DEPARTMENT OF INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Master's Thesis in Human Factors Engineering

# Software Ergonomics of Free and Open Source Software: Augmenting the Usability of JabRef Through User-Centered Design

Martin Simon

DEPARTMENT OF INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Master's Thesis in Human Factors Engineering

# Software Ergonomics of Free and Open Source Software: Augmenting the Usability of JabRef Through User-Centered Design

# Softwareergonomie von Freier Open Source Software: Verbesserung der Usability von JabRef mittels User-Centered Design

| | |
|---|---|
| Author: | Martin Simon |
| Supervisor: | Prof. Dr.-Ing. Jörg Ott |
| Advisor: | Linus Dietz, M.Sc. |
| Submission Date: | 15.01.2019 |

I confirm that this master's thesis in human factors engineering is my own work and I have documented all sources and material used.

Garching, 15.01.2019 Martin Simon

# Abstract

The goal of this thesis is to improve the usability of JabRef, a free and open source reference management software, which processes references in the BibTeX format and is mainly targeted at LaTeX users. To achieve that goal, we proceeded in accordance with the user-centered design process by involving users in our work and employing principles of software ergonomics. As the first step, we identified the most important features of JabRef from a user perspective, through an online questionnaire with 101 respondents, which revealed that JabRef's entry editor and group feature should be prioritized. We then performed a laboratory study with six participants, consisting of a thinking aloud experiment for both features and a card sorting task targeting the parts of the entry editor. Lastly, we performed an expert assessment of the JabRef interface to identify more general usability problems, but it also led to the main menu being prioritized in our future work. On the basis of our findings, suggestions for changes in JabRef's main menu, entry editor, and group feature were made in accordance with usability principles. To support the recommendations, high fidelity prototypes of the reworked user interface parts were created. To integrate our findings into the open source development process, we presented them in a standardized digital format on Github and divided redesign suggestions into multiple smaller packages.

# Kurzzusammenfassung

Das Ziel dieser Arbeit ist es, die Usability von JabRef zu erhöhen, einem Open-Source-Literaturverwaltungsprogramm, welches Zitationen im BibTeX-Format verwaltet und sich hauptsächlich an LaTeX-Nutzer richtet. Um dieses Ziel zu erreichen, gingen wir nach dem Prozess des *User-centered Design* vor, indem wir Nutzer in unsere Arbeit einbanden und Prinzipien der Softwareergonomie anwendeten. Im ersten Schritt unserer Arbeit identifizierten wir die wichtigsten Funktionsbestandteile von JabRef mittels eines Onlinefragebogens mit 101 Teilnehmern, dessen Ergebnisse offenbarten, dass der Entry Editor und die Gruppenfunktion von JabRef priorisiert werden sollten. Danach führten wir mit weiteren sechs Teilnehmern eine Laborstudie durch, die aus einem Thinking-Aloud-Experiment mit beiden der priorisierten Funktionen und einem Card Sorting für die Teile des Entry Editors bestand. Zuletzt nahmen wir eine Expertenbewertung der Benutzeroberfläche von JabRef vor, um weitere, allgemeine Usabilityprobleme zu identifizieren. Des Weiteren führte sie dazu, dass JabRefs Hauptmenü in unserer weiteren Arbeit priorisiert wurde. Auf Basis unserer bisherigen Ergebnisse wurden Änderungsvorschlage bezüglich des Hauptmenüs, des Entry Editors und der Gruppenfunktion gemacht, die sich nach Prinzipien der Usability richteten. Zur Unterstützung dieser Vorschläge wurden high-fidelity Prototypen der neu gestalteten Oberflächen erstellt. Um unsere Ergebnisse in den Open-Source-Entwicklungsprozess zu integrieren, präsentierten wir diese in einem standardisierten digitalen Format auf Github und teilten die Vorschläge zur Neugestaltung auf mehrere kleinere Arbeitspakete auf.

# Contents

# 1. Introduction

The statistical programming language *R*, the web browser Mozilla Firefox, and the text processing software OpenOffice are examples for free open source software (FOSS). All three of them represent cases, where an open source project managed to challenge its corporate competitors. There are many reasons aside from monetary ones why FOSS is chosen over the more expensive alternatives, like higher flexibility or the faster implementation of new features. A reason, which would probably not be listed in that context, is the higher usability of open source programs. On the contrary, usability is an attribute often overlooked in most communities of open source developers, which is also the case in the community of JabRef.

JabRef is a free open source reference management software (RMS) written in Java, first released in 2002. It saves and manages references in the BibTeX format, which is supported by LaTeX and, therefore, is used mostly in a university environment. Features which set it apart from its competitors include its capacity for big libraries, a large selection of quality assurance tools for entries, and an active support forum. While the JabRef community has an active group of contributors and maintainers, who continue to improve the software and add new functions, the usability of its user interface has never been evaluated professionally and no efforts in systematically improving it have been undertaken. Therefore, we do pioneer work in this thesis by evaluating and improving JabRef's UI with the methods of user-centered design, which can have multiple positive effects.

Most importantly, software with high usability tends to attract more users and to lose less of them, which in turn would most likely improve the reputation of JabRef and raise the number of developers who are interested in working on it. Additionally, the users would be more satisfied with JabRef's UI and would be able to use it with their own set of skills. This would cause the number of complaints filed with the support to go down, leaving the developers to focus on more pressing matters.

Alas, improving usability seems to have positive effects in every regard, which raises the question how we can measure and change it. The method of user-centered design provides a good development process model of how to design a highly usable system. It offers three major steps in the user-centered development process which are relevant for us: the analysis of the current status of a system, the specification of usability requirements which the system should fulfill, and the implementation of these requirements in a prototype.

However, JabRef presents us with additional challenges in our work. We have to consider that it is not a corporate product, which is developed by an organized team of workers but community software, to which numerous people contribute out of their own volition. Furthermore, JabRef is not a newly developed product but an existing software with countless functions, dialogs and other elements, which could all be subjected to our examination. However, the scope of this thesis limits the amount of work which can be done, raising the need to prioritize certain functions of JabRef before we begin our analysis.

Taking these considerations into account, the work of this thesis consists of four essential tasks. After reviewing literature on usability in open source software (OSS) and RMS, the first task will be the prioritization of JabRef's functions. Our second task will be the analysis of the highly prioritized functions and the examination of the way in which JabRef's users interact with them. In the third step, we will create requirements and recommendations for the improvement of JabRef's usability based on the knowledge that we have gathered, and the fourth task will be to implement these recommendations in a prototype of JabRef's user interface.

# 2. Literature Review

Looking at scientific literature, there is a vast corpus of findings regarding usability in OSS projects. In the following, we present challenges arising from the special work environment of those projects. Furthermore, we look at how to tackle these challenges, while maintaining the core aspects of user-centered design. The second part of this chapter focuses on the functionality and interfaces of current RMS in general, and on JabRef's in particular. The third part surveys different strategies of usability analysis techniques.

## 2.1. Usability in Open Source Projects

Literature consistently states that there might be a *usability problem* in open source communities, which describes the lacking user friendliness of OSS and the difficulties of bringing usability into OSS projects. Therefore, if we want to conduct a usability analysis and try to enhance the interface of OSS, firstly, we have to understand the special standing of usability in OSS projects. To do so, we examine the literature, to understand the usability problem in OSS. We also look at previous suggestions on how to overcome those challenges, so our endeavor can be successful.

### 2.1.1. Problems and Challenges

One problem of OSS is the knowledge gap between the developers of the software and its end users. While in the beginning OSS was developed *by hackers for hackers*, its target group now also encompasses a large group of casual end users, who speak an entirely different language from the IT terminology and do not recognize technical terms that are routinely used by developers (Nichols and Twidale, 2003). Moreover, they often lack the technical knowledge to cope with the complexity of some programs, while this is no problem for the more tech-savvy contributors. Therefore, software complexity is not addressed in the development process (Nichols and Twidale, 2003). Even if end users try to alert developers of usability issues, they often do not have the necessary means to do so effectively, as there are almost no low level reporting tools for usability issues (Viorres et al., 2007). Bug trackers exist mainly for functional problems and are often too complex to be used by casual users of software. Thus, the existing blind spots of developers cannot be covered by the additional viewpoint of an end user.

Apart from the lack of end user feedback, usability experts also tend to not get involved in OSS projects. There are probably two main reasons for this, the first being the lack of incentives, which a OSS project holds for usability professionals (Nichols and Twidale, 2003). The sense of achievement gained from implementing a new feature in a program is much higher than the one of improving the usability of a user interface, as those changes are often less visible. This intertwines with the second reason for software ergonomists to avoid OSS projects, whose social hierarchies resemble a meritocraty and are based on the past achievements of the contributor alone (Despalatović, 2013). Due to the subtle nature of usability changes, usability experts' contributions are welcome but often not rated as valuable as functional ones, leaving them with a low social standing in OSS communities (Andreasen et al., 2006).

The mentioned prioritization of functionality over usability is another problem specific to OSS. As Nichols and Twidale (2003) state, "developers work on the topics that interest them and this may well not include features for novice users" ("The incentives in OSS", para. 1). Adding functional features to a piece of software proves to be more rewarding for OSS contributors and is, therefore, chosen more frequently than increasing the usability of existing software, as developers themselves confirm (Andreasen et al., 2006). Furthermore, the decentralized way of OSS development requires a lot of modular coding, which often threatens the consistency of a single program and causes confusion for the user (Viorres et al., 2007). This is further fueled by the high frequency of unstable versions of software, which are often preferred by software veterans because of their enhanced functionality, but they cause distress for novice users, who are not able to cope with possible system crashes. While a roughly coded application might be sufficient for contributors themselves, it is too difficult to use for casual end users (Nichols and Twidale, 2003).

Another problem is that usability bugs are harder to discuss than functionality bugs and are often very time expensive to fix. This is especially the case in OSS projects, as the decentralized development further impedes the discussion of usability bugs. As Twidale and Nichols (2005) found when analyzing posts on Mozilla's bug tracker, usability bugs were often too dynamic and contextual for a simple issue tracker and can sometimes not even be sufficiently displayed with visuals. Andreasen et al. (2006) also discuss the notion of usability issues being different from functionality bugs and give the option of creating a separate infrastructure for them. The difficulty of fixing usability bugs lies in their often global nature. To fix an issue, it might be necessary to overhaul the entire interface of an application for the sake of consistency (Nichols and Twidale, 2003). This proves especially difficult for the modular contribution approach of OSS.

The main reason which managers of OSS projects name for the lack of usability efforts made in their projects is the lack of financial resources (Andreasen et al., 2006). As the work done in these projects is mainly voluntary and funding is in most cases very sparse, project leaders can not hire outside professionals for support (Nichols and Twidale, 2003). This becomes visible, when looking at the statistics obtained by Andreasen et al. (2006). Only 29% of the projects examined by them hired experts from the outside to evaluate their usability. 21% used remote usability evaluations and only 8% hired a usability laboratory.

### 2.1.2. Special Assessment Methods

As stated above, the disparity in the views of developers and casual end users leads to a lot of usability problems. The only way to avoid this seems to be the involvement of end users in the development process. There are some examples in literature where the evaluation of usability by the user was undertaken in OSS. Faaborg and Schwartz (2010) developed a distributed version of the heuristic evaluation, which was tested on Mozilla's bug tracker. Zhao et al. (2010) used exploratory inspection, which was meant to increase users knowledge of usability while they evaluated the system.

To get usability experts to be part of an OSS project and solve the problem of lacking resources, instead of hiring professionals from the outside, developers could collaborate with universities and other academical institutions. Especially students are often eager to partake in projects where they can increase their skills and have an impact on a real product at the same time (Nichols and Twidale, 2003). These incentives also hold potential to attract volunteering usability experts, as Andreasen et al. (2006) found out in an interview.

The problem of high complexity of usability bugs can hopefully be solved by better communication methods for usability issues and results of usability evaluations. An effort in that direction has been made with the markup language UsabML, which tries to convey usability findings that can automatically be imported into a bug tracker and which are more concise than a written report (Feiner and Andrews, 2012).

## 2.2. Reference Management Software

To improve the usability of JabRef, we first have to look at the functionality of current RMS to understand the workflow of their users. We examine the user interfaces of those programs to identify the ways of human-machine interaction on the screen. We then explore the functionality and UI of JabRef to thoroughly understand the software which we work with in this paper, and to be able to compare it to its competitors.

### 2.2.1. Current Functionality

Regarding functionality we can roughly categorize RMS into different types, reflecting the purpose for which the tool was built. A taxonomy of RMS has been described by Marino (2012), who evaluated different RMS on two dimensions. The first dimension was, whether the tool was desktop-based with a client independent from the web, or web-based and only accessible via an Internet connection. They also added a third category of hybrid systems, which have a standalone client as well as an Internet version and have the ability to save bibliographies locally and in the cloud (Marino, 2012). The fourth category were browser-based systems, which were directly integrated into a browser. With Zotero becoming a hybrid application, however, there are currently no members in that category.

The second dimension of categorization focuses more on the power and the purpose of the RMS. It discerns four different types of applications, two of which we omit here, because they are not fully fledged reference managers. These are bibliography generators lacking the saving and organizing capabilities of more powerful tools, and social bookmarking tools, which offer sharing of references, but often cannot generate them in a specific style for immediate use (Marino, 2012). The two remaining categories both focus on traditional tools, which describes RMS with a large set of features and multiple use cases. These tools can be divided into applications that have limited or no social capabilities, and RMS that allows for extensive sharing and collaborative work.

After categorizing the tools that we want to examine, we need to asses their quality. As RMS vary in type, they also vary in functionality, which makes it hard to define a set of necessary features. Gilmour and Cobus-Kuo (2011) specify eight functions that they expect from RMS but also state that not every tool can fulfill those requirements. A second evaluation from Fenner (2010) uses categories resembling eight of the requirements from Gilmour and Cobus-Kuo (2011), and adds some more advanced ones like full text search and integrated PDF viewers.

Table 2.1.: Features of three different RMS in comparison to JabRef

| Feature | Zotero | Citavi | RefWorks | JabRef |
|---|---|---|---|---|
| Search databases and websites | × | ○ | ○ | ○ |
| Import citations from databases | ○ | ○ | ○ | ○ |
| Organize citations | ○ | ○ | ○ | ○ |
| Annotate citations | ○ | ○ | ○ | ○ |
| Gather metadata from PDF | ○ | ○ | ○ | ○ |
| Search full text of PDF | × | ○ | × | × |
| View PDF | × | ○ | ○ | ○ |
| Share references | ○ | ○ | ○ | × |
| Share PDF | ○ | ○ | × | × |
| Export citations | ○ | ○ | ○ | ○ |
| Produce citations in certain style | ○ | ○ | ○ | ○ |
| Work with word processors | ○ | ○ | ○ | ○ |

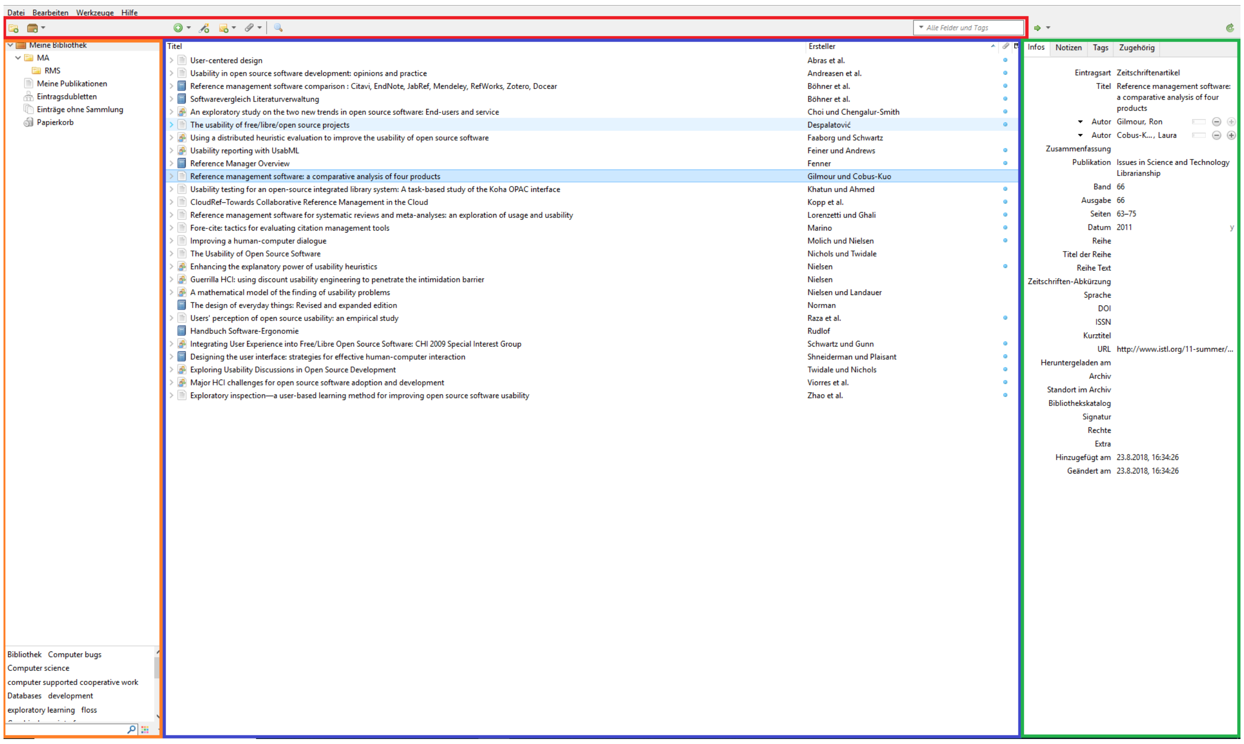
○ = Feature is available; ×= Feature is not available.

While these sources hold good criteria for assessing the quality of an RMS, the information they give is outdated, as software applications can change substantially in a short period of time. The most extensive overview of reference managers and their capabilities can probably be found on Wikipedia (Wikipedia contributors, 2018). Another recent comparative list including less applications, which is easier to read, is the one created by Böhner et al. (2018). A further current source is the comparison chart of the University of Illinois, that summarizes important features of five RMS (UIC, 2018).

To give a better example of the functionality of current RMS, we have chosen three applications, each representing a certain type according to the first dimension of our taxonomy: Citavi is a desktop-based software with limited social functions, RefWorks is a web-based tool with only extremely basic social functions, and Zotero is a hybrid tool with extended social functions. An overview of the functionality of these programs taking into account the expected functionality according to Gilmour and Cobus-Kuo (2011) as well as the feature list of Fenner (2010) is given in Table 2.1. The result shows that only Citavi is able to fulfill all of the listed requirements. Both Zotero and RefWorks lack the capabilities to interact with PDF-files attached to references. Additionally, in Zotero the user is not able to directly search databases from within the application.

### 2.2.2. Current User Interfaces

The sources regarding functionality cited above make only few statements on usability or user interaction of RMS. The UIC (2018) attests Zotero and RefWorks good learnability, but give no reason for it. Böhner et al. (2018) rate the usability of Zotero as *good* and Citavi's as *very good*. Wikipedia has no information on user interaction at all. Another study based on an interview with authors of meta analyses revealed some specific usability problems. Most frequent were problems with downloads of references and incorrectly formated output styles (Lorenzetti and Ghali, 2013).



| Titel | Ersteller |
|---|---|
| User-centered design | Abras et al. |
| Usability in open source software development: opinions and practice | Andreasen et al. |
| Reference management software comparison : Citavi, EndNote, JabRef, Mendeley, RefWorks, Zotero, Docear | Böhner et al. |
| Softwarevergleich Literaturverwaltung | Böhner et al. |
| An exploratory study on the two new trends in open source software: End-users and service | Choi und Chengalur-Smith |
| The usability of free/libre/open source projects | Despalatović |
| Using a distributed heuristic evaluation to improve the usability of open source software | Faaborg und Schwartz |
| Usability reporting with UsabML | Feiner und Andrews |
| Reference Manager Overview | Fenner |
| Reference management software: a comparative analysis of four products | Gilmour und Cobus-Kuo |
| Usability testing for an open-source integrated library system: A task-based study of the Koha OPAC interface | Khatun und Ahmed |
| CloudRef–Towards Collaborative Reference Management in the Cloud | Kopp et al. |
| Reference management software for systematic reviews and meta-analyses: an exploration of usage and usability | Lorenzetti und Ghali |
| Fore-cite: tactics for evaluating citation management tools | Marino |
| Improving a human-computer dialogue | Molich und Nielsen |
| The Usability of Open Source Software | Nichols und Twidale |
| Enhancing the explanatory power of usability heuristics | Nielsen |
| Guerrilla HCI: using discount usability engineering to penetrate the intimidation barrier | Nielsen |
| A mathematical model of the finding of usability problems | Nielsen und Landauer |
| The design of everyday things: Revised and expanded edition | Norman |
| Users' perception of open source usability: an empirical study | Raza et al. |
| Handbuch Software-Ergonomie | Rudlof |
| Integrating User Experience into Free/Libre Open Source Software: CHI 2009 Special Interest Group | Schwartz und Gunn |
| Designing the user interface: strategies for effective human-computer interaction | Shneiderman und Plaisant |
| Exploring Usability Discussions in Open Source Development | Twidale und Nichols |
| Major HCI challenges for open source software adoption and development | Viorres et al. |
| Exploratory inspection—a user-based learning method for improving open source software usability | Zhao et al. |

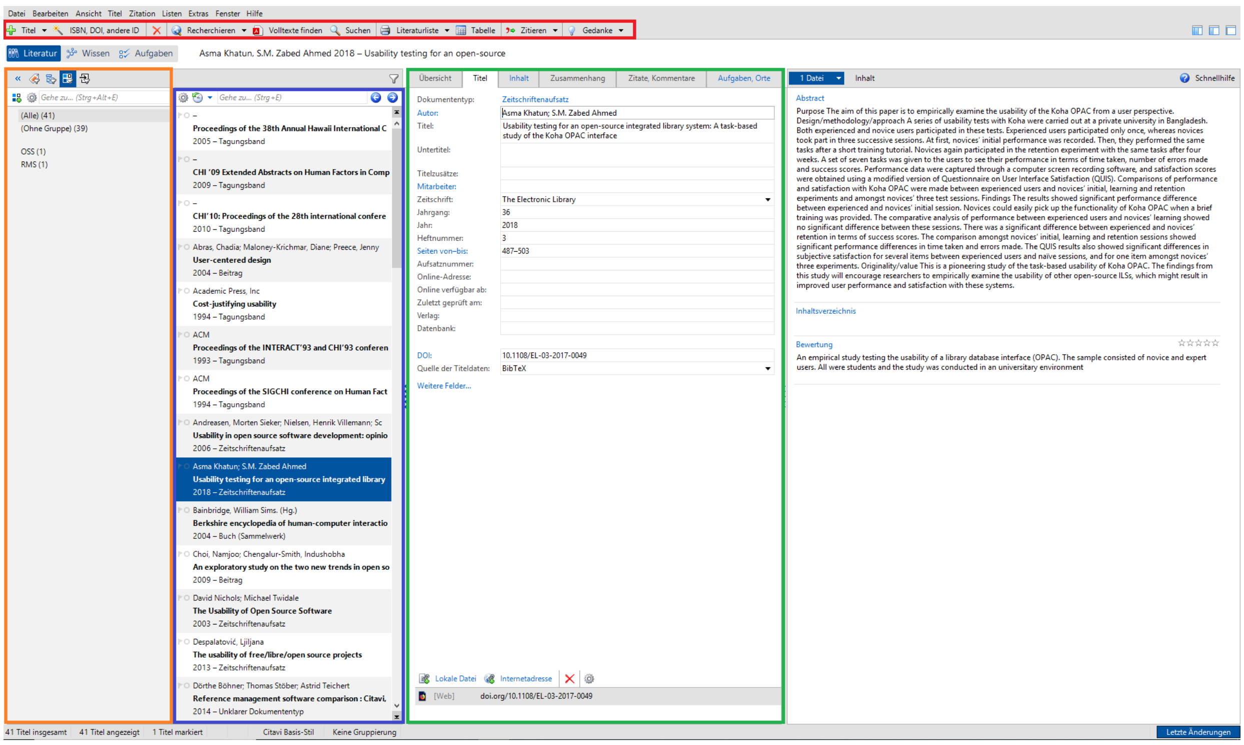
| | |
|---|---|
| Eintragsart | Zeitschriftenartikel |
| Titel | Reference management software: a comparative analysis of four products |
| Autor | Gilmour, Ron |
| Autor | Cobus-K..., Laura |
| Zusammenfassung | |
| Publikation | Issues in Science and Technology Librarianship |
| Band | 66 |
| Ausgabe | 66 |
| Seiten | 63–75 |
| Datum | 2011 |
| Reihe | |
| Titel der Reihe | |
| Reihe Text | |
| Zeitschriften-Abkürzung | |
| Sprache | |
| DOI | |
| ISSN | |
| Kurztitel | |
| URL | http://www.istl.org/11-summer/... |
| Heruntergeladen am | |
| Archiv | |
| Standort im Archiv | |
| Bibliothekskatalog | |
| Signatur | |
| Rechte | |
| Extra | |
| Hinzugefügt am | 23.8.2018, 16:34:26 |
| Geändert am | 23.8.2018, 16:34:26 |

Figure 2.1.: User interface of Zotero when bibliography entry is selected.

To evaluate the current designs of RMS user interfaces, we have to inspect the surface of our three example applications ourselves. All three share similar main elements in their layout. We will talk about these main elements in the following. Each application's main screen can also be viewed in the corresponding Figure (see Figure 2.1 – 2.3).

The colored areas on the screenshots represent one main element which has been marked in the same color in all figures. Consequently, the main elements of current reference managers are very similar to those of other products, meaning that a conventional interface structure exists. In red, we can see the toolbar, being arranged horizontally above the rest of the interface. In the middle of the screen (here in blue) is the list of references, which forms the core of Zotero and RefWorks. In Citavi, the reference list shares the visual center of the interface with the viewer and editor of the reference details (green). The other two applications have moved the detail viewer to the right of the screen. The last colored main element is the grouping menu to organize references (orange). All three reference managers put this feature on the far left of the interface in a vertical sidebar, with control options on its top. Citavi is the only one, where this sidebar is not opened by default and has a one-click-button to do so.

We would like to point out that all screenshots were taken with the default interface, which was provided when first opening the program, except for the sidebar in Citavi, which was made



Asma Khatun, S.M. Zabed Ahmed 2018 – Usability testing for an open-source

(Alle) (41)
(Ohne Gruppe) (39)

OSS (1)
RMS (1)

–
**Proceedings of the 38th Annual Hawaii International C**
2005 – Tagungsband

–
**CHI '09 Extended Abstracts on Human Factors in Comp**
2009 – Tagungsband

–
**CHI'10: Proceedings of the 28th international confere**
2010 – Tagungsband

Abras, Chadia; Maloney-Krichmar, Diane; Preece, Jenny
**User-centered design**
2004 – Beitrag

Academic Press, Inc
**Cost-justifying usability**
1994 – Tagungsband

ACM
**Proceedings of the INTERACT'93 and CHI'93 conferen**
1993 – Tagungsband

ACM
**Proceedings of the SIGCHI conference on Human Fact**
1994 – Tagungsband

Andreasen, Morten Sieker; Nielsen, Henrik Villemann; Sc
**Usability in open source software development: opinio**
2006 – Zeitschriftenaufsatz

Asma Khatun; S.M. Zabed Ahmed
**Usability testing for an open-source integrated library**
2018 – Zeitschriftenaufsatz

Bainbridge, William Sims. (Hg.)
**Berkshire encyclopedia of human-computer interactio**
2004 – Buch (Sammelwerk)

Choi, Namjoo; Chengalur-Smith, Indushobha
**An exploratory study on the two new trends in open so**
2009 – Beitrag

David Nichols; Michael Twidale
**The Usability of Open Source Software**
2003 – Zeitschriftenaufsatz

Despalatović, Ljiljana
**The usability of free/libre/open source projects**
2013 – Zeitschriftenaufsatz

Dörthe Böhner; Thomas Stöber; Astrid Teichert
**Reference management software comparison : Citavi,**
2014 – Unklarer Dokumententyp

| | |
|---|---|
| Dokumententyp: | Zeitschriftenaufsatz |
| Autor: | Asma Khatun; S.M. Zabed Ahmed |
| Titel: | Usability testing for an open-source integrated library system: A task-based study of the Koha OPAC interface |
| Untertitel: | |
| Titelzusätze: | |
| Mitarbeiter: | |
| Zeitschrift: | The Electronic Library |
| Jahrgang: | 36 |
| Jahr: | 2018 |
| Heftnummer: | 3 |
| Seiten von–bis: | 487–503 |
| Aufsatznummer: | |
| Online-Adresse: | |
| Online verfügbar ab: | |
| Zuletzt geprüft am: | |
| Verlag: | |
| Datenbank: | |
| DOI: | 10.1108/EL-03-2017-0049 |
| Quelle der Titeldaten: | BibTeX |

Weitere Felder...

[Web] doi.org/10.1108/EL-03-2017-0049

Abstract

Purpose The aim of this paper is to empirically examine the usability of the Koha OPAC from a user perspective. Design/methodology/approach A series of usability tests with Koha were carried out at a private university in Bangladesh. Both experienced and novice users participated in these tests. Experienced users participated only once, whereas novices took part in three successive sessions. At first, novices' initial performance was recorded. Then, they performed the same tasks after a short training tutorial. Novices again participated in the retention experiment with the same tasks after four weeks. A set of seven tasks was given to the users to see their performance in terms of time taken, number of errors made and success scores. Performance data were captured through a computer screen recording software, and satisfaction scores were obtained using a modified version of Questionnaire on User Interface Satisfaction (QUIS). Comparisons of performance and satisfaction with Koha OPAC were made between experienced users and novices' initial, learning and retention experiments and amongst novices' three test sessions. Findings The results showed significant performance difference between experienced and novices' initial session. Novices could easily pick up the functionality of Koha OPAC when a brief training was provided. The comparative analysis of performance between experienced users and novices' learning showed no significant difference between these sessions. There was a significant difference between experienced and novices' retention in terms of success scores. The comparison amongst novices' initial, learning and retention sessions showed significant performance differences in time taken and errors made. The QUIS results also showed significant differences in subjective satisfaction for several items between experienced users and naïve sessions, and for one item amongst novices' three experiments. Originality/value This is a pioneering study of the task-based usability of Koha OPAC. The findings from this study will encourage researchers to empirically examine the usability of other open-source ILSs, which might result in improved user performance and satisfaction with these systems.

Inhaltsverzeichnis

Bewertung ☆☆☆☆☆

An empirical study testing the usability of a library database interface (OPAC). The sample consisted of novice and expert users. All were students and the study was conducted in an universitary environment

41 Titel insgesamt | 41 Titel angezeigt | 1 Titel markiert | Citavi Basis-Stil | Keine Gruppierung

Figure 2.2.: User interface of Citavi when bibliography entry is selected.

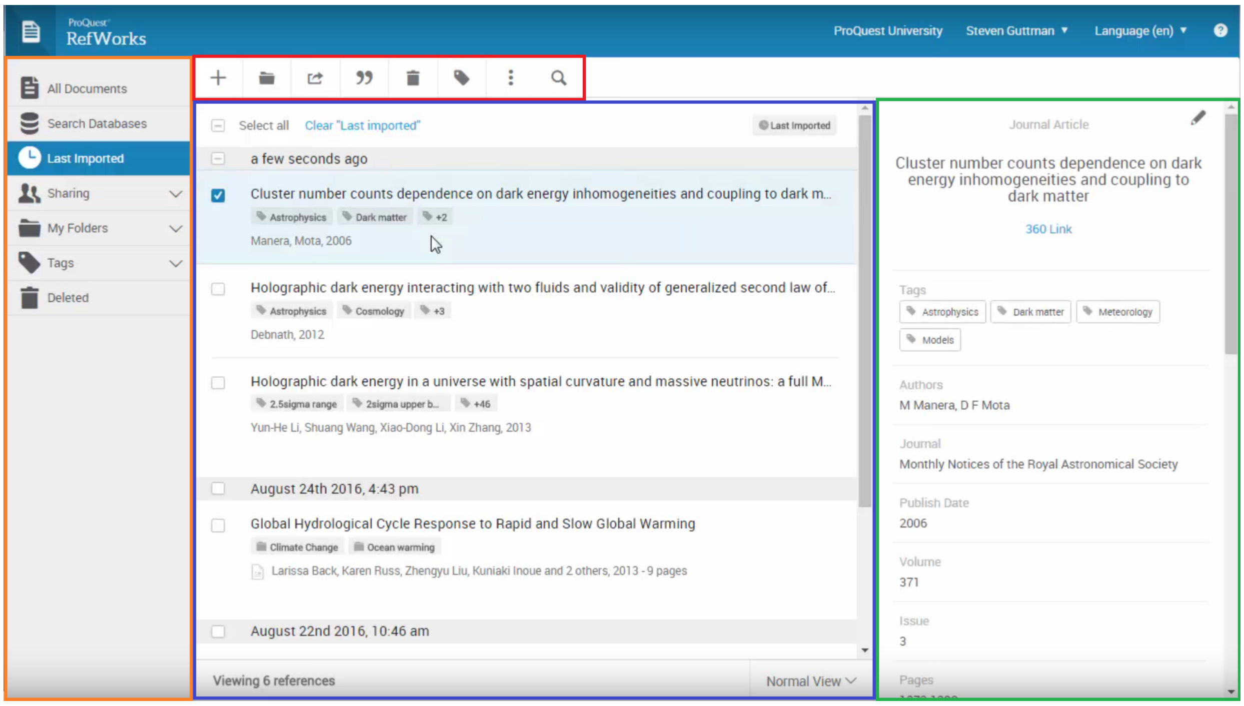
visible for consistency. We chose this approach, as the UI in this state reflects the developers thought process on how the user should experience it. However, in Citavi and Zotero, users have the opportunity to customize the interface by varying the size of the presented main elements or hiding them altogether. We cannot confirm the same for RefWorks, and from the material available to us, interface customization seems unlikely.

While we have established the similarity in the interfaces, there are also differences between the applications' UIs making them special. As already mentioned, Citavi's organizing sidebar is hidden per default in contrast to the other two programs. It is also the only one that has a preview window for external resources as part of its default UI. This window taking up about a third of the screen is located on the far right of the interface and enables users to immediately open PDF or online resources (see Figure 2.2).

Another exception can be found in RefWorks, which is the only web-based program of the three. While the other two applications have a horizontal main menu above its toolbar, as almost all desktop programs do, RefWorks has a broader page header with a link to its main page, as well as account and language options (see Figure 2.3). As its solely web-based nature suggests, RefWorks is designed like a website, not like a desktop program. Therefore, it follows different design conventions.

### 2.2.3. Features and UI of JabRef

After we examined the functionality and user interfaces of current RMS, we now inspect in detail, how JabRef works in particular and how it compares to current programs. The newest official release of JabRef to this date is version 4.3.1, the latest available verison, however, is the



Select all Clear "Last imported" Last Imported

a few seconds ago

[✓] Cluster number counts dependence on dark energy inhomogeneities and coupling to dark m...
Astrophysics | Dark matter | +2
Manera, Mota, 2006

Holographic dark energy interacting with two fluids and validity of generalized second law of...
Astrophysics | Cosmology | +3
Debnath, 2012

Holographic dark energy in a universe with spatial curvature and massive neutrinos: a full M...
2.5sigma range | 2sigma upper b... | +46
Yun-He Li, Shuang Wang, Xiao-Dong Li, Xin Zhang, 2013

August 24th 2016, 4:43 pm

Global Hydrological Cycle Response to Rapid and Slow Global Warming
Climate Change | Ocean warming
Larissa Back, Karen Russ, Zhengyu Liu, Kuniaki Inoue and 2 others, 2013 - 9 pages

August 22nd 2016, 10:46 am

Viewing 6 references | Normal View

Journal Article

Cluster number counts dependence on dark energy inhomogeneities and coupling to dark matter

360 Link

Tags
Astrophysics | Dark matter | Meteorology | Models

Authors
M Manera, D F Mota

Journal
Monthly Notices of the Royal Astronomical Society

Publish Date
2006

Volume
371

Issue
3

Pages

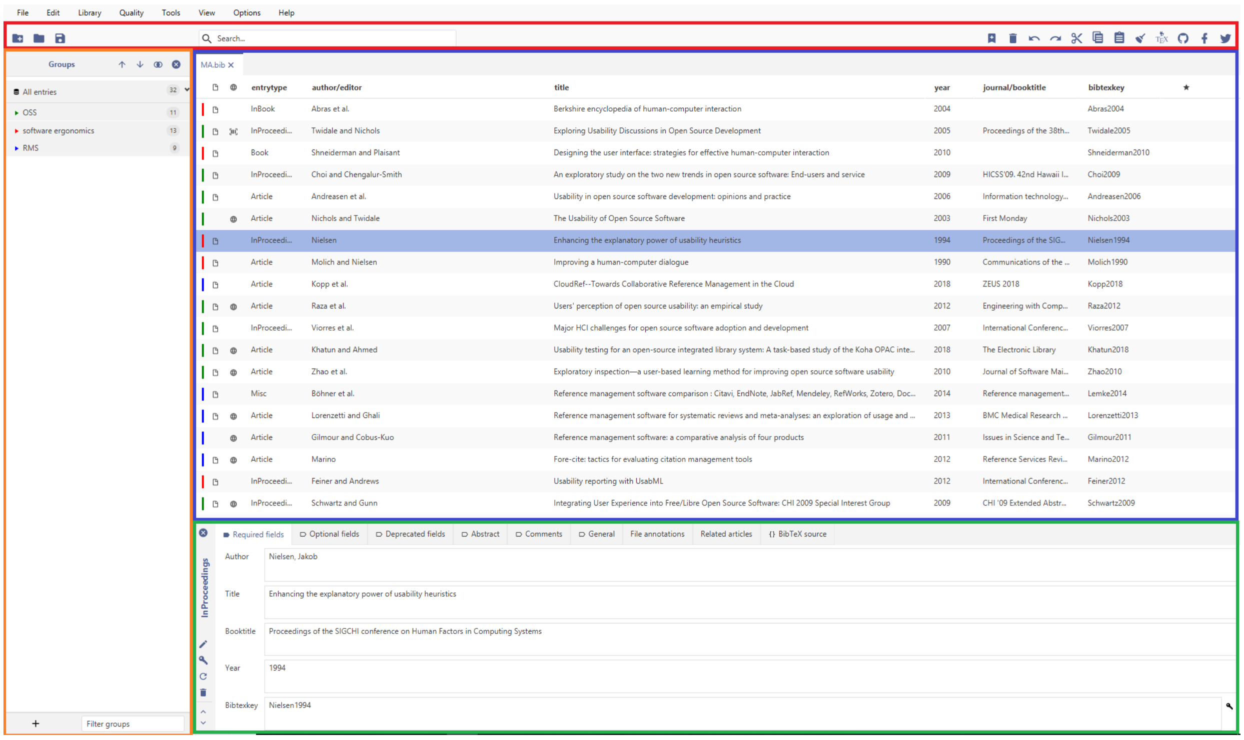
Figure 2.3.: User interface of RefWorks when bibliography entry is selected.

5.0 developer version. Regarding the taxonomy of Gilmour and Cobus-Kuo (2011), JabRef falls in the category of desktop applications that have no additional web interface available. It is a standalone Java application that runs on the three major OS: Windows, Linux and Mac OS X. Apart from database connectivity, JabRef has no social functions at all. Although there are social reference management applications which are based on JabRef (Kopp et al., 2018; Datta, 2010), JabRef itself does not offer integrated sharing of references.

The capabilities of JabRef in comparison to the three competitors, which we have already inspected in detail, support its profile of limited sharing functions (see Table 2.1). JabRef does not offer sharing of references or attached PDF documents. It is also not able to conduct full text searches of PDFs, but possesses all other expected functionalities.

Additionally to the basic functionality, JabRef has extended capabilities, which distinguish it from its competitors. Firstly, it is able to handle very large bibliographies (Böhner et al., 2018) and can therefore be used to organize a vast database of references. Additionally, JabRef gives access to a multitude of quality control tools. They make it possible to find and resolve duplicate references by merging them (Kopp et al., 2018), and give the user the opportunity to check the formal integrity of their references, list faulty entries, and automatically improve the quality by enforcing formal rules on them. Another characteristic, setting JabRef apart, is its easy interaction with LaTeX software applications. Bibliographies are stored in the BibTeX format and citations can be directly exported to an external LaTeX text editor.In this way, JabRef grants access to any citation style, which is available in combination with BibTeX.

Böhner et al. (2018) attest JabRef good usability and call it intuitive to use. An observation of JabRef's user interface reveals that it has similar main elements to the three applications, which were examined before (see Figure 2.4). Again, there is a toolbar at the top of the screen, with a horizontal main menu bar above it. The difference from other applications is that JabRef's toolbar is separated in two parts, one aligned right, one aligned left. In the middle of both parts is a search bar, which can be used to filter for specific text in a bibliography. On the left of the screen is a vertical grouping menu, as in other RMS, with controls on top of the folder view. Folders in the group menu are presented in hierarchical fashion, the number of references in a group folder is shown behind the folder name. From the center to the right of the screen, the list of references is presented in table form, with the columns representing some selected details of the entries. At the bottom of the screen, the detail viewer and editor extends horizontally beneath the list of references. This differs from the other applications, which all had a vertically arranged detail window, but like Citavi, JabRef uses tabs to partition the reference details shown and lets the user switch between these tabs.



| entrytype | author/editor | title | year | journal/booktitle | bibtexkey |
|---|---|---|---|---|---|
| InBook | Abras et al. | Berkshire encyclopedia of human-computer interaction | 2004 | | Abras2004 |
| InProceedi... | Twidale and Nichols | Exploring Usability Discussions in Open Source Development | 2005 | Proceedings of the 38th... | Twidale2005 |
| Book | Shneiderman and Plaisant | Designing the user interface: strategies for effective human-computer interaction | 2010 | | Shneiderman2010 |
| InProceedi... | Choi and Chengalur-Smith | An exploratory study on the two new trends in open source software: End-users and service | 2009 | HICSS'09. 42nd Hawaii I... | Choi2009 |
| Article | Andreasen et al. | Usability in open source software development: opinions and practice | 2006 | Information technology... | Andreasen2006 |
| Article | Nichols and Twidale | The Usability of Open Source Software | 2003 | First Monday | Nichols2003 |
| InProceedi... | Nielsen | Enhancing the explanatory power of usability heuristics | 1994 | Proceedings of the SIG... | Nielsen1994 |
| Article | Molich and Nielsen | Improving a human-computer dialogue | 1990 | Communications of the ... | Molich1990 |
| Article | Kopp et al. | CloudRef--Towards Collaborative Reference Management in the Cloud | 2018 | ZEUS 2018 | Kopp2018 |
| Article | Raza et al. | Users' perception of open source usability: an empirical study | 2012 | Engineering with Comp... | Raza2012 |
| InProceedi... | Viorres et al. | Major HCI challenges for open source software adoption and development | 2007 | International Conferenc... | Viorres2007 |
| Article | Khatun and Ahmed | Usability testing for an open-source integrated library system: A task-based study of the Koha OPAC inte... | 2018 | The Electronic Library | Khatun2018 |
| Article | Zhao et al. | Exploratory inspection—a user-based learning method for improving open source software usability | 2010 | Journal of Software Mai... | Zhao2010 |
| Misc | Böhner et al. | Reference management software comparison : Citavi, EndNote, JabRef, Mendeley, RefWorks, Zotero, Doc... | 2014 | Reference management... | Lemke2014 |
| Article | Lorenzetti and Ghali | Reference management software for systematic reviews and meta-analyses: an exploration of usage and ... | 2013 | BMC Medical Research ... | Lorenzetti2013 |
| Article | Gilmour and Cobus-Kuo | Reference management software: a comparative analysis of four products | 2011 | Issues in Science and Te... | Gilmour2011 |
| Article | Marino | Fore-cite: tactics for evaluating citation management tools | 2012 | Reference Services Revi... | Marino2012 |
| InProceedi... | Feiner and Andrews | Usability reporting with UsabML | 2012 | International Conferenc... | Feiner2012 |
| InProceedi... | Schwartz and Gunn | Integrating User Experience into Free/Libre Open Source Software: CHI 2009 Special Interest Group | 2009 | CHI '09 Extended Abstr... | Schwartz2009 |

| | |
|---|---|
| Author | Nielsen, Jakob |
| Title | Enhancing the explanatory power of usability heuristics |
| Booktitle | Proceedings of the SIGCHI conference on Human Factors in Computing Systems |
| Year | 1994 |
| Bibtexkey | Nielsen1994 |

Figure 2.4.: User interface of JabRef (version 5.0 dev.) when bibliography entry is selected.

## 2.3. Methods of Software Ergonomics

Trying to measure and enhance the usability of software, we have multiple tools and strategies, which we can employ. A commonly made differentiation regarding those strategies is the difference between user-based and expert evaluation. In the following section, we will explore the two families of methods which we will use in this study in detail.

### 2.3.1. Testing with the User

A common framework for software development based on involvement of the user in any step is user-centered design. Abras et al. (2004) claim that the term originated from the laboratory of Donald Norman in the 1980s. Norman himself describes human-centered design in his book *The Design of Everyday Things* as follows:

> It means starting with a good understanding of people and the needs that the design is intended to meet. This understanding comes about primarily through observation, for people themselves are often unaware of their true needs, even unaware of the difficulties they are encountering (Norman, 2013).

A more formalized definition of the user-centered development process can be found in newer sources (Rudlof, 2006; see Figure 2.5). Here, the process is separated into four distinct, iterative steps. In this paper, we will mainly focus on the steps 1 to 3 of the iterative process, namely the analysis of users and usage in context, the specification of usage requirements derived from the observations of usability problems, and the development of prototypes.

This means that representative users of the system, who are not involved in its development, are observed while using JabRef, and their performance and especially their problems while interacting with the user interface are recorded and evaluated. In the step of finding usage requirements, we as experts will try to comprehend the user needs and expectations and how the system has to react to meet both of them. We will then formally specify the appearance and behavior that JabRef should exhibit to improve understandability by its users, and make the proposed changes transparent for the OSS community in charge of implementing a software prototype. This focus on an interdisciplinary team of usability experts and software developers is inherent in the idea of user-centered design.

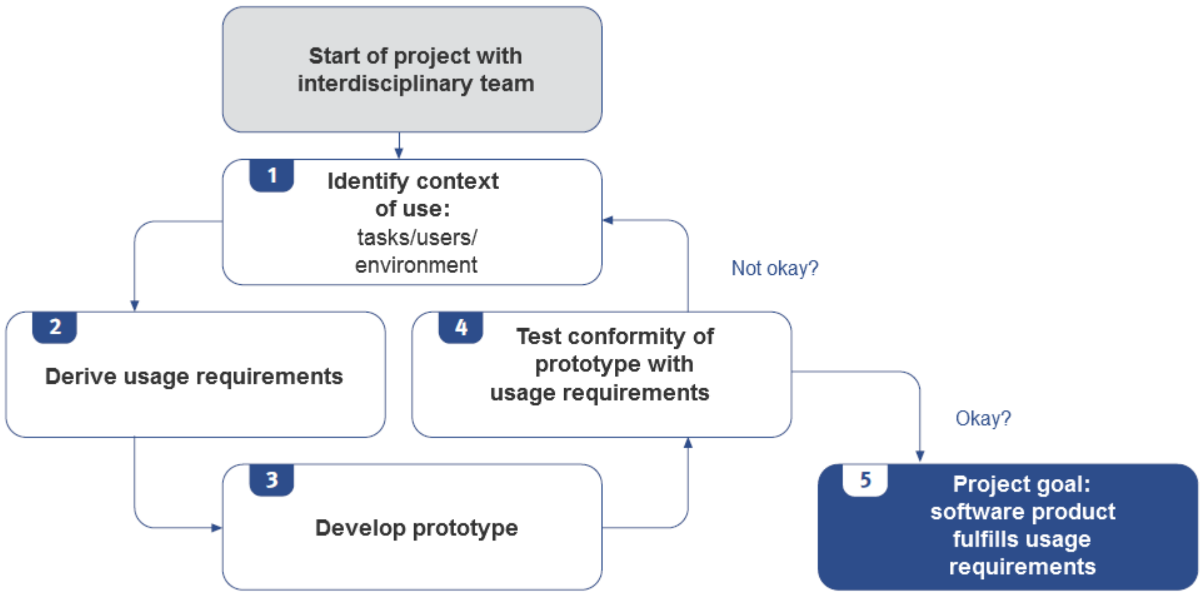

Start of project with interdisciplinary team
1
Identify context of use: tasks/users/ environment
Not okay?
2
Derive usage requirements
4
Test conformity of prototype with usage requirements
Okay?
3
Develop prototype
5
Project goal: software product fulfills usage requirements

Figure 2.5.: The user-centered design process (figure taken and translated from Rudlof, 2006).

As positively regarded as user testing is, there are obstacles in its way that surpass those of regular software testing. The biggest of them might be the budget allocated for testing. While the benefit of an university environment is low personnel cost in combination with access to reasonably high expertise, there is only a small budget available to pay users for their participation. Furthermore, time and efficiency are of more concern, as only a few people work on a single project, often with a restricted time frame.

For these situations, which are also common in smaller companies and open source communities, Nielsen (1994b) has developed the concept of discount usability engineering, trying to find the usability measures with the best cost-benefit ratio. He also describes a simplified version of the thinking-aloud method, and the use of scenarios as efficient methods of user testing. Moreover, he addressed a second problem of user studies, which is the number of participants. While a low number of observed users means that most statistical tests cannot be performed and not all usability problems will be discovered, it is stated that the maximum efficiency of user testing is already reached at the number of five participants (Nielsen and Landauer, 1993). If this is still not feasible, Nielsen (1994b) even regards three subjects as a sufficient sample to draw conclusions.

A decision we have to make when doing user tests is the one between subjective and objective measurements. As the quote from Norman says, users themselves do often not know their own expectations and needs. Observation can therefore be a way for usability professionals to deal with this problem. The principle of user-centered design, to let the users shape a product's design, does not mean that users should be the designers and have to be able to voice their opinion. Merely observing the user operate the system might be much more suited to learn of an individual's normal use of a piece of software.

On the other hand, collecting subjective data from users, be it through standardized questionnaires or interviews, can reveal information that is not accessible through observation alone. Interviews can reveal a more structured mental model of a user than the one found through interpreting user behavior. We will therefore use questionnaires to ask the users about their most urgent issues with JabRef, and will use card sorting techniques (Wood and Wood, 2008) in addition to observation to better comprehend users' understanding of JabRef.

### 2.3.2. Expert Evaluation

While user testing is always necessary in user-centered design, there are universal problems, which cannot only be found through user testing, but also through experts inspecting the software. A popular method to do this is the heuristic evaluation, developed by Nielsen in the 1990s. He describes ten heuristics, which were found to subsume a large amount of possible usability issues (Nielsen, 1994a), and can be used by experts to systematically identify problems in a piece of software. Although this heuristic evaluation cannot replace user testing, it can be a valuable addition to it, by using the knowledge of usability experts. As this study marks the first involvement of usability experts in the JabRef community, we hope to find basic usability problems with this method, which have remained in the program.

In spite of expert involvement being only an addition in the initial testing phase, the specification of usability requirements is mainly the work of usability professionals. Interpreting the results of inspections as well as user tests is the first step of that project phase. The second one is translating user needs and problems into system requirements, which means redesigning JabRef's UI according to user needs and expectations.

Sources for help with this task can be existing design recommendations for interface design. The current ISO standard for usable interactive systems includes seven basic principles that interactive systems should adhere to, including properties like fault tolerance and being self-descriptive (ISO, 2006). Shneiderman and Plaisant (2010) also discriminate between principles and guidelines, where principles describe the aforementioned type of rules. Guidelines, on the other hand, are more concise than this and often describe concrete parts of an interface and how they should be designed.

# 3. User and System Analysis

The first step in the enhancement of JabRef's usability is to understand the current status of the system and how it is used. This encompasses the functionality of JabRef on a detailed level, the attributes of its users, and their interaction with the system. Another influential factor are the tasks that users try to perform with JabRef, their goals, and their ways to reach them with this program. In short, we investigated the workflow of the users and how JabRef might help or hinder them in their endeavors. Our line of action is described in the following and also shown visually in Figure 3.1. The emphasis of this analysis lay on the user's perspective on the system, which is, why we chose to conduct an online survey to find the most important use cases of JabRef. To do so, telemetry data and feedback from the JabRef developers was integrated in the survey design process.

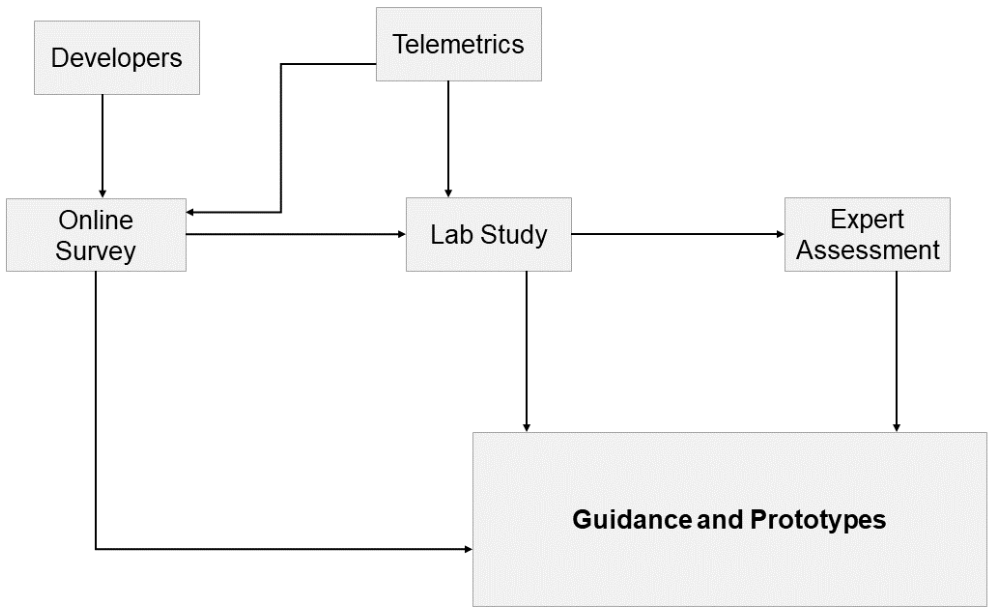


Figure 3.1.: Modus operandi in the system analysis of JabRef.

The survey results were then used to design an lab study with users participating in person. This study aimed to understand the users' problems with some selected use cases and features, and how they fit into their workflow. Additionally to these user-centered methods, we conducted an expert assessment of JabRef to find usability problems which could be examined without direct user involvement. Furthermore, we used insights gained through the lab study and the online survey to inspect features with high risk potential.

## 3.1. Identification of Important Use Cases

In this section, the design process of the online survey is described including the features which we asked about and how the questionnaire was structured. Furthermore, the results of the survey are presented as well as their interpretation, which constituted the basis for our laboratory user study.

### 3.1.1. User Survey

The first step in designing a user survey to find important features of JabRef was to create a list of fundamental functions, which users could then rate. Because of JabRef's size and complexity and the high granularity of the list of functions, it was necessary to preselect some of these features for the scope of the survey to be reasonable.

We used anonymous telemetric data, which was voluntarily collected from users working with the developer versions of JabRef. The data consisted of logged function calls in the JabRef code, all of which were calls of dialog functions marking a specific interaction between user and software by opening dialog boxes. In addition to the function name, the version of JabRef as well as the country and a timestamp of the call were logged. However, the latter two were not used in the analysis of the data.

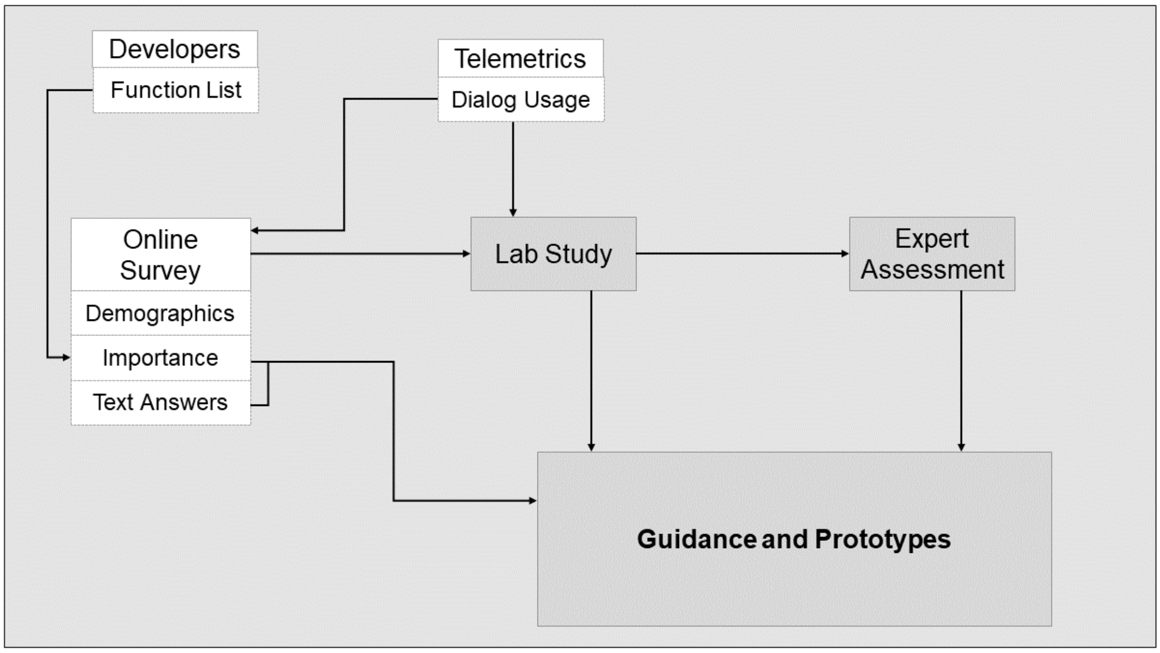


Figure 3.2.: Detailed overview of the online survey, the data used for it, and how it fits into the system analysis.

The sample consisted of 22149 function calls. We reduced it to 21720 by removing calls from JabRef development versions from further calculations. For the remaining sample, the frequencies of the different function calls were calculated to serve as the first hint towards important JabRef features. The number of different functions found was 36. The frequencies listed together with the name of the function and a description of the corresponding dialog can be found in appendix A. One of the examined functions was excluded because it was removed from JabRef in newer versions. By far the most opened dialog was the one for creating a new entry in a BibTeX file, followed by the import dialog for new references. Features associated with frequently appearing dialogs were given higher priority in the selection of functions, which would later be rated by the users in the online survey.

Additionally to the telemetric data, JabRef's main menu was searched for possibly important features because it seemed reasonable for essential functions to be accessible there. As a last source, a group of JabRef developers was asked to comment on the list of functions and suggest missing ones to be added to the list (see Figure 3.2). With this procedure, a list of 24 features was generated. After that, the collected features were sorted into a taxonomy similar to those referenced in chapter 2.2.1 to ensure that no vital aspect of JabRef's functionality was neglected in the survey.

To get the users' perspective on the importance of the gathered features, we constructed a five-point Likert scale, which asked the respondents to rate the importance of every feature in their personal work with JabRef. A sixth option was available on the scale for the case that the respondent had not known the feature before and therefore had no experience with it.

Along with the rating of importance, the survey contained two questions, where a text response was required. The first question asked to name further important features which the respondent used, but which had been omitted in the rating question. The second free text question was aimed towards users' negative experience with JabRef by asking about features that had frustrated the respondent before. The goal of this question was to already identify parts of the software with high potential for improvement.

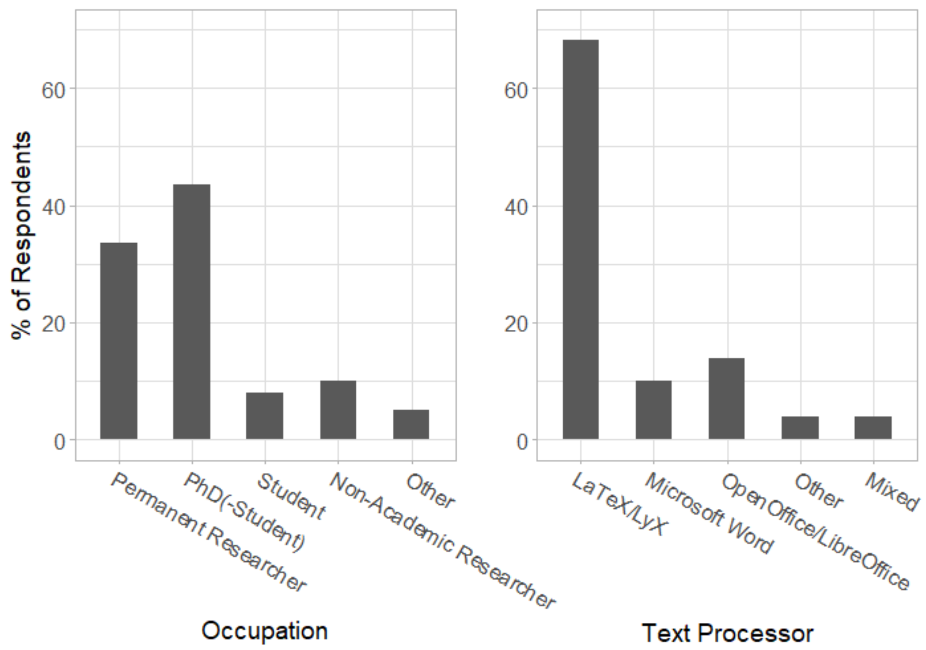


Figure 3.3.: Proportion of occupations and text processors in the sample.

At the end of the survey, demographical data connected with the use of JabRef was collected: the occupation of the respondent as well as their preferred text processing software and their current version of JabRef. Additionally, the number of entries in the respondent's largest BibTeX file was gathered to have some measurement for the intensity with which the respondent used JabRef. The survey was constructed with Google Forms and was advertised on universitary groups in social media as well as on the social media accounts of JabRef. The full questionnaire can be found in Appendix B.

### 3.1.2. Survey Results

The number of respondents to the survey was $N$ = 101. The largest group were PhD students and Post Docs with 44% of the answers, followed by academic researchers, who accounted for 34% of the respondents (see Figure 3.3).

With 68%, the majority of the sample used LaTeX/LyX as their text processing software. Only 4% used a program other than the three listed in the survey and another 4% used multiple text processors (see Figure 3.3).The JabRef version used by most participants was the current release, version 4.3.1, which 68% used. The current developer version was only used by 6%, the remaining 26% used an older version.

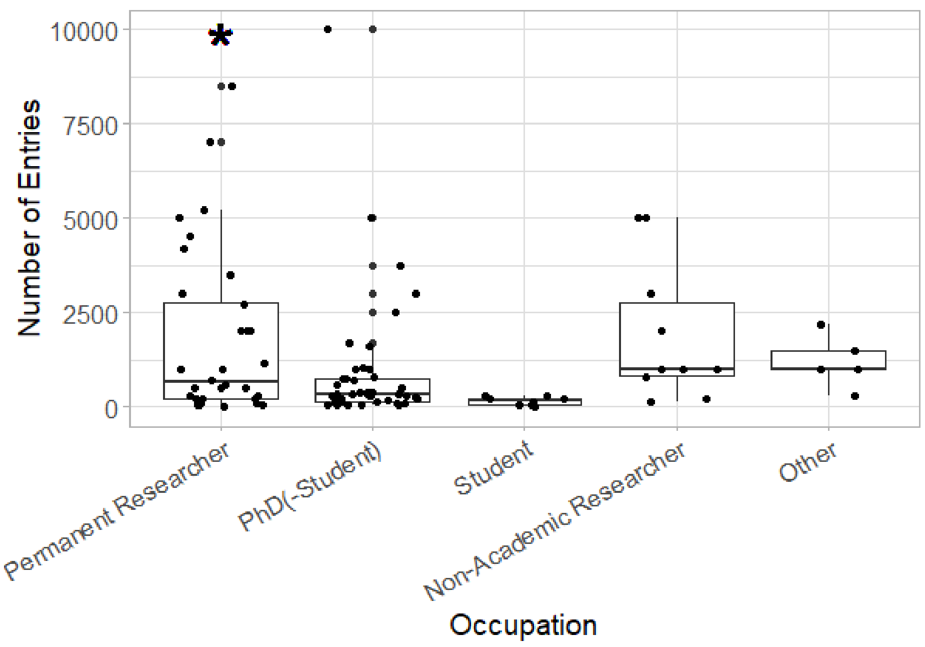


Figure 3.4.: Size of the biggest JabRef library by occupation with mean and median. The Asterisk marks three data points with a y-value of 20,000, which were not shown for visualization purposes. However, the data points were included in all statistical analyses.

The size of the users' biggest JabRef library greatly varied. It ranged from 10 to 20,000 entries with *M(SD)* = 1630(3213), while the median was at 400. This shows that the distribution of the size is right skewed, with numerous small values and few very high ones. That the intensity of the skew is different depending on the occupation of the respondent can be seen in

Figure 3.4, where the difference between mean and median is highest in the group of academic researchers and lowest in the student group with no skew at all.

The ratings of importance were transcoded into a numerical scale reaching from 0 (not important at all) to 4 (absolutely crucial), to make calculating mean values for the ratings possible. If the option *I do not know the feature* was chosen, the user's rating for this feature was excluded. The calculated means ranged from $M$ = 1.22 for the feature to rate entries with stars to $M$ = 3.64 for the feature to edit the details of an entry, e.g. the author. An overview of the rating distribution for all 24 features can be found in Figure 3.5, ordered from the feature with the highest mean rating to the one with the lowest.

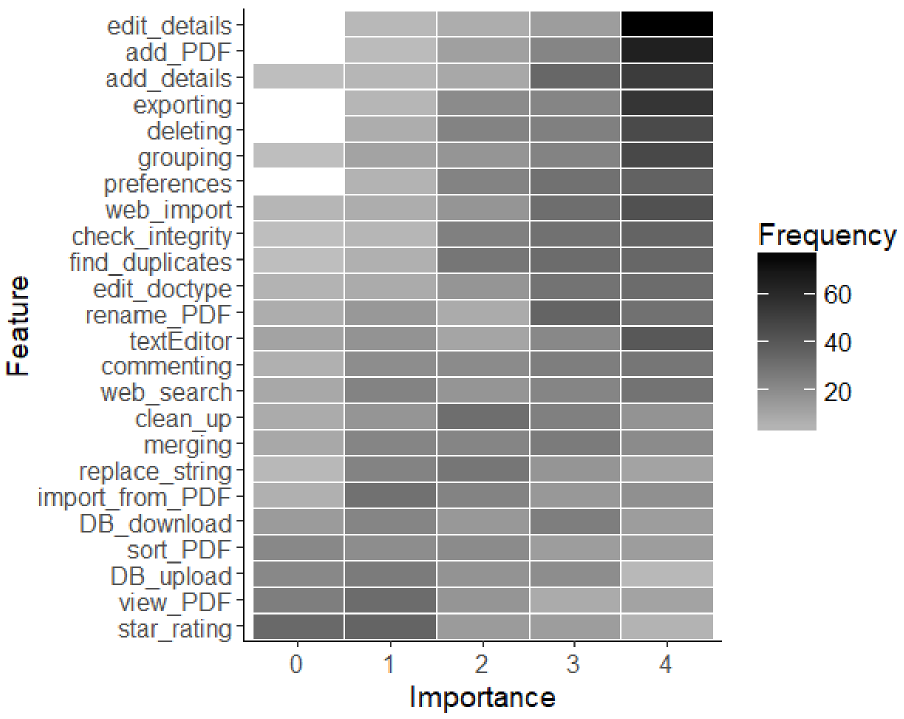


Figure 3.5.: Distribution of importance ratings for all 24 features.

The excluded ratings indicating the lacking awareness of a feature were summed up to represent the prominence of a certain function. While four functions were known by every user, one had not been noticed by 20% of the respondents before. Figure 3.6 hints towards a positive relationship between the awareness of a feature and its perceived importance, but especially features which had a high rating but were not that well known could show potential for better exposition.

Apart from the features which we listed in the survey, further features were found,which were important to the respondents. A first inspection of the free text questions showed that 35 users tried to list one or more further functions which were significant for their workflow. After excluding answers which did not fit the question as well as unintelligible responses, 18 remained for closer inspection. An informal analysis of these answers revealed three prominent functions: the interaction with file attachments beyond the ones queried in the survey, the use of keywords, and the automated generation of BibTeX keys.

The question about frustrating JabRef features received 46 responses, 38 of which were

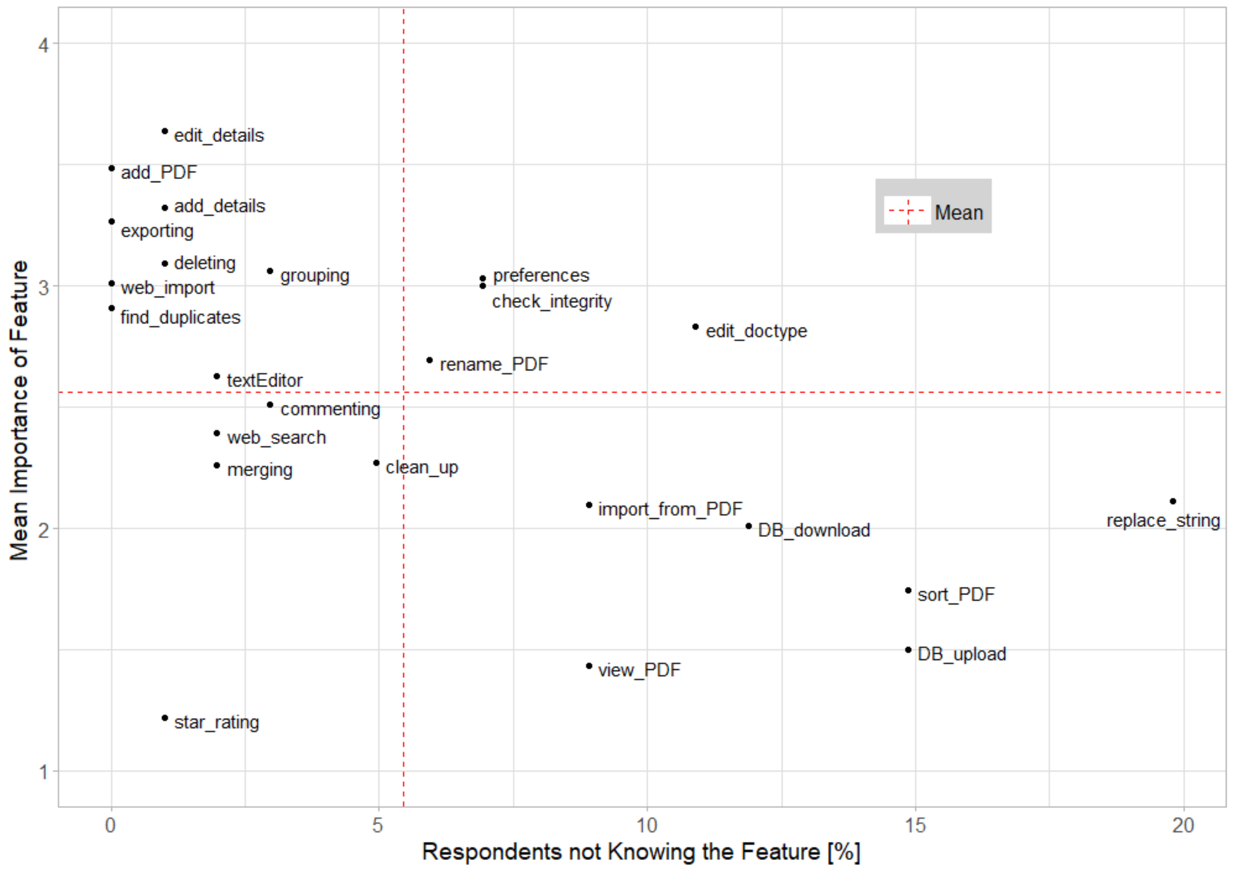


Figure 3.6.: Relationship between perceived importance and awareness of a feature. The dashed lines mark the mean of the displayed data points in the two axis dimensions.

valid answers to the posed question. The most frequently recurring topics were the grouping feature, the editing of entries, and the import of file attachments and references. As this question received more answers than the previous one, we decided to generate a word cloud from the response texts, which confirmed our first inspection of the text answers, especially the prominence of the group feature (see Figure 3.7).

### 3.1.3. Interpretation of Results

Based on the presented data, we can draw conclusions regarding the user group of JabRef and how the software is used. Additionally, we can integrate the text questions and importance ratings to find the JabRef features with the highest priority regarding improvement. The typical user seems to be an academic researcher or PhD student using LaTeX based text processing software. These two groups also seem to contain most of the *power users*, referring to respondents with extremely large bibliographies.

Regarding the ratings of importance, we can consult the additional data we gathered to put them into context. If a feature which is present in the telemetry frequency data also scores high on the importance scale, it evidently has an important role in the usage of JabRef. The highest mean importance rating was found for the feature of editing the detail information of

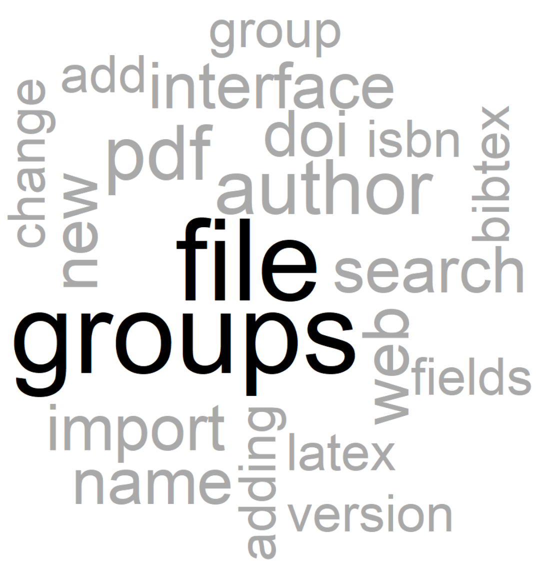


Figure 3.7.: Wordcloud of answers on frustrating JabRef features.

an entry. Almost equally important was the adding of information to an entry, all of which use the entry editor of JabRef. Moreover, the dialog to create a new entry, which precedes the addition of information, was the most frequent one found in our telemetric data. The entry editor also enables other features rating high on importance like the attaching of PDF files and the generation of BibTeX keys, which was repeatedly mentioned in the free text questions of the survey. The entry editor enables multiple significant actions and can therefore become a problem area if information is too condensed or not arranged in a way which is logical to the user. That such a problem exists is evident by the responses to the second free text question, which contained multiple instances of the entry editor being described as frustrating or confusing.

A second function of JabRef causing the users distress is the grouping feature. While telemetry and importance data confirm that the feature is perceived as a crucial part of JabRef and is used frequently as well, it was also mentioned as being frustrating more often than any other feature. A lot of users find the creation of groups too complex or fail at finding a specific grouping function. Users complaining about a missing feature which is actually present in a software is a strong indicator that said feature is lacking the usability to be used to its full potential. Therefore, the grouping feature along with the functions of the entry editor were prioritized in our further work.

## 3.2. Laboratory User Study

After we had found the JabRef features, which had high potential or were at risk of hindering users in their workflow, we constructed a laboratory study to inspect the participants' interaction with these features in detail (see Figure 3.8). We created a thinking aloud experiment to observe the user while they were working with JabRef. In a card sorting task, we let participants arrange elements from the entry editor to gain knowledge about how to display the right information in the right place. At last, the users were asked several interview questions aiming at their overall workflow and their experiences with the tested features.

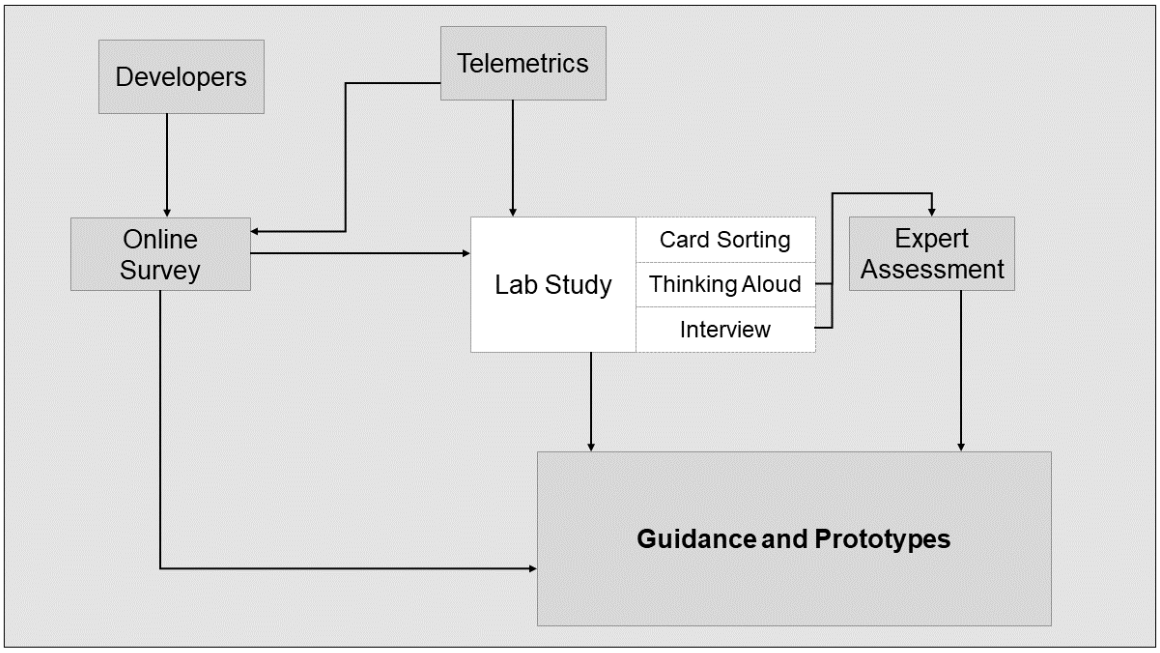


Figure 3.8.: Detailed overview of the laboratory study and how it fits into the system analysis.

The lab study was conducted in rooms of the Technical University Munich in individual sessions of about 30 minutes length. Six users with at least a basic level of experience with JabRef participated in the study. All were included in the data evaluation.

## 3.2.1. Thinking Aloud Experiment

For our thinking aloud experiment we designed two tasks, the *Editor Task* and the *Group Task*, and divided them into subtasks, which were easy to perform without much cognitive workload imposed by the question itself. The tasks had seven to eight subtasks and were solvable with a JabRef bibliography provided for the participants. No other software was required.

The Entry Task focused on features in the entry editor and actions which the user could perform in that part of the UI. Three subtasks required the manual alteration of the entry properties, which is the key function of the entry editor and should be easily accessible. The remaining four subtasks all encompassed functions located in the *General* tab of the entry editor (cf. Figure 2.4). Here, a lot of features concerning attachments and resources are gathered, which scored high on importance in the online survey. The participants were required to search for a PDF file of an existing entry, create an entry from a PDF file, search for the DOI of an entry, and extract further reference information from that DOI.

In the Group Task, users had to create two groups for entries of documents written before or after the year 2000. After that, they had to create two subgroups for the second group, and then add all articles to the respective group. This was to test if JabRef's grouping metaphor collided with the participant's mental model of the function. Another subtask was targeted at the creation of dynamic groups, which could cause the users confusion with the group feature through its complexity. Therefore, participants had to create a group of entries containing the keyword *survey*, which incentivized them to use dynamic grouping. A complete list of the tasks and subtasks can be found in Appendix C.1.

The participants worked on the tasks with JabRef's current release, 4.3.1, on Linux. They were given a tethered keyboard and mouse and a 24 inch monitor. They were instructed to finish the given tasks with as few mistakes as possible, and to verbalize their thoughts and expectations during the process. The two tasks were administered separately, but there was no interruption between the subtasks. Even though there was no time limit, users were given a hint towards the solution of a subtask if they had not progressed at all for a long amount of time. During the experiment, the instructor observed every move of the participant on a second monitor, and notes were taken on relevant behavior or verbalizations, especially if the participant was confused or met another problem in completing the task.

While working on task A, participants employed different methods and strategies to change the properties of an entry. For the generation of BibTeX keys, the button in the sidebar of the entry editor and the toolbar button were used. The entry type was changed in the right click menu as well as directly in the BibTeX source code.

Overall, there were two groups of users with their own strategies and problems arising from it. The first group mainly edited the source code of the entries, which was easier for them than to search for a button, but made more mistakes while manipulating the code. The high error rate can be attributed to the very basic layout of the source code field, which hinders the user from noticing their mistakes. The second group of users worked with buttons and menu options instead of manipulating the source code directly. This group often had difficulty finding the right button or text field for the property, which they wanted to change. Therefore, they took longer to complete a subtask, which indicates that the entry editor confuses the users with its arrangement of information.

One problem in the remaining subtasks seemed to be the lacking prominence of the features which needed to be used. Because their buttons were to small and inconspicuous (see Figure 3.9), the three features accessible in the General tab were often overlooked, even if the users searched all tabs of the editor. Once they found the buttons though, they returned to the General tab faster in the following subtasks. This indicates that the coupling of functions matches the users' mental model.

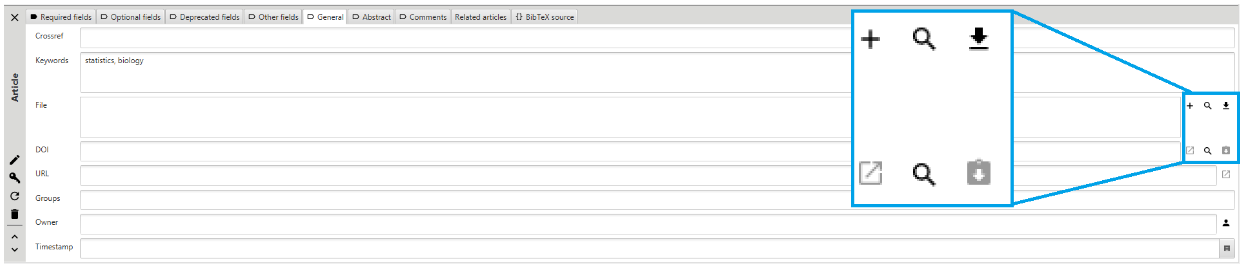
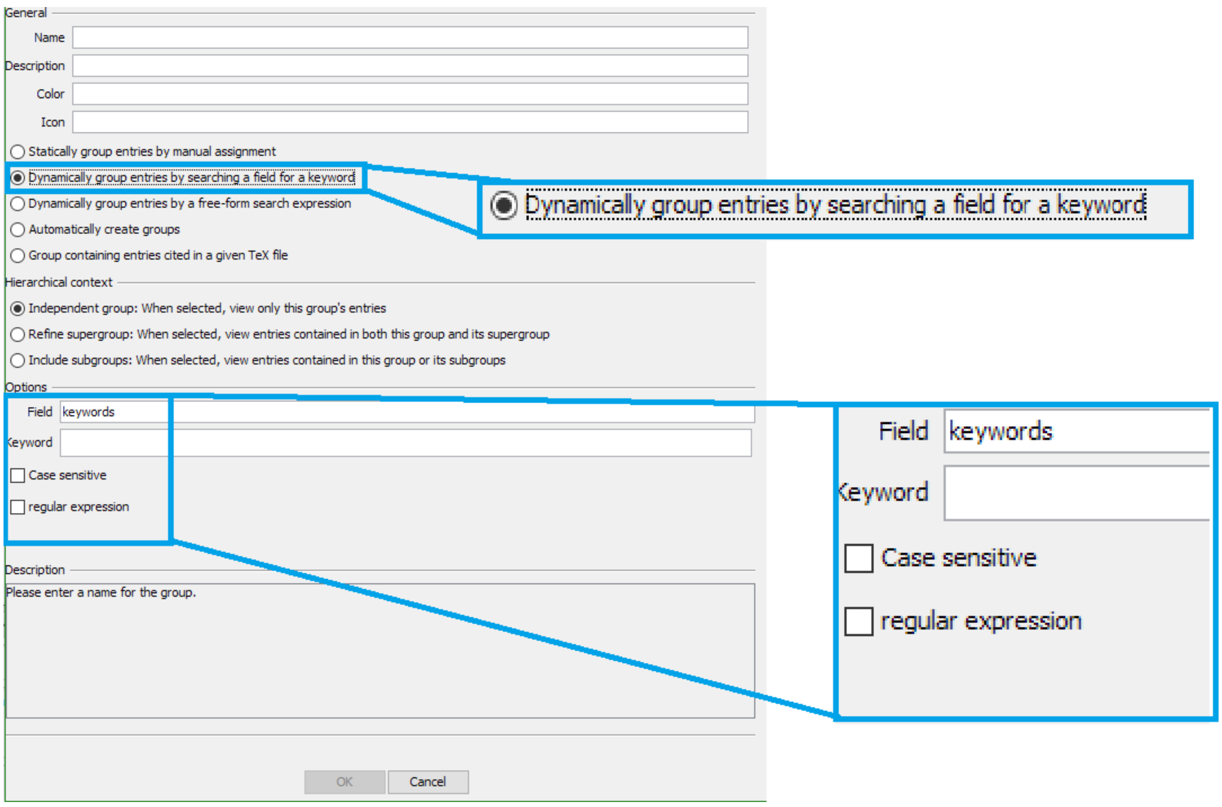


Figure 3.9.: JabRef entry editor with buttons for tested features highlighted.

Participants searching the main menu needed an equally long time to reach their goal, although most functions were available there as well. The amount of information together with its confusing arrangement prevented them to find the feature at first glance. When the users found the functions in the *Quality* tab, they voiced complaints over the confusing categorization of the main menu.

The first subtask of task B required the participants to create a group with no groups existing already. Some of them were confused by JabRef's model of grouping, which treats every group as a subgroup of the *All entries* group. Others had difficulty finding a way to add a group at all. This is a first sign that the creation of groups in JabRef is too hard to access and understand.

The same is true for the group creation dialog containing numerous options, e.g., for generating a dynamic group based on the content of a BibTeX field. This option, although viable in multiple subtasks, was only used in subtask 7, which was aimed specifically at this feature. We believe that the users ignored the group options because of their inherent complexity and the sophisticated phrasing used in them. Even in subtask 7, only half of the participants managed to create a dynamic group based on a keyword. Although the other half of participants had the right solution in mind, they could not execute it because of the options not pointing them in the right direction (cf. Figure 3.10).



General

| | |
|---|---|
| Name | |
| Description | |
| Color | |
| Icon | |

Statically group entries by manual assignment [✓ Dynamically group entries by searching a field for a keyword] Dynamically group entries by a free-form search expression Automatically create groups Group containing entries cited in a given TeX file

Hierarchical context

[✓ Independent group: When selected, view only this group's entries] Refine supergroup: When selected, view entries contained in both this group and its supergroup Include subgroups: When selected, view entries contained in this group or its subgroups

Options

| | |
|---|---|
| Field | keywords |
| Keyword | |

☐ Case sensitive
☐ regular expression

Description

Please enter a name for the group.

Figure 3.10.: JabRef group creation dialog with options for dynamic group creation highlighted.

The subtasks, where participants had to add entries to a group, were solved mostly with the drag and drop mechanic, allowing the user to mark all entries for a group and then move them to a group with one mouse movement. In all other cases, the right click menu was used for the task, but as soon as drag and drop was used once, the participants continued with that method. We deduced from this that drag and drop is the preferred method of moving entries to groups and that the availability of this method should be communicated.

### 3.2.2. Card Sorting

The goal of the card sorting was to rearrange the information in the entry editor, so that users could navigate it more easily. For the user not to be overwhelmed by the amount of information, they need to be given a clear structure, which facilitates a fast search for specific BibTeX fields. Because the fields in most tabs were dynamically generated depending on the entry or its type, we focused on those unchanged by it. This included the fields in the General tab, except *Timestamp* and *Owner*, the *Abstract* field and the *Comment* field. Timestamp and Owner were excluded, as they were planned to be removed in the coming JabRef release. Instead, the ISBN and ISSN fields were included as cards because of their special role as identifiers.

The method of card sorting that we used was the open card sort as described in Spencer et al. (2009), which allows participants to "create and label their own groups of cards" (Spencer et al., 2009, p.52). The only restriction given to them was to create a minimum of two and a maximum of five piles with their ten cards.The size of the piles, however, was not restricted.

Participants were seated in front of an empty table before the paper cards were spread in front of them, and they were told to sort them according to what information they would expect to find in the same place in JabRef. Prior to starting the task, they were asked if they knew all of the presented BibTeX fields and their meaning, and were requested to say if that was not the case. The users were given no time limit in sorting the cards and notified the instructor when they were content with the sort. Then they were asked to label the piles of cards in one or two words, again without imposing a time limit. The piles together with the corresponding labels were recorded as a picture and later transcribed as classification data.

Table 3.1.: Frequencies of numbers of cards in a pile

| Cards per Pile | Absolute Frequencies |
|---|---|
| 1 | 3 |
| 2 | 8 |
| 3 | 4 |
| 4 | 6 |
| 5 | 1 |

Total number of piles = 22. Sample size $N = 6$.

The evaluation showed that four participants had created four piles of cards. The remaining two participants grouped the cards into three piles. Most piles contained two cards, followed by piles with four cards, which can be seen in Table 3.1.
The cards, which were sorted into the same pile the most, were ISBN and ISSN, followed by DOI and Crossref. This seems reasonable because the partnered fields have a similar purpose. It was also apparent that all four of those features' cards were put into a pile multiple times, which stems from their shared role of identifiers (see Figure 3.11). Deviating from this rule was

the DOI, as it was also frequently grouped with the URL card. This could indicate that the features serve some shared purpose for JabRef users, e.g., as a source for additional reference information.

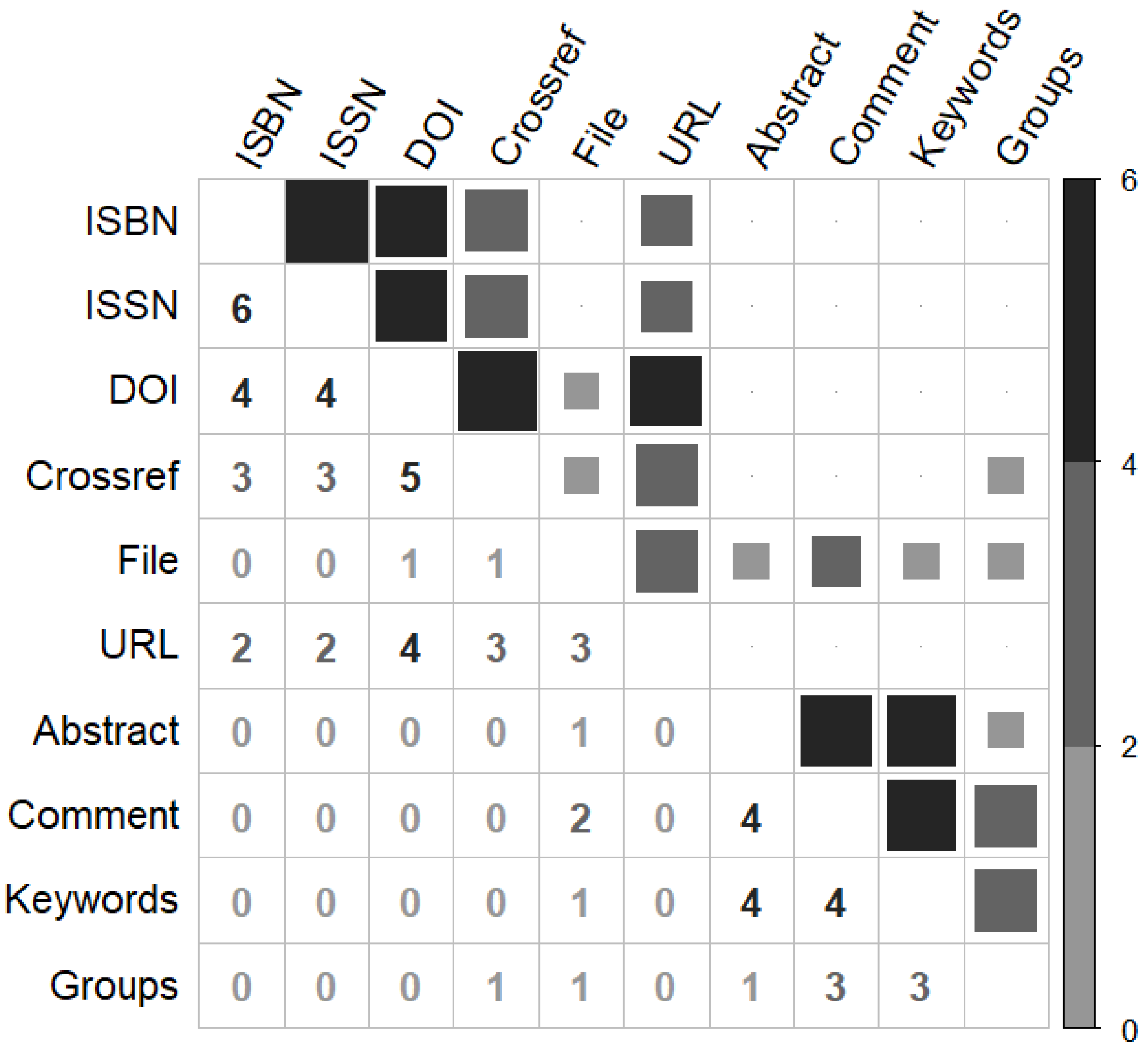


Figure 3.11.: Absolute frequencies of the co-occurence of two feature cards in the same pile. *N* = 6.

A second cluster of grouping contained the cards for abstract, comments, keywords and groups (see Figure 3.11). Although Abstract and Groups were only sorted together once, both were grouped with the remaining two features multiple times. The cluster therefore seems to unite content-related features and organization features.

### 3.2.3. Semi-Structured Interview

The interview questions at the end of the lab study served multiple purposes. Firstly, we gathered personal information, which we could put in context with our user survey to compare our participants with larger demographics. Secondly, we wanted users to voice their retrospective opinion on the tasks and tested features because we assumed to receive more detailed concerns than in the thinking aloud scenario itself, where time pressure limited the participants' verbalizations. Lastly, due to the semi-structured nature of the interview, we expanded our questions, when an opportunity allowed us, to gain insights into the users' workflow with JabRef. We did not only query them regarding awareness of the tested features, but also if they would use them, and how they fit into their workflow. The written outline used for the interview can be found in Appendix C.2.

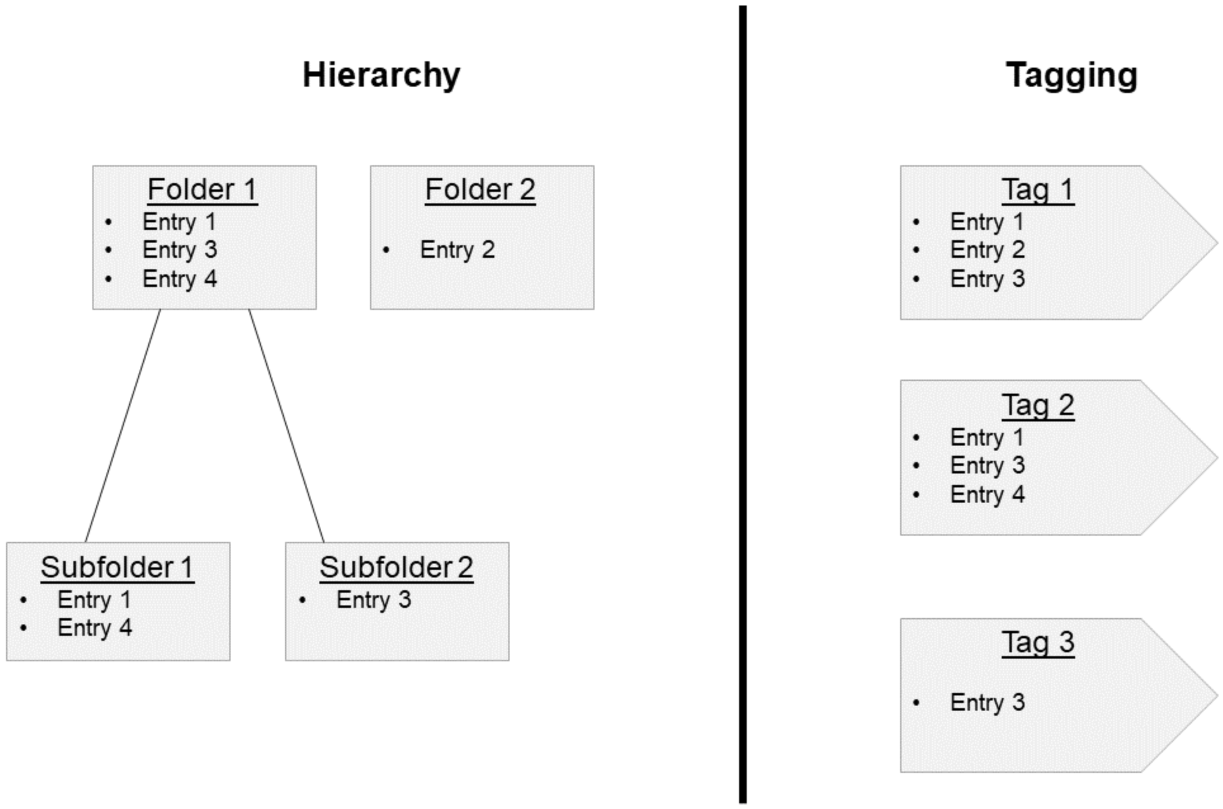


Figure 3.12.: Comparison of two different grouping metaphors. Left: Folder hierarchy, where the entries of the two subfolders are also in the folder above in the hierarchy. Right: Tagging system, where tags represent topics and each entry is part of multiple topics.

The sample of the lab study resembled the one of the online survey in so far as all six participants were university affiliates. Four were students and two were PhD students or post docs, suggesting a more unexperienced user group than the one in the survey, to which only a small percentage of students responded (cf. Figure 3.3). The student users also were less experienced with JabRef, having used it for 1 month to 3 years prior. The remaining two participants had used JabRef for at least 6 years before. This could explain why most participants had trouble with the features in task A of the thinking aloud, which were geared towards more experienced users of JabRef. Concerning the biggest JabRef library of the participants, numbers ranged from 20 to 250 entries, showing that unlike in the online survey, no power users with multiple thousands of entries did partake in the lab study.

When asked about the group feature of JabRef, all participants claimed that it worked like they expected it to. In contrast to that, only one of them knew of the feature to create groups by keywords beforehand. This might indicate that the participants were not confused by the process of grouping, but at the same time did not understand its potential entirely. Those who had not used automatic grouping in task B found the feature useful when told about it and stated that they would have used it, had they known about it before. Those who had discovered the feature during the study could also imagine using the feature in their further work with JabRef.

Another question asked in the interview aimed at the metaphor used to present the grouping feature. In our explanation, we differentiated between two of them. The hierarchy metaphor describes a system of nested folders, where every entry is located in only one folder, but entries of a subfolder are also part of the superfolder (cf. Figure 3.12). In contrast to that, the tagging metaphor uses not folders but tags, which represent a certain topic and are not hierarchically sorted, meaning an entry can be part of multiple tags at the same time (cf. Figure 3.12).

We inquired if participants preferred one of the two metaphors over the current implementation in JabRef, which also worked with a system of nested folders but had no hierarchy by default, causing folders and subfolders to be independent from each other. While most participants preferred the hierarchy metaphor, they stated that a parallel tagging system would be helpful to them, especially for exploratory literature research. The one user choosing the JabRef implementation over the others claimed it to be a useful workaround to have both a tagging and a hierarchy system to work with. We can therefore deduce that having both metaphors would be the optimized solution for the group feature.

Regarding the more advanced features in task A, the participants gave different answers depending on their workflow. Although none of the participants had used the features in the general tab regularly, most answered positively when asked about incorporating them into their work in the future. Here, the existence of two groups of users became evident.One group seemed to start their work by manually gathering most of the reference data for an information, including identifiers like the DOI, and then continued to find additional resources if possible. Most of the time, these participants had inspected a single article for a long time or searched for it specifically. This user group was open to use full text search to find a PDF for their reference and the entry update through the DOI, which would allow them to correct and complement their found data.The second group of users was more likely to have only few information about an article and try to expand their entry details, often starting just with a PDF file. These users would be glad to use the import of PDF metadata into JabRef to find some more information. The same goes for retrieving a DOI automatically and updating entry details with its help, which could be too time consuming if it had to be done manually.

Considering these new insights, all four of the mentioned functions have a purpose in the JabRef workflow and should gain more presence in the entry editor. Furthermore, the import of PDF metadata should be added to the group of features, as it serves a role in the process of creating and updating an entry. We also have to incorporate the different workflows into our considerations and not design an interface which leaves the user without enough freedom.

## 3.3. Expert Assessment of JabRef

While user involvement is the core of user-centered design, a system analysis profits from the involvement and opinion of experts as well. Therefore, to complement the user integration discussed in the previous sections, we conducted an expert assessment of JabRef covering three aspects where usability problems can be reliably detected and removed: visual form and presentation, information quality and quantity, and user-system interaction. While we based our inspection on our previously obtained results (see Figure 3.13), we also tried to expand our field of examination in this analysis for our results to be as thorough as possible. Because of the gravity of the UI change between version 4.3.1 and the 5.0 developer's version, we decided to use the latter for this evaluation.

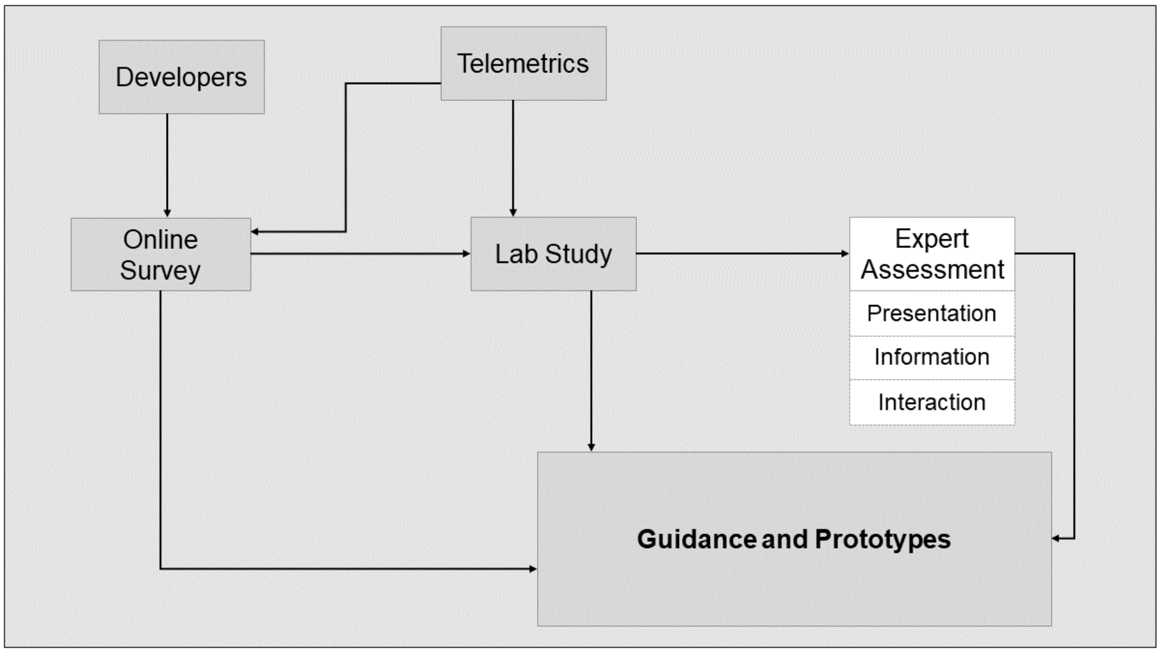


Figure 3.13.: Detailed overview of the expert assessment and how it fits into the system analysis.

Regarding the visual presentation in JabRef, we started by examining the font height and its conformity to current standards. The norm, to which we compared JabRef, was the ISO 9241 (ISO, 2008; ISO, 2011). In that document, the minimal allowed font height is defined by a viewing angle of 16″ (arc minutes) for the capital letter *H*, excluding ascenders, while the recommended height is 20-22″ (ISO, 2008; ISO, 2011). As a simplification, we assume a viewing distance of 600 mm, which conforms to the ISO standard and allows us to take clear measurements and give suggestions for the font height in millimeter. This results in a minimal height of 2.80 mm and a recommended height of 3.50 mm. JabRef's computed default font size was 9 pt, but our measurements showed that the minimum font height was not achieved with a font size lower than 12 pt, which corresponded to a font height of 3.00 mm. For the recommended font height, the computed font size had to be 14 pt or greater. As the variation of font sizes in JabRef is quite limited, this is a global problem, requiring a change of the default font size.

A further variable influencing the usability of software is the color contrast between fore- and background elements. Here, we tested against two standards, the ISO standard 9241, and the W3C standard (W3C World Wide Web Consortium, 2008), which is mainly applied to web pages but can be used for the evaluation of other software as well. For the font sizes present in JabRef, the W3C Consortium requires a contrast of 4.5:1 between font and background (W3C World Wide Web Consortium, 2008), while ISO 9241 only demands a 3:1 ratio (ISO, 2011). Out of 12 reviewed contrasts, all met the ISO requirements and ten met the W3C standard. However, the contrast between the background elements was found to be quite low. None of the four tested background contrasts met the W3C ratio of 3:1.

At last, we examined the icons used in the JabRef user interface. Although size seems to be an influential factor in the usability of icons (Grobelny et al., 2005), exact values depend on the specific set of icons, which is why we chose other metrics to evaluate JabRef's icon usage. Firstly, we inspected icons with high similarity, which slows down the user in identifying the right symbol (Trapp and Wienrich, 2018). An example would be the toolbar, which contains two pairs of similar icons. The *Open file* and *New file* symbols only differ in a small plus sign, and the copy icon resembles the paste icon (cf. Figure 3.14).

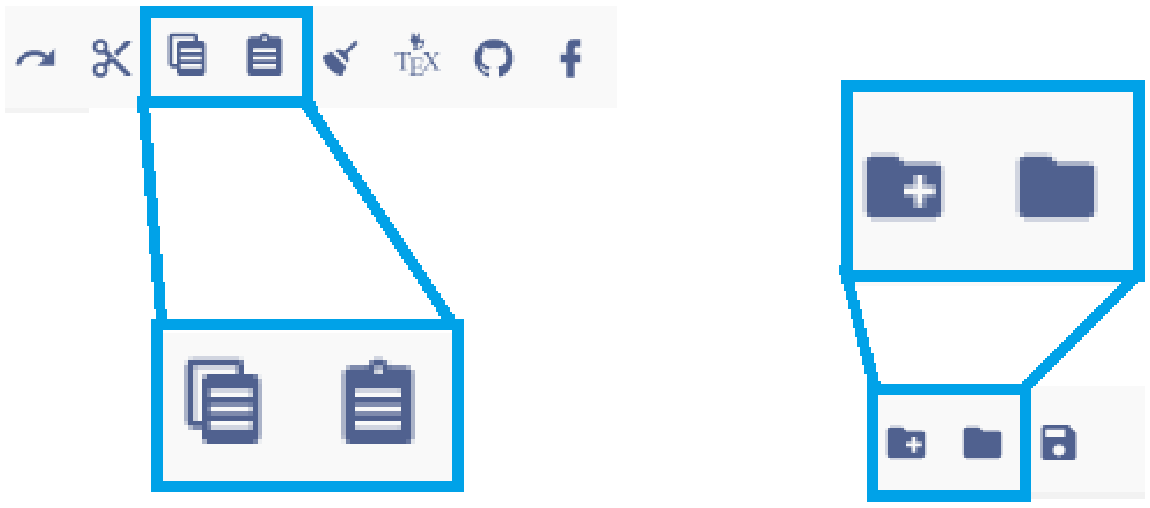


Figure 3.14.: Some toolbar icons of JabRef with icons highlighted, which are similar to each other.

We also identified icons, which were not consistently used and changed their meaning throughout the application, e.g., the web icon, which indicates *URL/DOI* in the entry list but *Web search* in the main menu. Finally, some icons lacking conciseness were found. One of these icons was the update button in the sidebar of the entry editor, which cues a search for additional reference information based on the entries identifiers. The symbol on the button is the refresh icon, which could be interpreted in multiple ways in that context and should therefore not stand without additional text.

Apart from visual presentation, we identified cases, where JabRef did not provide information the right way or provided it in the wrong quantity. As a taxonomy for our findings, we used the 10 Usability Heuristics described by Nielsen (1994a), which also helped us to take into account every aspect of JabRef's usability. While JabRef gives the user a lot of information to work with, we found some potential problems stemming from a lack of information. One global issue are missing input suggestions for text fields, which could give the user a hint at quality and form of a text input. Another information, which could improve interaction with the user as well, is a preview for complicated functions, like grouping or importing an entry from plain text.

In contrast to these cases, the problem of an information overload is present in JabRef as well. As has already been stated by Hick's Law, this leads to increased search times for the user. The tabs of the main menu contain too much entries, which are too detailed and not sufficiently nested. The same goes for the right click menu. Apart from the menus, the preference function contains an overwhelming number of unsorted preferences, which are again divided into numerous options.

Differing from an information overload is information which is present but unnecessary, e.g., the star ranking of an entry being shown in the entry list per default. Also falling in this category were the social media buttons in the toolbar, which did not match the features around it, as they are not used as regularly as the other toolbar function. Information which was necessary but presented in a confusing way was examined as well. The text fields in the entry editor were too big and made it difficult to get an overview of multiple entry details at a time. Equally confusing was the information display in some lesser used functions found in the main menu. The library import through LaTeX auxiliary files contained messages describing its results, which were ambiguous in their meaning and did not give the users constructive guidance on how to act. The same was the case for the check integrity dialog, which provided the user with an unstructured list of integrity violations but lacked information about the exact location of the violation and how to resolve it. Another potential problem was the layout of the plain text import for references, which made it impossible to preview the important fields in one window, resulting in an increased cognitive load on the user.

Another aspect of information quality deemed insufficient was the naming in JabRef. Sometimes names were not concise or correct like the *Quality* tab of the main menu, the content of which did not fit its subordinated funtions. Inconsistent naming also lead to confusion, e.g., in the grouping by a keyword. Here, JabRef uses the term *keyword* for the BibTeX field as well as for a search term used for grouping. This can provoke human error and should be avoided by unique naming of concepts.

For a third group of usability issues, we checked potential errors in the interaction between JabRef and its users. A common theme is the miscommunication between system and user, where one sends a signal which the other interprets differently from its intended meaning. These signals are not necessarily explicit messages but can be patterns which are incorporated into an interface element, hinting towards a specific interaction.

In JabRef, problems occur in the main menu, where functions which apply to a whole library are presented in the same way as functions applying only to selected entries. This prevents the user from establishing a working mental model for the use of these functions. Additionally, the handling of input by the user is also different from function to function. While some of them will just execute an action for all entries of a library, other features prompt them to select one or more entries.

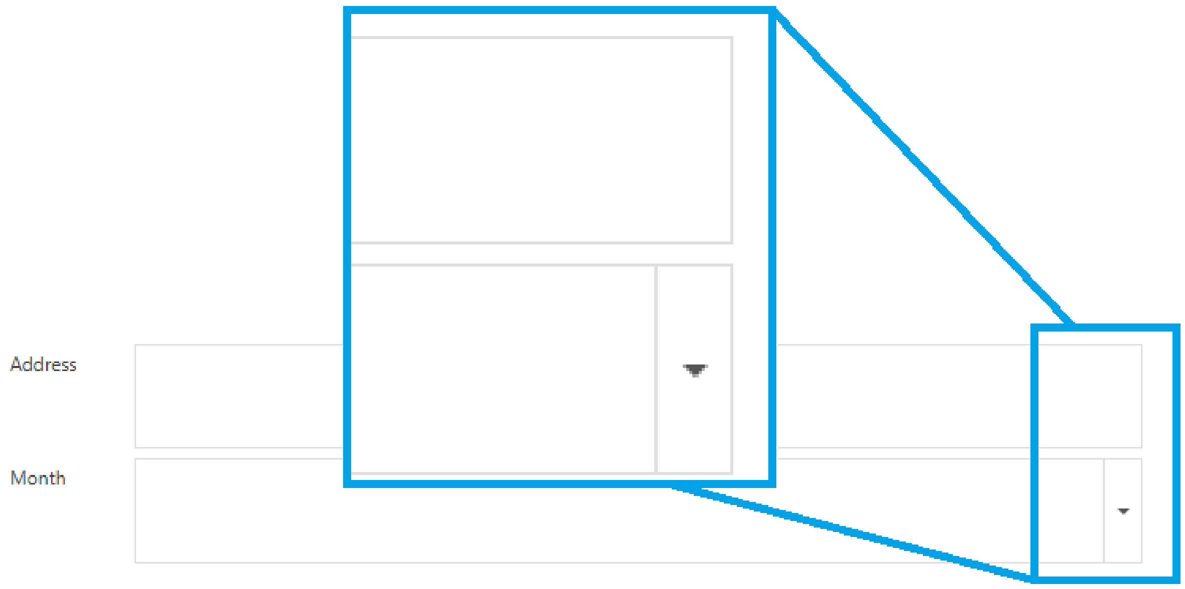


Figure 3.15.: Text field and drop down field in the entry editor of JabRef with difference highlighted.

Miscommunication also occurs if the system does not send any feedback for its actions, leaving the user without any knowledge of the system status. There is a considerable amount of functions in JabRef which give little or no feedback concerning their progress and their results. The most important examples are the full text search, which lacks any display that the function is in progress, and the search for document identifiers, which does not only miss giving feedback while in progress but also fails to notify the user of its results. In these cases, appropriate and consistent display of feedback has to be included.

At last, we examined the input modes which JabRef required from the user, and if they were appropriate. A lot of times, text input was used for the user to name a certain BibTeX field. The number of BibTeX fields is, however, limited and their names are registered in JabRef. Therefore, a drop down list of elements to choose from would avoid spelling mistakes and not require the user to recall the names of BibTeX fields. Additionally, JabRef fields which have already had the drop down functionality did not reflect it in their design but looked nearly identical to text fields (cf. Figure 3.15). In cases like these, the appearance of the input field should reflect its mode of interaction more clearly.

# 4. Guidance for Improving the JabRef User Interface

Following the user-centered design process, we have identified the user group of JabRef, their workflows and needs, and we have already located observed and potential causes for usability problems. Furthermore, we have specified some usability requirements in a general manner, which are based on our observations, guidelines, and expert knowledge. In the following chapter, we consolidate these requirements in the form of concrete guidelines and visually support them with non-functional high fidelity prototypes. Our efforts concentrate on four aspects of the JabRef user interface, which have all proven to be in need of our attention over the course of our analysis (see Figure 4.1).

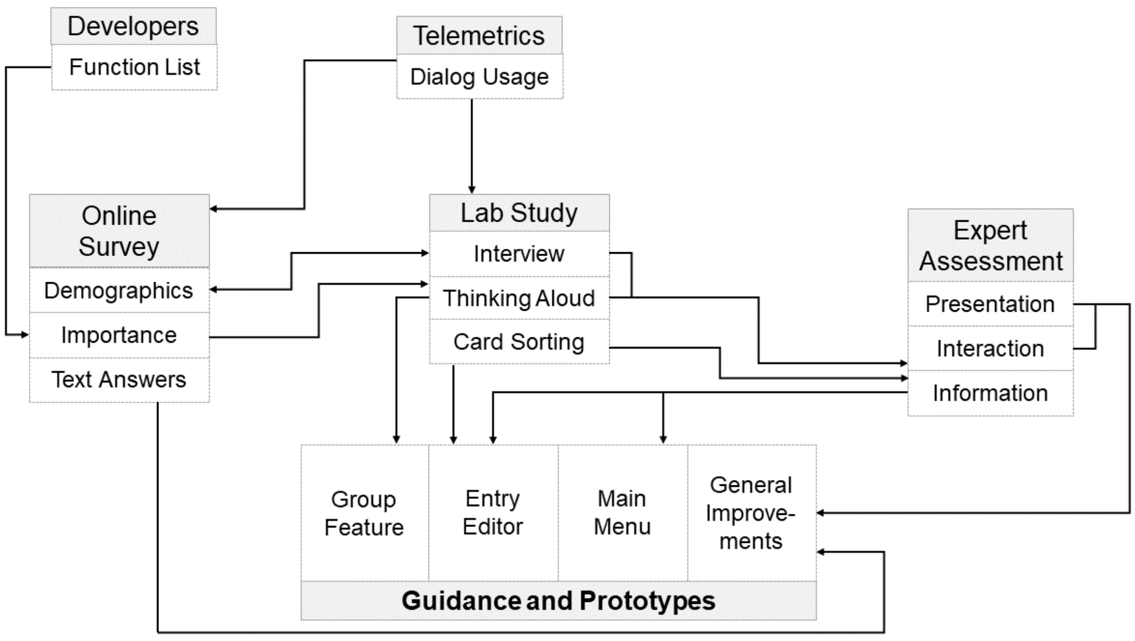


Figure 4.1.: Detailed overview of the system analysis and the influence of its parts on the guidance and the prototypes.

We decided to give suggestions an general ergonomical improvements, concerning either the complete JabRef UI or multiple parts of it. These improvements are on one hand influenced by the text responses we received in our online survey, which contained criticism on the overall look of the UI or smaller issues distributed in multiple features, like missing suggestions in text input fields. On the other hand, the guidance is based on expert knowledge and thorough interaction with the interface, e.g., concerning contrast and font size. In our expert assessment, we also inspected parts of the UI with which the participants of the thinking aloud experiment had problems, like feedback on search functions.

Our second block of suggestions targets the main menu of JabRef. While the online survey only showed that a lot of functions in the main menu were important for the users, the lab study revealed how difficult it was for them to find and access the function they were searching for, due to the poor structuring and overwhelming amount of information in the main menu. This finding was supported by our expert assessment (see Figure 4.1), which induced us to restructure the main menu and improve it by better organizing its content.

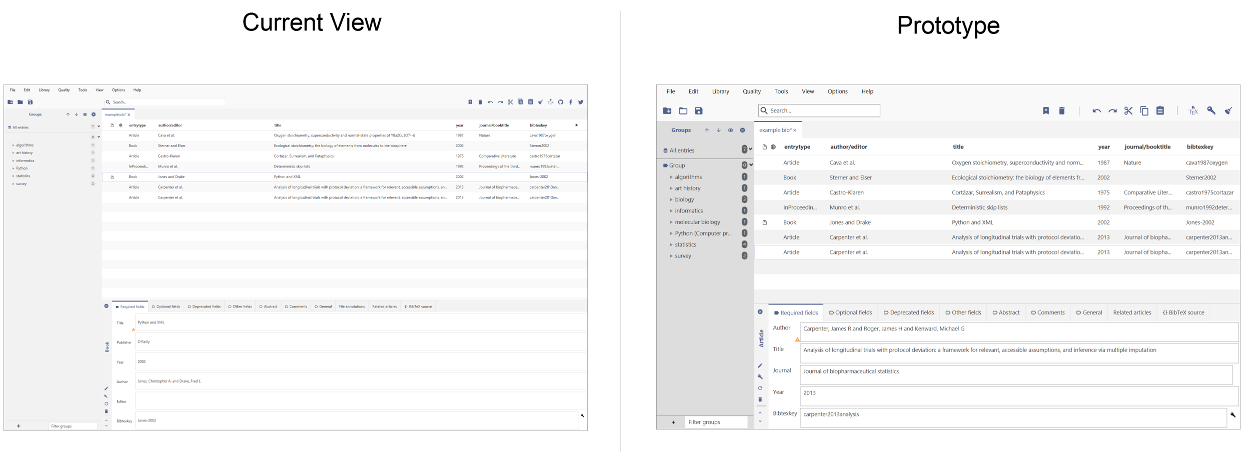
The third area for improvement was already found early in our analyses, the entry editor. The importance of its functions was rated highest in the online survey, showing its relevance for the use of JabRef. Therefore, we decided to prioritize the entry editor in our lab study by dedicating the card sorting to it, and adding the Editor Task task in the thinking aloud experiment (see Figure 4.1). Through that, we learned that the structure of information in the entry editor had potential for improvement and that its functions concerning attached files and online resources needed more exposure. The expert assessment revealed further room for improvement in the amount and presentation of information in the entry editor, like the size of text fields and the information density of the General tab.

The last feature, which was examined in great detail, was the group feature. The dialog usage in the telemetric data already showed frequent use of the feature (see Appendix A) paired with a high potential for frustration, as users attested the feature in the online survey. Therefore, we decided to investigate the source of the users' frustration directly in the Group Task of the thinking aloud experiment (see Figure 4.1). There, we found that the complexity of the feature prevented the users to delve deeper into the feature, although it was called useful in the interview of the lab study. The participants also expressed the wish for a tagging system in addition to the available grouping feature.

## 4.1. General Ergonomical Improvements

For making general suggestions for changes in JabRef, general principles are the foundation, e.g., readability of text. Therefore, an increase in font size should be applied to match existing standards, which has already been suggested in the expert assessment (see chapter 3.3). We recommend to increase JabRef's default font size from 9 pt to 14 pt, which equals a letter height of 20″. Other font sizes should be used in relative measure to the default, but should not leave the range between 12-15 pt. The realization of this suggestion can be found in the prototype (see Figure 4.2), which uses a font size of 14 pt. The optimal font size, however, depends on the size of the container elements as well and has to fit a layout to be practical. It should be noted that a change in font size should always indicate high or low importance, as its size causes the text to be more or less prominent.

Regarding the technical implementation, alternative font sizes can be specified in CSS by using the unit *em*, or they can be realized through size variables containing the absolute values for large, medium, or small sized text. Caution should be exercised if the font size is given in pixel (px) because this unit depends on the pixel size of the display device, making a scalable font size necessary.



Current View

| entrytype | author/editor | title | year | journal/booktitle | bibtexkey |
|---|---|---|---|---|---|
| Article | Cava et al. | Oxygen stoichiometry, superconductivity and normal-state properties of YBa2Cu3O7−δ | 1987 | Nature | cava1987oxygen |
| Book | Sterner and Elser | Ecological stoichiometry: the biology of elements from molecules to the biosphere | 2002 | | Sterner2002 |
| Article | Castro-Klaren | Cortázar, Surrealism, and Pataphysics | 1975 | Comparative Literature | castro1975cortazar |
| InProceedi... | Munro et al. | Deterministic skip lists | 1992 | Proceedings of the third... | munro1992deter... |
| Book | Jones and Drake | Python and XML | 2002 | | Jones-2002 |
| Article | Carpenter et al. | Analysis of longitudinal trials with protocol deviation: a framework for relevant, accessible assumptions, an... | 2013 | Journal of biopharmace... | carpenter2013an... |
| Article | Carpenter et al. | Analysis of longitudinal trials with protocol deviation: a framework for relevant, accessible assumptions, an... | 2013 | Journal of biopharmace... | carpenter2013an... |

| | |
|---|---|
| Title | Python and XML |
| Publisher | O'Reilly |
| Year | 2002 |
| Author | Jones, Christopher A. and Drake, Fred L. |
| Editor | |
| Bibtexkey | Jones-2002 |

Prototype

| entrytype | author/editor | title | year | journal/booktitle | bibtexkey |
|---|---|---|---|---|---|
| Article | Cava et al. | Oxygen stoichiometry, superconductivity and norm... | 1987 | Nature | cava1987oxygen |
| Book | Sterner and Elser | Ecological stoichiometry: the biology of elements fr... | 2002 | | Sterner2002 |
| Article | Castro-Klaren | Cortázar, Surrealism, and Pataphysics | 1975 | Comparative Liter... | castro1975cortazar |
| InProceedin... | Munro et al. | Deterministic skip lists | 1992 | Proceedings of th... | munro1992deter... |
| Book | Jones and Drake | Python and XML | 2002 | | Jones-2002 |
| Article | Carpenter et al. | Analysis of longitudinal trials with protocol deviatio... | 2013 | Journal of biopha... | carpenter2013an... |
| Article | Carpenter et al. | Analysis of longitudinal trials with protocol deviatio... | 2013 | Journal of biopha... | carpenter2013an... |

| | |
|---|---|
| Author | Carpenter, James R and Roger, James H and Kenward, Michael G |
| Title | Analysis of longitudinal trials with protocol deviation: a framework for relevant, accessible assumptions, and inference via multiple imputation |
| Journal | Journal of biopharmaceutical statistics |
| Year | 2013 |
| Bibtexkey | carpenter2013analysis |

Figure 4.2.: Prototype of the overall JabRef layout with usability improvements, in comparison to the current layout under version 5.0 dev.

The contrast of text and background in JabRef was appropriate in all places, making no changes of font color necessary. However, the contrast between user interface components and their background was too low in some cases. In our prototype, we darkened the borders of text input boxes and the group size indicators (see Figure 4.2) to fit the contrast of 3:1, recommended by W3C World Wide Web Consortium (2008). Furthermore, we suggest to increase the contrast between different layout parts, making it easier for the user to find the UI element for which they are searching. This can be seen in the color change of the group sidebar, which is more clearly separated from its neighboring elements in the prototype than in the current UI. Additionally, the color marking alternative rows in the main table was darkened to facilitate the search for a specific entry.

While JabRef contains defined color schemes and a set of font colors, which are used depending on the contrast with the background, UI components should follow the same rules and be changed automatically if the contrast is too low. Another possibility would be to construct an exhaustive list with pairs of font and background colors, which can then be used in styling every element. In addition to color contrast, shadow is a well documented way to elevate certain UI elements and separate them from each other. Especially Google has set standards for using shadow in their Material Design guidelines (Google, 2018).

Apart from contrast and font size, JabRef's toolbar was examined and changed in our prototype. Firstly, some of the icons in the toolbar should be changed, namely the *Open library* icon and the *Copy* icon (cf. Figure 4.2). The icons for JabRef's social media presence should be removed, but this measurements alone would cause the toolbar to be full of unused whitespace, which we utilized in two different ways, the first one being the separation of icons into groups. Grouping icons provokes participants to regard the functions in a group as belonging together, which we achieved by separating icons with a vertical line (see Figure 4.2). If our groups of icons match the user's mental model, it might speed up the search for a specific function. The second way of using whitespace was the addition of other icons (and functions) to the toolbar. The BibTeX key generation icon was added because of it being used multiple times in our thinking aloud experiment, even though other ways of key generation were available. While no other icons were added in the prototype, it should be done depending on the users' wishes, or the toolbar could be made customizable by the developers to let the users decide for the available functions themselves.

## 4.2. The Main Menu

To give guidance for improving the main menu of JabRef, we created prototypes of all menu tabs, except the *Options* and *Help* tabs, the first of which was excluded because of its complexity and was deemed to need an own in-depth analysis, which would go beyond the scope of this thesis. The Help tab did not require a rework from our perspective. All other tabs were disassembled into their single menu items, which were then newly sorted according to multiple criteria and distributed to the tabs differently than before. The names of the tabs were not changed in the process for continuity's sake, only their content and the wording and icons of some menu items.

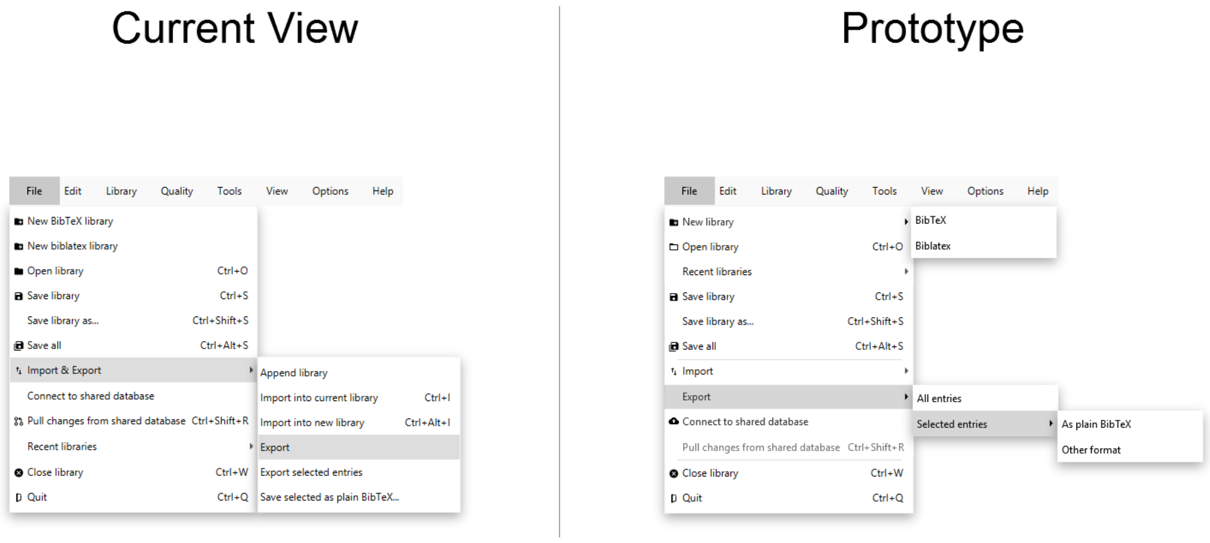


Figure 4.3.: Prototype of the *File* tab of JabRef's main menu with usability improvements, in comparison to the current version (5.0 dev).

For the redistribution, we found three criteria, which should be considered. First, menu items should be grouped according to their semantic meaning, especially inside a menu tab. In our prototype in Figure 4.3, we moved the function to open recent files closer to the *Open* function because they have a similar effect. This should be considered, especially in adding new menu items to a tab.

Another factor to consider is the workflow of the users, e.g., with the BibTeX key generation. Although the function is used in assuring the quality of a library, it has a more general use as well, for example when a new article was added. Therefore, we moved the function from the *Quality* tab to *Tools*, where it was grouped with other more general functions (see Figure 4.4). If information about the users' workflow is available or if a function depends on another one, this should be considered in the menu structure.

The third criterion for grouping functions was their application scope, meaning if they had a global effect or were applied to one, multiple, or even all entries. In our prototype, we moved two global functions away from the Tools tab (see Figure 4.4): adding entries based on an

AUX file and opening the LibreOffice and OpenOffice connection interface. They were moved to *Library* and *View* respectively (see Figure 4.5). If possible, the user should know by its location if a function has a local or global effect, and developers should design functions as liberal as possible, changing their application scope based on the context and user input.

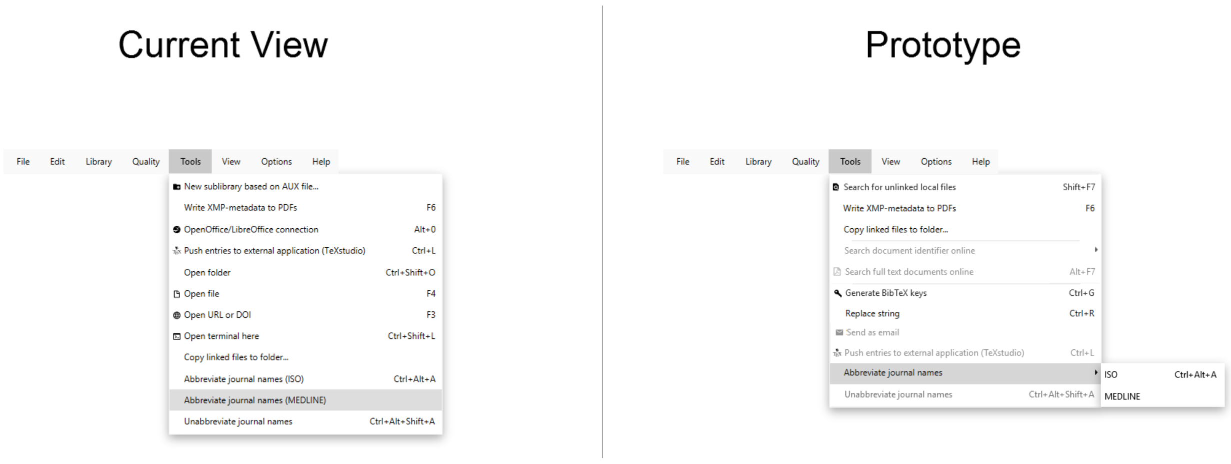

Current View
File
Edit
Library
Quality
Tools
View
Options
Help
New sublibrary based on AUX file...
Write XMP-metadata to PDFs F6
OpenOffice/LibreOffice connection Alt+0
Push entries to external application (TeXstudio) Ctrl+L
Open folder Ctrl+Shift+O
Open file F4
Open URL or DOI F3
Open terminal here Ctrl+Shift+L
Copy linked files to folder...
Abbreviate journal names (ISO) Ctrl+Alt+A
Abbreviate journal names (MEDLINE)
Unabbreviate journal names Ctrl+Alt+Shift+A
Prototype
File
Edit
Library
Quality
Tools
View
Options
Help
Search for unlinked local files Shift+F7
Write XMP-metadata to PDFs F6
Copy linked files to folder...
Search document identifier online
Search full text documents online Alt+F7
Generate BibTeX keys Ctrl+G
Replace string Ctrl+R
Send as email
Push entries to external application (TeXstudio) Ctrl+L
Abbreviate journal names
ISO Ctrl+Alt+A
MEDLINE
Unabbreviate journal names Ctrl+Alt+Shift+A

Figure 4.4.: Prototype of the Tools tab of JabRef's main menu with usability improvements, in comparison to the current version (5.0 dev).

Apart from the reallocation of information, there exist other measures to facilitate the navigation of the main menu. One of them is introducing hierarchy into the menu, which reduces the number of items on the first hierarchy level, effectively reducing search times. It is, however, important that the addition of one or more levels does not hide functions from the user by covering them up with an inappropriate umbrella term. Examples for possible hierarchies in JabRef can be found in the Figures 4.3–4.5.

Additionally, grouping without hierarchy is possible as well and can be used to visually link functions which are semantically connected. Here, we used horizontal lines to group the menu items enclosed in them together (cf. Figures 4.3–4.5). This method should be used carefully and only if the grouped functions are semantically related. If too many groups are created, the cognitive load on the user is not decreased as intended, but raised by the additional information instead.

A suitable information structure can be supported by contextual computing. The method of progressive disclosure describes that menu functions which are currently unavailable are grayed out to signal their inactivity (Lowdermilk, 2013). This lets the user focus on the important, active menu items on the one hand, and on the other hand, notifies them of the inactivity of functions which they might want to use, making error messages obsolete. The best example can be found in Figure 4.4, where functions are disabled, which can only be used when an entry is selected. Progressive disclosure is a tool with two purposes regarding usability and should be used as long as the requirements to activate a disabled function are known to the user.

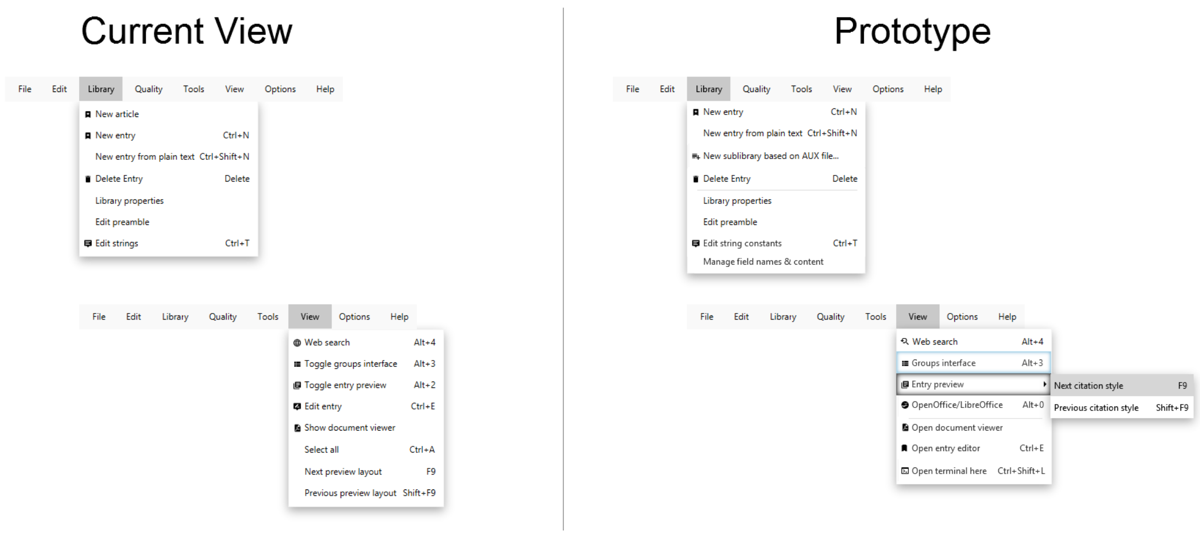

Current View
Prototype
File Edit Library Quality Tools View Options Help
New article
New entry Ctrl+N
New entry from plain text Ctrl+Shift+N
Delete Entry Delete
Library properties
Edit preamble
Edit strings Ctrl+T
New entry Ctrl+N
New entry from plain text Ctrl+Shift+N
New sublibrary based on AUX file...
Delete Entry Delete
Library properties
Edit preamble
Edit string constants Ctrl+T
Manage field names & content
Web search Alt+4
Toggle groups interface Alt+3
Toggle entry preview Alt+2
Edit entry Ctrl+E
Show document viewer
Select all Ctrl+A
Next preview layout F9
Previous preview layout Shift+F9
Web search Alt+4
Groups interface Alt+3
Entry preview
Next citation style F9
Previous citation style Shift+F9
OpenOffice/LibreOffice Alt+0
Open document viewer
Open entry editor Ctrl+E
Open terminal here Ctrl+Shift+L

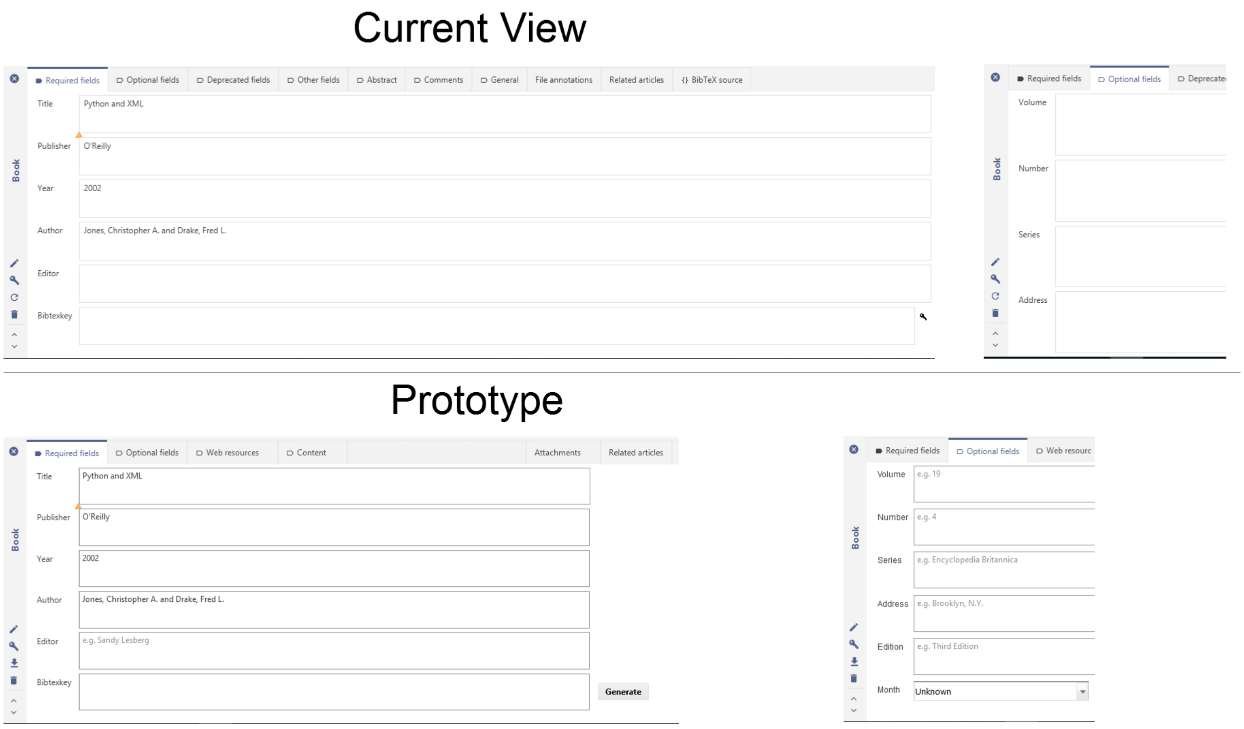
Figure 4.5.: Prototype of the Library tab (top) and the View tab (bottom) of JabRef's main menu with usability improvements, in comparison to the current versions (5.0 dev).

Lastly, some icons and wordings were changed in the main menu to help the user in finding the function they need and in understanding what it does. As was already mentioned previously, icons should be recognizable and be used consistently throughout the application. For the first reason, we removed an icon in the File tab, which was used for the *Pull changes from shared database* function, which was too small to be identified. Instead, we added the cloud icon for the function to connect to a shared database, which is easily discernible and is based on a common metaphor (cf. Figure 4.3). Icons should only be used in the main menu if they are discernible at a small size and have a unique meaning or if a function should gain attention. Otherwise, they could lead to misunderstandings about a function or make the information search harder for the user.

To avoid such misunderstandings, concise wording of menu items is equally necessary. Therefore, they should be written in the language of the user, and not rely on technical terms, which is why we removed the word *toggle* from the functions in the View tab (see Figure 4.5). Instead, we propose to implement the menu options themselves as toggle buttons, which are a common element of interaction. Apart from vocabulary issues, functions should be named as descriptive as possible, while still having an appropriate length. We abode by this principle in our prototype, where we renamed the function *Look up full text documents* by adding *online* to the end, specifying that the function performs a web search, not a local one. In the same manner, we added *local* to the function searching files locally (see Figure 4.4).

## 4.3. The Entry Editor

The entry editor of JabRef was shown to have an information structure which doesn't fit the users mental models. Furthermore, it contained too much information, which did not help the user in their work, leading to extensive changes in our prototype. Because the *Required fields* and *Optional fields* tabs were dynamically generated, those tabs were only updated concerning their layout and visual properties, while other tabs were restructured or even dismissed in the prototype to facilitate the users' work. The tabs *Other fields* and *Desprecated fields* were removed completely (see Figures 4.6–4.9).



Current View

Title: Python and XML
Publisher: O'Reilly
Year: 2002
Author: Jones, Christopher A. and Drake, Fred L.
Editor:
Bibtexkey:

Volume:
Number:
Series:
Address:

Prototype

Title: Python and XML
Publisher: O'Reilly
Year: 2002
Author: Jones, Christopher A. and Drake, Fred L.
Editor: e.g. Sandy Lesberg
Bibtexkey:

Volume: e.g. 19
Number: e.g. 4
Series: e.g. Encyclopedia Britannica
Address: e.g. Brooklyn, N.Y.
Edition: e.g. Third Edition
Month: Unknown

Figure 4.6.: Prototype of the Required fields tab and part ofthe Optional fields tab of JabRef's entry editor with usability improvements, in comparison to the current versions (5.0 dev.).

Although these two tabs conformed to the BibTeX or Biblatex standard, the difference between them and the Optional fields tab is not important for the workflow of the user. Instead, they are bound to the citation styles required for publications, as most users are researchers at a university. Additionally, the two removed tabs often just contained copies of fields in other tabs, which adds unnecessary information and could potentially confuse the user. Apart from removing tabs, multiple tabs were removed in name only and their contents were reallocated, which will be described in the following paragraphs. Through this measure, the number of tabs was reduced from up to ten tabs to just seven tabs (see Figure 4.6), which in turn reduces the time of searching for a specific information.

For the same reason, the layout of tab headers was changed too (cf. Figure 4.6). Two groups of tabs were created, one representing the information of an entry itself, the other one containing possibilities to further work with the entry. These groups visualize two different workflows of JabRef users, which either use JabRef just as a database for references, or want to work with these references and its corresponding documents. It should be noted that, in spite of the changes, the order of tabs was preserved to keep the continuity with previous versions of JabRef.

The changes in the Required and Optional tabs transformed their appearance and interaction with the user. The borders of text fields were darkened to match the overall changes mentioned in Chapter 4.1. Furthermore, text input fields in the Required tab were shortened, those in the Optional tabs were reduced in height and width (see Figure 4.6). We did this to enable users to ingest multiple entry details at once, without needing to continuously shift their gaze between them. While big input fields make it, so the text is completely visible regardless of its length, the resulting space also makes it more difficult for the user to get an overview of all information if multiple input fields are presented. A way to counter this can be the implementation of a maximum size for input fields. While minimal width and height for input and display fields exist to keep the text readable even at small interface sizes, a maximum of those parameters would make sure that a stretched interface does not cause impractically large input fields.

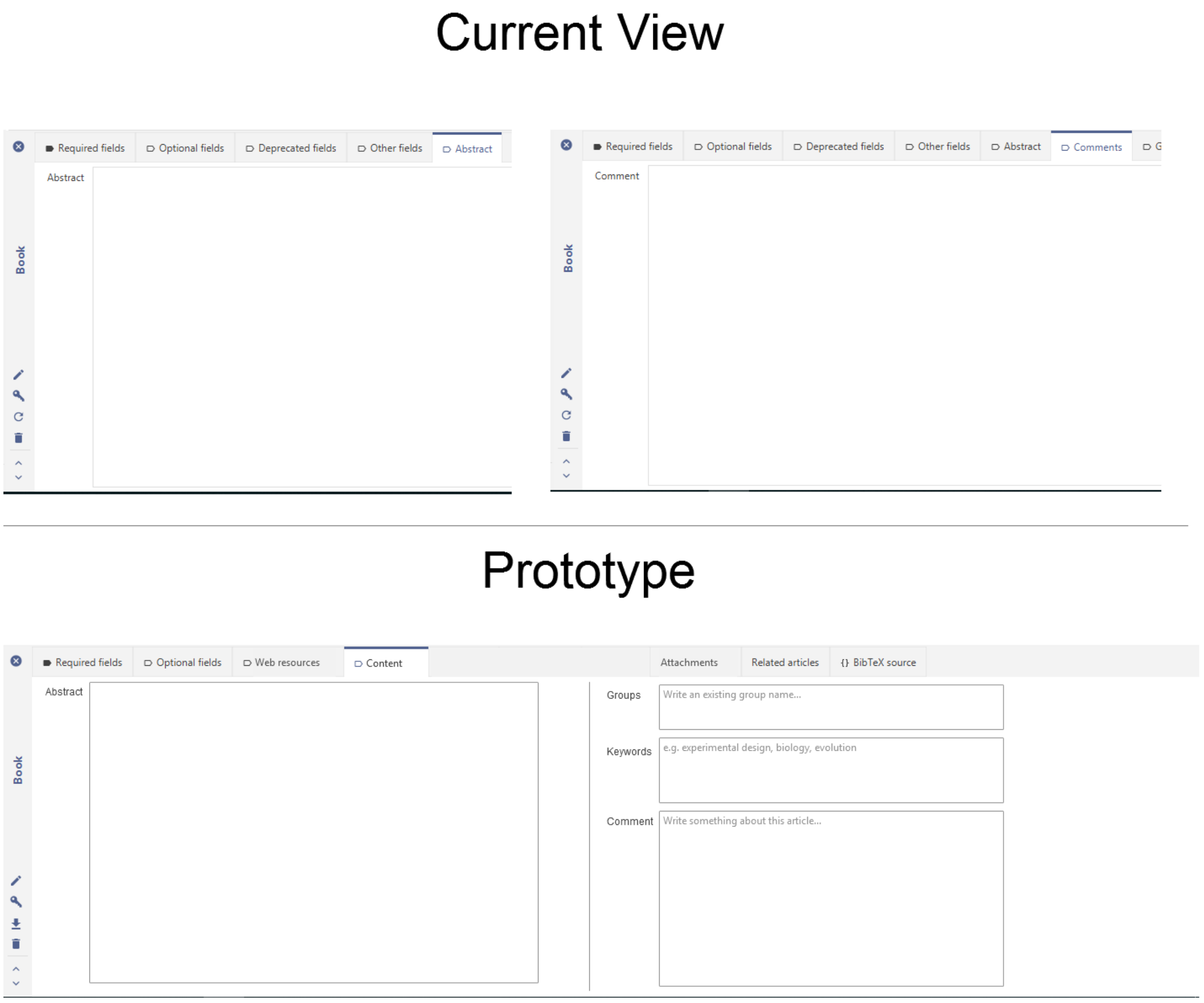
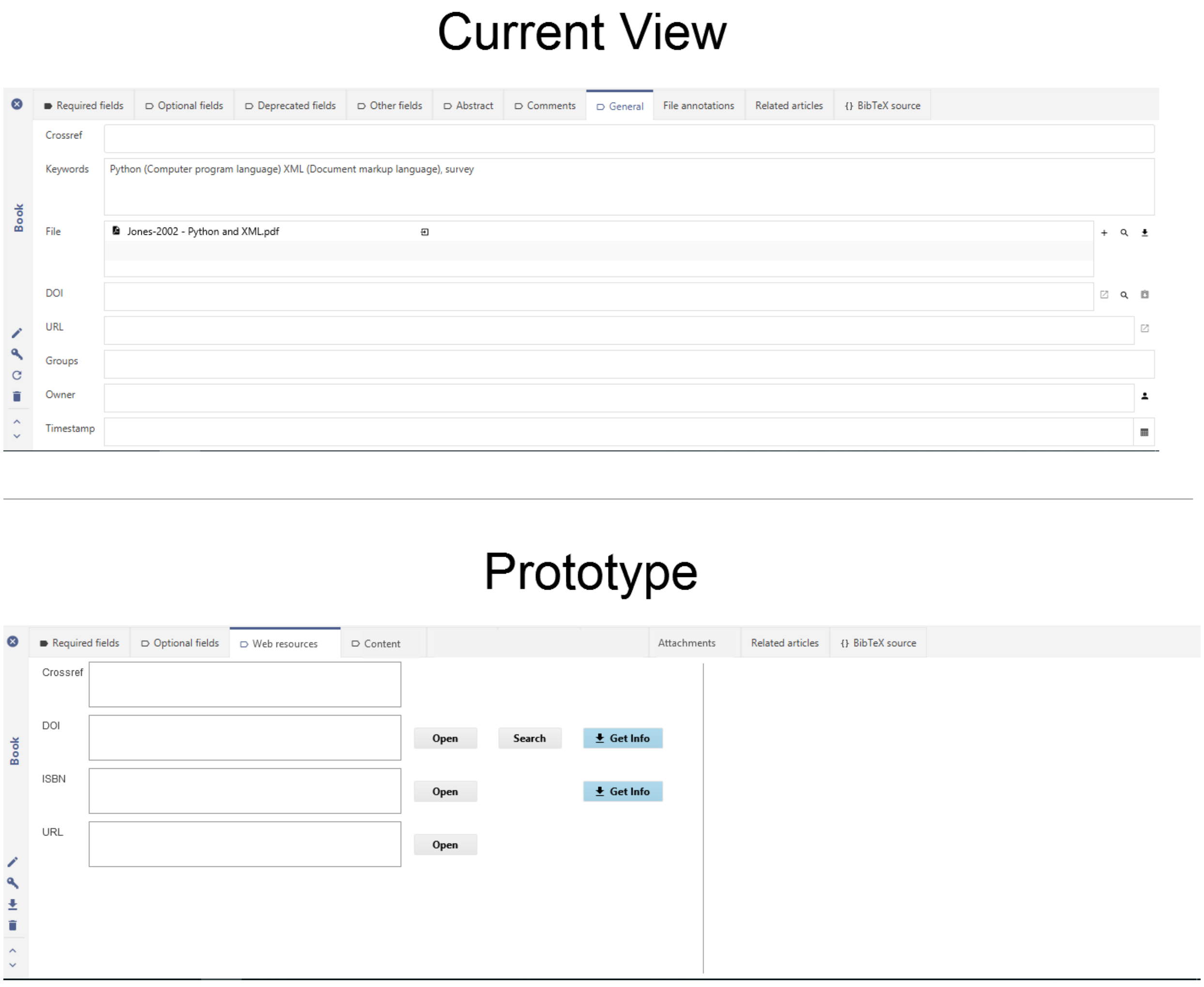


Figure 4.7.: Prototype of the Content tab of JabRef's entry editor, in comparison to parts of the current *Abstract* and *Comment* tabs in JabRef version 5.0 dev.

A change which is also targeted at the input fields is the suggestion text, which is visible in gray letters in empty input fields and gives the user examples for the content of the respective field. As soon as an input is added, the text vanishes. The main purpose of this text is to inform the user about the input expected by a field and its form. In our prototype, for example, the suggestion text of the field *Edition* shows the user that an ordinal number is the intended content of the field and that it should ideally be entered in words (see Figure 4.6). When using suggestion text, it is important to make the text appear different from a regular input, which could confuse the user. For that reason, apart from graying out the text, we began our suggestions with *e.g.*, to leave no doubt about the function of our text. If the purpose of an input field is not fixed and its input varies too much to give a fitting example, a simple invitation to write something can also be used (cf. Figure 4.7).



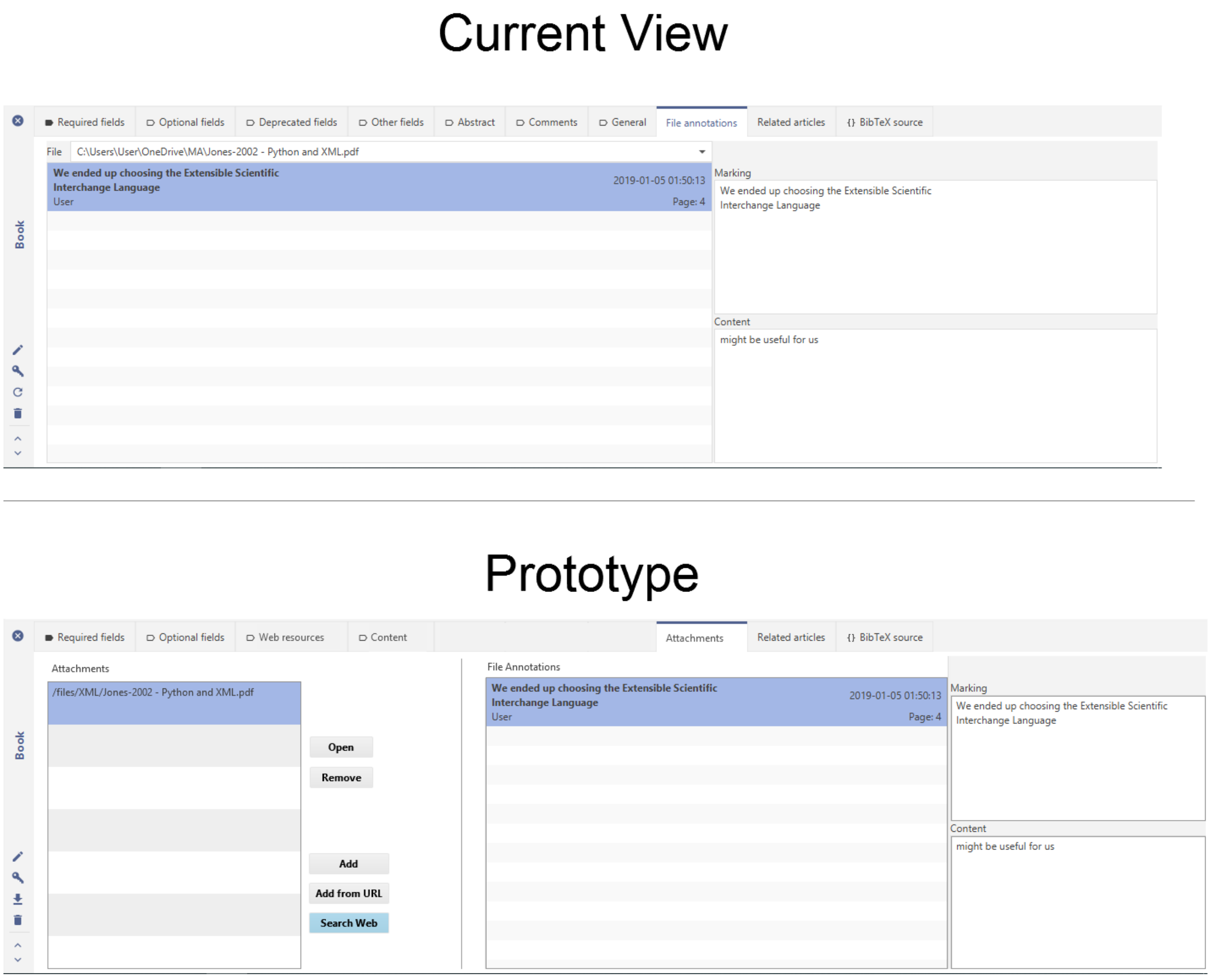
Keywords: Python (Computer program language) XML (Document markup language), survey

File: Jones-2002 - Python and XML.pdf

Figure 4.8.: Prototype of the *Web resources* tab of JabRef's entry editor, in comparison to the current *General* tab in JabRef version 5.0 dev.

Another method to facilitate the input of entry details for the user is the use of drop down fields, whenever the user has to choose between a limited number of preset values, as is the case in the current version and the prototype with the *Month* field (see Figure 4.6). Choosing a drop down field instead of text input eliminates spelling mistakes and gives the user a selection of all options without him having to remember them. If it should stay possible to add new options to the list, a combination field of text and drop down input is also possible. However, the drop down attribute of the input field should be obvious from its design, which should differ from that of normal text fields.

At last, we modified the buttons tied to a single text field, which were flat and very small in the 5.0 dev. version. We increased their size and visually elevated them by coloring the button, which is a technique from Material Design (Google, 2018). Making them prominent and elevated signals the element's interactivity and helps the user to recognize it as a button (cf. Figure 4.6).



File C:\Users\User\OneDrive\MA\Jones-2002 - Python and XML.pdf

We ended up choosing the Extensible Scientific Interchange Language
User
2019-01-05 01:50:13
Page: 4

Marking
We ended up choosing the Extensible Scientific Interchange Language

Content
might be useful for us

Attachments
/files/XML/Jones-2002 - Python and XML.pdf

File Annotations
We ended up choosing the Extensible Scientific Interchange Language
User
2019-01-05 01:50:13
Page: 4

Marking
We ended up choosing the Extensible Scientific Interchange Language

Content
might be useful for us

Figure 4.9.: Prototype of the *Attachments* tab of JabRef's entry editor, in comparison to the current *File Annotations* tab in JabRef version 5.0 dev.

The Content tab in our prototype contains both the *Abstract* and the *Comment* tab, which had only one field each with the same name as the tab (see Figure 4.7). Additionally, we moved the *Keywords* and *Groups* fields there from the General tab (see Figure 4.8), which is an arrangement backed by the data from the card sorting. We adjusted the height of the text fields according to the expected length of the text displayed by them.

For the new tab *Web resources* we included the fields DOI, Crossref and URL from the General tab and all other identifiers located in the dynamically generated tabs (see Figure 4.8).

To gather all these fields in one tab also fits our card sorting results and it enables us to bundle all interaction possible with these fields. This includes opening a resource in the browser, searching for the resource online and updating the entry from the resource's information. Although only the DOI currently supports all three actions, the button layout was the same for every field, the only difference being that the buttons for unavailable functions were missing (see Figure 4.8). The identical layout requires the user to only commit one layout to memory and still be able to navigate all others just as fast. Additionally, the user can see at first glance if one of the functions is available or not.

In the same way as in the *Required fields* tab, we transformed the buttons to be recognizable as such and used text labels instead of icons, which were much more concise in their meaning. Using icons as labels only should be limited to cases, where little space is an issue or if the icon is completely unambiguous. Preferably, an icon should be paired with text to draw attention and signal the exact meaning of the button at the same time. The icon in the *Get Info* button was added mainly to give a clear meaning to the corresponding icon in the sidebar of the entry editor (see Figure 4.8).

For the left side of the new Attachments tab, the *File* field from the General tab was used and expanded, and for the right side, the content of the File annotations tab was compressed horizontally and used without big changes (see Figure 4.9). The only difference in the list of annotations is the removed drop down selector for an attached document, which is obsolete due to the presence of the list of attached files on the left. Selecting a document there will automatically show the file annotations of that file if possible, which demonstrates the advantages of managing files and their annotations in one view.

The file list, however, was changed to a greater extent (cf. Figure 4.8). The fields height was increased and its width decreased, so it would fit the new layout. For the interaction with the files, a vertical row of buttons was added. We split the buttons into two groups with different purposes. The lower group of buttons contained functions to add new attachments, while the upper one consisted of buttons for managing existing attachments (see Figure 4.9). Apart from the space between them, the size of the buttons was used to differentiate between the two groups. Apart from the difference between the groups, the button *Search Web* is colored in JabRef's accent color. A special size or color of a button should always be accompanied by a special function, which it has. In this tab and the Web resources tab, we used an accent colored button to mark the most intrusive function, adding an unknown attachment or unknown entry information respectively.

## 4.4. The Group Feature

While improving the entry editor was mainly based on removing and restructuring information display, the group feature suffered from its extreme functional complexity. Therefore, we decided to reduce the functions hidden in the *Add group* dialog and give more content and interaction to the group sidebar itself.

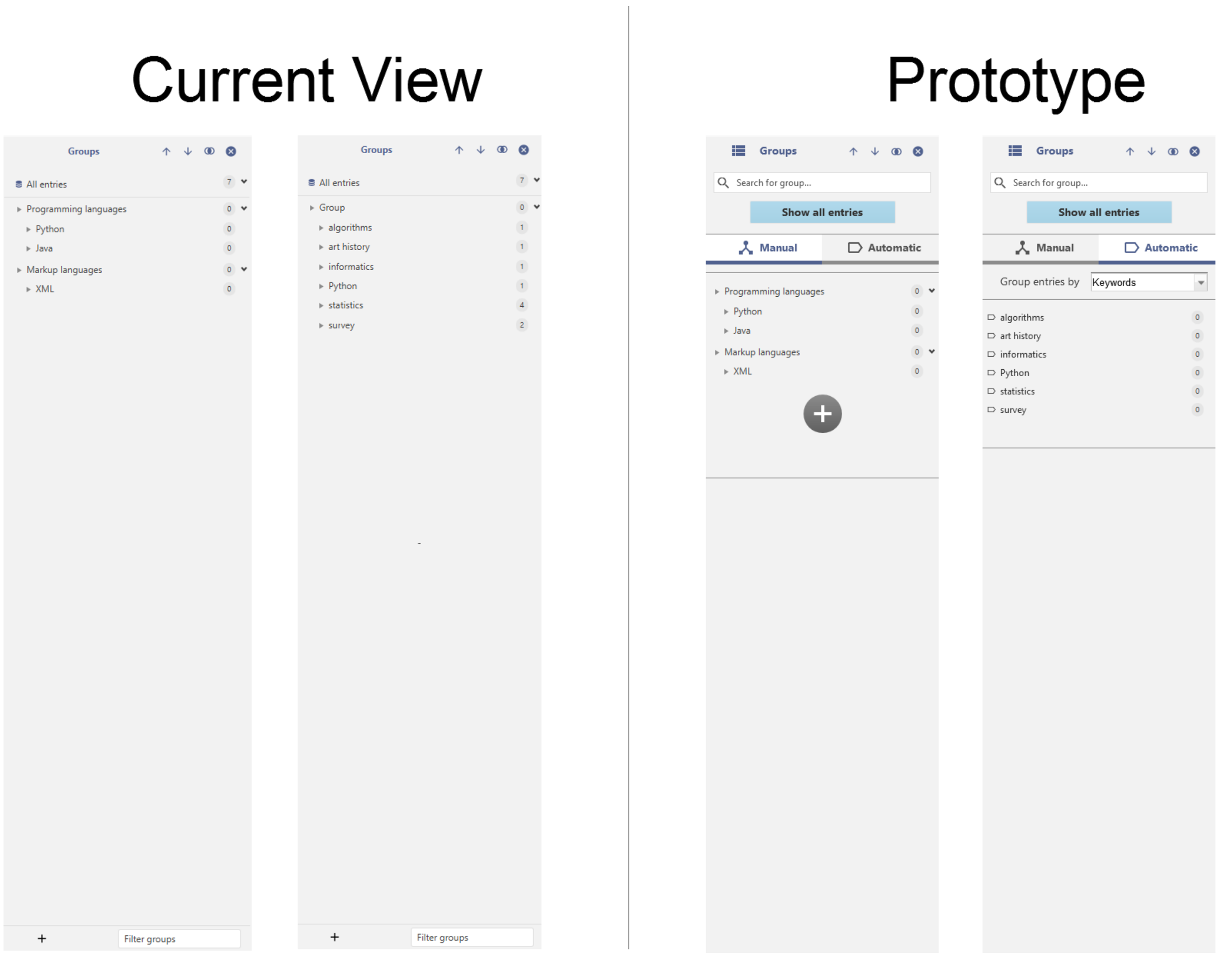
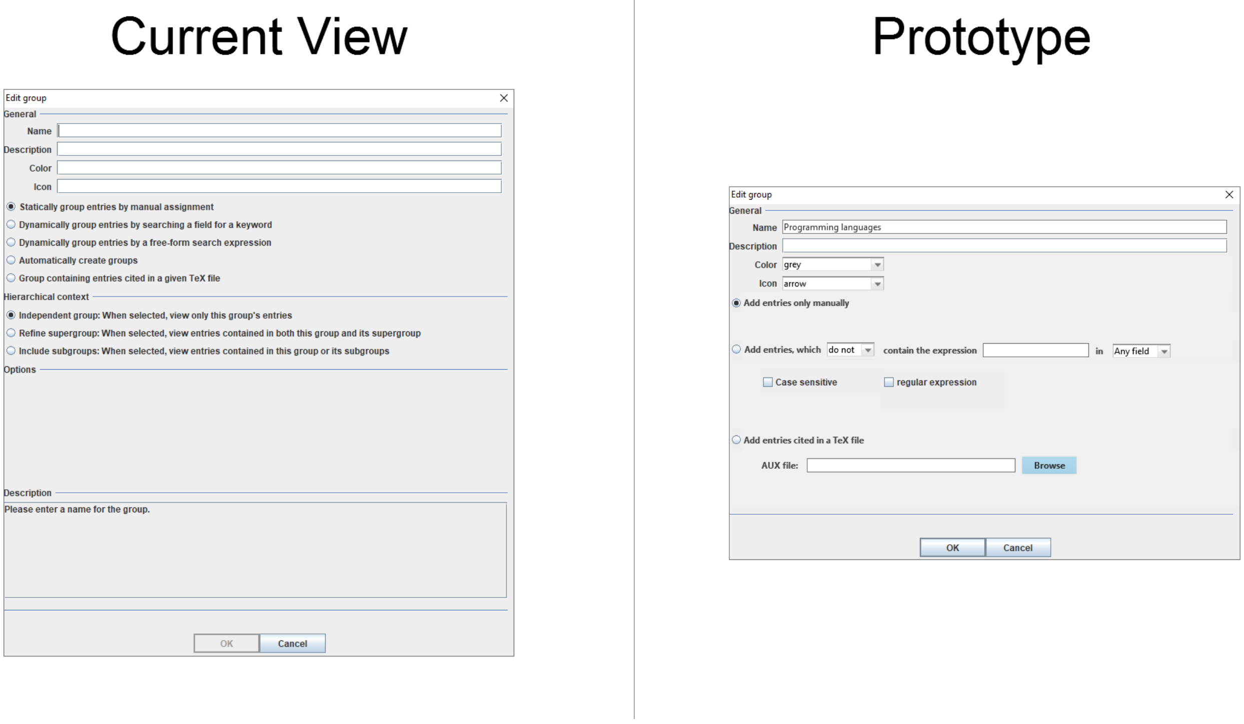


Figure 4.10.: Prototype of the two views of the group sidebar with manual grouping (left) and automatic grouping (right), compared to the current sidebar (version 5.0 dev.) mimicking the two views of the prototype.

Our suggested changes for the sidebar started with the header, where the group icon used in the main menu was added for consistency. We then moved the filter function for groups to the top of the grouping menu and renamed it *Search for group* (see Figure 4.10) for two reasons. Firstly, the convention of search bars is for them being on the top of the application, which should be complied with. Secondly, the function being on the top moves it nearer towards the group list, its area of effect, which allows the user to evaluate the result of their search query faster.

Next up, we removed the *All entries* group, representing the supergroup of all added groups, because it did not fit the users' understanding of groups. Although it matches the technical implementation of a grouping system, human users generally do not regard an entity subsuming all elements of something a group. Here, the machine model conflicts with the mental model of the user. Additionally, by not forcing hierarchy in the grouping process, we can use groups without any hierarchy, which will be important in the following sections. Although the All entries group did not provide a suitable metaphor, it also had the use of removing the selection of a group and displaying all entries again. To keep this function available for the user, we added a button performing the same feature below the search bar (see Figure 4.10).

Our main task regarding the group feature was to implement two parallel grouping systems, which the users had wished for. We decided to draft two tabs for the group sidebar, between which the users could freely switch. One tab was called *Manual* and was supposed to offer similar functions as the existing feature, with which entries could freely be added to and removed from groups. The *Automatic* tab was supposed to let entries be grouped by JabRef, which would not allow entries to be moved manually between groups. Instead, they were assigned to the groups based on the values in one BibTeX field, which the user could choose and change to their liking. The style of the tabs was inspired by Material Design (Google, 2018) with font colored in JabRefs theme color in their active state, and in gray when they were inactive (cf. Figure 4.10).



Current View

Edit group

General

Name:
Description:
Color:
Icon:

[✓ Statically group entries by manual assignment]
Dynamically group entries by searching a field for a keyword
Dynamically group entries by a free-form search expression
Automatically create groups
Group containing entries cited in a given TeX file

Hierarchical context

[✓ Independent group: When selected, view only this group's entries]
Refine supergroup: When selected, view entries contained in both this group and its supergroup
Include subgroups: When selected, view entries contained in this group or its subgroups

Options

Description

Please enter a name for the group.

OK | Cancel

Prototype

Edit group

General

Name: Programming languages
Description:
Color: grey
Icon: arrow

[✓ Add entries only manually]

Add entries, which do not contain the expression in Any field

Case sensitive
regular expression

Add entries cited in a TeX file

AUX file: Browse

OK | Cancel

Figure 4.11.: Prototype of the *Add group* dialog compared to the current version in JabRef 5.0 dev.

As already mentioned, in the Manual tab, users could add their own groups and subgroups and add entries to them like in a hierarchic folder system. This system used the hierarchy metaphor, which is visualized in Figure 3.12, but added an exception for groups of the highest level of the hierarchy. With the removal of the All entries group, the groups on that level had no group above them and, therefore, could contain entries that are in other groups on the same hierarchy level. This exception was important to keep the consistency with the metaphor, which we used to better match the users' mental model. While the group list does not visually differ from the existing one, the button to add groups was taken from the bottom of the sidebar and moved directly to the bottom of the group list for the same reason as for the search bar:

to bring it closer to its area of effect. The button was also colored, to be more prominent and placed in the middle of the sidebar.

In the Automatic tab, groups could not be added by the user one by one. Instead, JabRef created a group for every unique value in a certain BibTeX field, which could be chosen by the user. The default was the keyword field, so for every keyword used in any entry a group was created, containing the entries with that specific keyword. The groups had no hierarchy and used the tagging metaphor (cf. Figure 3.12), allowing an article to be in multiple groups at once. The group list differs from the existing one only in the absence of the *Add group* button and in the icon used for the groups. We chose the same icon as for the *Manage keywords* function in the main menu, which fit the default field used for the automatic grouping. The biggest difference to the Manual tab, however, is the drop down field on top of the group list which lets the user switch the field by which the groups are created (see Figure 4.10). The field does not stand alone but is preceded by a phrase that is completed by the value of the drop down field, which is a natural way to signal the result of a change in that field.

Additionally to the changes in the sidebar, the *Add group* dialog was reduced, as its complexity seemed to be a major source of confusion with the group feature altogether. By updating the sidebar in the way we did, we were also able to simplify the dialog in multiple ways. The hierarchical context options were removed completely because the *include subgroups* context was chosen as the default hierarchy mode and to select it anew for every group was deemed unnecessary (see Figure 4.11). While freedom of choice is important in fulfilling user needs, that freedom can also hinder the user if the number of required decisions becomes to high. Therefore, setting good default values can decrease the cognitive load on the user and at the same time make some decisions obsolete. This does not imply that user flexibility has to be discarded completely. In this case, the hierarchy options could be added to JabRef's general options, the *Preferences*.

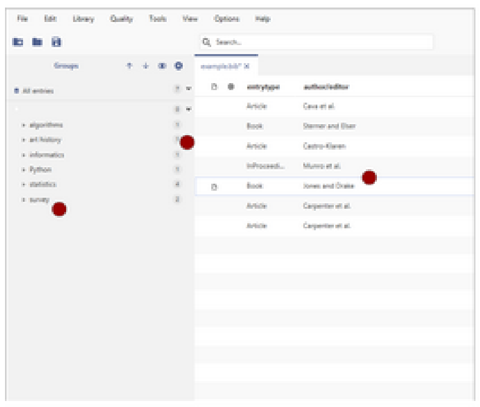
Concerning the five optional group creation methods in the dialog, we managed to reduce the complexity as well. The option to automatically create groups was made obsolete by the Automatic tab in the sidebar. The methods to group entries which contained a certain expression either in a particular BibTeX field or just somewhere in the entry were combined into one, and the process to use that method was made more linear. This was achieved by embedding the various options in a sentence, which would then describe, which entries were added to a group (see Figure 4.11). The method to add entries cited in an AUX file to a group was not modified, except for the added *Browse* button, so users could add a file path without using text input.

Through the changes made to the group creation methods, we were able to cut the *Options* section of the dialog (cf. Figure 4.11). Instead, the options for the selected method were displayed directly beneath it, which enables the user to explore multiple options without constantly shifting their gaze from the method selection to the Options section and back. The space saved by this could be used to unfold all options and not hide them until a creation method is selected. This way, the user has an overview of all possibilities available to them at one glance. We also removed the description section, which was found to not aid the user in understanding the group creation better, but rather just takes more time for the user to comprehend. Even though well designed descriptions can be a useful tool to help the user with extensive or inherently complex functions, they should be a last resort for when all effort for simplification had been unsuccessful.

## 4.5. Communication of Results

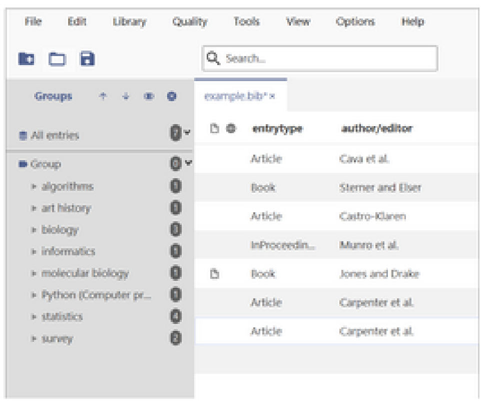
While the guidelines for improving JabRef are the centerpiece of this thesis, communicating these insights to the JabRef developers needs to be addressed as well.

Current View



| entrytype | author/editor |
|---|---|
| Article | Cava et al. |
| Book | Sterner and Elser |
| Article | Castro-Klaren |
| InProceedi... | Munro et al. |
| Book | Jones and Drake |
| Article | Carpenter et al. |
| Article | Carpenter et al. |

Prototype

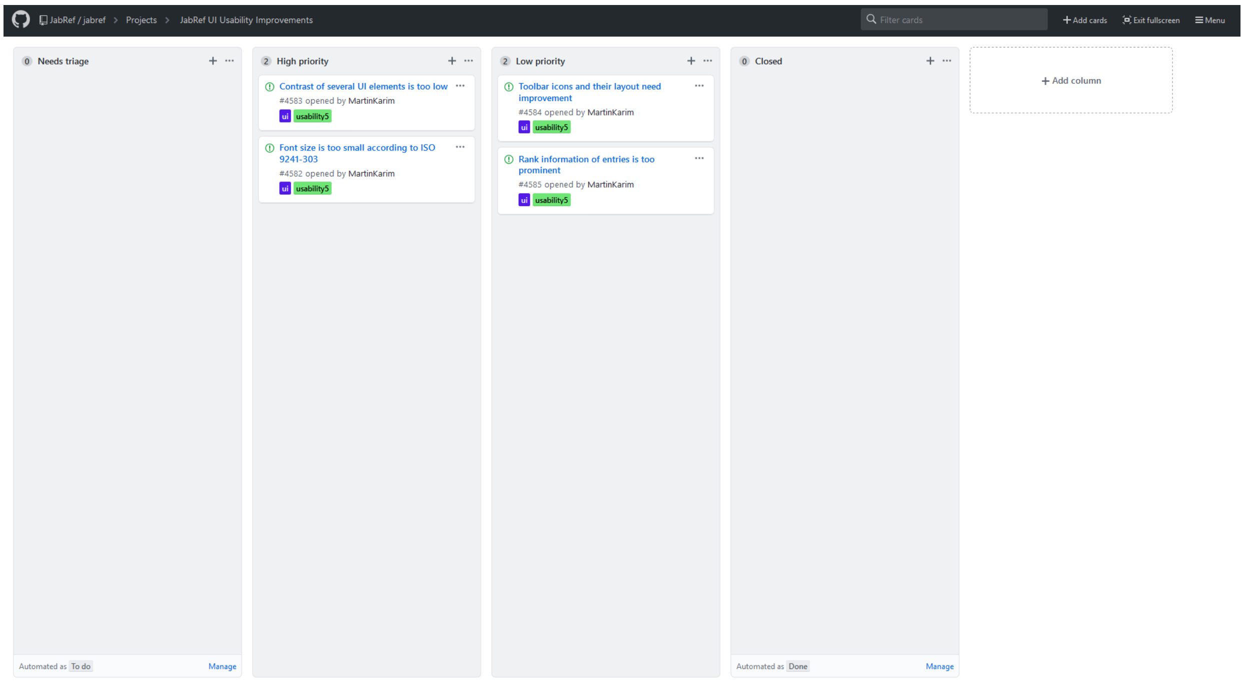


| entrytype | author/editor |
|---|---|
| Article | Cava et al. |
| Book | Sterner and Elser |
| Article | Castro-Klaren |
| InProceedin... | Munro et al. |
| Book | Jones and Drake |
| Article | Carpenter et al. |
| Article | Carpenter et al. |

| Description of Issue | Possible Solution | Additional Remarks |
|---|---|---|
| Contrast between entry counters in sidebar and background is too low; Sidebar & toolbar not visually separated; Contrast between alternative rows in entry table not high enough; | Darker background color of sidebar; Darken color of entry counters to contrast of 3:1; Darker alternative row background | Contrast of font should still be high enough; |

Figure 4.12.: Example of a github issue with prototype images and information table.

With interdisciplinary communication being difficult even in a well organized corporate environment, it presents an even greater challenge in the more flexible ecosystem of an open source community like JabRef's, where personal contact with the contributors is rare and every developer is only bound by their own ambition. Therefore, we need to ensure that the documentation of our results meets two criteria. First, it has to be straightforward and must be comprehensible even for people, who are not usability professionals, and second, it needs to incentivize developers to work on these issues, by highlighting the rewarding result of solving them. To fulfill this second requirement, we have an advantage in reporting usability issues over the implementation of most software features, which is our high fidelity prototypes. By using imagery of the desired look of the software, developers do not only have a task to fulfill but are also given a clear goal, towards which they can work. However, the use of images alone would neglect our first criterion, as the prototypes merely give a copy of an outcome, but can not draw attention to what parts in it are actually important. Therefore, we decided to support imagery with structured text, referencing the crucial parts of the interface and explaining the existing problem and its solution.



0 Needs triage

2 High priority

- Contrast of several UI elements is too low — #4583 opened by MartinKarim — ui usability5
- Font size is too small according to ISO 9241-303 — #4582 opened by MartinKarim — ui usability5

2 Low priority

- Toolbar icons and their layout need improvement — #4584 opened by MartinKarim — ui usability5
- Rank information of entries is too prominent — #4585 opened by MartinKarim — ui usability5

0 Closed

Figure 4.13.: Github project dashboard showing four issues based on the results of this thesis. The two columns in the middle are used to assign high or low priority to an issue.

The platform we chose for reporting our results was the Github repository of JabRef, where a bug reporting infrastructure was already present in the form of an issue tracker. Although there exists an official website of JabRef, by presenting our findings directly on Github it was likely that developers would find and attend to our results faster because most exchange and coordination between JabRef developers takes place there. Therefore, we divided our suggestions for changes into smaller packages and created an issue in JabRef's issue tracker for them. An example of such an issue can be seen in Figure 4.12. Like in this thesis, an image of the current user interface was shown at the top, opposed by the improved prototype. Beneath it, an additional table with text was located. The table described the usability problem in the current layout as short and precise as possible. If the problem was confined to a specific area of the UI, the location was marked in the image of the current user interface with red markers. Along with the description of the problem, possible solutions or approaches to the problems were listed, which conformed to the changes implemented in the prototype. The last piece of information were additional remarks, which could contain warnings about potential pitfalls in the implementation of the suggested solution or a reference to an external source, e.g., design guidelines.

Additionally to generating issues, we created a github project in the JabRef repository for all issues that stemmed from our findings. The project dashboard (see Figure 4.13) provides an overview over all issues filed for our results, and enables the maintainers of JabRef to sort them by their priority. In an attempt to complement this very detailed overview of issues, we created a more general one, which could serve as a pint of orientation for the managers of the project to identify and prioritize more pressing issues over less crucial ones (see Table 4.1).

Table 4.1.: List of prototype views with an urgency rating for their content

| Prototype View | UI Part | Urgency |
|---|---|---|
| General Issues | All | Low |
| File Tab | Main Menu | Medium |
| Library Tab | Main Menu | Low |
| Tools Tab | Main Menu | High |
| View Tab | Main Menu | Medium |
| Required Fields | Entry Editor | Low |
| Optional Fields | Entry Editor | Low |
| Content Tab | Entry Editor | Low |
| Web Resources Tab | Entry Editor | High |
| Attachments Tab | Entry Editor | Medium |
| Manual Grouping | Group Sidebar | High |
| Automatic Grouping | Group Sidebar | Medium |
| Add Group Dialog | Group Sidebar | High |

Our table contains every view of our prototype, stating the part of the UI that it displays, and the urgency of the usability measures suggested by it. Combined with the project dashboard, an easy prioritization of issues is possible.

# 5. Conclusion

The aim of this thesis was the improvement of JabRef's usability with the methods of user-centered design. In our literature review, we discovered that JabRef's functionality could compete with other programs of its sort, while usability had been disregarded in the past, which made the amount of necessary evaluations too high for the scope of our work. Therefore, we decided to focus our efforts only on the most important functions of JabRef, leaving us with four major goals. First, we aimed at the identification of important JabRef features and achieved this goal by conducting an online survey. The second goal of our research was the exploration of user needs and the interaction between the user and these important features. This was accomplished in the laboratory study and complemented by an expert examination of JabRef. The last two goals were the specification of possible usability improvements for the JabRef UI and the implementation of that guidance in a prototype. They were accomplished in the previous chapter, where we discussed our recommendations in detail and presented a high fidelity prototype of an improved JabRef UI.

We can conclude that all the goals of this thesis were achieved and that the results look promising. We were able to present the results of an in-depth analysis of JabRef's core features, which could be a reference point for future projects as well. Additionally, our prototypes are developed to a high fidelity stage, enabling developers to follow our guidance without much effort. As this was the first involvement of usability professionals in the JabRef community, we expect great changes based on our suggestions. One requirement for this, the successful communication of our results to the JabRef open source community was fulfilled by presenting our results on Github in a format that developers were familiar with. Therefore, we did not only generate ways to improve JabRef's usability, but we also ensured that our findings would be implemented, by adapting to the special circumstances of open source development.

There are, however, aspects of our work, which leave room for improvement in future work on the usability of JabRef. While the observational data on the users' interaction with JabRef was quite insightful, it would have been desirable to gather more data about the workflow of the users through our interview or in the online survey, instead of collecting only basic demographical data. Furthermore, the value of our prototype would be even higher, if we had used existing design guidelines to construct them, e.g., Material Design (Google, 2018), as we could have referenced these guidelines in the usability report and given the developers better guidance with exact documentation rather than with just conveying basic usability principles to them.

Albeit its shortcomings, this work can hopefully be a stepping stone for future usability work. After our suggestions have been functionally implemented, user tests have to be conducted to show if the recommended changes actually improved the usability of JabRef. Furthermore, the results of the online survey hint at several more important features of JabRef which would be in need of a usability check, but covering them in our research did exceed the scope of this thesis. One of these features would be the *Preferences*, which contain the most settings for JabRef in an unstructured and confusing way, deterring most users from adapting the software to their needs. We believe that a lot can be gained from building on the work of this thesis, and we encourage usability professionals to do so in the future and engage in making JabRef more usable.

# List of Tables

# List of Figures

# Bibliography

Abras, C., Maloney-Krichmar, D., and Preece, J. (2004). *Berkshire encyclopedia of human-computer interaction*, volume 2, chapter User-centered design. Berkshire Publ. Group, Great Barrington, Mass.

Andreasen, M. S., Nielsen, H. V., Schrøder, S. O., and Stage, J. (2006). Usability in open source software development: opinions and practice. *Information technology and control*, 35(3).

Böhner, D., Stöber, T., and Teichert, A. (2018). Softwarevergleich literaturverwaltung.

Datta, A. (2010). Soja: Collaborative reference management using a decentralized social information system. In *6th International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom 2010)*, pages 1–9.

Despalatović, L. (2013). The usability of free/libre/open source projects. *Int. J. Comput. Inf. Technol*, 2(05):958–963.

Faaborg, A. and Schwartz, D. (2010). Using a distributed heuristic evaluation to improve the usability of open source software. In *CHI'10: Proceedings of the 28th international conference on Human factors in computing systems*.

Feiner, J. and Andrews, K. (2012). Usability reporting with usabml. In *International Conference on Human-Centred Software Engineering*, pages 342–351. Springer.

Fenner, M. (2010). Reference manager overview.

Gilmour, R. and Cobus-Kuo, L. (2011). Reference management software: a comparative analysis of four products. *Issues in Science and Technology Librarianship*, 66(66):63–75.

Google (2018). *Material Design*. Google Inc.

Grobelny, J., Karwowski, W., and Drury, C. (2005). Usability of graphical icons in the design of human-computer interfaces. *International Journal of Human-Computer Interaction*, 18(2):167–182.

ISO (2006). Ergonomics of human-system interaction – part 110: Dialogue principles. Standard ISO 9241-303:2006, International Organization for Standardization, Geneva, CH.

ISO (2008). Ergonomics of human-system interaction – part 302: Terminology for electronic visual displays. Standard ISO 9241-302:2008, International Organization for Standardization, Geneva, CH.

ISO (2011). Ergonomics of human-system interaction – part 303: Requirements for electronic visual displays. Standard ISO 9241-303:2011, International Organization for Standardization, Geneva, CH.

Kopp, O., Breitenbücher, U., and Müller, T. (2018). Cloudref–towards collaborative reference management in the cloud. *ZEUS 2018*, page 56.

Lorenzetti, D. L. and Ghali, W. A. (2013). Reference management software for systematic reviews and meta-analyses: an exploration of usage and usability. *BMC Medical Research Methodology*, 13(1):141.

Lowdermilk, T. (2013). *User-Centered Design*. O'Reilly Media, Inc, USA.

Marino, W. (2012). Fore-cite: tactics for evaluating citation management tools. *Reference Services Review*, 40(2):295–310.

Nichols, D. and Twidale, M. (2003). The usability of open source software. *First Monday*, 8(1).

Nielsen, J. (1994a). Enhancing the explanatory power of usability heuristics. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, pages 152–158. ACM.

Nielsen, J. (1994b). Guerrilla hci: using discount usability engineering to penetrate the intimidation barrier. In *Cost-justifying usability*, pages 245–272. Academic Press, Inc.

Nielsen, J. and Landauer, T. K. (1993). A mathematical model of the finding of usability problems. In *Proceedings of the INTERACT'93 and CHI'93 conference on Human factors in computing systems*, pages 206–213. ACM.

Norman, D. (2013). *The design of everyday things: Revised and expanded edition*. Constellation.

Rudlof, C. (2006). *Handbuch Software-Ergonomie*.

Shneiderman, B. and Plaisant, C. (2010). *Designing the user interface: strategies for effective human-computer interaction*. Addison-Wesley.

Spencer, D., Garrett, J. J., and ProQuest (2009). *Card Sorting: Designing Usable Categories*, chapter Choose the Method, pages 51–60. Rosenfeld Media, Brooklyn, N.Y.

Trapp, A. K. and Wienrich, C. (2018). App icon similarity and its impact on visual search efficiency on mobile touch devices. *Cognitive Research: Principles and Implications*, 3(1):39.

Twidale, M. B. and Nichols, D. M. (2005). Exploring usability discussions in open source development. In *Proceedings of the 38th Annual Hawaii International Conference on System Sciences*, pages 198–198.

UIC (2018). Refworks and other citation management tools.

Viorres, N., Xenofon, P., Stavrakis, M., Vlachogiannis, E., Koutsabasis, P., and Darzentas, J. (2007). Major HCI challenges for open source software adoption and development. In *International Conference on Online Communities and Social Computing*, pages 455–464. Springer.

W3C World Wide Web Consortium (2008). WCAG2.0 HTML techniques. Standard, W3C.

Wikipedia contributors (2018). Comparison of reference management software — Wikipedia, the free encyclopedia. [Online; accessed 28-August-2018].

Wood, J. R. and Wood, L. E. (2008). Card sorting: Current practices and beyond. *J. Usability Studies*, 4(1):1–6.

Zhao, L., Deek, F. P., and McHugh, J. A. (2010). Exploratory inspection—a user-based learning method for improving open source software usability. *Journal of Software Maintenance and Evolution: Research and Practice*, 22(8):653–675.

# Appendices

# A. Telemetric Frequency Data

| Frequency | Description | Comment |
|---|---|---|
| 11618 | create new document | |
| 4059 | import external file | automatically opens at input of non .bib files |
| 1316 | merge duplicates | |
| 1282 | merge entry from web search | |
| 1060 | install updates | not relevant |
| 522 | add/edit groups | |
| 500 | import entry PDF metadata | |
| 412 | merge edits of an externally modified entry | |
| 408 | choose preferences | |
| 126 | show found online documents | |
| 100 | define string constants | |
| 78 | change bibliography-specific options | |
| 73 | create change fields for an entry type | |
| 60 | create an entry with unformatted text | |
| 60 | change fields shown in tabs Abstract, General, Comments | |
| 58 | search unlinked PDFs | |
| 53 | add jstyle file | |
| 42 | connect to database | |
| 36 | add individual import filters | |
| 35 | merge multiple libraries | |
| 35 | show selected preferences in overview | |
| 32 | ??? | not present anymore |
| 30 | add export filter | |
| 28 | change font options | |
| 21 | replace strings in entries | |
| 17 | generate library from AUX file | |
| 17 | add import filters | |
| 16 | edit library preamble | |
| 14 | merge two entries | |
| 14 | edit the type of a linked external file | |
| 11 | create keywords for entries | |
| 7 | change BibTeX key pattern | |
| 5 | show metadata written to PDF | |
| 2 | merge edits of an externally modified entry | |

| | |
|---|---|
| 1 | add a file with protected terms for import |
| 1 | confirm writing XMP-metadata for selected entries |

# B. User Survey

## Questionnaire for JabRef Users

Welcome, JabRef user!

Thank you for taking part in our survey concerning the features of JabRef.

To better understand how people use JabRef, we need your learn about how you use the software. This will help us to improve the functionality, especially the user interface of JabRef.

This study is conducted in the scope of a Master's thesis at the Technical University of Munich (TUM) in collaboration with the JabRef Developers.

If you have any questions or concerns please contact Linus Dietz <linus.dietz@tum.de>, who is a JabRef Maintainer and advisor to this thesis. For any general JabRef-related issues such as bug reports please use the issue tracker on Github https://github.com/JabRef/jabref or ask your question in the forums http://discourse.jabref.org/.

Please answer the questions on the following pages. It will only take ca. 5 MINUTES.

* Required

### Important Features of JabRef

In the following questions, we would like to ask, how IMPORTANT the following features of JabRef are to you.
Importance could mean that you use that feature FREQUENTLY, or that it is NECESSARY for your work.
Basically, the more VALUE a feature has to you personally, the more important it is.
If you do not know the feature, please check the option on the FAR RIGHT.

1. **How important are these features to you?** *
   *Mark only one oval per row.*

| | not important at all | not very important | somewhat important | pretty important | absolutely crucial | I DON'T KNOW THE FEATURE |
|---|---|---|---|---|---|---|
| Ranking references with a star rating | ◯ | ◯ | ◯ | ◯ | ◯ | ◯ |
| Downloading references from bibliographic databases/websites | ◯ | ◯ | ◯ | ◯ | ◯ | ◯ |
| Searching databases and websites from within JabRef | ◯ | ◯ | ◯ | ◯ | ◯ | ◯ |
| Sorting references into groups | ◯ | ◯ | ◯ | ◯ | ◯ | ◯ |
| Writing Comments on references | ◯ | ◯ | ◯ | ◯ | ◯ | ◯ |
| Replacing strings in references | ◯ | ◯ | ◯ | ◯ | ◯ | ◯ |

2. *

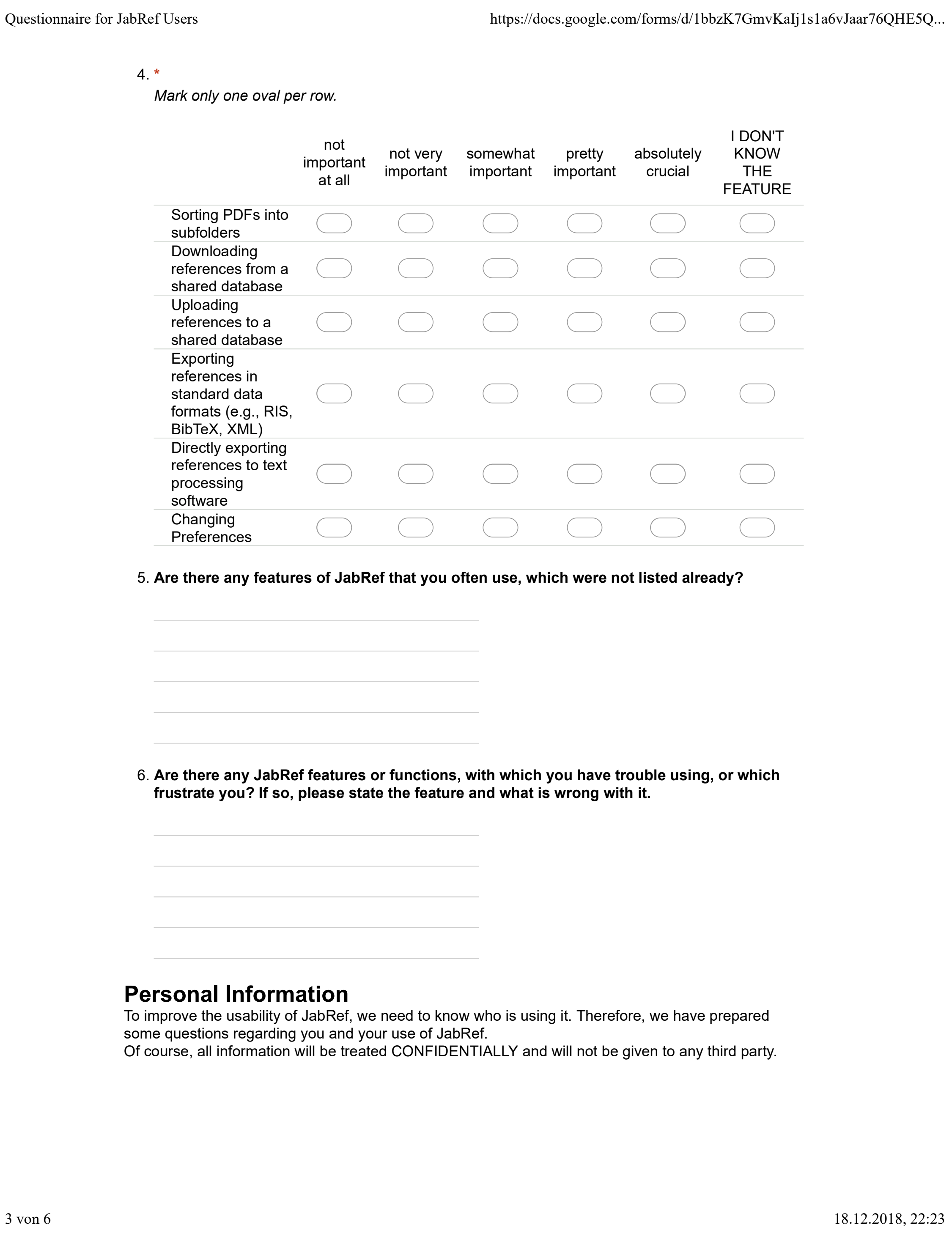
*Mark only one oval per row.*

| | not important at all | not very important | somewhat important | pretty important | absolutely crucial | I DON'T KNOW THE FEATURE |
|---|---|---|---|---|---|---|
| Adding reference based on PDF metadata | ◯ | ◯ | ◯ | ◯ | ◯ | ◯ |
| Adding a PDF/URL to a reference | ◯ | ◯ | ◯ | ◯ | ◯ | ◯ |
| Editing reference details (e.g author, year) | ◯ | ◯ | ◯ | ◯ | ◯ | ◯ |
| Editing document type | ◯ | ◯ | ◯ | ◯ | ◯ | ◯ |
| Renaming attached PDFs according to a pattern | ◯ | ◯ | ◯ | ◯ | ◯ | ◯ |
| Adding new details to reference (e.g. ISBN, page numbers) | ◯ | ◯ | ◯ | ◯ | ◯ | ◯ |

3. *

*Mark only one oval per row.*

| | not important at all | not very important | somewhat important | pretty important | absolutely crucial | I DON'T KNOW THE FEATURE |
|---|---|---|---|---|---|---|
| Cleaning up references automatically | ◯ | ◯ | ◯ | ◯ | ◯ | ◯ |
| Viewing PDF in JabRef | ◯ | ◯ | ◯ | ◯ | ◯ | ◯ |
| Checking formal correctness of references | ◯ | ◯ | ◯ | ◯ | ◯ | ◯ |
| Merging two references into one | ◯ | ◯ | ◯ | ◯ | ◯ | ◯ |
| Deleting references | ◯ | ◯ | ◯ | ◯ | ◯ | ◯ |
| Searching for duplicate references | ◯ | ◯ | ◯ | ◯ | ◯ | ◯ |



Questionnaire for JabRef Users https://docs.google.com/forms/d/1bbzK7GmvKaIj1s1a6vJaar76QHE5Q...

4. *

*Mark only one oval per row.*

| | not important at all | not very important | somewhat important | pretty important | absolutely crucial | I DON'T KNOW THE FEATURE |
|---|---|---|---|---|---|---|
| Sorting PDFs into subfolders | ◯ | ◯ | ◯ | ◯ | ◯ | ◯ |
| Downloading references from a shared database | ◯ | ◯ | ◯ | ◯ | ◯ | ◯ |
| Uploading references to a shared database | ◯ | ◯ | ◯ | ◯ | ◯ | ◯ |
| Exporting references in standard data formats (e.g., RIS, BibTeX, XML) | ◯ | ◯ | ◯ | ◯ | ◯ | ◯ |
| Directly exporting references to text processing software | ◯ | ◯ | ◯ | ◯ | ◯ | ◯ |
| Changing Preferences | ◯ | ◯ | ◯ | ◯ | ◯ | ◯ |

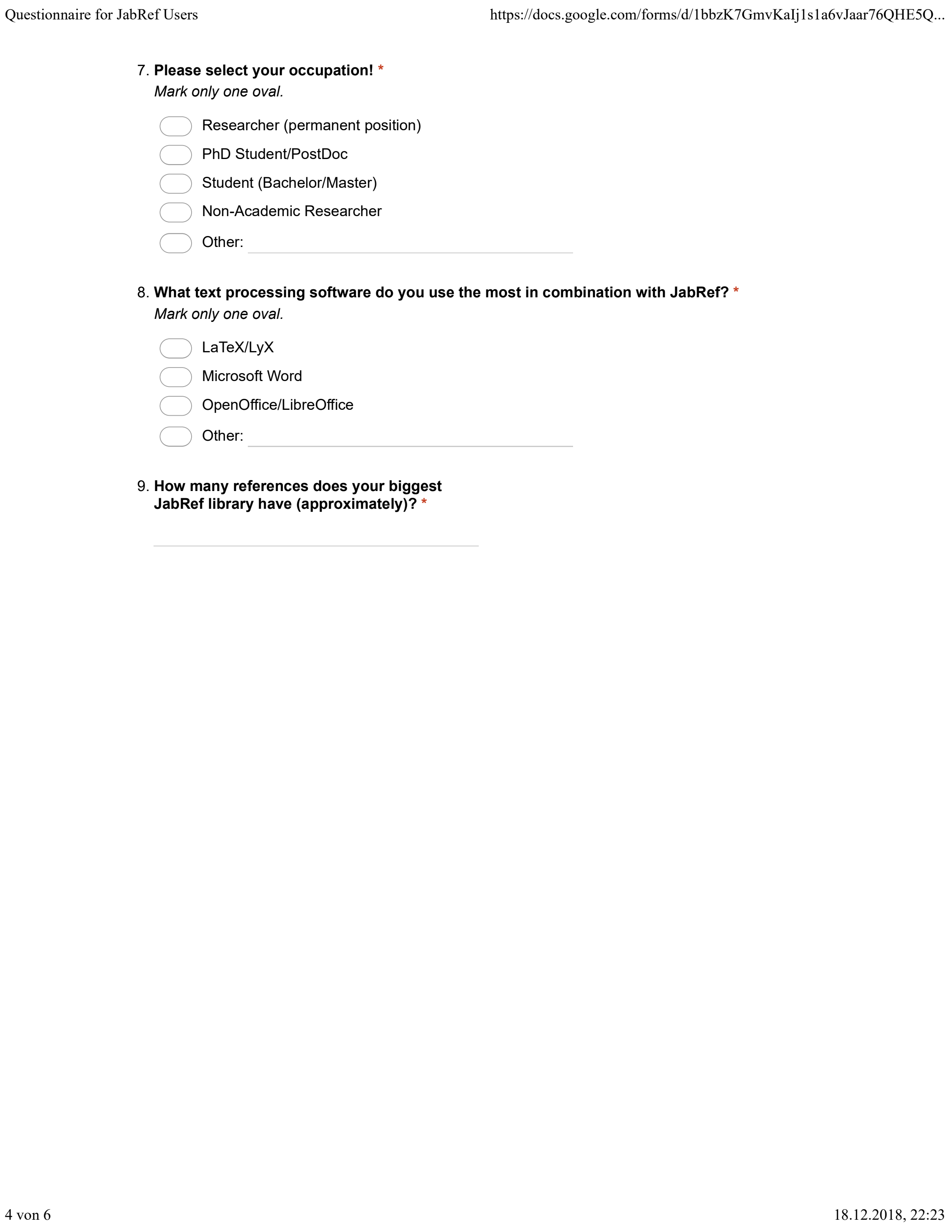
5. **Are there any features of JabRef that you often use, which were not listed already?**

6. **Are there any JabRef features or functions, with which you have trouble using, or which frustrate you? If so, please state the feature and what is wrong with it.**

## Personal Information

To improve the usability of JabRef, we need to know who is using it. Therefore, we have prepared some questions regarding you and your use of JabRef.
Of course, all information will be treated CONFIDENTIALLY and will not be given to any third party.

3 von 6 18.12.2018, 22:23



Questionnaire for JabRef Users https://docs.google.com/forms/d/1bbzK7GmvKaIj1s1a6vJaar76QHE5Q...

7. **Please select your occupation!** *
*Mark only one oval.*

- Researcher (permanent position)
- PhD Student/PostDoc
- Student (Bachelor/Master)
- Non-Academic Researcher
- Other: ______________________

8. **What text processing software do you use the most in combination with JabRef?** *
*Mark only one oval.*

- LaTeX/LyX
- Microsoft Word
- OpenOffice/LibreOffice
- Other: ______________________

9. **How many references does your biggest JabRef library have (approximately)?** *

______________________

4 von 6 18.12.2018, 22:23

10. **Select the version of JabRef, that you use at the moment...** *
*Mark only one oval.*

- ◯ 5.0 dev
- ◯ 4.3.1
- ◯ 4.3
- ◯ 4.2
- ◯ 4.1
- ◯ 4.0
- ◯ 3.8.2
- ◯ 3.8.1
- ◯ 3.8
- ◯ 3.7
- ◯ 3.6
- ◯ 3.5
- ◯ 3.4
- ◯ 3.3
- ◯ 3.2
- ◯ 3.1
- ◯ 3.0
- ◯ 2.11.1
- ◯ 2.11
- ◯ 2.10
- ◯ 2.9.2
- ◯ 2.9.1
- ◯ 2.9
- ◯ 2.8.1
- ◯ 2.8
- ◯ 2.7
- ◯ 2.6
- ◯ 2.5
- ◯ 2.4
- ◯ 2.3
- ◯ 2.2
- ◯ 2.1
- ◯ I don't know

11. **Would you like to tell us anything else?**

Powered by Google Forms

# C. Laboratory Study

## C.1. Thinking Aloud Scenarios

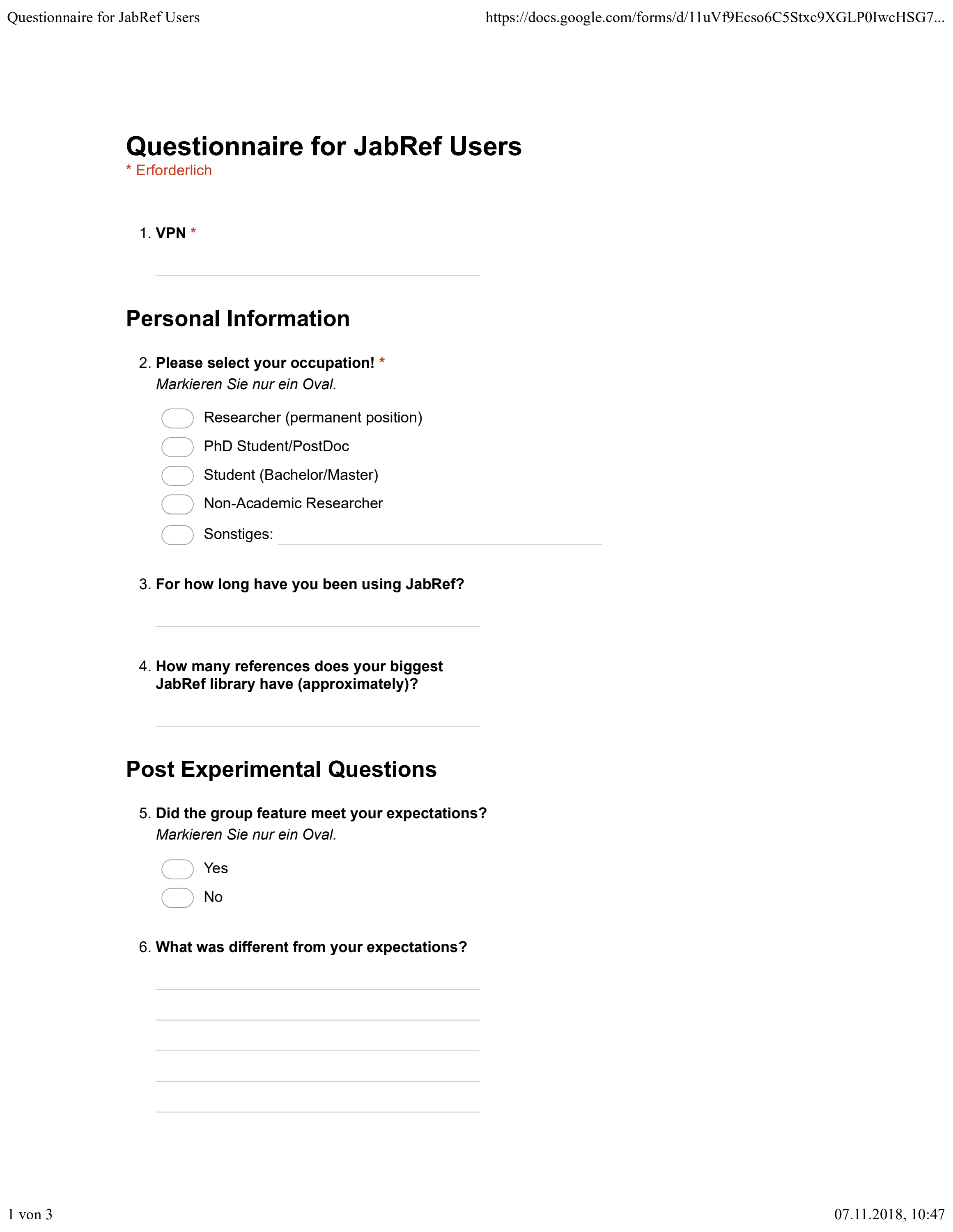
TASK A

1. Add “November” as the month of publishing to Cava1987
2. Search the web for the full text PDF for the entry Munro1992
3. Search for unlinked PDFs in the default directory and import the one named “Import.pdf” using its metadata
4. Change the document type of the new entry to “Article”
5. Search for the DOI of the new entry
6. Update the new entry with the BibTeX data from the DOI
7. Generate a BibTeX key for the new entry

TASK B

1. Create a new group called “last century”
2. Move all entries published before the year 2000 to that group (if not already done)
3. Create another group called “this century”
4. Create two subgroups of “this century” called “before 2010” and “recent articles”
5. Move all entries published between 2000 and 2009 to the subgroup “before 2010”
6. Move all entries published 2010 or later to the subgroup “recent articles”
7. Create a new group consisting of articles with the keyword “survey”
8. Verify if all entries are grouped correctly

## C.2. Interview Outline



Questionnaire for JabRef Users https://docs.google.com/forms/d/11uVf9Ecso6C5Stxc9XGLP0IwcHSG7...

### Questionnaire for JabRef Users

* Erforderlich

1. **VPN** *

#### Personal Information

2. **Please select your occupation!** *
   *Markieren Sie nur ein Oval.*
   - ( ) Researcher (permanent position)
   - ( ) PhD Student/PostDoc
   - ( ) Student (Bachelor/Master)
   - ( ) Non-Academic Researcher
   - ( ) Sonstiges: ____________

3. **For how long have you been using JabRef?**

4. **How many references does your biggest JabRef library have (approximately)?**

#### Post Experimental Questions

5. **Did the group feature meet your expectations?**
   *Markieren Sie nur ein Oval.*
   - ( ) Yes
   - ( ) No

6. **What was different from your expectations?**

1 von 3 07.11.2018, 10:47

7. **What kind of grouping would you prefer?**
   *Markieren Sie nur ein Oval.*

   - Strict Hierarchy (An entry can only be in one group)
   - Tag System (An entry can have multiple tags)
   - JabRef System (An entry can be in multiple groups and subgroups)

8. **Did you know that you could automatically create groups by a keyword?**
   *Markieren Sie nur ein Oval.*

   - Yes
   - No

9. **(IF NO) Would you have used that, to solve the "survey task"?**
   *Markieren Sie nur ein Oval.*

   - Yes
   - No

10. **(IF YES) Why didn't you use it to solve the "survey task"?**
    *Markieren Sie nur ein Oval.*

    - Yes
    - No

11. **Have you used the import through PDF metadata before?**
    *Markieren Sie nur ein Oval.*

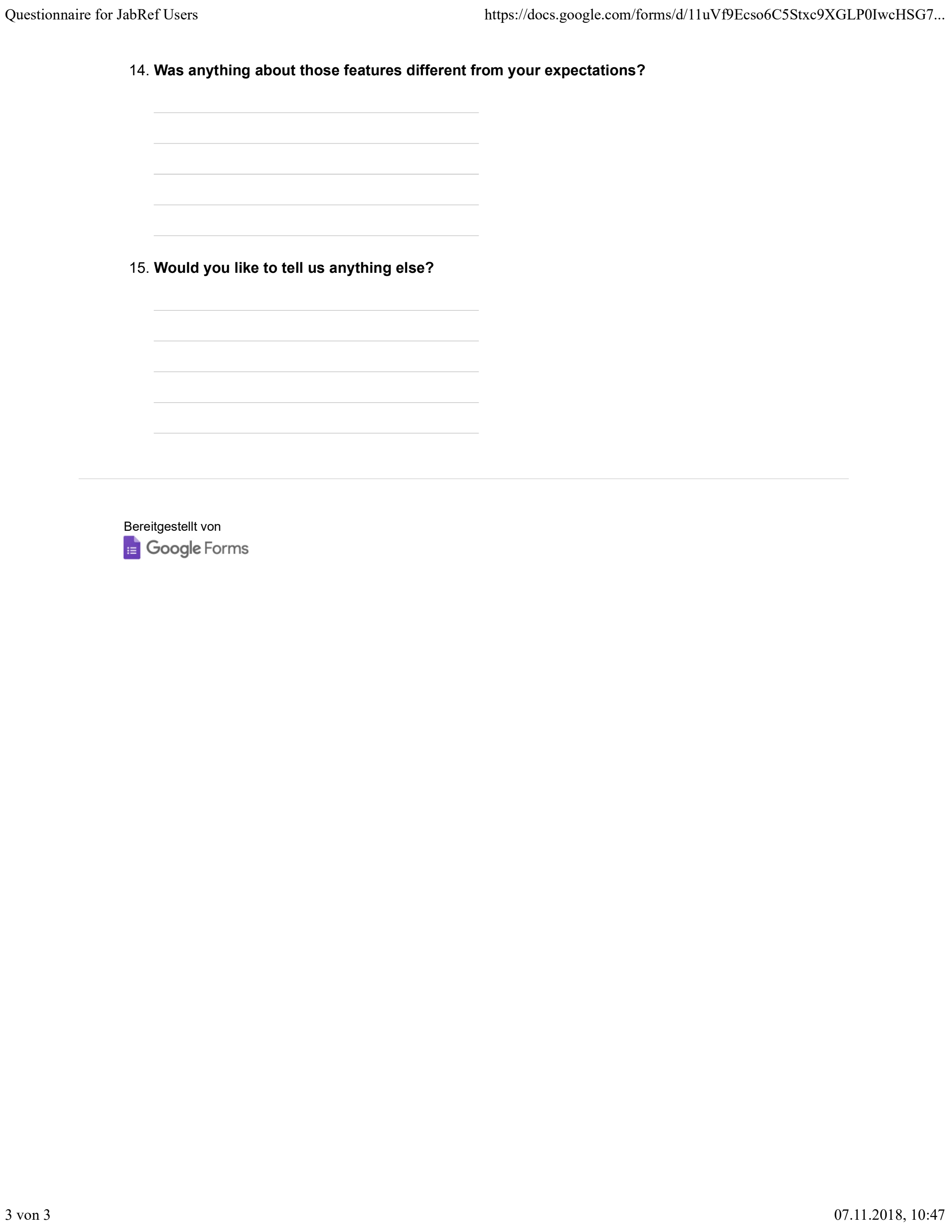
    - Yes
    - No

12. **Have you searched for the DOI of an article before?**
    *Markieren Sie nur ein Oval.*

    - Yes
    - No

13. **Have you updated an article with the information from the DOI before?**
    *Markieren Sie nur ein Oval.*

    - Yes
    - No



Questionnaire for JabRef Users https://docs.google.com/forms/d/11uVf9Ecso6C5Stxc9XGLP0IwcHSG7...

14. **Was anything about those features different from your expectations?**

15. **Would you like to tell us anything else?**

Bereitgestellt von
Google Forms

3 von 3 07.11.2018, 10:47