hiccups that would prevent the actual evaluation session from running smoothly. In this session (figure 8), firstly the features of the dashboard were explained and showcased. Then the researcher was asked to connect to the global IP address of the host, on which the LogUI instance was running, and start up the dashboard. He was given six tasks (Appendix C) which he was asked to complete. These tasks ranged from acquiring a plot only containing specific sessions and metrics, to figuring out which search result was clicked on using only the screen recording. During the experiment, the researcher could ask questions in case of confusion. After each task was completed, he was asked to rate the ease at which the task was completed on a scale of 1 (very difficult) to 5 (very easy). If a task was difficult he was asked to explain why. The researcher was then asked to shortly describe what aspects of the dashboard he found useful and which were not useful or should be adapted. Finally, he was asked if he would use the dashboard for future studies. The



(a) LogUI Dashboard Box Plot showing the QUERY_RETURNED, the QHW_QUERY_CLICK, and the QHW_SCROLL_START event



(b) LogUI Dashboard Time Series Plot showing the QUERY_RETURNED, the QHW_QUERY_CLICK, and the QHW_SCROLL_START event



(c) The LogUI Dashboard Event Timeline showing the QUERY_RETURNED, the QHW_QUERY_CLICK, and the QHW_SCROLL_START event

Figure 6: LogUI Dashboard Visuals

Figure 7: LogUI Session Dashboard. The Statistics Table (A) contains the values for the session for all metrics within the corresponding flight. The Graph View (B) shows visuals for this session. The Screen Recording (C) shows the screen recording of the session. The Logs (D) show all logs from this session. Clicking on a log makes the Screen Recording show the four seconds surrounding that event.

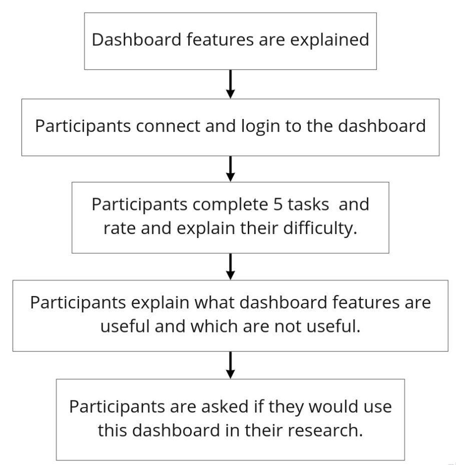


Figure 8: Diagram showing the flow of the dashboard evaluation.

pilot session transpired smoothly, so the same method was repeated in a single session with all other researchers present.

All researchers managed to complete the six tasks. The only task which was found moderately difficult (2.8/5 on average) was a task in which they had to analyze the event timeline to find out which session completed the tutorial first. The researchers noted that the timeline gets cluttered and unclear, and isolating a single metric is not very intuitive. All other tasks were rated between 3.4 and 5 on average. There were quite a few comments on the fact that there is the *Filter* feature for the tables and that the visuals have their own filter feature. Also, the *Filter* feature does not allow double-clicking to exclude all other metrics, while the visual's filter feature does. This appears to cause confusion and makes filtering a tiring task. There are also some requests for a search option for session IDs, metrics, and log events. Next to the group option for sessions, a filter checkbox was also requested to make it easier to include and exclude sessions. A comment was also made that anyone not familiar with JSON might be confused by the table containing logs in the Session Dashboard. Despite these comments, five out of five researchers said they would use the dashboard in future research. They will mostly use it for creating visuals for papers and for replaying what a user did at specific moments with the screen capture recording.

5 RESPONSIBLE RESEARCH

The expansion of the LogUI framework described in this report was done in close cooperation with a total of six IIR researchers. These researchers all agreed to participate in the surveys, interviews, and evaluation sessions. Any personal data gathered was deleted after finishing the project. The researchers were not in a subordinate position to the host.

All work carried out in this paper is that of the author, Hugo van Dijk, unless stated otherwise. Credits were given when concepts were taken from other literary works, and any code that was not solely the author's work has been accompanied by its source.

The results described in section 3.1 only serve as an indication and might not be completely reproducible as compression effectiveness greatly depends on the differences between the frames of a video. This means the precise numbers are presumably not repeatable. However, we do believe the overall order of the settings in terms of which value results in lower file size is reproducible.

The requirement elicitation process of the dashboard expansion was done with five IIR researchers. As stated in section 4, the wishes differed between researchers. If this process were repeated with other researchers, the required metrics and visuals will presumably differ.

6 DISCUSSION

6.1 Discussion of the screen capture feature

The results in section 3.1 indicate that there are three measures that influence the file size of a recording, and thus the bandwidth usage between client and server. The MIME type is responsible for the compression of the videos. The H.264 codec results in the best compression, followed by VP9. The VP8 codec performed the worst. There is both supporting [14] and contradicting [9] research for this result. Both studies compared compression efficiency using the objective video quality measure peak signal-to-noise ratio, where we subjectively judged video quality. That might explain the difference in results. The lower the bitrate, the lower the resulting file size. However, a low bitrate can result in unusable recordings due to poor video quality and thus this setting should be left untouched unless experimented with for the use case. Recording a browser window or tab as opposed to the entire screen results in a smaller file size of the recordings. When there is not a lot of activity on the recorded scene, a single tab recording is significantly more bandwidth-efficient. When there is much activity, a single tab recording can be less bandwidth efficient than a window recording. Web search sessions are often a combination of activity and inactivity, so single tab recording will be the best choice.

Each of these experiments was run once. Running multiple repeats of the same experiment presumably won't change the overall ranking of codecs and captured area in regards to their effect on file size. However, it will improve the accuracy of the results.

6.2 Discussion of the dashboard

As a result of the user studies, we can determine that the dashboard should include the metrics dwell time, time between queries, and session duration, as well as occurrence counts for all logging events. Required visuals are tables, box plots, time series plots, and an event timeline. Additionally, the easy relation of logs in a table to their occurrence in the screen recordings is also highly beneficial. All researchers said they would use the dashboard in their research. However, some practice is required for them to be fully familiar with all functionality. Search functionality and a (de)select all option for the filters would improve the dashboard's user experience.

Some researchers showed interest in additional functionality, like flagging outlying values or more visuals. Due to time constraints, those were not included in this paper. Additionally, as the user studies for the dashboard were only performed with five participants, their results might not be as elaborate as if more participants were used. However, we do believe the dashboard does appear to be a good foundation for data analysis and visualization for IIR research.

7 CONCLUSIONS

The purpose of this report was to find out how screen capturing can efficiently be integrated into LogUI (RQ1), and what metrics and visuals are required for a dashboard with LogUI user logs (RQ2). Both were done in the context of IIR research. To do so, we looked at existing tools for screen recording, and actions to take to limit the bandwidth usage between client and server. We also elicited the dashboard requirements with 5 IIR researchers. Then after finishing the final product, we again used 5 IIR researchers for its evaluation.

For efficient integration of screen capturing (RQ1), we need to limit bandwidth usage between client and server. The H.264 codec proved to be most effective compared to the VP8 and VP9 codec. Recording a single tab as opposed to the browser window or entire screen is most bandwidth-efficient for web search sessions. Bitrate settings do influence bandwidth usage, but should not be touched as that can result in undesirable video quality.

Our results also show that every IIR researcher required different metrics and visuals for the dashboard (RQ2). Having accumulated the wishes of all interviewed researchers, the metrics dwell time, session duration, time between queries, and event occurrence counts should be included. Required visuals are tables, box plots, and time series plots. An event timeline can be valuable as well but can become cluttered and unclear. However, additional visuals are still called for. A filter and grouping feature make sure researchers can include only what they want in the visuals. For more information regarding a log, it can be related to their occurrence in the screen recording. All researchers said they would use the dashboard in their research, so it is clear that averts the need for researchers to implement their own scripts for data analysis. However, adding search functionality and filters for sessions will improve the user experience.

For future work, the robustness of the screen capture feature should be evaluated when network congestion occurs. An example experiment would be to have many machines send screen recordings to the server simultaneously, and see at what point the server starts dropping packets. It is also worth looking into adding screen capture functionality for mobile browsers as well, as SearchX and many web applications have both a desktop and a mobile version. As LogUI now has screen capturing capability, machine learning models could be used to detect patterns, odd behavior, or other interesting observations in recorded sessions. Some IIR research has been done using eye-tracking [11, 33], so recording and analyzing a user's webcam in addition to their screen might improve analysis of user behavior.

An extension on the dashboard to have it also make predictions, for example about a user's result evaluation strategy [29] and human performance [30], and include *indirect observable* metrics would be beneficial to IIR researchers as well. Also, including additional visuals, like the Sankey diagram and correlation matrix explained in section 2.3, would make it practicable for a wider variety of researchers.

At the moment, the dashboard takes a significant time to load when a flight contains many logs. Its contents aren't cached which means that switching back and forth between the dashboard and *Session Dashboard* accumulates to a lot of waiting time. Speeding up the fetching and calculation process, and caching the contents should result in a much better user experience. The dashboard also does not update in real-time, meaning that when new data comes in the page had to be reloaded for it to appear in the dashboard. Adding an interval-based fetching process might also benefit LogUI users.

REFERENCES

[1] [n.d.]. Documents. https://docs.mongodb.com/manual/core/document/
[2] [n.d.]. GridFS. https://docs.mongodb.com/manual/core/gridfs/
[3] [n.d.]. Plotly JavaScript Graphing Library. https://plotly.com/javascript/
[4] [n.d.]. Screen Capture API - Web APIs: MDN. https://developer.mozilla.org/en-US/docs/Web/API/Screen_Capture_API
[5] [n.d.]. Screen Resolution Stats Worldwide. https://gs.statcounter.com/screen-resolution-stats
[6] [n.d.]. The WebSocket API (WebSockets) - Web APIs: MDN. https://developer.mozilla.org/en-US/docs/Web/API/WebSockets_API
[7] 2021. Binary-to-text encoding. https://en.wikipedia.org/wiki/Binary-to-text_encoding
[8] Kate Beard, Heather Deese, and Neal R Pettigrew. 2008. A framework for visualization and exploration of events. *Information Visualization* 7, 2 (2008), 133–151.
[9] Juraj Bienik, Miroslav Uhrina, Michal Kuba, and Martin Vaculik. 2016. Performance of H. 264, H. 265, VP8 and VP9 Compression Standards for High Resolutions. In *2016 19th International Conference on Network-Based Information Systems (NBiS)*. IEEE, 246–252.
[10] Pia Borlund. 2013. Interactive Information Retrieval: An Introduction. *Journal of Information Science Theory and Practice* 1 (09 2013). https://doi.org/10.1633/JISTaP.2013.1.3.2
[11] Georg Buscher, Andreas Dengel, Ralf Biedert, and Ludger V Elst. 2012. Attentive documents: Eye tracking as implicit feedback for information retrieval and beyond. *ACM Transactions on Interactive Intelligent Systems (TiiS)* 1, 2 (2012), 1–30.
[12] W. Bruce. Croft, Donald Metzler, and Trevor Strohman. 2010. *Search engines: information retrieval in practice.* Addison-Wesley.
[13] Stephen Few. 2006. *Information dashboard design.* OReilly.
[14] Dan Grois, Detlev Marpe, Amit Mulayoff, Benaya Itzhaky, and Ofer Hadar. 2013. Performance comparison of h. 265/mpeg-hevc, vp9, and h. 264/mpeg-avc encoders. In *2013 Picture Coding Symposium (PCS)*. IEEE, 394–397.
[15] Frederick Hartwig and Brian E Dearing. 1979. *Exploratory data analysis.* Number 16. Sage.
[16] Claudia Hauff. 2021. Logging events · Issue 38 · searchx-framework/searchx-frontend. https://github.com/searchx-framework/searchx-frontend/issues/38#issuecomment-811865806
[17] Diane Kelly. 2009. Methods for Evaluating Interactive Information Retrieval Systems with Users. *Foundations and Trends in Information Retrieval* (Jan 2009). https://doi.org/10.1561/9781601982254
[18] Muaz Khan. [n.d.]. RecordRTC.js: Live Demo. https://recordrtc.org/
[19] Muaz Khan and Thijs Triemstra. 2017. muaz-khan/RecordRTC. https://github.com/muaz-khan/RecordRTC
[20] Muaz Khan, Vladimir Yazykov, Anders Stalheim Øfsdahl, Philipp Hasper, and Rhema Linder. 2017. streamproc/MediaStreamRecorder. https://github.com/streamproc/MediaStreamRecorder
[21] David Maxwell. 2021. Configuration-Object. https://github.com/logui-framework/client/wiki/Configuration-Object
[22] David Maxwell and Claudia Hauff. 2021. LogUI: Contemporary Logging Infrastructure for Web-Based Experiments. In *Advances in Information Retrieval (Proc. ECIR)*. 525–530.
[23] Felipe Rikarno Moraes, Sindunuraga Rikarno Putra, Nirmal Rikarno Roy, and Kilian Rikarno Grashoff. 2019. searchx-framework/searchx-frontend/README.md. @misc{moraes_putra_roy_grashoff_2019,title={searchx-framework/searchx-frontend},url={https://github.com/searchx-framework/searchx-frontend},journal={GitHub},author={Moraes,FelipeandPutra, SindunuragaRikarnoandRoy,NirmalandGrashoff,Kilian},year={2019},month={Nov}}
[24] Sindunuraga Rikarno Putra, Felipe Moraes, and Claudia Hauff. 2018. SearchX: Empowering Collaborative Search Research.. In *SIGIR*. 1265–1268.
[25] Gareth Renaud and Leif Azzopardi. 2012. SCAMP: a tool for conducting interactive information retrieval experiments.. In *IIiX*. 286–289.
[26] N. Roy, A. Câmara, D. Maxwell, and C. Hauff. 2021. Incorporating Widget Positioning in Interaction Models of Search Behaviour. In *Proc. 7^{th} ACM ICTIR.*
[27] Gerard Salton and Donna Harman. 2003. *Information Retrieval.* John Wiley and Sons Ltd., GBR, 858–863.
[28] Nick Schonning, Peter Bengtsson, Schalk Neethling, Jean-Yves Perrier, Florian Scholz, Anton Bershanskiy, and Ben Bucksch. 2021. MediaRecorder() - Web APIs: MDN. https://developer.mozilla.org/en-US/docs/Web/API/MediaRecorder/MediaRecorder
[29] Mark D Smucker. [n.d.]. An Analysis of User Strategies for Examining and Processing Ranked Lists of Documents. ([n. d.]).
[30] Mark D Smucker. 2009. Towards timed predictions of human performance for interactive information retrieval evaluation. *Proc. of HCIR* (2009), 3.
[31] Renato Toasa, Marisa Maximiano, Catarina Reis, and David Guevara. 2018. Data visualization techniques for real-time information — A custom and dynamic dashboard for analyzing surveys results. *2018 13th Iberian Conference on Information Systems and Technologies (CISTI)* (2018). https://doi.org/10.23919/cisti.2018.8398641
[32] Elaine G Toms, Luanne Freund, and Cara Li. 2004. WiIRE: the Web interactive information retrieval experimentation system prototype. *Information Processing & Management* 40, 4 (2004), 655–675.
[33] Vu Tuan Tran and Norbert Fuhr. 2012. Using eye-tracking with dynamic areas of interest for analyzing interactive information retrieval. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in Information Retrieval*. 1165–1166.
[34] John W Tukey. 1977. *Exploratory data analysis.* Addison Wesley Publishing Company.
[35] user15175033 and Kaiido. 2021. Why does screen recording of the entire screen vs a window vs a single tab result in significantly different file sizes? https://stackoverflow.com/questions/67556335/why-does-screen-recording-of-the-entire-screen-vs-a-window-vs-a-single-tab-resul
[36] H.J.P. van Dijk. 2021. LogUI client. https://github.com/hjpvandijk/client.
[37] H.J.P. van Dijk. 2021. LogUI server. https://github.com/hjpvandijk/server.
[38] H.J.P. van Dijk. 2021. SearchX frontend. https://github.com/hjpvandijk/searchx-frontend.

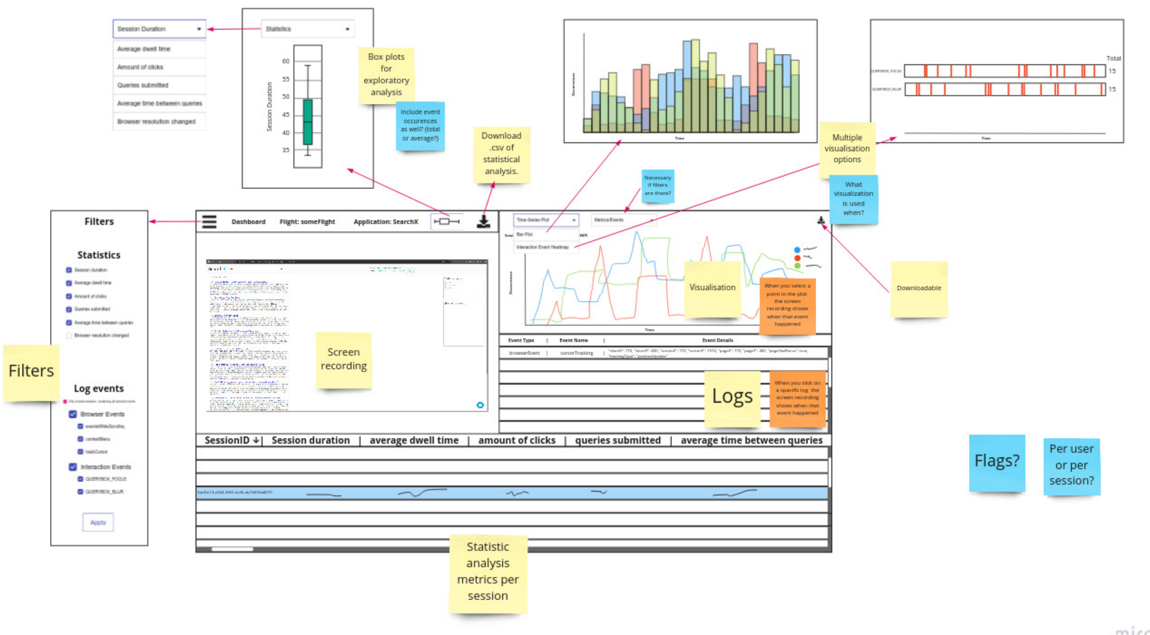
A REQUIREMENTS ELICITATION SURVEY QUESTIONS

- Could you describe the scenarios in which you use user logs gathered through LogUI and how they are used in your work?
- What type of metrics (like cursor location, elements clicked or hovered over, dwell time, idle time, etc.) do you typically look at when analyzing LogUI logs?
- What type of visualizations (if any) do you typically create for analysis of user logs?
- What type of conclusions do you try to make from the user logs? Are there specific predictions you try to make or aspects of SearchX you try to evaluate? Please also describe how you would come to these conclusions/predictions (for example by relating them to the aforementioned metrics).
- I'm also working on adding a screen capture feature to LogUI, which will capture the user's screen during (specific moments of) a session. These videos should be available just like how the logs are downloaded in the current version of the framework, while clearly showing to which session and logs they correspond. A possible addition to this, if time

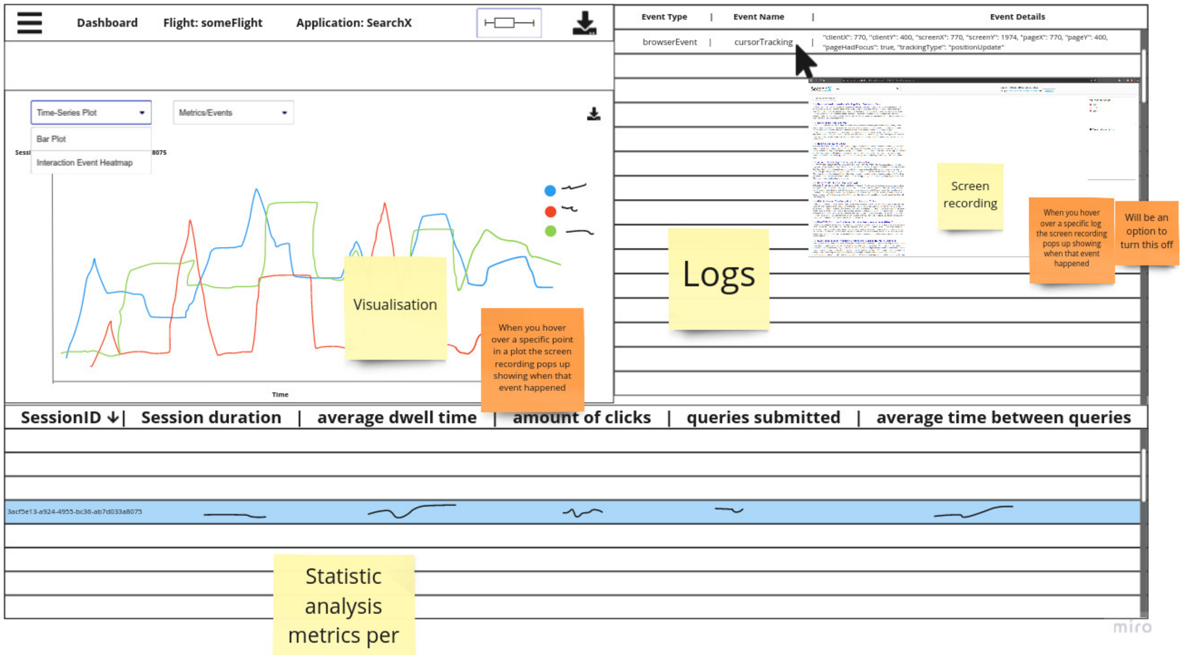
permits, would be flagging specific captures in case of special user events (like when there is no movement for a few minutes). How would you see yourself using these videos and what would you expect from this feature?

- In what way would it be beneficial to you that these captures are integrated into the dashboard?
- In general, what would you wish for in a dashboard and how do you see yourself using it in your work.
- Do you have any general remarks or recommendations you think might help me in this design process?

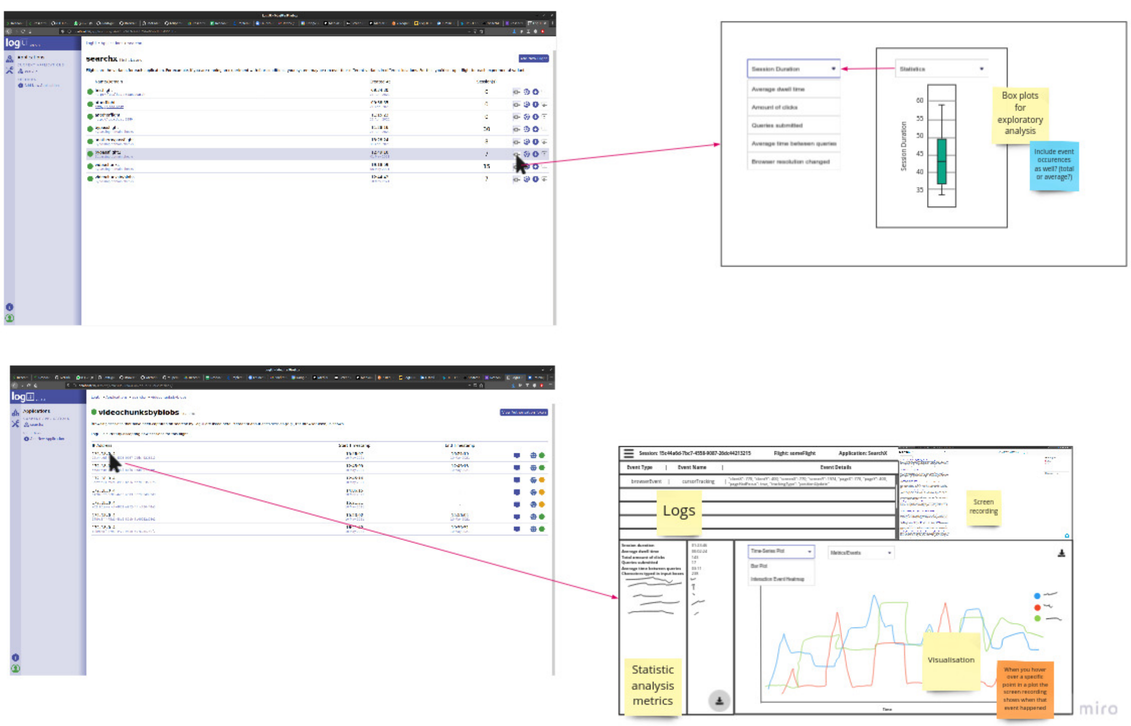
B DASHBOARD PROTOTYPES



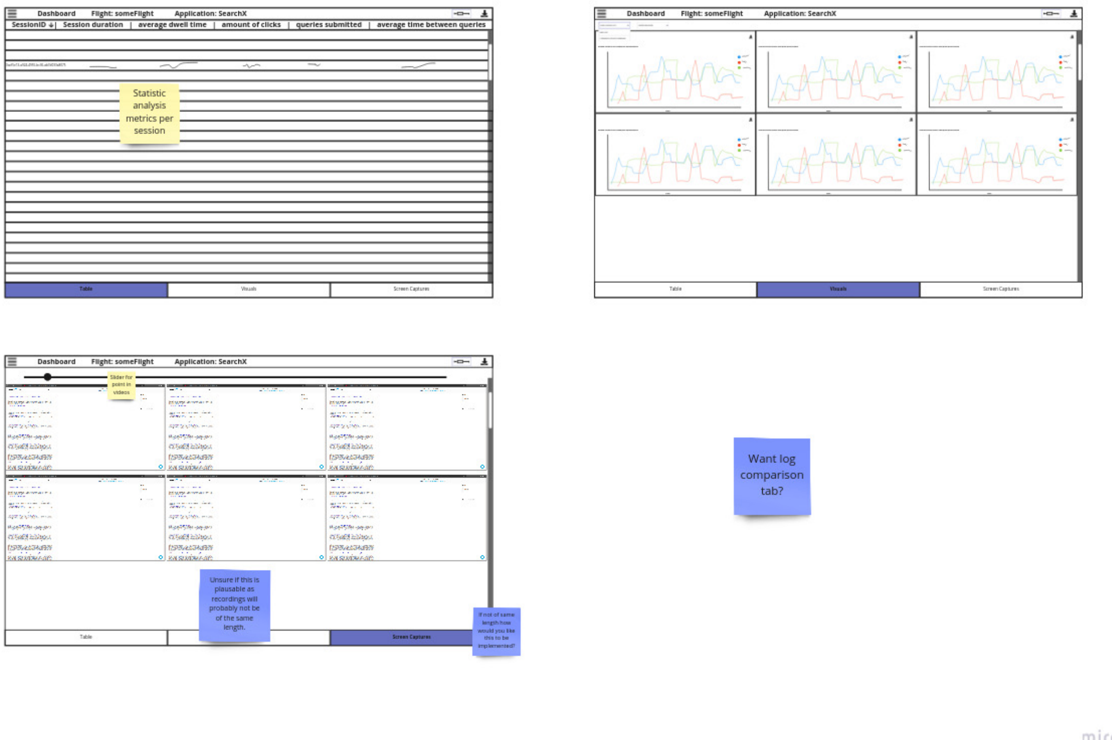
(a) LogUI dashboard prototype 1



(b) LogUI dashboard prototype 2



(c) LogUI dashboard prototype 3



(d) LogUI dashboard prototype 4

Figure 9: LogUI dashboard prototypes

C DASHBOARD EVALUATION TASKS AND QUESTIONS

Evaluation LogUI dashboard Firstly go to: http://80.112.171.123/ in your browser and log in with the following credentials username: hugo password: rp_project

Here is some LogUI terminology in case you are unfamiliar with it: Application: The application (SearchX in your case) that experiments are performed on. Flight: A flight is an aggregation of sessions for a specific experiment (with a certain condition/feature added for example). Session: A specific user session within a flight. Whenever a user loads into a web page tracked by LogUI, this user is assigned a specific session ID. However, when doing collaborative search within SearchX, all collaborating users share the same LogUI session ID.

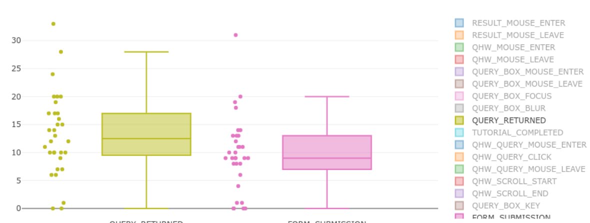
Go to Manage Applications, and click on the SearchX-QHW application. Go to the dashboard of the april25-variantTL flight by clicking the icon highlighted below in that row. The dashboard might 15-30 seconds to load.

Task 1: Exclude the dwell_time_total and dwell_time_average metric from the dashboard.

- How easy was it to complete task 1?
- If it was difficult to complete this task please explain why.

Task 2: Create two box plots for the QUERY_RETURNED event and the FORM_SUBMISSION event side by side. and download this plot.

You should get something that looks like this.



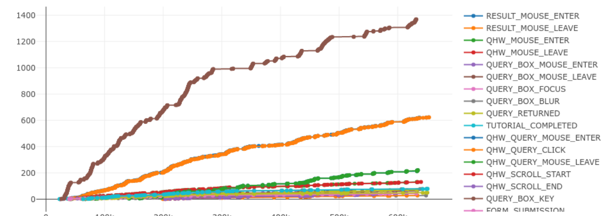
- How easy was it to complete task 2?
- If it was difficult to complete this task please explain why.

difficult to complete this task please explain why.

Task 3: Create a time series plot with all events, only including the following four sessions:

```
0592c8e9-e65a-4bab-af88-7f3eaa1b2cd0
f7da1af3-9703-4805-b9ae-6ca6757af465
da10c17d-1b8b-443e-8ec1-ca73b6b013cf
e9cffee1-b6c9-4d78-ade4-e71d93679239
```

You should get something that looks like this.



- How easy was it to complete task 3?
- If it was difficult to complete this task please explain why.

Task 4: While still having selected the same group of sessions, select the event timeline. Find out which session completed the tutorial first and what the first logged event after tutorial completion was.

- How easy was it to complete task 4?
- If it was difficult to complete this task please explain why.

Task 5: Download a .csv file of the bottom table, only the three metrics: RESULT_MOUSE_ENTER, RESULT_MOUSE_LEAVE, and QHW_MOUSE_ENTER. The .csv file should look like this:

```
sessionID ,RESULT_MOUSE_ENTER,RESULT_MOUSE_LEAVE,QHW_MOUSE_ENTER
0655e035-c55b-4444-aa5e-419fa35be2af ,0 ,0 ,0
0592c8e9-e65a-4bab-af88-7f3eaa1b2cd0 ,124 ,124 ,14
27b7e70f-b136-42ce-85e1-b74e3824235a ,46 ,46 ,10
f7da1af3-9703-4805-b9ae-6ca6757af465 ,94 ,94 ,41
59d5fcb3-f88a-4011-85e9-25247eeb43a2 ,17 ,17 ,7
8fa047c9-2a14-470b-b275-5fc11fb50bd0 ,0 ,0 ,0
31558038-089f-490c-8548-bfd852823026 ,0 ,0 ,1
81235e0f-9216-450a-9a20-41acf5ba4e39 ,54 ,54 ,20
da10c17d-1b8b-443e-8ec1-ca73b6b013cf ,194 ,194 ,37
8abfae1a-1f49-4d3d-a1bf-a3043f99be23 ,227 ,227 ,34
e9cffee1-b6c9-4d78-ade4-e71d93679239 ,211 ,211 ,39
141b92c0-2d10-4fe4-aa63-fecd1de1ee96 ,226 ,225 ,44
09f44511-87a0-4bad-a3c4-9d05aff587f3 ,0 ,0 ,0
96b69529-45e1-43c8-9bb4-79f184b1b6b9 ,98 ,98 ,18
f72e089d-e8a9-4da2-ba68-8e1e3b0b1234 ,153 ,153 ,30
36f319f7-1aae-4d9a-b6e5-7b64233956b7 ,194 ,193 ,45
66f08d95-54c8-48e7-87df-d77f65800360 ,176 ,176 ,18
0aae05e0-f050-4cf5-855f-de7ce70dc5e9 ,178 ,177 ,44
a5fca6b1-fd71-43e7-8c00-4d584ef94a8c ,70 ,70 ,28
6a9eff8f-8811-480e-bdc5-943a894bafd0 ,82 ,82 ,18
5aaefc3e-3cb2-4c16-b4ea-b82224902324 ,211 ,211 ,43
e81e57ca-9a30-42ee-af09-0f576b902b90 ,235 ,235 ,50
531e841b-6507-408e-af0e-54b93c568218 ,81 ,81 ,23
c9c25e11-734d-47bf-8edd-a609a340aa49 ,71 ,71 ,10
3db939ae-1498-488f-ab3d-99213dbe741c ,136 ,136 ,28
2ff3856d-444a-4377-9f1b-ec996e94306c ,100 ,100 ,24
759ea38d-5357-464f-bf44-a625aba31ef5 ,152 ,151 ,27
c1e0fd9b-a58c-4a64-85c9-2a68cae7b7fe ,47 ,47 ,12
afdca406-aad5-4a0a-94e8-eaed7896135f ,16 ,16 ,6
0a5387df-1cf4-487b-a448-9ef9352a376d ,83 ,83 ,21
eba7dc9c-62a6-4c9d-82b2-b386587e509a ,109 ,109 ,20
ba65e91c-e72d-4524-adea-242d045e365e ,293 ,291 ,54
fad45eb3-03d1-4bbb-b59e-be7b1d586af6 ,110 ,110 ,22
b9189922-ec9a-4746-8f6a-4b38e94a65f8 ,384 ,384 ,76
dbe713d0-5c02-43bb-9aff-6435d6d97af9 ,0 ,0 ,0
65f08b23-0c9f-42fa-8ce2-39febe549915 ,0 ,0 ,0
e0ef4586-6bc9-4ce2-ab1a-50264e57f5ea ,0 ,0 ,0
abb88f2f-d518-41bb-9112-b4e9d0df2d1f ,0 ,0 ,0
df5c0418-a5f3-46cc-9e6c-1a372444abb3 ,0 ,0 ,0
c6962231-fd26-4f4d-ae09-2e6e2bc1f986 ,0 ,0 ,0
b498ad9b-7c5f-4c84-a4ca-09436ee0bee4 ,0 ,0 ,0
70a85461-bf09-4ac2-bc5b-fb332a7a0759 ,0 ,0 ,0
5b168e32-8adf-4ca4-8f2a-d41f0eb29009 ,0 ,0 ,0
c563ea67-0cfc-49a9-a0de-baa0db61bf63 ,0 ,0 ,0
74bcd2a7-c956-4f63-973c-2635775e190d ,0 ,0 ,0
```

- How easy was it to complete task 5?
- If it was difficult to complete this task please explain why.

Go back to the flight overview and go to the dashboard of the "evaluation" flight and go to the session dashboard of the only session in that flight.

Task 6: Find out through the screen capture which result was clicked on for the "MODAL_DIALOG_SHOW" event to be logged.

- How easy was it to complete task 5?
- If it was difficult to complete this task please explain why.

- What features of the dashboard do you find useful, and why?
- What features of the dashboard do you not find useful, and why?

- Would you use the dashboard in your future research?