

Unplagged Developers Manual

Building the Plagiarism Detection Cockpit

Term paper for the master project I

Mentoring Teacher: Prof. Dr. Debora Weber-Wulff

Department Economics II

HTW Berlin – University of Applied Sciences

Elsa Mahari (s0534556) <Elsa.Mahari@gmx.de>

Tien Nguyen (s0512510) <s0512510@htw-berlin.de>

Dominik Horb (s0534217) <dominik.horb@googlemail.com>

Benjamin Oertel (s0522720) <contact@benjaminoertel.com>

Heiko Stammel (s0534218) <heiko.stammel@googlemail.com>

Contents

Introduction	v
Chapter Overview	VI
Conventions	VII
1. A Plagiarism Primer	1
1.1. Plagiarism definition	1
1.1.1. History of plagiarism	3
1.1.2. Progress and social acceptance	5
1.2. Basic Classification of Plagiarisms	7
1.2.1. Copy&paste	7
1.2.2. Copy, shake&paste	7
1.2.3. Patchwriting (rewording)	7
1.2.4. Structural plagiarism	7
1.2.5. Translations	7
1.3. How to detect plagiarism	8
1.3.1. Commercial software systems	9
1.3.2. Free tools and techniques	30
1.4. Vroni Plag	31
1.4.1. Activities	32
1.4.2. Pros vs. Cons	33
2. Project Workflow and Requirements	37
2.1. The Workflow	38
2.1.1. Product Owner — “The Debbie Meetings”	39
2.1.2. Team Meetings	40
2.2. Target Group	41
2.3. User roles	41

2.4. Basic functionalities	42
2.5. Document Parser	42
2.6. Detection Modes	42
2.7. Plugin Architecture	42
2.8. Use Cases	42
3. Developing Unplugged	43
3.1. Development Environment	44
3.1.1. Git	44
3.1.2. Local Deployment	49
3.1.3. Netbeans	51
3.1.4. Additional Software	52
3.1.5. Continuous Integration	52
3.2. User Interface / User Experience	55
3.2.1. Responsive Layout using CSS3 Media Queries	56
3.2.2. Javascript and fallbacks	58
3.3. Frameworks	60
3.3.1. Zend Framework	60
3.3.2. Doctrine	62
3.4. Architectural Goals	63
3.4.1. Progressive Enhancement	63
3.4.2. Test Driven Development	63
A. Minutes	64
B. Logged Time As Of March 11, 2012	65
C. Mockups	80
C.1. Hand-Drawn	80
C.2. Digitalized	85

List of Figures

1.1.	4-stage plagiarism detection process	9
1.2.	Overview of Commercial Detection Systems	9
1.3.	Plagaware Website	11
1.4.	Plagaware overview	13
1.5.	Plagaware side-by-side view	14
1.6.	Turnitin Website	15
1.7.	Turnitin - A lot of little matches can't be found, if the sensibility has not been raised.	16
1.8.	Turnitin - A lot of spam-sites are reported. Not all sufficient to use at work. This is one of the harmless examples.	17
1.9.	Ephorus Website	18
1.10.	Ephorus report - gives a great overview of the results	19
1.11.	Ephorus Problem with umlauts	20
1.12.	Plagscan Website	21
1.13.	Plagscan report is clear and tidy.	22
1.14.	Plagscan reports are not self-explanatory.	23
1.15.	Urkund Website	24
1.16.	Urkund List view	26
1.17.	Urkund Report	27
1.18.	Google Search	30
1.19.	Page numbers	33
1.20.	Barcode analysis	34
1.21.	fragments	35
1.22.	colors	36
2.1.	Redmine Roadmap	38
2.2.	Scrum Meeting	39

2.3. User Stories	40
3.1. Deployment workflow	53
3.2. Mockup – New case – hand-drawn	55
3.3. Mockup – New case – digitalized	56
3.4. Initial Screen PSD	57
C.1. Mockup – Compare results – digitalized	80
C.2. Mockup – Media list – digitalized	81
C.3. Mockup – New case – digitalized	82
C.4. Mockup – New fragment – digitalized	83
C.5. Mockup – New project – digitalized	84
C.6. Mockup – New case – digitalized	85
C.7. Mockup – List fragments – digitalized	86
C.8. Mockup – New fragment – digitalized	87
C.9. Mockup – Show fragment for approval – digitalized	88
C.10. Mockup – New report – digitalized	89

Introduction

After Minister Guttenberg had to resign, because of the plagiarisms found in his doctoral thesis, the big media coverage and interest in plagiarism in Germany has very much subsided (?). However, the initial idea for the creation of the “Unplagged” project, whose development approach will be described here, can be found in this very case of plagiarism. Related to it were the formation of the [GuttenPlag](#) and its descendant [VroniPlag](#). Both are Wiki-based communities that are collaboratively discovering and collecting plagiarism in their respective cases and are kind of the role models for the way the Unplagged system is developed.

The project idea and context were provided by our professor Dr. Debora Weber-Wulff and the two-term master project, every media informatics student at the [HTW-Berlin](#) has to take. Professor Weber-Wulff is a well known expert in Germany on the topic of plagiarism. As she has also done research in this field for over ten years and is actively involved in the VroniPlag community under her synonym “WiseWoman”(?), she came up with the idea to build a dedicated system — a “Plagiarism Detection Cockpit”(?) — that is modeled after the experiences that were made with the workflow used in VroniPlag and GuttenPlag.

So, to put it in a catchy marketing phrase, here is what Unplagged aims to become:

**Unplagged is a simple, web-based, collaborative system to help
discover, collect and document plagiarism in scientific papers.**

To make things a bit more conceivable, we also often refer to it as a mixture of a very specialised text editor, with a focus on comparing texts and marking passages and a modern project management tool like [Redmine](#) or [JIRA](#), to manage the collaborative

aspects of the system. The big distinction we make to other plagiarism software on the market is, that the approach is not to autodetect plagiarism, but focused on aiding the workflow of the users while searching for plagiarized fragments inside a scientific paper, a homework assignment or any other kind of probable textual plagiarism.

This present document will be the handbook that gets you started if you are interested in helping us with the development of this open source project, which is licensed under the [GNU GPLv3](#).

Chapter Overview

One of the biggest problems we faced at the start was, that none of the team members had written a longer scientific text than a bachelors thesis and therefore the experience we got with actual scientific writing was very limited and very specific to the field of computer science. We understand the ethical problems, that come with the betrayal of good scientific practice of plagiarists, but we simply can not relate easily to the amount of work that has to be put into a PhD., or be as passionate about plagiarism as Prof. Weber-Wulff always is, because we never experienced it ourselves.

That is why we had a lot of catching up to do on the most important history behind VroniPlag, the different types of plagiarism, different citation styles and the research Prof. Weber-Wulff and others had already done on systems that try to help finding plagiarism. Chapter 1, [A Plagiarism Primer](#), will give a brief overview of the most important topics to get you up to speed with the domain of the software, if you are not already familiar with it.

Chapter 2, [Project Workflow and Requirements](#), will be the place, where the development process is described and a collection and description of the parts of the system, that already exist or that we identified as necessary parts of Unplagged will be given. As the system is developed with an agile project management style, this will be done primarily based on the product backlog.

If you know all those things already and simply want to get started working and coding,

you should probably jump to [3, Developing Unplugged](#). This chapter will give the technical insights into the system, the basic installation steps and all necessary tools for you as a developer.

And if this still isn't enough reading material for you, the appendices show you..

Conventions

To markup important words in the text, the following typographical conventions are used:

Italic

First used technical terms

Constant Width

Programm code, file names, paths

Bold Constant Width

Variables that have to be changed by the user

1. A Plagiarism Primer

In this chapter we will give a brief overview of the most important topics to get you up to speed with the domain of the software, if you are not already familiar with it.

1.1. Plagiarism definition

What is Plagiarism?

Many people think of plagiarism as copying another's work, or borrowing someone else's original ideas. But terms like "copying" and "borrowing" can disguise the seriousness of the offense: According to the Merriam-Webster Online Dictionary, to "plagiarize" means

to steal and pass off (the ideas or words of another) as one's own to use (another's production) without crediting the source to commit literary theft to present as new and original an idea or product derived from an existing source.

In other words, plagiarism is an act of fraud. It involves both stealing someone else's work and lying about it afterward. But can words and ideas really be stolen?

According to U.S. law, the answer is yes. The expression of original ideas is considered intellectual property, and is protected by copyright laws, just like original inventions. Almost all forms of expression fall under copyright protection as long as they are recorded in some way (such as a book or a computer file). All of the following are considered plagiarism:

turning in someone else's work as your own copying words or ideas from someone else without giving credit failing to put a quotation in quotation marks giving incorrect information about the source of a quotation changing words but copying the sentence structure of a source without giving credit copying so many words or ideas from a source that it makes up the majority of your work, whether you give credit or not (see our section on "fair use" rules)

Most cases of plagiarism can be avoided, however, by citing sources. Simply acknowledging that certain material has been borrowed, and providing your audience with the information necessary to find that source, is usually enough to prevent plagiarism. See our section on citation for more information on how to cite sources properly.

— Plagiarism and collusion in students' assessed work are issues of increasing concern to the academic community as a whole. By plagiarism we mean the submission of part or all of another person's work as if it were one's own, without the knowledge of the author, and with intention to deceive. Collusion, on the other hand is the submission of work as one's own when (at least some of) that work has been done partly or wholly by another person, and that other person is party to the deception. 1 Plagiarism and Collusion Detection using the Smith-Waterman Algorithm

Plagiarism occurs when a student, with intent to deceive or with reckless disregard for proper scholarly procedures, presents any information, ideas or phrasing of another as if they were his/her own and/or does not give appropriate credit to the original source. Proper scholarly procedures require that all quoted material be identified by quotation marks or indentation on the page, and the source of information and ideas, if from another, must be identified and be attributed to that source. Students are responsible for learning proper scholarly procedures.¹

This definition suggests that there are two kinds of plagiarism: one that is committed with the intent to deceive (intentional plagiarism) and one resulting from the disregard for proper scholarly procedures (unintentional plagiarism). 1The Duke Community Standard in Practice: A Guide for Undergraduates 2011-12, Pages 18 and 20. Examples of Intentional Plagiarism

Purchasing a pre-written paper (either by mail or electronically). Letting someone else write part or all of a paper for you. Paying someone else to write part or all of a paper for you. Submitting as your own someone else's unpublished work (including a computer program or algorithm), either with or without permission. Submitting as your own, work done jointly by a group in which you may have participated. Submitting work done by you, but for another class or another purpose without documenting that it was previously used. Creating phony citations.

Examples of Unintentional Plagiarism:

Failure to cite a source that is not common knowledge. Failure to "quote" or block quote author's exact words, even if documented. Failure to put a paraphrase in your own words, even if documented. Failure to put a summary in your own words, even if documented. Failure to be loyal to a source.

- **Copy&paste**
- **Copy, shake&paste**
- **Patchwriting (rewriting)**
- **Structural plagiarism**
- **Translations**

Kurze Definition und Begriffserklärung

1.1.1. History of plagiarism

Geschichtlicher Hintergrund: von den Roemern zu Wild usw.

Twentieth-century dictionaries define plagiarism as "wrongful appropriation," "close imitation," or "purloining and publication," of another author's "language, thoughts,

ideas, or expressions,” and the representation of them as one’s own original work,[1][2] but the notion remains problematic with nebulous boundaries.[3][4][5][6] There is no rigorous and precise distinction between imitation, stylistic plagiarism, copy, replica and forgery.[3][4][5][6]

In the 1st century, the use of the Latin word *plagiarius* (literally kidnapper), to denote someone stealing someone else’s work, was pioneered by Roman poet Martial, who complained that another poet had ”kidnapped his verses.” This use of the word was introduced into English in 1601 by dramatist Ben Jonson, to describe as a *plagiary* someone guilty of literary theft.[7][18]

The derived form plagiarism was introduced into English around 1620.[19] The Latin *plagi?rius*, ”kidnapper”, and *plagium*, ”kidnapping”, has the root *plaga* (”snare”, ”net”), based on the Indo-European root *-plak, ”to weave” (seen for instance in Greek *plekein*, Bulgarian ”?????” *pleta*, Latin *plectere*, all meaning ”to weave”).

The modern concept of plagiarism as immoral and originality as an ideal, emerged in Europe only in the 18th century, particularly with the Romantic movement.[7][11][12] Romantic aesthetic and ideology, still retains a significant strength in the 20th century, and encourages attacks against all that violates its values of genius, originality and individuality.[20] From the Romantic perspective, artistic techniques like parody are considered parasitic.[20] For centuries before, not only literature was considered ”*publica materies*,” a common property from which anybody could borrow at will, but the encouragement for authors and artists was actually to ”copy the masters as closely as possible,” for which the closer the copy the finer was considered the work.[7][8][13][21][22] This was the same in literature, music, painting and sculpture. In some cases, for a writer to invent their own plots was reproached as presumptuous.[7] This stood at the time of Shakespeare too, when it was common to appreciate more the similarity with an admired classical work, and the ideal was to avoid ”unnecessary invention.”[7][9][10]

The modern ideals for originality and against plagiarism appeared in the 18th century, in the context of the economic and political history of the book trade, which will be exemplary and influential for the subsequent broader introduction of capitalism.[23] Originality, that traditionally had been deemed as impossible, was turned into an obligation by the emerging ideology of individualism.[10][13] In 1755 the word made it into Johnson’s

influential A Dictionary of the English Language, where he was cited in the entry for copier ("One that imitates; a plagiary; an imitator. Without invention a painter is but a copier, and a poet but a plagiary of others."), and in its own entry denoting both A thief in literature ("one who steals the thoughts or writings of another") and The crime of literary theft.[7][24]

Later in the 18th century, the Romantic movement completed the transformation of the previous ideas about literature, developing the Romantic myth of artistic inspiration, which believes in the "individualised, inimitable act of literary creation", in the ideology of the "creation from nothingness" of a text which is an "autonomous object produced by an individual genious." [5][12][22][25][26][27][28] Plagiarism has often been used as a derogatory term for parodies.[29][30]

Despite the 18th century new morals, and their current enforcement in the ethical codes of academia and journalism, the arts, by contrast, not only have resisted in their long-established tradition of copying as a fundamental practice of the creative process,[12][13][14][31] but with the boom of the modernist and postmodern movements, this practice has been accelerated, spread, increased, dramatically amplified to an unprecedented degree, to the point that has been heightened as the central and representative artistic device of these movements.[15][16][12] Plagiarism remains tolerated by 21st century artists.[13][14] An early rebuttal to Romantic aesthetic in this respect, came from Russian formalism.[20]

1.1.2. Progress and social acceptance

(1 Seite) Entwicklung und gesellschaftliche Akzeptanz Anfang mit Gutenberg / Vroni Steuber usw.

Gesellschaftliche Probleme Universitäre Probleme

vroni-plag – VroniPlag ist ein Wiki, d.h. jeder kann etwas beitragen. Und jeder, der etwas beitragt, tut dies aus ganz persönlichen Motiven. Es ist deshalb schwierig eine allgemeingültige Antwort zu geben - trotzdem scheint bei vielen regelmässigen

VroniPlag Autoren eher die Sorge um die Wissenschaft im Vordergrund zu stehen und nicht parteipolitische Ueberlegungen. Siehe auch Frage zum Thema Motivation.

1.2. Basic Classification of Plagiarisms

1.2.1. Copy&paste

1.2.2. Copy, shake&paste

1.2.3. Patchwriting (rewriting)

1.2.4. Structural plagiarism

1.2.5. Translations

1.3. How to detect plagiarism

Plagiarism detection means a lot of effort and hard work, because a lot of documents, books and papers must be found, scanned and compared line for line. Often it is really complicated to find available sources in libraries or scientific database systems and plenty of pages must be copied by hand. During the last ten years a lot of software companies came into market to automatize this process and developed algorithms to find plagiarisms automatically. These systems are called "Plagiarism detection systems (PDS) ". There are two different approaches for plagiarism detection systems to identify plagiarism in text documents:

1. Corpus based analysis

means to "compare suspicious documents against a set of potential original documents" PAN'07 to find similar text passages.

2. Intrinsic analysis

"identifies potentially plagiarized passages by analyzing the suspicious document with respect to changes in writing style". PAN'07

These computer assisted detection systems alone are not appropriate to find all plagiarisms without human judgement.

We found out that the most practical way is to combine both approaches - the first step is to use the computer-based detection to find similarities between the suspicious documents and the original papers. The second step is to examine the results, validate them and continue the search in a deeper level. (see figure 4 stage)

(Culwin, Fintan; Lancaster, Thomas (2001), "Plagiarism issues for higher education", Vine 31 (2): 36-41, doi:10.1108/03055720010804005)

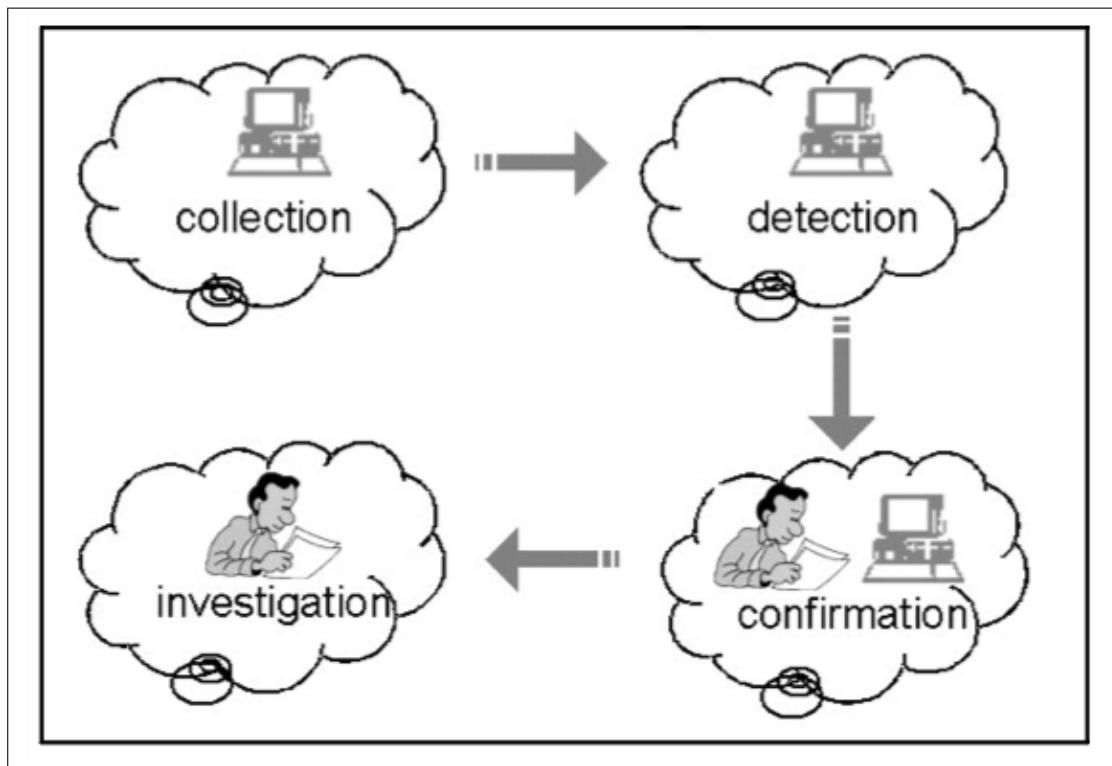


Figure 1.1.: 4-stage plagiarism detection process

1.3.1. Commercial software systems

Name	No free access	Free version available	Free demo available	Website
Plagaware		X		www.plagaware.de
Turnitin			X	www.turnitin.com
Ephorus			X	www.ephorus.com
Plagscan	X			www.plagscan.com
Urkund			X	www.urkund.com

Figure 1.2.: Overview of Commercial Detection Systems

Several computer companies developed commercial software systems to facilitate the detection of plagiarism. They offer different terms of pricing and online as a service or offline programs. The software system compare digital content from the internet or different types of databases. They seek for similarities, report suspicious parts and try to answer the question if the present text is a plagiarism or not.

This chapter covers parts of the results of the big "Plagiarism Detection System Test 2010". The following benchmarking was done by Prof. Debora Weber-Wulff at the University of Applied Sciences HTW Berlin and her plagiarism team in 2010. The entire benchmarking includes 26 of the 47 available systems on the market and gives an overview of the strengths and weaknesses of these systems in finding plagiarism.

We used the top 5 "partially useful" software systems in this benchmarking to overview the market and find the best usable features for our own system to simplify the daily work of plagiarism finders.

PlagAware

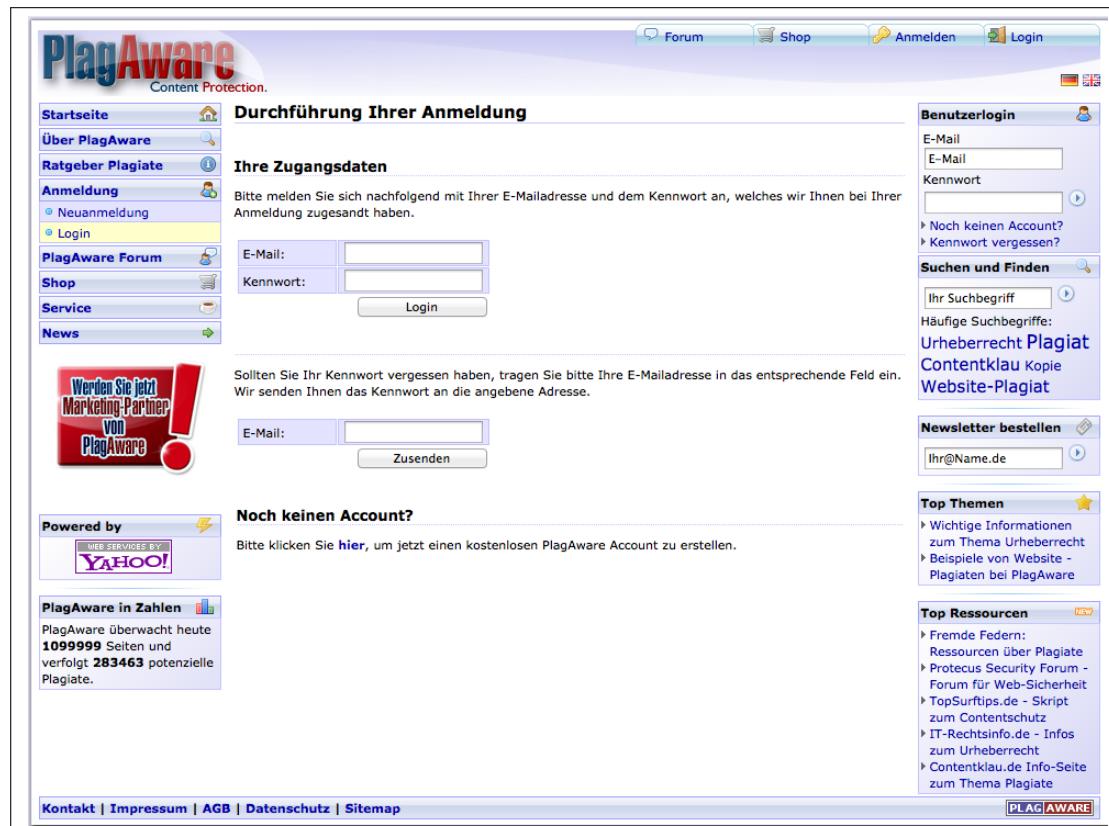


Figure 1.3.: Plagaware Website

Plagaware Business Promotion

"PlagAware is an online-service, which offers services around the topics Searching, finding, analysing and tracing of plagiarisms. The central element of PlagAware is a search engine, which is specialised in detecting identical contents of given texts. Contrary to the plagiarism scanning with classical search engines, the places of finding are not directly transferred to the user, but analysed on the rate and the type of analogy, before a message to the user is written. By this the differing result reports of PlagAware allow to recognise very fast the percentage and the distribution of the copied text contents, thus permitting an efficient and secure rating of a possible plagiarism."

“PlagAware Business Promotion”(?) [PlagAware](#)

PlagAware is software company from Ulm in Germany. The website is online for nearly 5 years and is the top-ranked system in the HTW ”Plagiarism Detection System Test” 2010, but in fact it still detect only 61,11% of the plagiarism cases. Although PlagAware ”produces excellent documentation of the plagiarism found, highlighting the commonalities in a side-by-side presentation. However, its usefulness at university is limited, as each file must be uploaded individually - no ZIP file or student-submission is possible. The system was not designed to be used in a university setting, but rather to find plagiarisms of online texts, which is important for sites trying to optimize their search machine ranking, as plagiarism will contribute to downranking.” [Plagiat Website](#)

Figure Overview shows an overview of all fragments and the results of the plagiarism detection. In figure Side-by-side shows a nice side-by-side fragment view, where all found plagiarisms are shown with different colors. Anomalies in the text are highlighted and the barcode-view is available.

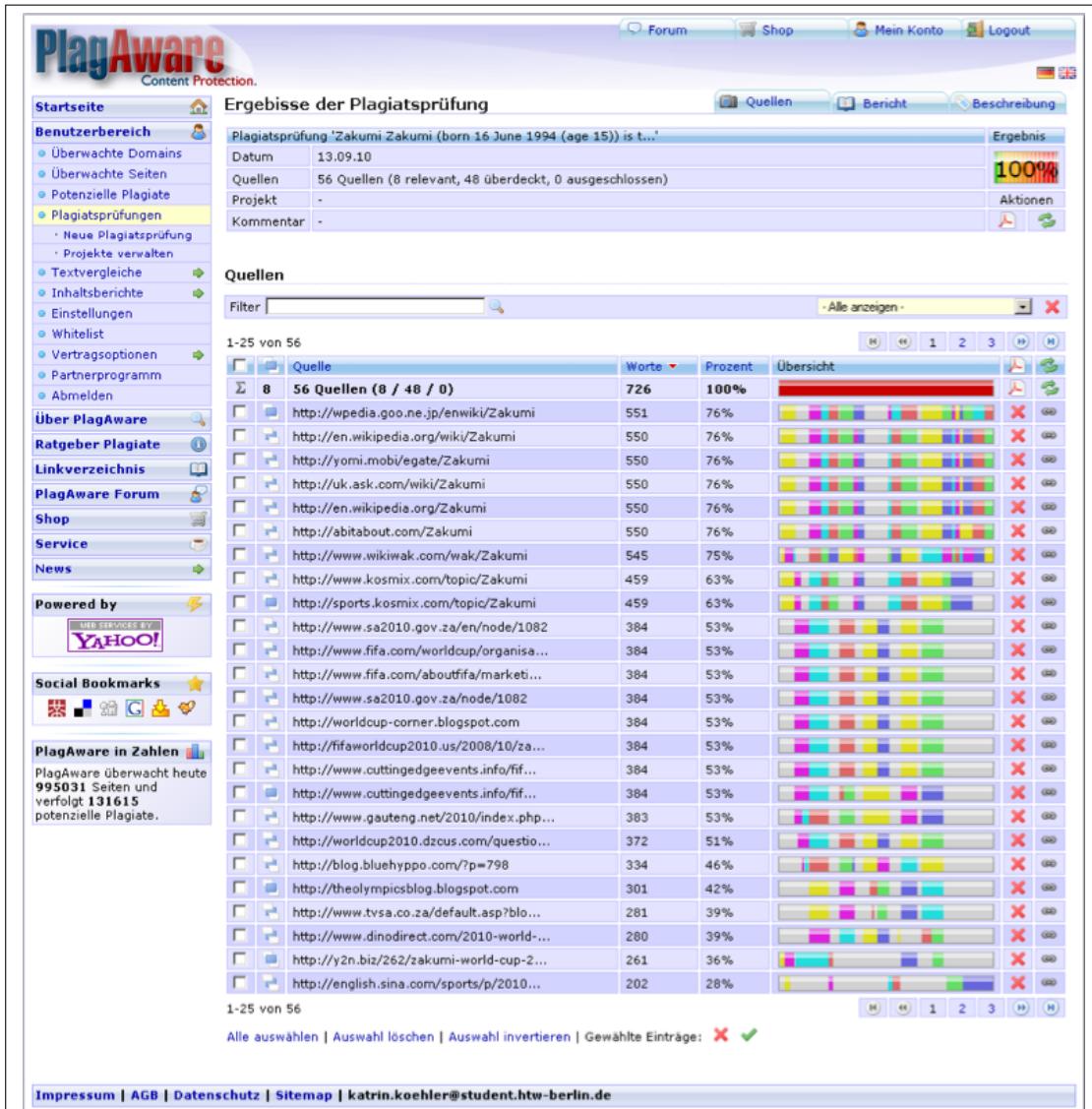


Figure 1.4.: Plagaware overview

PlagAware Costs

PlagAware has four different payment-models:

1. Free

30 scans/month for free. Every additional scan costs 3,0 ct.

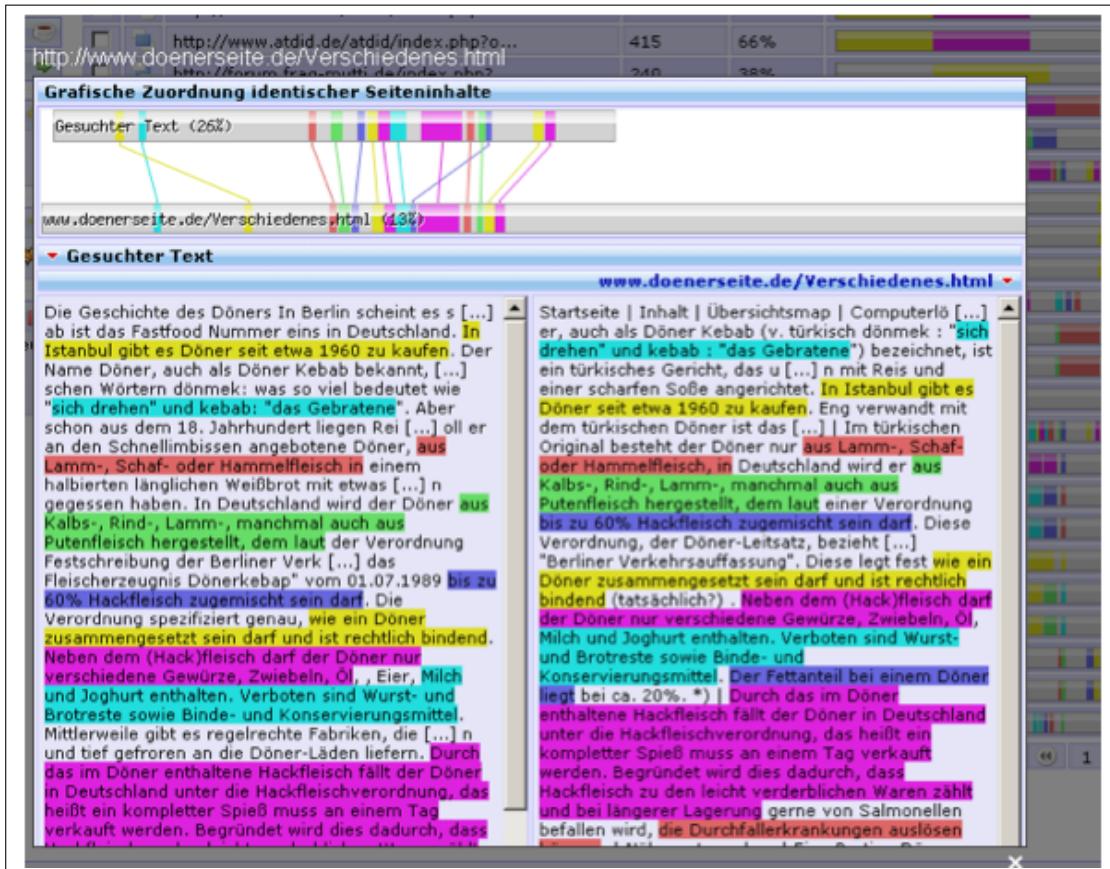


Figure 1.5.: Plagaware side-by-side view

2. Light

EUR 2,99/month. 150 scans/month included. Every additional scan costs 2,0 ct.

Mininum term of 6 month.

3. Standard

EUR 7,49/month. 500 scans/month included. Every additional scan costs 1,5 ct.

Mininum term of 6 month.

4. Premium

EUR 14,99/month. 1500 scans/month included. Every additional scan costs 1ct.

Mininum term of 6 month.

Turnitin

The screenshot shows the Turnitin homepage. At the top, there is a navigation bar with links for 'Startseite', 'Produkte', 'Kunden', 'Partner', 'Support und Schulung', 'Über uns', and a search bar. On the right side of the header are fields for 'E-mail' and 'Passwort' with buttons for 'ANMELDEN', 'Account erstellen', and 'Passwort zurücksetzen'. Below the header, there is a language selector set to 'Deutsch' and a 'suchen...' search input field.

The main content area features a large banner with the text 'Die umfassende Lösung zur Evaluierung von Texten' (The comprehensive solution for evaluating texts). It lists three benefits: '✓ Originalität sicherstellen', '✓ Arbeiten benoten', and '✓ Peer-Reviews erleichtern'. Below this, there are two red buttons: 'ERFAHREN SIE MEHR' and 'FORDERN SIE EIN ANGEBOT AN'. To the right of the banner, there is another section titled 'Plagiatsprävention mit Turnitin' (Plagiarism prevention with Turnitin) for 'Pädagogische Hochschule Zürich' on March 28, 2015, from 12.15 - 17.00 Uhr, with a red 'ANMELDEN' button.

Below the banner, there is a testimonial in German: "Ich habe Stunden damit zugebracht, mithilfe von Google nach ungewöhnlichen Formulierungen zu suchen, wenn ich vermutete habe, dass ein Student eine Arbeit nicht selbst verfasst hat. Jetzt suche ich einfach schnell mit Turnitin!" followed by a quote from 'LEHRKRÄFT'.

On the right side, there is a section titled 'Turnitin – Resultate auf einen Blick' (Turnitin – Results at a glance) with a list of statistics:

- über 200 Millionen archivierte Arbeiten
- über 90.000 Magazine und Bücher
- über 1 Million aktive Lehrkräfte
- über 17 Milliarden gecrawlte Webseiten
- 10.000 Bildungseinrichtungen
- über 20 Millionen lizenzierte Studenten
- 126 Länder

Below this, there is a link 'ENTDECKEN SIE, WAS EINE HALBE MILLION LEHRKRÄFTE SCHON WISSEN' (Discover what half a million teachers already know) and a note about a recent statistical analysis showing reduced plagiarism over time, with a link to download the PDF.

At the bottom of the page, there are links for 'Startseite', 'Datenschutz', 'Nutzungsrichtlinien', 'Kontakt', and copyright information: 'Copyright © 1998 - 2012 iParadigms, LLC. Alle Rechte vorbehalten.' and a small logo.

Figure 1.6.: Turnitin Website

Business Promotion

"Our award-winning solution discourages plagiarism and facilitates rich, meaningful feedback that improves writing skills, promotes critical thinking, and streamlines grading."

"Turnitin"(?)[Turnitin](#)

Turnitin is a product by a company called iParadigms. It is a well-known US plagiarism software system and one of the most used plagiarism detection systems in the education sector. The website is online for nearly 13 years and the system is at the second position in the HTW "Plagiarism Detection System Test" 2010.

Die besten Ergebnisse erhaelt man durch Material, das bereits in deren Datenbank vorhanden ist. Plagiatsauffälligkeiten werden auf dem Text farblich markiert und die Hauptquelle wird angezeigt. Was nicht so gut ist, ist das auch viele winzige Uebereinstimmungen gefunden werden, wie in einem englischen Text das englische Wort "of".

In the past they had a lot of problems to deal with umlauts, and having a complex setup. They improved a lot of parts especially the german translation. Still a big problem for european countries is the copyright policy of Turnitin. They still storing copies of user material in their database without a permission. [Plagiat Website](#)

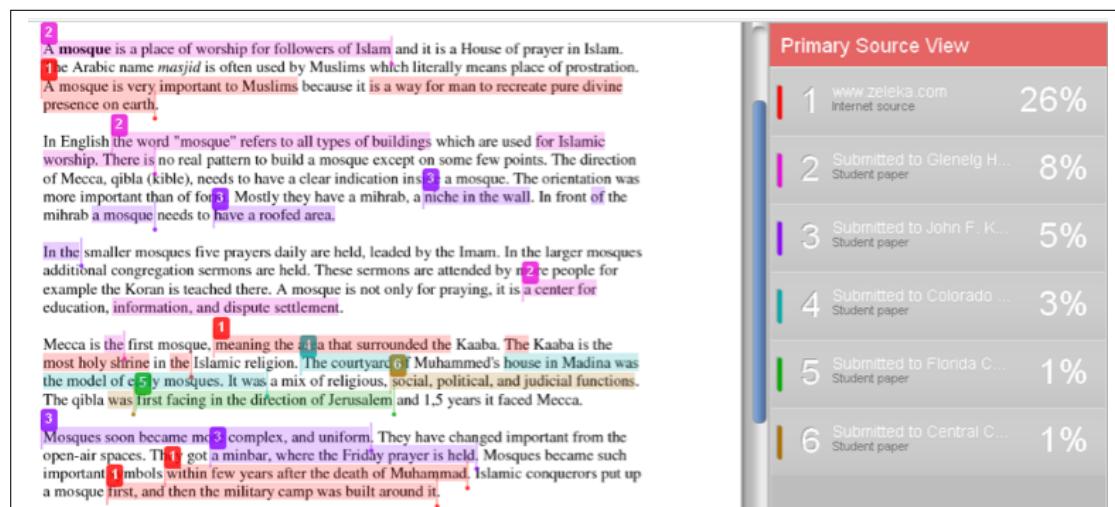


Figure 1.7.: Turnitin - A lot of little matches can't be found, if the sensibility has not been raised.

In 2008 the system was placed at the 13th position. The reason of this change is that the other systems have gotten worse. Turnitin has still problems of flagging spam sites especially when these sites are not safe for work (e.g. site with pornography content). (figure porn) On the other hand the search algorithm of Turnitin is storing sites in their database although they are still not exist. [Plagiat Website](#)

Costs

There are different license models for the education sector and the cost depends on the amount of users .



Figure 1.8.: Turnitin - A lot of spam-sites are reported. Not all sufficient to use at work.
This is one of the harmless examples.

Ephorus



The screenshot shows the homepage of the Ephorus website. At the top left is the Ephorus logo, which consists of the word "ephorus" in orange lowercase letters, where each letter is composed of a grid of smaller, semi-transparent letters. To the right of the logo is a collage of three images: a spiral notebook, a person's hand writing on a document, and a pile of papers. Below the header is a navigation bar with links for "Start", "Demo", "Produkte", "Referenzen", "Support", and "Kontakt". A language dropdown menu is set to "Deutsch". On the left side, there is a sidebar with a "Start" button and two links: "Für wen?" and "Über uns". The main content area features a heading "Ephorus - Marktführer in Europa" and a paragraph of text. The text discusses preventing plagiarism without extra effort, mentioning that Ephorus is a market leader in anti-plagiarism and offers best results at low prices. It also invites users to try a free demonstration. At the bottom of the page is a "Login" button with a lock icon. The footer contains a copyright notice: "© Ephorus 2012 | Privacy statement | Disclaimer".

Figure 1.9.: Ephorus Website

Business Promotion

"Never search for plagiarism yourself again? An end to all irritations and qualitatively better papers? No problem. With Ephorus, you can prevent plagiarism with no extra effort. Moreover, with this anti-plagiarism market leader, you will be assured of the best service and the lowest prices. With Ephorus, teaching will be fun again! Would you like to try out Ephorus?

"Ephorus Business Promotion"(?) Ephorus

The third position in the HTW "Plagiarism Detection System Test" 2010 is Euphorus. It's a plagiarism detection system from the netherlands and the website is online for nearly 8 years. 2007 it took the first place in the test, 2008 it was only position 8. Now they redesigned and reorganized the system and old problems were solved. The usability of reports and the whole handling of the system very good. (figure: report) But their still problems with umlauts and the european copyright problematic like in Turnitin. (figure: umlauts)

Eingereichtes Dokument - Download

68% 32-zakumi.pdf - Umbenennen 02-09-2010 | 15:08

Quellenliste

68% Gesamte Übereinstimmungen

<input type="checkbox"/> 51%	<input checked="" type="radio"/> http://worldcup-corner.blogspot.com/
<input type="checkbox"/> 51%	<input type="radio"/> http://www.cbn.co.za/pressoffice/2010/fullstory/1028.htm
<input type="checkbox"/> 51%	<input type="radio"/> http://www.sa2010.gov.za/node/1082
<input type="checkbox"/> 37%	<input type="radio"/> http://www.tvsouthafrica.net/default.asp?blogname=shugesblogiwood&articleid=9369
<input type="checkbox"/> 22%	<input type="radio"/> http://www.southafrica.info/2010/zakumi.htm
<input type="checkbox"/> 16%	<input type="radio"/> http://en.wikipedia.org/wiki/Zakumi

Bericht

Zakumi
Zakumi (born 16 June 1994 (age 15)) is the Official Mascot for the 2010 FIFA World Cup. He is a cheerful and sporty leopard with green hair, presented on 22 September 2008. His name comes from "ZA", the ISO 3166-1 alpha-2 code for South Africa, and "kumi", a word that means ten in several African languages.

Eingereicht:
One thing is for sure, Zakumi will be first on the dancefloor and last off it at the biggest party in the world - the 2010 FIFA World Cup South Africa™. He wants to dance and entertain as many people as he can. He is an animator for fans, players and officials, for schoolchildren, teenagers and big kids alike!

Gefunden:
One thing is for sure, Zakumi will be first on the dancefloor and last off it at the biggest party in the world - the 2010 FIFA World Cup South Africa™. He wants to dance and entertain as many people as he can. He is an animator for fans, players and officials, for schoolchildren, teenagers and big kids alike!

Eingereicht:
Zakumi is a jolly, self-confident, adventurous, spontaneous, and actually quite shrewd little fellow. He loves to perform and always follows his instinct and intuition, yet sometimes has the tendency to exasperate a bit. You will often find him fooling about and teasing

Gefunden:
Zakumi is a jolly, self-confident, adventurous, spontaneous, and actually quite shrewd little fellow. He loves to perform and always follows his instinct and intuition, yet sometimes has the tendency to exasperate a bit. You will often find him fooling about and teasing

Figure 1.10.: Ephorus report - gives a great overview of the results

Ephorus costs

Not stated.

Buchbesprechung (Originaltitel: „The Handmaid's Tale“) In einem fiktiven Staat im Jahr 2195 in Nordamerika haben religiöse Fundamentalisten die totalitäre Republik Gilead errichtet. Nach einer atomaren Verseuchung ist ein großer Teil der weiblichen Bevölkerung unfruchtbar. Frauen werden entmündigt und in drei Gruppen eingeteilt: Ehefrauen von Führungskräften, Dienerinnen und Mägde. Letztere werden zur Fortpflanzung rekrutiert und sollen nach biblischem Vorbild für unfruchtbare Ehefrauen Kinder empfangen. Können sie ihre Aufgabe als Gebärmaschine nicht erfüllen, werden sie in entfernte Kolonien zu gefährlichen Arbeiten wie Giftmüllentsorgung abgeschoben. Desfred, Hauptfigur und Erzählerin, lebt in einer Welt, in der sie nur geduldet wird, weil sie gesund ist und gebären kann. Sonst wäre sie für ihren Widerstand gegen die herrschende Diktatur schon lange erhangt worden. Ihr Glück ist, das gebärfähige Frauen Mangelware sind: die Umwelt ist so verseucht, daß die meisten Menschen unfruchtbar sind. Darum wird sie "Magd" bei einem altilchen Offiziersehepaar, und ihre einzige Aufgabe ist, bei den rituellen Befruchtungsakten stillzuhalten. Und bald schwanger zu werden, aber das ist nicht so einfach. Desfred wird dem Kommandanten Fred als Zweitfrau in dessen Haushalt zugewiesen, wo regelmäßig die entwürdigende Prozedur des Geschlechtsakts in Gegenwart der Ehefrau durchgeführt wird. Ihr Zimmer darf Desfred nur zu seltenen Einkäufen und zu öffentlichen Hinrichtungen verlassen. Sie hat sich dem totalitären Regime unterworfen, dem sie nur durch Zufall entfliehen kann. Desfreds Tonbandaufzeichnungen schildern in einfachen Worten den Alltag in einer entmenschlichten Gesellschaft mit vollständiger Überwachung und grausamer Unterdrückung. Unterbrochen werden die Beschreibungen von Desfreds Erinnerungen an die alte Zeit: an Mann und Kind, aber auch an die Errichtung des Überwachungsstaats. Wie es dazu kam, ist eigentlich noch interessanter: nach und nach wurden die Rechte der Frauen beschnitten. Erst funktionierten ihre Kreditkarten nicht mehr, dann durften sie nicht mehr arbeiten, usw. Die Parallelen zu Nazideutschland ist subtil, aber vorhanden: auch den Juden beschritt man nach und nach ihre Freiheiten, so langsam, daß jeder Schritt als gerade noch erträglich schien. Eindringlich entsteht das Bild der totalen Repression, das umso bedrückender auf die Leser wirkt, als es eine geringe Distanz zur Gegenwart aufweist: mit alltaglichen Requisiten wie Computer und Kreditkarte, die hier in den Dienst der Republik Gilead gestellt werden. Trotz offenkundiger Parallelen zu 1984 von George R. Orwell ist „Der Report der Magd“ aber weniger als Utopie denn als Stellungnahme zu aktuellen politischen Strukturen und Diktaturen zu verstehen.

Figure 1.11.: Ephorus Problem with umlauts

PlagScan

Business Promotion

PlagScan stands for professionalism

- All documents are treated 100% confidential
- You control whether your document is checked against others, or not
- Integration via API in your existing CMS or learning management system possible

Plagiarism check as easy as pie: PlagScan

- Annotations directly in the document, check without additional work
- No installation - complete functionality in every browser
- All popular formats can be processed

Save time with PlagScan

Figure 1.12.: Plagscan Website

- Check several documents in parallel.
- Fully automated document analysis.
- No use of your resources, all computation is carried out on our servers.

“Plagscan Business Promotion”(?) Plagscan

Plagscan is a software company from Mainz, Germany. It placed at position number 4 in the HTW "Plagiarism Detection System Test" 2010. The website is online for 3 years and in the preview check 2008 it came to the 10th position. As a user you have to buy "Plag Points" (PP). One test costs 1 PP per 100 words. The administrator sets up users and assigns them points for use. There are three different kinds of reports - a list of possible sources with links to click on, the submitted document with the suspicious areas linked to a possible source, and a docx file with the sources in comments. There's no side-by-side presentation, so it's not possible to compare the fragments. Although there are still problems, PlagScan was first place in usability, but only 8th place in overall effectiveness with only 60% of the points awarded for finding plagiarisms.

The screenshot shows the Plagscan dashboard. On the left, a sidebar includes a greeting ("Hallo Bernhard Blütentraum!"), user information ("Institution: Test HTW Berlin", "Ihr Guthaben: 1913"), and links for "Texteingabe", "Hilfe anzeigen", "Admin fragen", and "Einstellungen". A "Plagiat Level" section uses a color-coded legend: green for 0-1%, yellow for 1-5%, and red for 5-100%. The main area displays a table of 10 analyzed documents, each with a checkbox, word count, plagiarism percentage (in green, yellow, or red), and a link to the report. The table has columns for "Alle auswählen", "Alle abwählen", "Ausgewählte Dokumente", "Analysieren", "Löschen", "Inhalt", and "Bericht". At the bottom of the page, there are links for "Copyright © PlagScan.com 2010", "Kontakt & Impressum", and "AGB & Datenschutz".

		Ausgewählte Dokumente	Analysieren	Löschen	Inhalt	Bericht	
<input type="checkbox"/>	18_vikinger.pdf - Die Vinland-Karte, die isla*ndische	879 Worte	Inhalt	Bericht>>	2010-08-03 10:36:58		
<input type="checkbox"/>	17_squaredance.pdf - Square Dance Ein Square D:	531 Worte	Inhalt	7.2% Bericht>>	2010-08-03 10:36:56		
<input type="checkbox"/>	16_jelenik.pdf - Elfriede Jelinkel ist am 20.10.1946 in	468 Worte	Inhalt	50.2% Bericht>>	2010-08-03 10:36:55		
<input type="checkbox"/>	15_beduerfnisanstalt.pdf - Bedu*fnisanstalten, im w	721 Worte	Inhalt	30.9% Bericht>>	2010-08-03 10:36:53		
<input type="checkbox"/>	14_schmeling.pdf - Max Schmeling war ein deutsche	706 Worte	Inhalt	0% Bericht>>	2010-08-03 10:36:50		
<input type="checkbox"/>	13_piment.pdf - Viele von den uns heute bekannten	494 Worte	Inhalt	0% Bericht>>	2010-08-03 10:36:48		
<input type="checkbox"/>	12_mikrobrauereien.pdf - Eine Mikrobrauerei, oder i	720 Worte	Inhalt	0% Bericht>>	2010-08-03 10:36:46		
<input type="checkbox"/>	11_mankell.pdf - Vor dem Einsetzen des Frostes ist	670 Worte	Inhalt	57.2% Bericht>>	2010-08-03 10:36:44		
<input type="checkbox"/>	10_fraktur.pdf - Die Fraktur wird oft als "alte Schrift"	1131 Worte	Inhalt	5.1% Bericht>>	2010-08-03 10:36:42		
<input type="checkbox"/>	19_blogs.pdf - Der Begriff Blog geht zurück auf der	1062 Worte	Inhalt	1.2% Bericht>>	2010-08-03 10:36:40		

Figure 1.13.: Plagscan report is clear and tidy.

Plagscan costs

PlagScan has four different payment-models without a contract:

1. 9 Euro

500 Plagpoint - 50.000 words - 200 Sites.

Dateiname: 21.doc Datum: 2010-08-03 12:14 Ergebnisse der Plagiarismus Analyse vom 2010-08-03 12:16:42	
127 Treffer von 43 Quellen, von denen 42 Onlinequellen sind. Plagiat Level: 	
Im Dokument anzeigen Docx-Dokument herunterladen	
Zeige besten Treffer pro Quelle - klicken Sie ihn an um alle Treffer anzuzeigen oder auf die URL für die Quelle selbst.	
29 Treffer von ein PlagScan Dokument datiert vom 2010-08-03 10:40	
<p><input type="checkbox"/> Wenige Tage nach den ersten Protesten begannen gewaltsame Ausschreitungen vorwiegend ... Tibeter in Lhasa dann auch in anderen Teilen des Landes die sich sowohl gegen chinesische Zivilisten als auch gegen die staatlichen ... und deren Einrichtungen richteten</p>	
16 Treffer von http://de.wikipedia.org/wiki/Tibetische_Uhrufen_2008	
<p><input type="checkbox"/> Wenige Tage nach den ersten Protesten begannen gewaltsame Ausschreitungen vorwiegend ... Tibeter in Lhasa dann auch in anderen Teilen des Landes die sich sowohl gegen chinesische Zivilisten als auch gegen die staatlichen ... und deren Einrichtungen richteten</p>	
14 Treffer von http://forum.gaming-universe.de/index.php?showtopic=14315&mode=threaded	
<p><input type="checkbox"/> März meldete zu diesem Vorfall die amtliche Nachrichtenagentur Xinhua unter Berufung auf Polizeiquellen beim Einsatz seien vier Demonstranten durch Schüsse verletzt worden</p>	
7 Treffer von http://www.mogelpower.de/forum/thread.php?thread_id=233840	
<p><input type="checkbox"/> Ausländische Webseiten werden in China in der Regel umgehend geblockt wenn sie regierungskritische Inhalte verbreiten</p>	
6 Treffer von http://www.computerwoche.de/nachrichtenarchiv/1858744	
<p><input type="checkbox"/> Wie der heimische Anbieter Baidu unterwerfen sich auch ausländische Unternehmen wie Google einer umstrittenen Selbstzensur in dem sie eigenhändig systemkritische Inhalte aus den Suchergebnissen herausfiltern um in China weiterhin Geschäft machen zu können</p>	
5 Treffer von http://www.adhs-anderswelt.de/index.php?topic=21220.20	

Figure 1.14.: Plagscan reports are not self-explanatory.

2. 19 Euro

1.250 Plagpoints - 125.000 words - 500 Sites.

3. 29 Euro

2.000 Plagpoints - 200.000 words - 800 Sites. .

4. 69 Euro

5.000 Plagpoints - 500.000 words - 2.000 Sites.

Urkund

The diagram illustrates the Urkund system architecture. It shows a central 'Transport' component with 'XML' and 'JSON' options. To the left is a 'Browser' box labeled 'Client Objects'. To the right is a 'Server' box labeled '.Net Objects'. Arrows indicate the flow of data between them, labeled 'Serialize' and 'Deserialize'. The 'Transport' box also has arrows pointing towards both the 'Browser' and 'Server' boxes.

URKUND

HOME | ABOUT URKUND | SERVICES | SUPPORT | STUDENTS | LOGIN | REGION

DEMO

Contact information

Continental Europe

URKUND France
Tel: +33 970 447 884
Fax: +33 1 73 72 91 75
savoirplus@urkund.fr

URKUND España
Tel: +34 902 001 288
Fax: +34 902 001 289
conocer@urkund.es

URKUND Nederland
Tel: +31 (0)10 3 400 666
info@urkund.eu
www.urkund.eu

URKUND Belgium
Tel: +32 (0)2 6118715
info@urkund.be
www.urkund.be

URKUND Deutschland
Tel: +49 69 59 603 605
wissen@urkund.de

URKUND Türkiye
Etkin Proje Yönetim Dan., Tic. Ltd. Şti.
info@urkund.com.tr
Tel: +90 (0)216 325 83 89
Fax: +90 (0)216 340 19 57

URKUND Polska
Tel: +48 510 099 134
support@urkund.pl
www.urkund.pl

Central and South America

Grupo Difusión Científica
Av. Emiliano Zapata No. 285
(Eje 7A Sur)
Tel: +55 5090 2800
+55 5090 5300

Scandinavia / Others

PrioInfo (HQ)
Tel: +46 8 7385200
info@urkund.se
Box 3217
10364 Stockholm, Sweden

URKUND is the leading system for plagiarism control in the Nordic region. It is designed with user friendliness in mind and is designed to help teachers and examiners. The system helps improve the quality of education by providing pedagogical support in the writing process and to save time otherwise spent on verifying the contents of written assignments.

URKUND can be integrated into external systems, allowing users of a Learning Management System to use URKUND as part of an already familiar environment. The new web service (2011) has all the nuts and bolts to make the interaction between our partners and us both more modern and flexible. The new web service uses completely different technology and we have therefore in connection with the revamp also improved major parts of both the flow and the logic. Among other things, we move away from e-mail as a method of transmitting documents in favour of a REST-based protocol. The new integration service is not an SOAP web service. It uses a REST inspired protocol using XML or JSON over HTTP.

URKUND is already integrated with a number of common Learning Management Systems, such as Moodle, Blackboard, Fronter, SharePoint, PingPong, Vklass and others.

Integration with It's Learning is coming up during 2011.

Figure 1.15.: Urkund Website

Business Promotion

"URKUND was born from the academic world. A team of teachers developed the idea of a web based service that would help them detect and deter plagiarism and URKUND was born in the fall of 2000. The problem of plagiarism received much attention in the media and more and more began realise the scope of the problem and the need of a tool to support the pedagogical work. URKUND continued to grow and develop over the years and came to be recognised as Sweden's foremost anti plagiarism service.

Today, URKUND is present in our neighbouring countries and continental Europe as well as the USA, Asia and the Middle East.

URKUND is a natural part of the educational work of the academic world today. Both faculty and students are aware of the immediate and long term benefits of our system."

["Urkund Business Promotion"\(?\) Urkund](#)

Urkund is the last system in our comparison of partially useful systems. It ranked at the 5th position in the HTW "Plagiarism Detection System Test" 2010. The company is from sweden and started their business in 2000. In this test it ranked high in effectiveness but on the other side it's not easy to use. It has problems in the translation and after the redesign 2008 the usability was going worse. Overall "the navigation is confusing, the layout at times catastrophic with texts overlapping fields, the printed reports could be better, the error messages are cryptic, and the link descriptions are unclear." "Plagiat Team HTW"(?)

Urkund costs

Not stated.

Datum:Zeit	Dok Nr.	Name des Dokuments	Absender-Antragsteller	Reg	A
2010-08-05 13:12	D2897850	24.pdf	cassis, cornelia		
2010-08-05 13:12	D2897851	26.pdf	cassis, cornelia		
2010-08-05 13:12	D2897852	27.pdf	cassis, cornelia		
2010-08-05 13:12	D2897853	28.pdf	cassis, cornelia		
2010-08-05 13:12	D2897854	29.pdf	cassis, cornelia		
2010-08-05 13:12	D2897855	30.pdf	cassis, cornelia		
2010-08-05 13:12	D2897856	31-pickles.pdf	cassis, cornelia		
2010-08-05 13:12	D2897857	32-zakumi.pdf	cassis, cornelia		
2010-08-05 13:12	D2897858	33-eyjafjallajoeku...pdf			
2010-08-05 13:12	D2897859	33-eyjafjallajökul...pdf			
2010-08-05 13:12	D2897860	34-stieglersson.pdf			
2010-08-05 13:12	D2897861	35-xss.pdf			
2010-08-05 13:12	D2897862	36-champagne.pdf			
2010-08-05 13:12	D2897863	37-mosque.pdf			

zweiter Urkundtest

Attachment: pdf.ZIP (1387433 bytes)

Figure 1.16.: Urkund List view

Resume commercial software systems

Es gibt zwar von Jahr zu Jahr mehr Softwareprodukte, die von sich behaupten, einen Text zuverlaessig daraufhin zu ueberpruefen, ob es sich um ein Plagiat handelt oder nicht. Aber: Diese Systeme haben nach temporaerer Besserung inzwischen wieder an Qualitaet verloren. Zu diesem ernuechternden Ergebnis kommen wir bei der vierten Untersuchung von Softwareprodukten zur Plagiatserkennung. Besonders pikant: Einige der Systeme werden von hoechst zweifelhaften Unternehmen angeboten, darunter auch von solchen, die Ghostwriting anbieten.

Im Test von 2010 - die vorherigen Tests fanden 2004, 2007 und 2008 statt - haben wir 26 Plagiaterkennungssysteme unter die Lupe genommen. Es gab auf dem Markt zwar 47 die wir finden konnten, aber wir mussten aus Zeitgrunden vorerst die Kollusionserken-

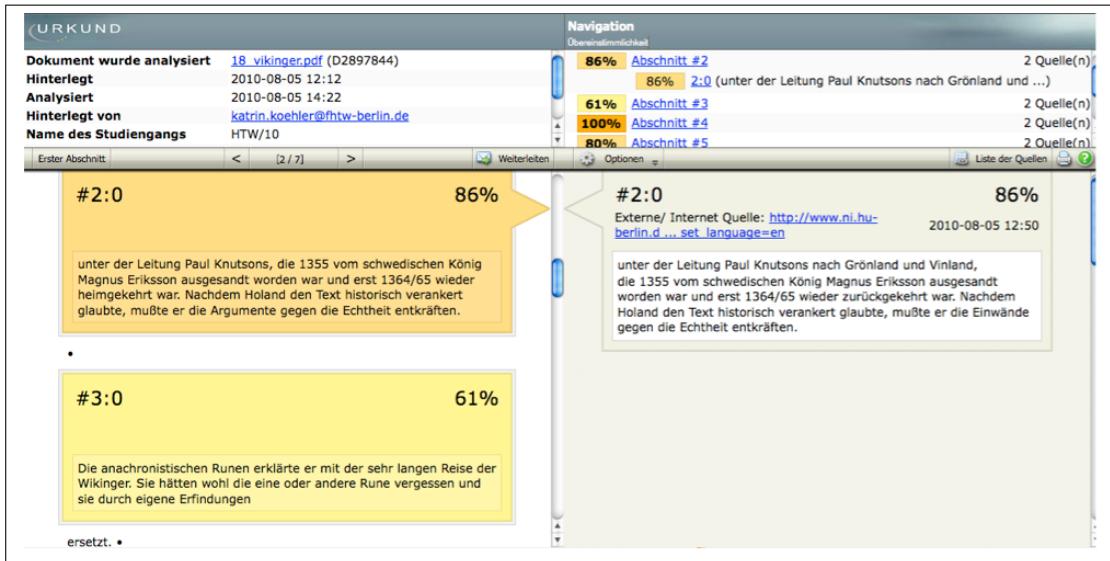


Figure 1.17.: Urkund Report

nungssysteme und die Systeme zur Erkennung von Plagiaten in Programmcode zurueckstellen. Auch Systeme, fuer die wir kein Zugang bekommen konnten (unter anderem Systeme, die veraergert waren, dass sie in vergangenen Tests so schlecht abgeschnitten haben), haben wir nicht testen koennen. Fuer die Pruefung wurden neue Testfaelle in Englisch und Japanisch entwickelt und jedes System mit 42 kurzen Essays konfrontiert. Darueber hinaus wurden die Benutzerfreundlichkeit der Systeme und die Professionalitaet der Unternehmen bewertet.

Fazit: Die auf dem Markt angebotenen Plagiatserkennungssysteme lassen sich derzeit drei Kategorien zuordnen: teilweise nuetzlich, kaum brauchbar und nutzlos. Teilweise nuetzliche Systeme koennen dann gut verwendet werden, wenn mit Hilfe einer Suchmaschine und drei bis fuenf Woertern eines verdaechtigen Absatzes bereits erste Plagiatsindizien erbracht wurden. In diesen Faellen helfen Systeme wie PlagAware, Turnitin, Ephorus, PlagScan oder Urkund dabei, groessere Sicherheit zu gewinnen. Doch selbst die besten Systeme finden hoechstens 60 bis 70 Prozent der plagierten Anteile. Eine ganze Reihe von weiteren Systemen sind hingegen kaum brauchbar. In der Kategorie der nutzlosen Systeme haben wir eine Reihe von Betruegern ausfindig gemacht.

Unsere Empfehlung: Plagiatserkennungssysteme sollten nur bei konkrem Verdacht verwendet werden, statt die Studierenden unter Generalverdacht zu stellen. Parallel dazu

sollte der Focus an Hochschulen und Schulen stärker auf der Aufklärung liegen: Was ist ein Plagiat, warum darf nicht plagiert werden und wie arbeitet man richtig? Dies sei sinnvoller als zuviel Zeit in die Durchsuchung von eingereichten Arbeiten bzw. in die Vermeidung von Fehlalarmen zu investieren.

— Prof. Dr. Debora Weber-Wulff, professor for media and computing at the University of Applied Sciences HTW Berlin has tested plagiarism detection systems in 2004, 2007, and 2008 and has published widely on the topic. In this fourth test series that was just completed at the end of 2010, 26 systems out of 47 available systems were closely examined. Particular focus was given to seeing how well the systems detect a known amount of plagiarism, and how they react when offered original material.

In 2008 the systems fared slightly better on the test, but in 2010 many have lost the ability to detect plagiarisms that have been slightly edited - word orders switched, words dropped or added, or synonyms used. The best systems only reached a grade of C-, not quite reaching 70% of the possible points.

The 2010 test included not only short essays in German, but also ones in English and Japanese. Additionally, a usability metric was calculated that took into account aspects such as design, language consistency, navigation, print quality of the reports, and how well the system fits into the workflow of a university. A new professionalism metric includes giving a real street address and the name of a contact person; not advertising for paper mills or ghostwriting services; answering the phone during normal business hours and not installing malware on the computer under the guise of installing the detection software.

The systems were categorized as partially useful, barely useful, and useless for university purposes. The best systems with between 60 and 70% effectiveness are PlagAware, Turnitin, Ephorus, PlagScan and Urkund.

Our recommendation: Only use these systems when suspicions of plagiarism arise that cannot be found with 3-5 words in a search machine. The focus should be on teaching students about plagiarism and how to avoid it instead of investing time in using software. Most of the work involved later is in preparing a plagiarism case and dealing with the plagiarist, and for this good documentation is needed. Very few systems provide good

documentation.

1.3.2. Free tools and techniques

simtext

Google-Search



Figure 1.18.: Google Search

1.4. Vroni Plag

Erklaerung was ist Vroni Plag....woraus entstanden, welche Mitglieder usw. Das ist nicht genau bekannt, da man sich fuer eine Mitarbeit nicht anmelden muss. Es gab am 18.3.2012 6335 Seiten im Wiki, 189 verschiedene Accounts haben schon mal editiert, 21 Leute sind als Admins eingetragen, und es gibt 3 Buerokraten, die Rechte vergeben koennen.

Das VroniPlag Wiki ist ein am 28. Maerz 2011[2] auf Wikia gegrundetes Wiki, das verschiedene Hochschulschriften - hauptsaechlich Dissertationen - untersucht, die unter Plagiatsverdacht geraten sind. Die Untersuchungen fuehrten in mehreren Faellen zur Aberkennung des Doktorgrades.

VroniPlag Wiki, das die Idee des GuttenPlag Wikis adaptiert, ist nach Edmund Stoibers Tochter Veronica benannt, deren Dissertation als erste untersucht[3] und der infolge dessen der Doktorgrad aberkannt wurde.[4] Bis Februar 2012 erhoehte sich die Zahl der namentlich genannten untersuchten Arbeiten auf 18, darunter einige von Politikern.

Der Gruender von VroniPlag ist Martin Heidingsfelder.

Alte und Junge, Maenner und Frauen, Wissenschaftler und Nicht-Wissenschaftler aus allen moeglichen Orten.

Die Wiki-Beitragenden haben unterschiedliche Motivationen und Ziele, manche behalten diese fuer sich. Viele Wiki-Beitragende wenden sich mit der Plagiatsdokumentation gegen akademisches und wissenschaftliches Fehlverhalten. Durch die Plagiatsdokumentation erlangt das Thema "Plagiat" Aufmerksamkeit. Diskussionen werden angeregt, und es entsteht ein Problembewusstsein, das der qualitativen Verbesserung von wissenschaftlichen Arbeiten dienlich ist. Textsynopsen dienen der Meinungsbildung und legen unterschiedliche Vorgehenstechniken bei woertlichen und sinngemaessen Plagiaten offen. In der speziellen Art, im Umfang, und in der Haeufigkeit von Plagiaten offenbaren sich verschiedene Defizite in der Qualitaet von Dissertations und Habilitationsschriften, Defizite in deren Bewertung und Defizite in der Betreuung der Kandidaten. Einige Wiki-Beitragende leiten aus der oeffentlichkeit der Dokumentation eine Praeventionswirkung

ab, welche hilft, Plagiatsfaelle in der Zukunft zu vermeiden. Manche interessieren sich nur fuer ganz bestimmte Plagiatsfaelle. Der/die eine oder andere mag ein besonderes Interesse an der Aberkennung eines zu Unrecht verliehenen Doktorgrads haben. Auch wissenschaftliche Interessen (Plagiatsforschung) werden vertreten. Im Sinne einer Rezension klaert die Plagiatsdokumentation ueber die Zitierbarkeit von Werken und die Originalitaet von deren Inhalten auf.

Wissenschaftsplagiate sind unzureichend kenntlich gemachte woertliche und sinngemaesse uebernahmen aus anderen Texten in wissenschaftlichen Schriften. Oft wird dadurch ein wissenschaftlicher Erkenntnisgewinn vorgetaeuscht oder fruehere Ideen fremder Autoren erscheinen als vermeintlich neue eigene Ideen. Wissenschaftsplagiate fuehren dazu, dass wissenschaftliche Arbeiten ihren wesentlichen Zweck verfehlten: Neue Erkenntnisse herzovzubringen. Sie fuehren dazu, dass falsche Autoren zitiert und verfaelschte, verwaesserte, unreflektierte, und manchmal unrichtige Inhalte wiedergegeben werden. Sie konterkarieren den Fortschritt. Sie fuehren zur Verschwendung von Forschungsgeldern. Sie haben auch zur Folge, dass ungeeignete Personen wissenschaftliche Fuehrungspositionen erlangen. Aus diesen Gruenden ist Redlichkeit in der Wissenschaft so wichtig.

1.4.1. Activities

Das VroniPlag Wiki ist ein am 28. Maerz 2011[2] auf Wikia gegrundetes Wiki, das verschiedene Hochschulschriften - hauptsaechlich Dissertationen - untersucht, die unter Plagiatsverdacht geraten sind. Die Untersuchungen fuehrten in mehreren Faellen zur Aberkennung des Doktorgrades.

VroniPlag Wiki, das die Idee des GuttenPlag Wikis adaptiert, ist nach Edmund Stoibers Tochter Veronica benannt, deren Dissertation als erste untersucht[3] und der infolge dessen der Doktorgrad aberkannt wurde.[4] Bis Februar 2012 erhoehte sich die Zahl der namentlich genannten untersuchten Arbeiten auf 18, darunter einige von Politikern.

Der Gruender von VroniPlag ist Martin Heidingsfelder.

Ergbnisreport

Fuer die Vorlage an den Ausschuss

Haupttext
001 002 003 004 005 006 007 008 009 010 011 012 013 014 015 016 017 018 019 020
021 022 023 024 025 026 027 028 029 030 031 032 033 034 035 036 037 038 039 040
041 042 043 044 045 046 047 048 049 050 051 052 053 054 055 056 057 058 059 060
061 062 063 064 065 066 067 068 069 070 071 072 073 074 075 076 077 078 079 080
081 082 083 084 085 086 087 088 089 090 091 092 093 094 095 096 097 098 099 100
101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120
121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140
141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160
161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180

Figure 1.19.: Page numbers

Barcode analysis

Plagiarism page analysis

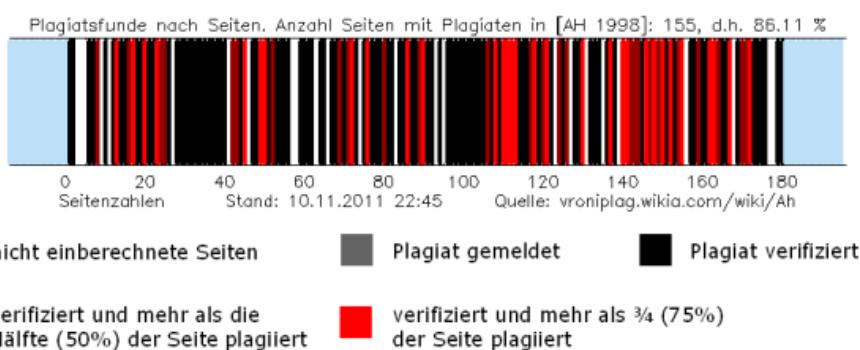
1.4.2. Pros vs. Cons

Eine kritische Auseinandersetzung mit der Dissertation von Dr. Arne Heller: Unternehmensführung und Unternehmenskontrolle unter besonderer Berücksichtigung der Gesamtverantwortung des Vorstands

Dissertation zur Erlangung der Doktorwürde an der Fakultät für Rechtswissenschaften der Universität Hamburg.

Gutachter: Prof. Dr. Klaus-Peter Martens, Prof. Dr. Detlev Joost. Veröffentlicht: Heidelberg 1998. → Nachweis

Deutsche Nationalbibliothek → ISBN 3-7685-1698-9.



Interpretationshilfe für die Prozentzahl über dem Barcode: Sie drückt den Anteil der Seiten aus, die Plagiatsanteil enthält, nicht den Plagiatsanteil am Fließtext.

Figure 1.20.: Barcode analysis

[1.] Ah/Fragment 105 06		Ah, BauernOpfer, Fragment, Geprüft, Mertens
Zuletzt bearbeitet: 2011-09-28 19:37:00 Hindemith		1989, Schutzlevel sysop
Kategorie BauernOpfer	Quelle Mertens 1989	Sichtung Guckar
Untersuchte Arbeit: Seite(n): 105, Zeilen: 6-14	Quelle: Seite(n): 45, Zeilen: 15-23	<input type="button" value="Farbig"/>
<p>Die von Hommelhoff vertretene extensive Pflicht zur Konzernbeherrschung ist jedoch weder mit dem geltenden Konzernrecht, das im wesentlichen Minderheiten- und Gläubigerschutzrecht ist und, wie §§ 17, 18 AktG zeigen, aus der Abhängigkeit keine Pflicht zur Konzernbildung ableitet, noch mit der Verfassung der Aktiengesellschaft vereinbar, die der Hauptversammlung, wie sich aus §119 Abs. 2 AktG ergibt, keine Geschäftsführungskompetenz zuweist; überdies ist sie auch wirtschaftsrechtlich höchst bedenklich, da sie zur Konzentrationsforderung und zur Erstreckung des Vollmachtsstimmrechts der Banken hinein in Sachfragen der Unternehmensleitung tendiert [FN 482].</p>		Diese These ist weder mit dem geltenden Konzernrecht vereinbar, das im wesentlichen Minderheiten- und Gläubigerschutzrecht ist und - wie §§ 17, 18 zeigen - aus der Abhängigkeit keine Pflicht zur Konzernbildung ableitet, noch mit der Verfassung der Aktiengesellschaft, die der Hauptversammlung aus guten Gründen keine Geschäftsführungskompetenz zuweist (vgl. auch oben Rn. 40 f., 51 ff.). Sie ist überdies auch wirtschaftsrechtlich höchst bedenklich; denn sie tendiert zur Konzentrationsförderung und zur Erstreckung des Vollmachtsstimmrechts der Banken in Sachfragen der Unternehmensleitung hinein.

Figure 1.21.: fragments



Figure 1.22.: colors

2. Project Workflow and Requirements

First of all, we've got a confession to make: Unplagged is like a big playground of new workflows and technologies for us, as we are aiming to incorporate “best-practices” wherever possible, or at least what we currently consider to be best-practices.

We believe this approach is necessary, because of the fact, that we are essentially trying to incubate Unplagged as a real open source project and this will only work if it is well crafted and if cutting-edge workflows and technologies are used. Nearly all of the team members are also working in some kind of web related side job, so we all got enough experiences with the problems that can occur during the maintenance of badly designed software.

Most of the times this works pretty well, but sometimes we are still trying to figure out how to get everyone up to speed with every technology and part of the system or how to divide the responsibilities carefully.

To start this project, we opted to use *Scrum*¹ as our agile development approach. If you are familiar with this methodology, you may notice, that there could be a few problems when considering, that the team is working mostly distributed without a common office and with very different time tables for each of the members.

We struggled a bit to tweak the workflow that is required by Scrum to fit the situation we faced, but you will see in the following what we came up with.

¹A nice introduction into Scrum is “The Scrum Primer” of the Scrum Alliance: <http://www.scrumalliance.org/resources/339>

2.1. The Workflow

To make it possible to work efficiently together in this kind of environment, we chose to use [Redmine](#) as our project management tool, which you can access under:

- <http://tickets.unplagged.com>

If you register there, an administrator should grant you access to the tickets and the wiki, so that you can participate in solving the problems at hand.

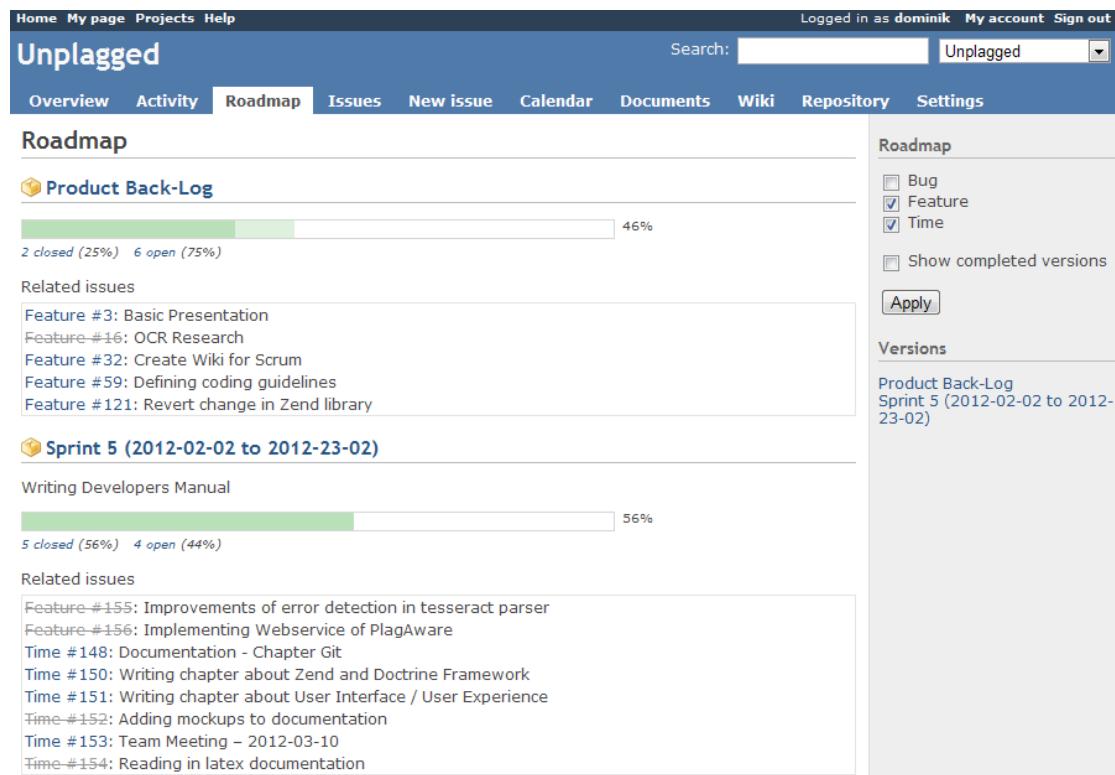


Figure 2.1.: Redmine Roadmap

What we are doing there is to map every *Sprint* and the *Product Backlog* to Redmine's notion of "Version" and every identified *User Story* to an "Issue".

You can see in figure 2.1 the view of the roadmap, with the current sprint 5 designated to create this very document at the bottom and a not very well filled product backlog at

the top.

Normally every identified user story that is not part of the current sprint should be in the product backlog (and we got plenty), but as we were still working in our small group at this point, the hassle of filling in all the tickets seemed unnecessary. This is something that will be fixed in the near future, so that you are able to see where the development is going.

Currently we are working mostly with four week long sprints, to overcome the problem that we are not working fulltime on the tickets, which is something that scrum normally assumes.

To have a nice statistical overview and more planning security for the “scrums”, it is required to log the time that was spent on an issue within redmine.

2.1.1. Product Owner — “The Debbie Meetings”

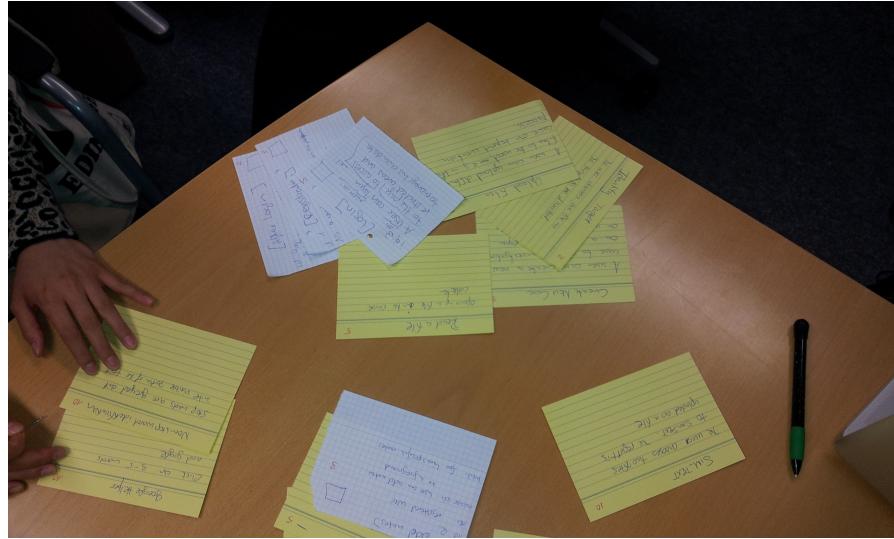


Figure 2.2.: Scrum Meeting

To figure out the user stories we mostly rely on what we internally call “Debbie Meetings”. Normally at the end of every sprint, the members of the team meet with Prof. Weber-Wulff, who we state to be our *Product Owner*. We simply sit down there and talk about

what should be implemented in the next sprint and collect it on cards as you can see in figure 2.3.

We consider this to be just a temporary way of handling this, because we hope that when we eventually have a prototype that we consider to have enough “business value” to be shown to more people, the focus will shift away from Prof. Weber-Wulff to a more broader understanding of the product owner.



2.2. Target Group

2.3. User roles

As the Unplagged system will provide a permission based user system, our goal is to make it possible, to create custom user roles from an administration area and make it possible for users to have multiple user roles in one case and also different roles for different cases. The standard roles which will be provided by the system are:

Guest A user without a valid login can only see the parts of cases that are set to be public.

Registered Registered users can get “promoted” to higher roles and contribute to publicly editable cases.

Collaborator Collaborators are registered users who were granted access to a specific case. Collaborators can access and edit these projects.

Case-Manager Case-Managers can set up new cases and manage collaborators for their cases and project versions. They may have the permission to add or remove project members.

Admin An admin owns all permissions, such as user administration or project administration. They also have the ability to block/unblock an existing case.

2.4. Basic functionalities

2.5. Document Parser

2.6. Detection Modes

2.7. Plugin Architecture

2.8. Use Cases

3. Developing Unplagged

Coming from the [Project Workflow and Requirements](#) here we have yet another set of requirements, before we can start with the actual description of the way you can help us develop the system. This time it's about what we believe will be helpful or sometimes even necessary prerequisites.

First of all, the programming languages mostly used in Unplagged are PHP and JavaScript, both of which in conjunction with a framework. Teaching programming languages is, as you probably can imagine, well beyond the scope of this document, but we will at least try to cover the most important concepts of the frameworks as they occur.

The used frameworks are [jQuery](#) for Javascript and [ZEND](#) for PHP respectively. jQuery is kind of the industry standard for unobtrusive scripting with about 50% of the Top 10.000 websites using it according to [?](#) and the Zend framework is also well established and brings a lot of features, that are useful to this project.

For most of the other topics, we will give you some (hopefully) helpful resources on the way, if it isn't covered thoroughly by us. But just to let you know, here is a list of the buzzwords, er technologies that will be mentioned:

- Scrum
- HTML5 and CSS3
- Continuous Integration
- Responsive Webdesign

- Progressive Enhancement
- Git, Netbeans, Redmine
- Tesseract, Imagemagick
- Simtext

As said in section ??, the system is developed, so that it should work on multiple platforms. This makes it sometimes difficult to describe certain installation processes in a way that would work for everybody. As it's often most problematic, to get some Linux software running on Windows, we will mostly concentrate on the way those things are done on this platform and give the instructions for other operating systems as an aside if necessary.

3.1. Development Environment

The following parts will mostly focus on the way you can get a development version of unplagged up and running on your system.

3.1.1. Git

The source code and files of all parts of the Unplagged project are managed through Git, with the repository hosted at [Github](#). Git is a distributed version control system, that exists since 2005 and gained more and more track in recent years. Many developers prefer it over other version control systems, because it is much easier to create different branches and merge them again or simply initialize local repositories. This made it also interesting for us to use it for Unplagged.

However, none of the team members had ever used Git before in a bigger context so it was a challenge to get it running on all the systems. But we took it, to explore all the features Git offers. .

Installing Git Bash

First of all let's find out, how to install the Git console application, called Git Bash. Unfortunately all the GUIs we were evaluating didn't work consistently, so we decided to use it from the command line only. A very good instruction on how to install the Git Bash can be found on the website of the github project:

Windows: <http://help.github.com/win-set-up-git/>

Linux: <http://help.github.com/linux-set-up-git/>

Mac OS X: <http://help.github.com/mac-set-up-git/>

Getting the source code of the unplagged project

Now it is time to get the project source code on your machine. As said before, the whole unplagged project is hosted on github, so if you want to be able to contribute source code later on, you first need to create an account there:

- <https://github.com>

This isn't necessary, if you simply want to look into the source code, which can be accessed via the repository URL:

- <https://github.com/benoertel/unplagged>

If you haven't been granted write access to the above mentioned repository by a project member (which is very likely when you are reading this document for the first time), you will need to do a fork of the Unplagged project right at github, like described in:

- <http://help.github.com/fork-a-repo/>

After this, the following steps are mostly the same for everybody, with the distinction of the project URIs, which should be the one of your newly created fork.

Open up the Git Bash and switch to the directory where you want the project to be located and clone the repository as you can see in listing 3.1.

Listing 3.1: Cloning a repository

```
1 cd Sites/unplagged.local  
2 git clone  
    https://<username>@github.com/benoertel/unplagged.git
```

After this you should have a local copy of all the repository data in the specified directory.

The most important git commands

You are now ready to use Git! Here are some more instructions on the most important commands and how to properly use it. However, if the given instructions in this manual are not enough, feel free to checkout the whole Git manual on:

- <http://schacon.github.com/git/user-manual.html>

The unplagged project consists of several branches, which are used to develop and store code independently of the other developers. Once a new feature is done, it is merged into the master branch. The master branch usually includes only fully tested and deployable source code.

As a new developer, it is important to create an own branch before doing anything else and switch to it.

Listing 3.2: Creating branches

```
1 git branch mynewfeature  
2 git checkout mynewfeature
```

Now anything in the repository can be changed. At any point changes can be versioned in the repository by using the `git commit` command. If new files were created, `git add` has to be executed as well.

Listing 3.3: Committing

```
1 git add .
2 git commit -m "A message that describes the changes."
```

When the feature is fully working and approved, it has to be merged back to the master branch, in order to get deployed to the staging environment. To do this, the master branch has to be checked out, updated with `git pull` and then all changes have to be merged from the new feature into the master branch. The feature branch can then be removed.

Listing 3.4: Merging branches

```
1 git checkout master
2 git pull
3 git merge mynewfeature
4 git branch -d mynewfeature
```

In comparison to Subversion for example, Git has one more step to really write back to the remote source repository. After a `git commit`, a `git push` has to be executed, each push can include multiple commits.

Listing 3.5: Pushing to the server

```
1 git push origin master
```

This is nearly it, the changes to the repository have been pushed to the master branch. The only thing, that probably has to be done now, is to open up a pull request on github, if you developed on your own fork of the project. This means, that you are asking the project members who have access to the “real” Unplugged github account, to integrate your changes into the actual project sources. A nice description of how this process is done can be found at github again:

- <http://help.github.com/send-pull-requests/>

Handling conflicts in merging process

It is possible, if two developers were working on the same part of file, that a conflict is found during the merge. Such a conflict could look like this:

Listing 3.6: Merge conflict

```

1 CONFLICT (content): Merge conflict in readme.txt
2
3 To https://github.com/benoertel/unplagged.git
4 ! [rejected]          master -> master (non-fast-forward)
5 error: failed to push some refs to 'https://github.com/
       benoertel/unplagged.git'
6 To prevent you from losing history, non-fast-forward updates
   were rejected
7 Merge the remote changes (e.g. 'git pull') before pushing
   again. See the
8 'Note about fast-forwards' section of 'git push --help' for
   details.
9
10 # Unmerged paths:
11 #   (use "git add/rm <file>..." as appropriate to mark
12 #     resolution)
13 #   both modified:      readme.txt
14 #

```

To resolve the issues, open the files listed in the error message, in this case *readme.txt* and decide how the correct version should look like, by removing all the “<<<<< << HEAD” and “>>>>>> b478801d68267ef479acc5ca54544634c52c545c” parts accordingly or using a dedicated merge tool, that is able to show you the changes that were made

Here is an example of how this process would work:

Listing 3.7: Conflicted file

```
1 <<<<< HEAD
2 The goal of this project is the creation of an easy-to-use,
   web-based
3 system to document and detect plagiarism in scientific papers.
4
5 hello world
6 =====
7
8 The goal of this project is the creation of an easy-to-use,
   web-based
9 system to document and detect plagiarism in scientific papers.
10
11 >>>>> b478801d68267ef479acc5ca54544634c52c545c
12 Just a change for educational purposes.
```

Could look like this after merging:

Listing 3.8: Fixed conflict after merging

```
1 The goal of this project is the creation of an easy-to-use,
   web-based
2 system to document and detect plagiarism in scientific papers.
3
4 hello world
5
6 Just a change for educational purposes.
```

3.1.2. Local Deployment

This subsection will describe how to configure a virtual host properly. A virtual host is a domain that is mapped to the local web server. It is assumed that Apache, MySQL and PHP are already running on the machine. If not, here are some tutorial to get them all running:

Windows:

<http://www.apachefriends.org/de/xampp-windows.html#1098>

Mac OS:

<http://www.djangoproject.com/blog/2011/07/24/installation-of-mysql-server-on-mac-os-x-lion/>

<http://www.quarkstar.at/index.php/2009/05/18/webserver-aktivieren-und-konfigurieren-in-mac-os-x/>

Most Linux distribution should already have this kind of server stack installed.

The main goal is to create a local domain and add the virtual host from listing 3.12 to the vhost config.

In Max OS X and Linux this can be done via the command line:

Listing 3.9: Mac OS X: Creating virtual host

```
1 sudo vi /private/etc/hosts
2 #add the following line:
3 "127.0.0.1 unplagged.local"
4
5 sudo vi /private/etc/apache2/extr/httpd-vhosts.conf
```

On Windows you need to open up your *hosts* file, which is mostly located in *C:\WINDOWS\system32\drivers\etc\hosts*, and add the following line on the bottom:

Listing 3.10: New host declaration

```
1 127.0.0.1 unplagged.local
```

Now you need to open your apache configuration file *C:\xampp\apache\conf\httpd.conf* and remove the hash symbol(uncomment) from the following line

Listing 3.11: httpd.conf

```
1 #Include conf/extr/httpd-vhosts.conf
```

Add the following configuration to the httpd-vhosts.conf file you just included:

Listing 3.12: Apache configuration

```
1 <VirtualHost *:80>
2   ServerName unplagged.local
3   DocumentRoot "/Users/me/Sites/unplagged.local/public"
4   SetEnv APPLICATION_ENV "development"
5
6   <Directory "/Users/benjamin/Sites/unplagged.local/public">
7     Options +Indexes +FollowSymLinks +ExecCGI
8     DirectoryIndex index.php
9     AllowOverride All
10    Order allow,deny
11    Allow from all
12  </Directory>
13 </VirtualHost>
```

You can tryout your new configuration by entering *unplagged.local* in your browser.

3.1.3. Netbeans

Configuring Tests

Documentation

Unplagged uses [?](#) for the generation of a HTML page of all the source code documentation comments, because of it's superior and much more beautiful user interface in comparison to the older [?](#).

Sadly the automatic generation is not yet supported by Netbeans, but as it will be soon(?), we are currently only generating this server-side as described in section ??.

This section will be enhanced, when the Netbeans user interface becomes available. If you are interested, you could install the software for yourself and use it over the command line.

3.1.4. Additional Software

As we currently have no installer or script that checks for installed software, you still have to install some additional dependencies to make some parts of the system work. Those are mainly command line tools that we use for the optical character recognition or text comparison.

Most of the times the software wouldn't break completely if those dependencies were not installed, but some parts would silently fail, which is of course one of the more annoying problems to debug.

Tesseract

Tesseract is an open source OCR¹ software, that is used to

Simtext

Imagemagick

3.1.5. Continuous Integration

To always have a running version of the latest code, we use an automated workflow, that always deploys everything that has been pushed to the Github repository on the Unplugged staging server. The machine this is done with, is a simple Ubuntu web server, that is also used for hosting our collaboration tools and the webpage.

As you can see in figure 3.1 the mechanism used for this is a callback, the *post-receive hook* of git, which github employs to let it's users enter a *post-receive URL* to call a URL after someone has pushed to the repository. The URL that gets called is located on our staging server and gets answered by a Redmine plugin called “redmine_github_hook”,

¹Optical character recognition

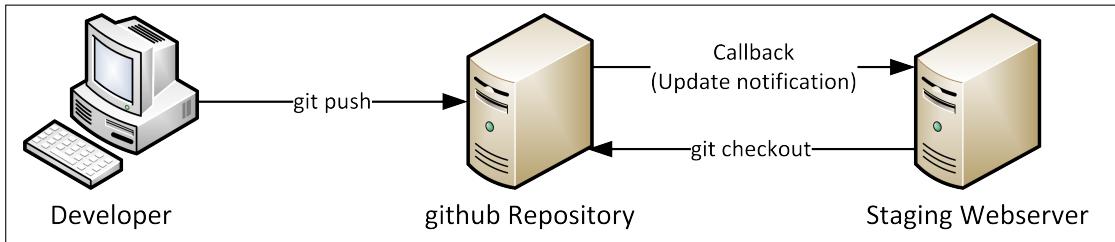


Figure 3.1.: Deployment workflow

that would normally only call a checkout on the server-side repository, so that the newest sources can be seen via the Redmine web frontend.

Listing 3.13: Changes to redmine_github_hook.rb

```

1 # Fetches updates from the remote repository
2 def update_repository(repository)
3   command = git_command('fetch origin', repository)
4   if exec(command)
5     command = git_command("fetch origin '+refs/heads/*:refs/
6       heads/*'", repository)
7     exec(command)
8   #custom checkout to preview area
9   system('sh /usr/local/etc/scripts/buildUnplaggedPreview.sh
10 ')
11 end
12 end

```

However, we tweaked the source code of this plugin slightly, as you can see on line 9 in listing 3.13 so that it also calls the below bash script(listing 3.14) to initiate the deployment process.

Listing 3.14: Deployment script

```

1 #!/bin/bash
2
3 cd /var/git/unplagged.git/
4 GIT_WORK_TREE=/var/www/preview.unplagged.com git checkout -f
5
6 cd /var/www/preview.unplagged.com

```

```

7 #generate phpdoc
8 apigen -s application/ -s library/Unplagged/ -d docs/phpdoc --
    title "Unplagged Documentation" --todo yes
9
10 #run database build scripts
11 cd scripts/build
12 php initdirectories.php
13 php doctrine_staging.php
14
15 cd /var/www
16 chown www-data:www-data preview.unplagged.com

```

The bash script is then used to do a “clean checkout”(without hidden .git folders) of the repository and to run “Apigen”, an engine to process the PHPDoc comments inside the project. Those two things can be accessed by the already prepared vHosts on the server:

- <http://preview.unplagged.com/>
- <http://phpdoc.unplagged.com/>

If you would like to get access to the preview areas, you need to obtain the password and username from a team member.

Possible Improvements

The above described workflow is, as we believe, already on a good way, but it still has a lot of room for improvement. First of all, it would be nice to only let the deployment go through, if the unit tests ran successfully on the server and to have some sort of email notification mechanism if this wasn’t the case.

Another improvement would also be to have a separation into a staging environment with the newest commits and an actual preview environment, that can be deployed to a known stable state/commit of the system in a simple manner.

3.2. User Interface / User Experience

This chapter will explain the progress and development of the user interface of our project. As we tried to follow the typical project workflow, we first drew a lot of mockups in the beginning, which represented the main features of Unplugged. At first the wireframes, or also called mockups were drawn by hand, before we digitalized them. As an example the mockups of the 'new case' page are shown below. All the other mockups can be found in the appendix.

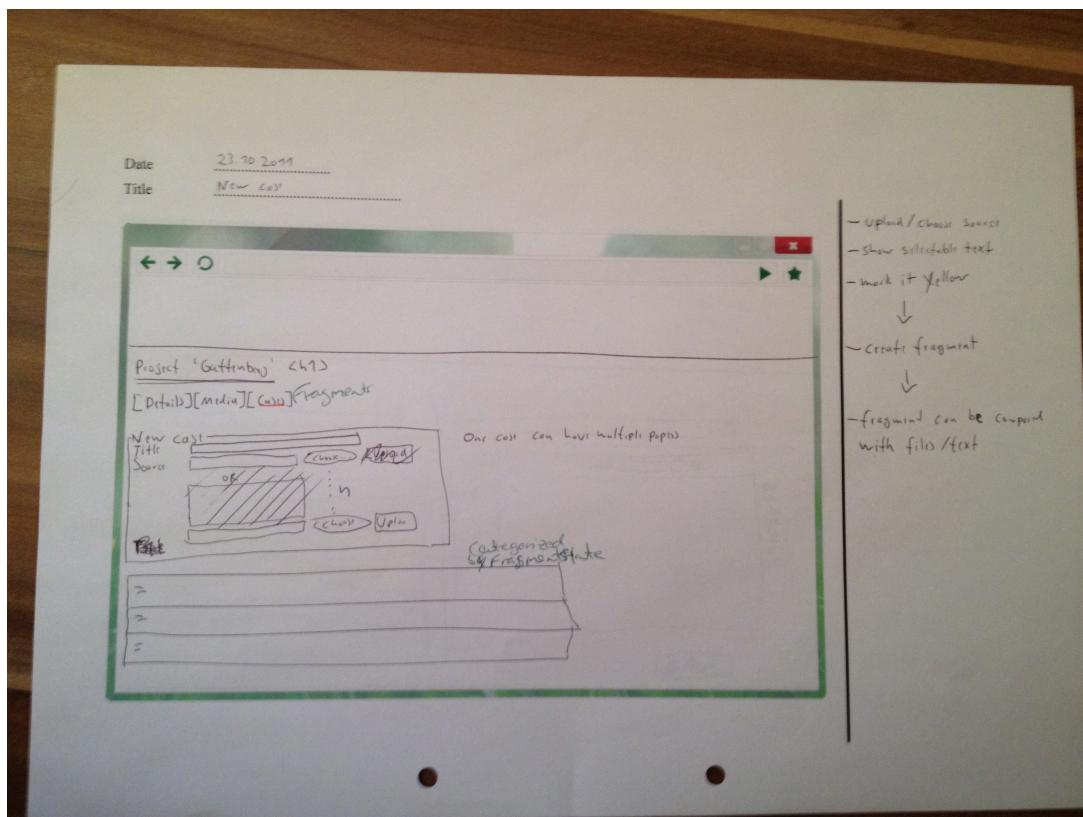


Figure 3.2.: Mockup – New case – hand-drawn

After we had a basic idea how the main interface should be structured, we created a first screen in Photoshop. Therefore we got several helpful suggestions from the website PremiumPixels: <http://www.premiumpixels.com/>.

The next step, before the HTML template got created, we defined the key features, our user interface should take care of:

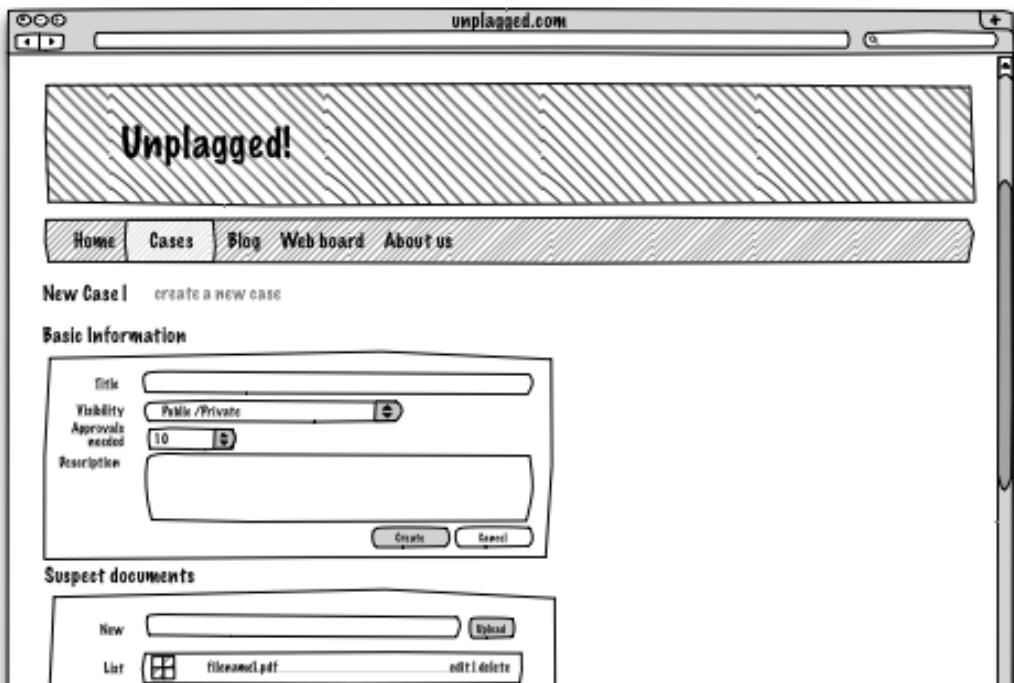


Figure 3.3.: Mockup – New case – digitalized

- Responsive Layout – optimized layouts for different devices
- Cross-Browser-Compatibility
- Light-weight and w3c-conform XML Markup
- Progressive enhancement with CSS3 – CSS instead of images where possible

3.2.1. Responsive Layout using CSS3 Media Queries

Since the worldwide amount of smartphones and tablets is growing very fast, website developers should optimize for these devices as well. And so do we. Some functionality as uploading a file, doesn't work on iOS at all, but at least all functions that are working on mobile devices, should work. So the goal is creating a user interface that uses the same markup, but displays differently on different devices. Therefore CSS media queries can be used, these are basically conditions that execute a part of CSS only if the condition is

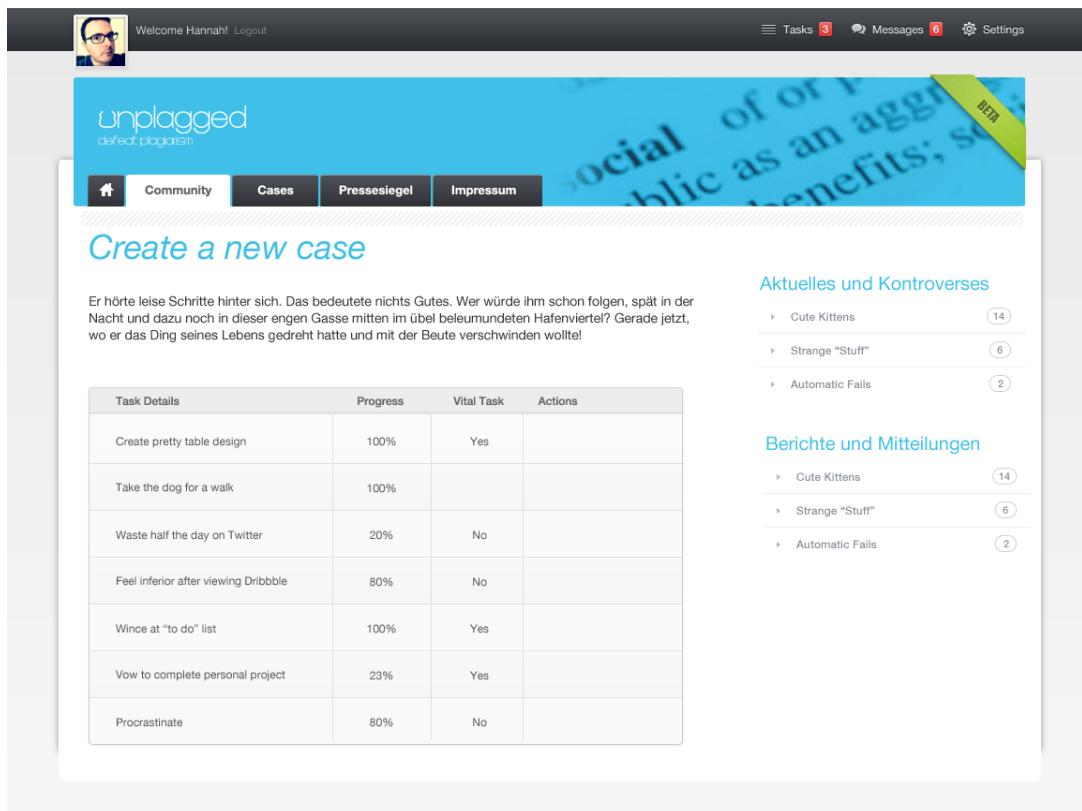


Figure 3.4.: Initial Screen PSD

true. The most prominent conditions are the following:

Listing 3.15: CSS Media Queries

```
1 max-width: 600px  # max browser width 600px
2 min-width: 300px  # min browser width 300px
3 orientation:landscape # current orientation landscape mode
4 orientation:portrait # current orientation portrait mode
5 -webkit-min-device-pixel-ratio: 2 # min pixel ration (iPhone
   4, Retina)
```

These media queries can be combined in any order to display an optimized page for each device. Since only the CSS changes, the HTML does not have to be touched. An example query looks like this:

Das entsprechende CSS Media Query sieht wie folgt aus:

Listing 3.16: CSS Media Query

```
1 @media only screen and (max-width: 600px) { }
```

Media queries are a new feature of modern browsers, except IE9. Though, this isn't a problem, because browsers that don't support them, just display the default css for desktop browsers.

3.2.2. Javascript and fallbacks

Even though many websites require Javascript as mandatory for using the whole functionality range of the page, it is important to provide as much functionality as possible, when Javascript is turned off. A short example will show how to implement a pagination with and without Javascript.

Usually when Javascript is disabled, the whole page will be reloaded, when the user changes to another page of the paginated content. The URL in this case will look something like this: <http://unplugged.local/document/list/page/2>. When Javascript is enabled, it

is much faster to only refresh the area of the page, that really needs to reload, in this case the table with the content of the next page. It is only possible to change the hash of an URL, the part after the hash key (#), using Javascript, the changed URL will be: <http://unplugged.local/document/list/#page/2>. An event called 'hashchange' can trigger a change of this part of the url and then reload the content through an AJAX request.

The pagination has the same HTML markup with and without Javascript, but with Javascript enabled it is much more convenient.

Listing 3.17: Javascript Pagination

```
1 $(".pagination a").live("click", function() {
2     var href = $(this).attr("href");
3     if(href) {
4         var substr = href.split('/');
5         var hash = substr.substr.length-2] + "/" + substr[
6             substr.length-1];
7         window.location.hash = hash;
8     }
9     return false;
10 });
11
12 $(window).bind('hashchange', function(){
13     var newHash = window.location.hash.substring(1);
14
15     if (newHash) {
16         var substr = newHash.split('/');
17         var hash = substr.substr.length-2] + "/" + substr[
18             substr.length-1];
19
20         var url = window.location.pathname;
21         if(url.charAt(url.length-1) != '/') {
22             url += '/';
23         }
24         url += hash;
25         $("#main-wrapper").load(url + " #main");
26     };
27 }
```

3.3. Frameworks

When we first discussed which programming language and frameworks, the Unplagged project should be built on, we figured out, that everyone was familiar with PHP. Since programming in a group requires a much better structure, than programming on your own, we needed a framework that requires a comfortable Model-View-Controller structure, we decided to use the Zend Framework. And to get rid of all the database issues as well as getting a flexibility in the used database system, we decided to use an Object-Relational-Mapping framework, called Doctrine. What an ORM is, will be discussed later on.

3.3.1. Zend Framework

The Zend Framework is a typical PHP Framework using the Model-View-Controller pattern. Due to it's pre-defined directory structure, it is easy to get well seperated code.

The directories and their meaning:

- application — includes controller, models and view
- data — currently only includes i18n stuff (language stuff)
- docs — PHP Documentation and Developers Manual
- library — extrnal and internal frameworks and extension to the Zend framework
- public — files that are accessible directly through the browser

- scripts — build scripts, deploying scripts
- temp — data that is overridden at any deployment
- tests — PHP Unit tests

The Zend framework offers REST-ful URLs that follow a fixed pattern: [unplugged.local/controller/action/key/value/key2/value2](#). Each controller is defined as ControllerName-Controller.php in the application/controllers directory and includes all possible actions. For example a file controller will look like this:

Listing 3.18: Persisting an object to the database in Doctrine

```

1 class FileController extends Zend_Controller_Action{
2     public function init() {
3     }
4
5     public function indexAction() {
6     }
7
8     public function uploadAction() {
9     }
10
11    public function listAction() {
12    }
13
14    public function downloadAction() {
15    }
16 }
```

By default, if no action is defined, the indexAction is called. For each action the appropriate view is by default rendered in the application/views/scripts/controllerName/actionName.phtml file.

The models directory includes all the objects, this will be discussed in more detail in the following chapter about Doctrine.

3.3.2. Doctrine

The whole database connection management of Unplagged is implemented using the Doctrine Framework in version 2. It consists of two layers, a database abstraction layer (DBAL) and an object relational mapping framework (ORM). The DBAL uses PDO, a PHP framework for encapsulating database statements. The DBAL manages the communication with any kind of SQL database and offers an own query syntax. This has the advantage, that the database behind the framework can be changed from MySQL, to OracleSQL, PostgreSQL or SQLite at any time. The DBAL is the agent between PDO and the ORM. The ORM is the connection between PHP objects and the DBAL.

Before the use of Doctrine is described, it will be explained, how the database on a new machine can be created and how the database structure can be updated, whenever something changed in the structure. Actually this is very easy, it is required to have a local my sql database at this point having 'root' as a username, no password and a database called 'unplagged'. It is also possible to create a new configuration in the application/configs/application.ini file, although this step will not be described in this chapter. If the database is created, the build script can be executed:

Listing 3.19: Updating database structure

```
1 php unplagged.local/scripts/build/doctrine.php
```

Now, the database is created or updated. If the database already existed, the data in it will not be removed! As described below, the big advantage of ORM is, that the programmer can stay in the PHP object context at any time. The only thing that has to be done additionally, is adding comments to the member variables of a class, that define the fields in the database:

Listing 3.20: Defining a class in Doctrine

```
1 /** @Entity */
2 class UserClass
3 {
4 /** @Column(type="integer") */
5 private $id;
6 /** @Column(length=30) */
```

```
7 private $username;  
8 }
```

The whole syntax documentation of doctrine can be found here: <http://docs.doctrine-project.org/projects/doctrine-orm/en/latest/index.html>

To persist a new object to the database, the persist method on this object has to be called, this writes the object into the doctrine cache. It stays in the cache, until the flush method is called, which actually executes all the previous operation since the last flush. These can be deleting, updating, or editing an object.

Listing 3.21: Persisting an object to the database in Doctrine

```
1 $user = new User();  
2 $user->setUsername('Max');  
3 $em->persist($user);  
4 $em->flush();
```

As the previous examples show, no database programming is necessary to create or persist a new object to the database, everything can be done in the PHP object context.

3.4. Architectural Goals

3.4.1. Progressive Enhancement

3.4.2. Test Driven Development

A. Minutes

B. Logged Time As Of March 11, 2012

The following tables are some example reports generated from the logged time in Redmine. To find the most recent version of these reports or to generate custom data analysis you can use the “Report” tool found in Redmine on the “Overview” page.

Table B.1.: Overview By Member and Month

Member	2011-10	2011-11	2011-12	2012-1	2012-2	2012-3	Total
Dominik Horb	27.30	10.00	55.75	51.00	28.00	16.80	188.85
Benjamin Oertel	34.00	29.00	19.00	57.50	4.50	29.00	173.00
Elsa Mahari	9.50	10.00	23.50	14.50	7.00		64.50
Tien Nguyen	1.00	24.00	13.00	12.00			50.00
Heiko Stammel	16.00	28.00	13.50	11.00			68.50
Total	91.30	101.00	124.75	146.00	39.50	45.80	548.35

Table B.2.: Overview By Member and Issue

Issue	Member	Total
none		18.30
Feature #1: Trac aufsetzen	Dominik Horb	2.00
Feature #5: Create Wiki List of Interesting Plagiarism Papers	Dominik Horb	1.00
Feature #6: Logo	Dominik Horb	1.00
Feature #8: Configure Email Notification in Trac	Benjamin Oertel Elsa Mähari	7.00 1.00
Feature #9: Github in Trac integrieren	Dominik Horb	1.50
Bug #11: Umstellung von Trac auf Redmine	Dominik Horb	1.00
Bug #12: Zeitmanagement in Trac integrieren	Dominik Horb	3.50
Bug #13: Initialize basics in wiki and issues	Dominik Horb	4.80
Bug #14: Setup Github in Redmine	Dominik Horb	1.00
	Dominik Horb	2.50
	Dominik Horb	2.50

Table B.2.: Overview By Member and Issue

Issue	Member	Total
Bug #15: Fremde Federn finden	Dominik Horb	3.00
Feature #16: OCR Research	Dominik Horb	3.50
Feature #20: Create Wiki Page for UserRoles	Dominik Horb	3.50
Time #21: Meeting 24.10.11	Tien Nguyen	1.00
	Dominik Horb	12.50
	Benjamin Oertel	2.50
	Elsa Mahari	2.50
	Tien Nguyen	2.50
	Heiko Stammel	2.50
Time #22: Meeting 17.10.11		10.00
	Dominik Horb	2.00
	Benjamin Oertel	2.00
	Elsa Mahari	2.00
	Tien Nguyen	2.00
	Heiko Stammel	2.00
Time #23: Meeting 10.10.11		10.00
	Dominik Horb	2.00
	Benjamin Oertel	2.00
	Elsa Mahari	2.00

Table B.2.: Overview By Member and Issue

Issue	Member	Total
Time #24: Debbie-Meeting 18.10.11	Tien Nguyen Heiko Stammel	2.00 2.00
	Dominik Horb	7.00
	Benjamin Oertel	1.00
	Elsa Mahari	2.00
	Tien Nguyen	1.00
	Heiko Stammel	2.00
Feature #28: Resume: Projectmeeting with HTW-Plag-Team		4.00
Feature #29: Projectmanagement Research	Heiko Stammel	4.00
Feature #30: Scrum Research	Heiko Stammel	7.00
Feature #31: Create Wiki for Project Management Steps	Heiko Stammel	7.00
Feature #32: Create Wiki for Scrum	Heiko Stammel	6.00
Bug #33: Mockup hand-drawing	Heiko Stammel	2.00
Bug #34: Mockup digital version	Benjamin Oertel	1.00
	Benjamin Oertel	6.00
	Benjamin Oertel	11.00
	Benjamin Oertel	11.00

Table B.2.: Overview By Member and Issue

Issue	Member	Total
Bug #35: Textvergleich-Algorithmen rechechieren	Tien Nguyen	1.00
Time #39: Meeting 31.10.11		12.50
	Dominik Horb	2.50
	Benjamin Oertel	2.50
	Elsa Mahari	2.50
	Tien Nguyen	2.50
	Heiko Stammel	2.50
Time #43: Meeting 2011-11-04 (Martin Heidingsfelder)		8.00
	Benjamin Oertel	2.00
	Elsa Mahari	2.00
	Tien Nguyen	2.00
	Heiko Stammel	2.00
Feature #44: Setting up initial Zend Framework + Doctrine environment		2.00
Feature #45: User Stories	Benjamin Oertel	2.00
Time #47: Meeting 07.11.11	Tien Nguyen	1.00
		10.00
	Dominik Horb	2.00
	Benjamin Oertel	2.00
	Elsa Mahari	2.00
	Tien Nguyen	2.00

Table B.2.: Overview By Member and Issue

Issue	Member	Total
Time #48: Meeting 14.11.11	Heiko Stammel	2.00
	Dominik Horb	2.00
	Benjamin Oertel	2.00
	Elsa Mahari	2.00
	Tien Nguyen	2.00
	Heiko Stammel	2.00
Time #49: Wiki Editing	Dominik Horb	2.00
Bug #50: Meeting Plagiatssteam HTW 3.11.11	Dominik Horb	6.00
	Benjamin Oertel	1.50
	Heiko Stammel	1.50
Time #53: Debbie Meeting 15.11.11	Dominik Horb	3.00
	Benjamin Oertel	1.50
	Elsa Mahari	1.50
	Tien Nguyen	1.50
	Heiko Stammel	1.50
Feature #56: User registration	Benjamin Oertel	6.50
Feature #57: Profile information update		3.00

Table B.2.: Overview By Member and Issue

Issue	Member	Total
Time #58: Meeting 28.11.11	Benjamin Oertel Dominik Horb	3.00 2.50
	Benjamin Oertel Elsa Mahari	2.50 2.50
	Tien Nguyen Heiko Stammel	2.50 2.50
Feature #59: Defining coding guidelines	Benjamin Oertel	0.50
Feature #61: Meeting 24.11.11 - Setting up developing environment	Benjamin Oertel Dominik Horb	0.50 12.00
	Benjamin Oertel Elsa Mahari	3.00 3.00
	Tien Nguyen	3.00
Feature #62: Identify target	Benjamin Oertel	0.50
Feature #63: Read a file	Benjamin Oertel	1.00
Feature #64: Upload files	Elsa Mahari	8.00
Feature #65: User Login	Benjamin Oertel	6.50
		1.00

Table B.2.: Overview By Member and Issue

Issue	Member	Total
Feature #66: Create a new case	Tien Nguyen	5.50
	Dominik Horb	2.00
Feature #67: Setting up PHPUnit	Dominik Horb	7.50
	Benjamin Oertel	2.00
	Elsa Mahari	2.00
Time #68: Meeting 05.12.11		4.00
	Dominik Horb	2.00
	Elsa Mahari	2.00
Time #69: Reading DocBook-XML	Dominik Horb	4.00
	Benjamin Oertel	1.00
Feature #70: Writing git tutorial		1.00
Time #71: Meeting Online 07.12.11	Benjamin Oertel	1.00
	Dominik Horb	16.00
	Benjamin Oertel	4.00
	Elsa Mahari	4.00
	Tien Nguyen	4.00
Time #72: Meeting preparation PHPDoc		2.00
Feature #73: Layout design	Dominik Horb	2.00
		10.50

Table B.2.: Overview By Member and Issue

Issue	Member	Total
Bug #74: protocoll meeting minuts 05.12.2011	Benjamin Oertel	10.50
	Elsa Mahari	0.50
Time #75: Creation of Preview area on server		3.00
Time #76: Meeting 12.12.2011	Dominik Horb	3.00
	Dominik Horb	7.50
	Benjamin Oertel	1.50
	Elsa Mahari	1.50
	Tien Nguyen	1.50
	Heiko Stammel	1.50
Time #77: create Wiki page for MeetingMinutes 15.11.2011		0.50
	Tien Nguyen	0.50
Feature #78: Setting up the unplagged-blog		17.00
Feature #79: Google Helper	Heiko Stammel	17.00
Feature #80: SIM-Text	Elsa Mahari	10.00
		10.00
Feature #81: Correct/edit OCR-scanned Text	Dominik Horb	3.00
	Tien Nguyen	10.00
	Benjamin Oertel	4.00

Table B.2.: Overview By Member and Issue

Issue	Member	Total
Feature #82: De-hyphenator	Benjamin Oertel	8.00
Feature #83: Non-stop word identification	Dominik Horb	8.00
Feature #84: Display text	Heiko Stammel	2.00
Feature #85: Standard OCR	Dominik Horb	6.00
Feature #86: Change file list view columns	Benjamin Oertel	3.50
Feature #87: Decrease line-height in tables.	Dominik Horb	1.00
Time #89: create Wiki page for MeetingMinutes 13.12.2011	Benjamin Oertel	2.50
Time #90: Prepare automatic preview area	Tien Nguyen	18.00
Time #101: Preparation of Database Mocks	Dominik Horb	12.00
Feature #102: Basics for responsive layout	Benjamin Oertel	6.00
	Benjamin Oertel	0.50
	Benjamin Oertel	0.50
	Benjamin Oertel	0.50
	Tien Nguyen	0.50
	Dominik Horb	11.25
	Dominik Horb	3.50
	Dominik Horb	3.50
		9.00

Table B.2.: Overview By Member and Issue

Issue	Member	Total
Time #103: Meeting 10.01.12	Dominik Horb	9.00
		3.00
	Dominik Horb	1.50
	Elsa Mahari	1.50
Bug #104: Skype Meeting 4.01.12	Elsa Mahari	1.00
		3.00
Bug #105: Switching Menu to Zend_Navigation	Dominik Horb	3.00
		10.00
Feature #106: Add Imagick to convert to different formats	Dominik Horb	6.00
	Benjamin Oertel	4.00
Bug #107: Meeting 2012-01-09		1.50
	Tien Nguyen	4.00
Time #108: Skype Meeting 12.01.12	Dominik Horb	2.00
	Elsa Mahari	2.00
		2.50
Time #109: Cleaning up directory structure	Dominik Horb	2.50
		13.50
Feature #110: Implementing Zend_Acl for user access control	Dominik Horb	11.50
	Benjamin Oertel	2.00
Feature #111: Improving interface workflow		8.00

Table B.2.: Overview By Member and Issue

Issue	Member	Total
Time #115: Debbie Meeting 18.01.12	Benjamin Oertel	8.00
	Dominik Horb	6.00
	Elsa Mahari	3.00
Feature #116: Logger	Benjamin Oertel	3.00
	Dominik Horb	4.00
Feature #117: Create Paging mechanism	Benjamin Oertel	2.00
Bug #118: create Wiki page for MeetingMinutes 18.01.2012	Benjamin Oertel	2.00
Bug #119: Fixing several css issues	Tien Nguyen	6.00
Feature #120: Beautifying forms, tables and buttons	Benjamin Oertel	0.50
Time #122: 23.01.12 Meeting	Benjamin Oertel	0.50
	Dominik Horb	5.00
	Elsa Mahari	5.00
Time #123: 30.01.12 Meeting	Benjamin Oertel	5.00
	Dominik Horb	6.00
	Benjamin Oertel	2.00

Table B.2.: Overview By Member and Issue

Issue	Member	Total
Time #128: LaTeX preparation	Elsa Mahari	2.00
Time #142: License research and inclusion	Dominik Horb	2.00
Bug #143: Fixing tests after last refactorings	Dominik Horb	6.00
Time #144: Reading Zend Doku for Bootstrap Plugins	Dominik Horb	1.50
Time #145: Meeting 06.02.12	Dominik Horb	4.00
Time #146: Basic TOC and split up of LaTeX document	Dominik Horb	2.00
Feature #147: Preface Developers Manual	Dominik Horb	1.50
Time #148: Documentation - Chapter Git	Dominik Horb	7.00
Time #149: Meeting 04.03.12	Benjamin Oertel	4.50
Time #150: Writing chapter about Zend and Doctrine Framework	Benjamin Oertel	5.00
	Benjamin Oertel	5.00

Table B.2.: Overview By Member and Issue

Issue	Member	Total
Time #151: Writing chapter about User Interface / User Experience	Benjamin Oertel	6.00
Time #152: Adding mockups to documentation	Benjamin Oertel	2.00
Time #153: Team Meeting – 2012-03-10	Benjamin Oertel	2.00
Time #154: Reading in latex documentation	Benjamin Oertel	1.50
Feature #155: Improvements of error detection in tesseract parser	Benjamin Oertel	3.00
Feature #156: Implementing Webservice of PlagAware	Benjamin Oertel	2.00
Bug #157: Zend Acl – allowing/denying access to actions, not only controllers	Benjamin Oertel	1.00
Total		548.35

Table B.3.: Overview By Sprints

Version	2011-10	2011-11	2011-12	2012-1	2012-2	2012-3	Total
none	7.00		28.75	47.00	33.00	19.30	135.05
Sprint 1 (2011-10-01 - 2011-11-14)	81.80	67.50					149.30
Product Back-Log	2.50	0.50	3.00				6.00
Sprint 2 (2011-11-15 to 2011-12-13)		33.00	72.00				105.00
Sprint 3 (2011-12-14 to 2012-01-17)			21.00	63.00			84.00
Sprint 4 (2012-01-18 to 2012-02-01)				36.00	2.00		38.00
Sprint 5 (2012-02-02 to 2012-23-02)					4.50	26.50	31.00
Total	91.30	101.00	124.75	146.00	39.50	45.80	548.35

C. Mockups

C.1. Hand-Drawn

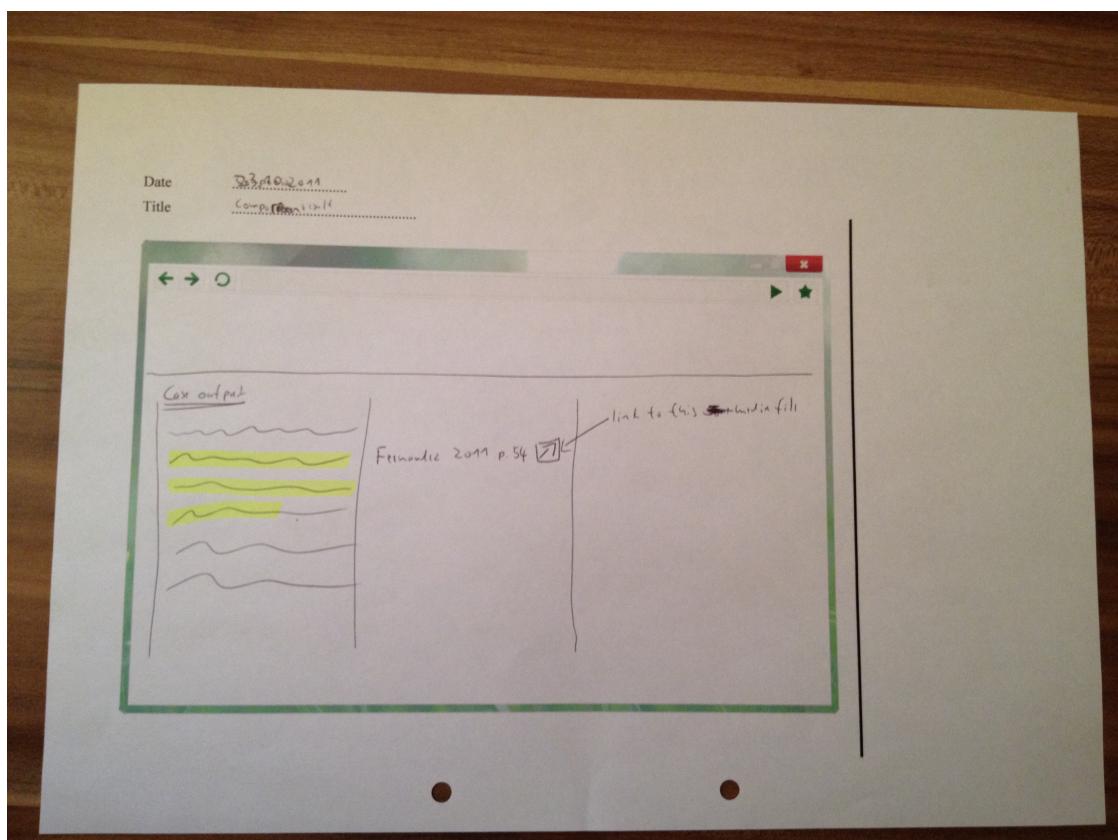


Figure C.1.: Mockup – Compare results – digitalized

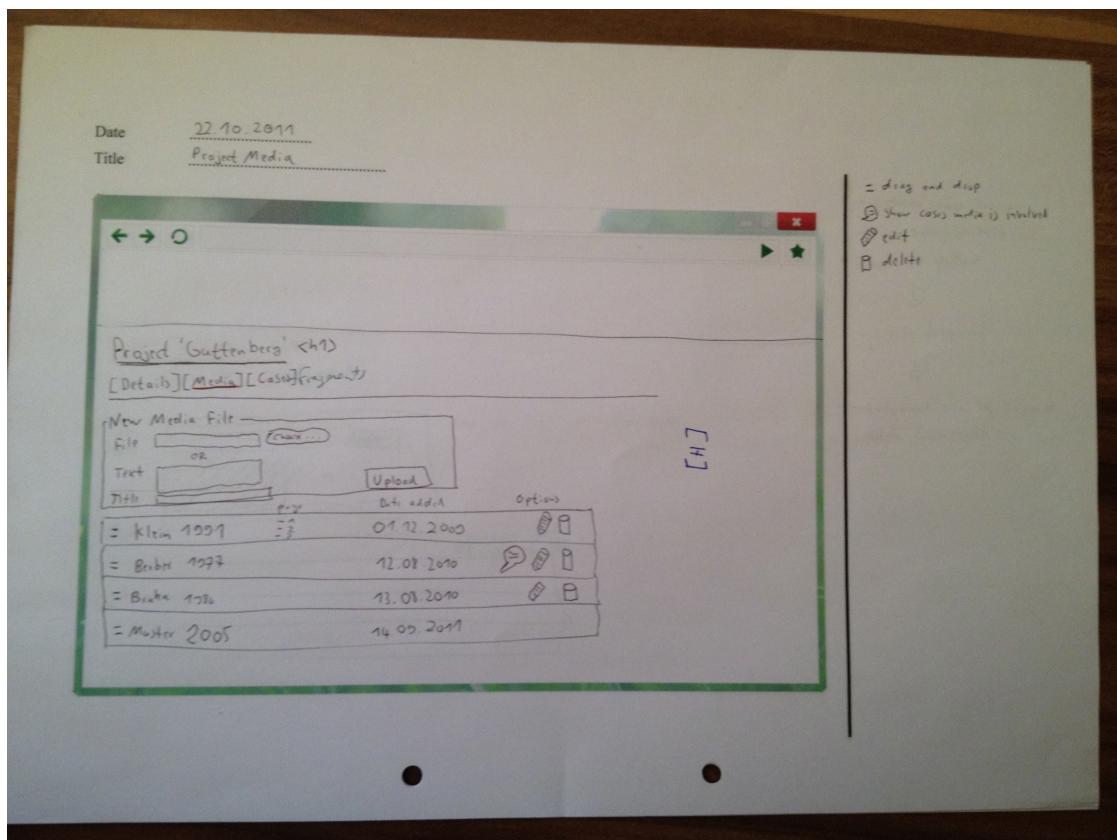


Figure C.2.: Mockup – Media list – digitalized

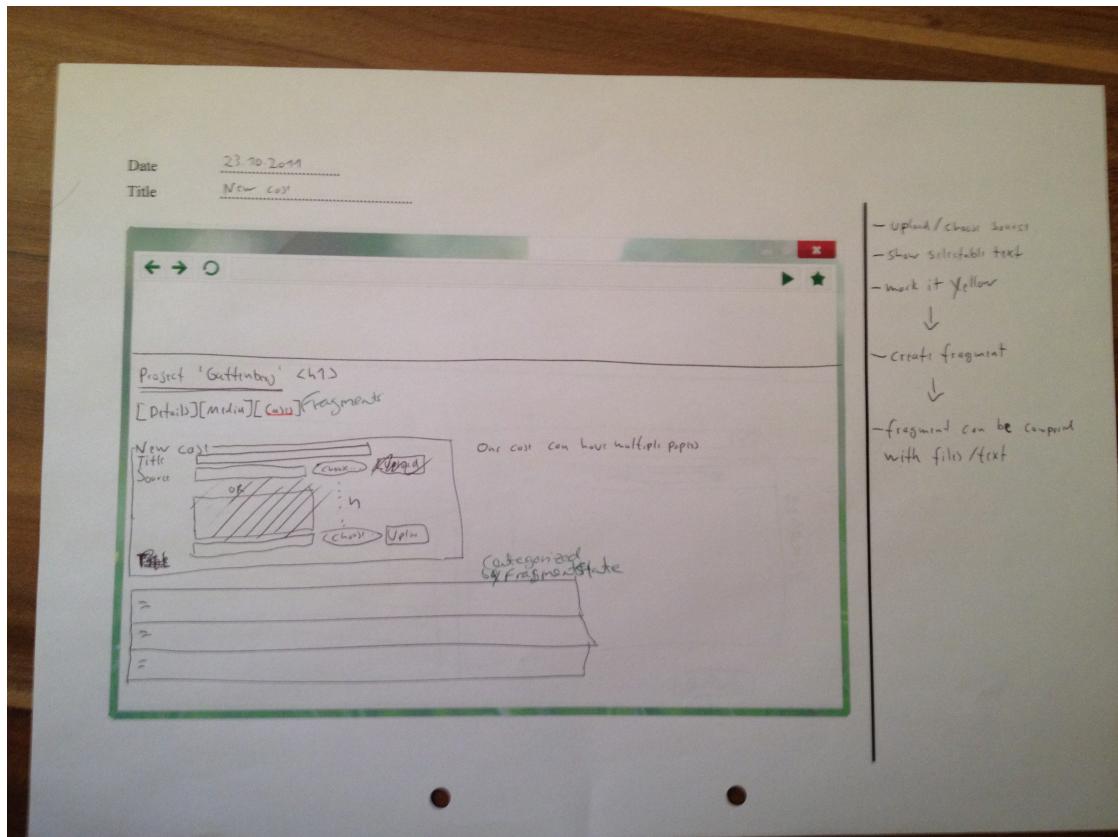


Figure C.3.: Mockup – New case – digitalized

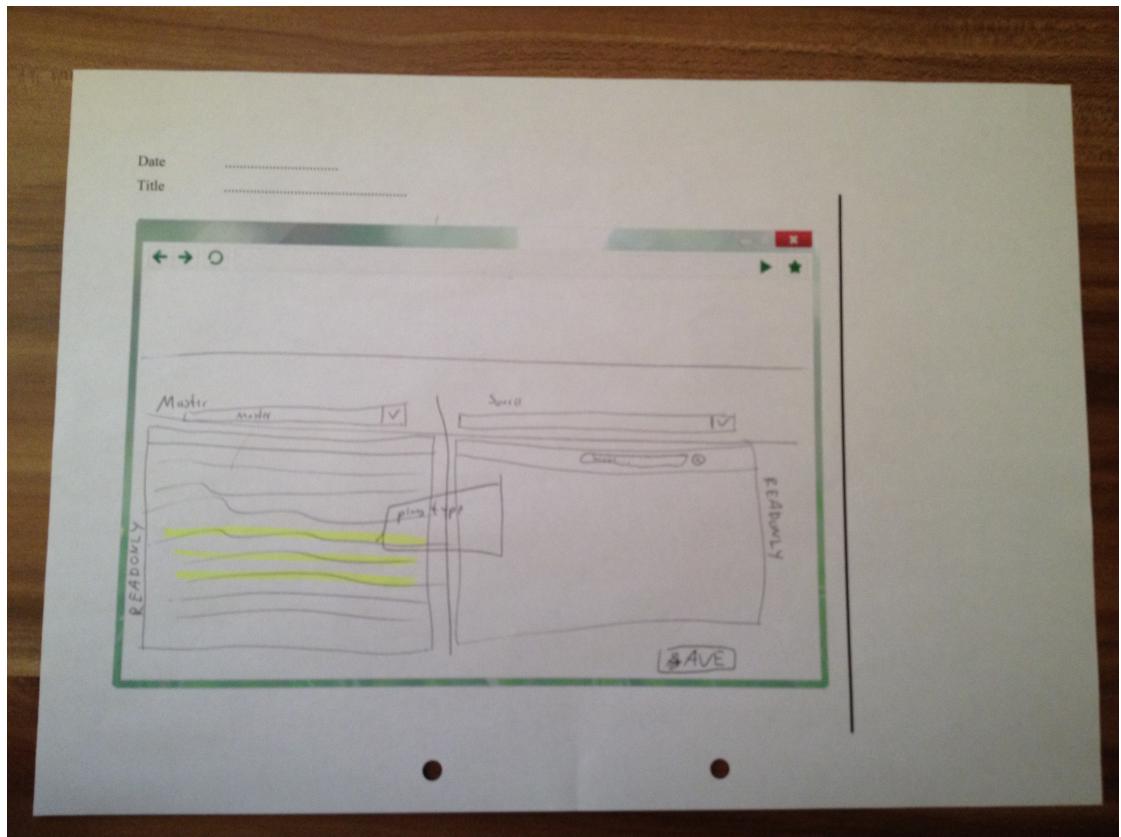


Figure C.4.: Mockup – New fragment – digitalized

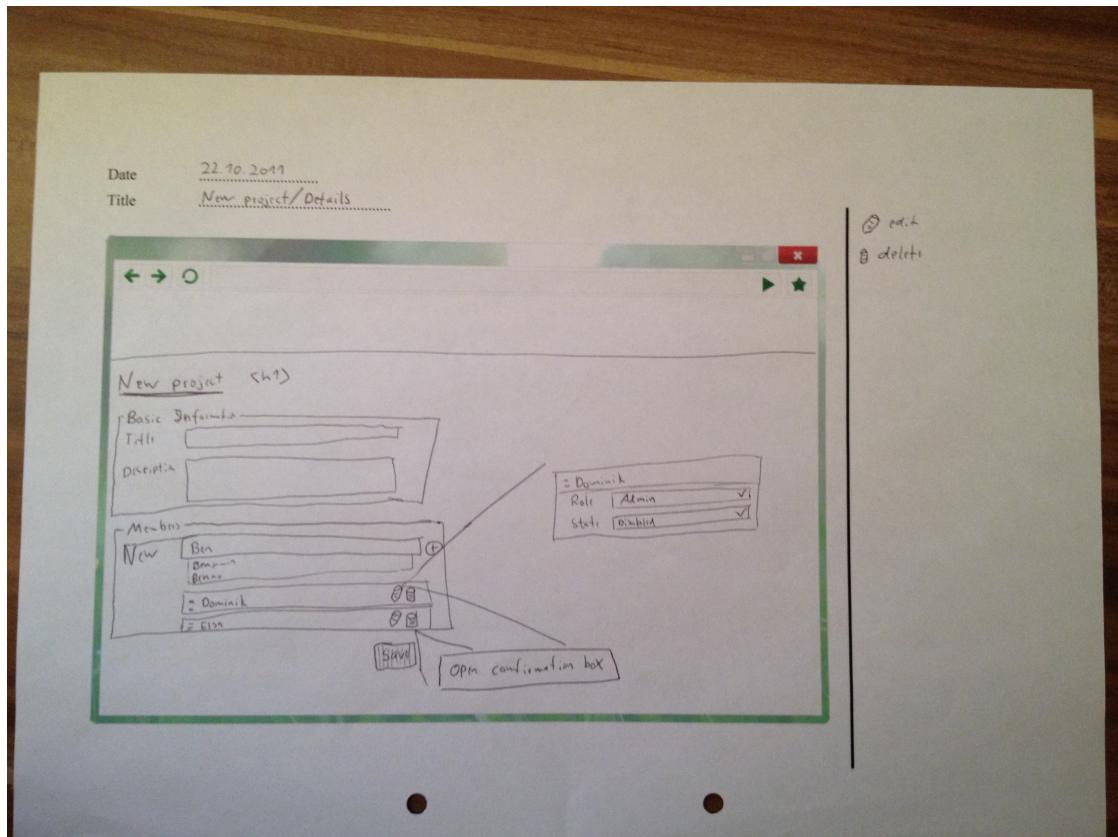


Figure C.5.: Mockup – New project – digitalized

C.2. Digitalized

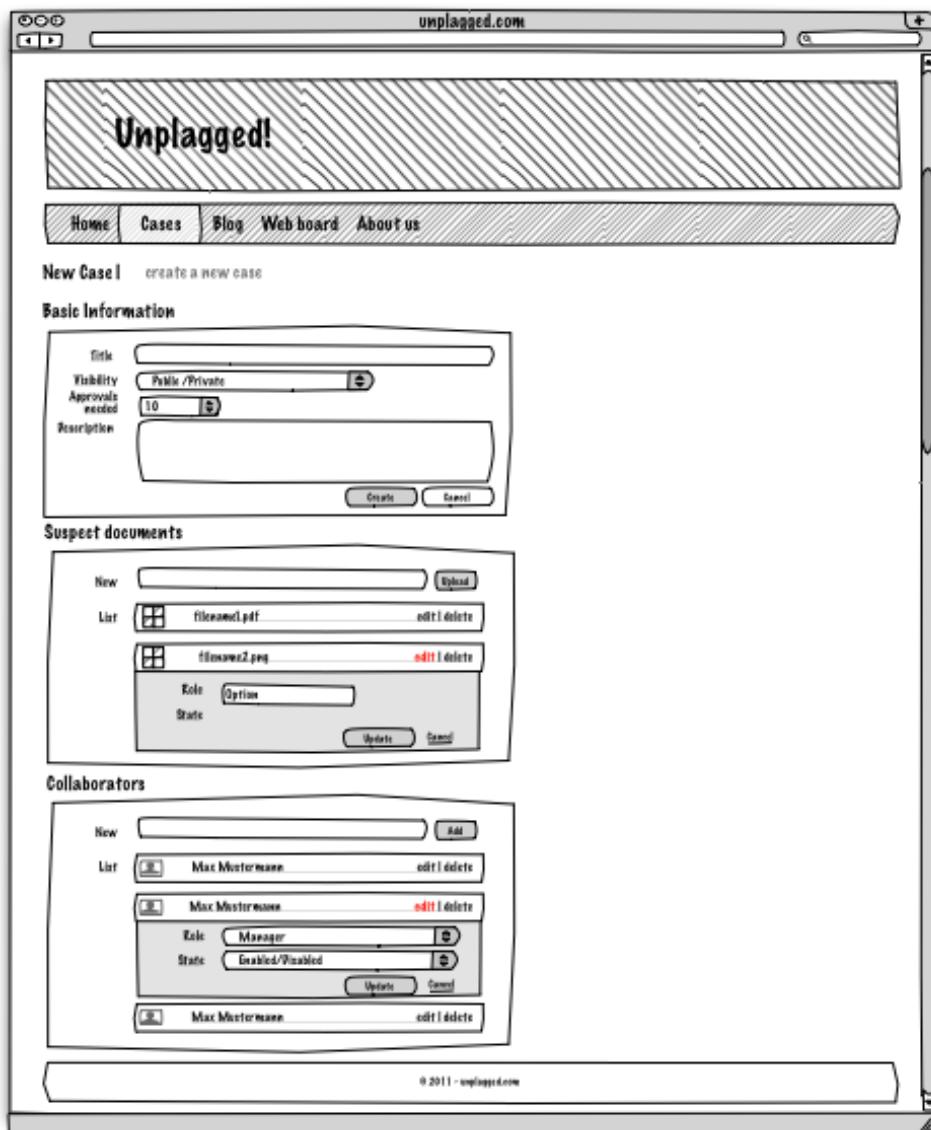


Figure C.6.: Mockup – New case – digitalized

The screenshot shows a web-based application for managing document fragments. At the top, a header bar contains the URL "unplugged.com" and a navigation menu with links for "Home", "Cases", "Blog", "Web board", and "About us".

The main content area is titled "Fragments | all fragments in case 'Gutenberg'". It features a "New Fragment" form on the left and a "Fragments by page" table on the right.

New Fragment Form:

- Document: Gutenberg.pdf
- Suspected source: Kafka
- Buttons: Create, Cancel

A yellow callout box next to the "Create" button provides the tip: "clicking create opens a new page with suspect document and search pre-selected".

Fragments by page Table:

Page	Content	Fragments	Latest activity	Options
(+) Page 1		0 2 1 7 1 5	2011-08-25	
(+) Page 2		0 2 1 7 1 5	2011-08-25	
(+) Page 3		0 1 1 1 2 0	2011-08-25	
Line 19-25 Kompletthyperlink		approved	2011-08-25	edit show approve
Line 29-44 Kompletthyperlink		waiting approval	2011-08-28	edit show approve
Line 55-55 Kompletthyperlink		waiting approval	2011-08-25	edit show approve
(+) Page 5		0 2 1 7 1 5	2011-08-25	
(+) Page 6		0 2 1 7 1 5	2011-08-25	

At the bottom of the page, a footer bar contains the text "© 2011 - unplugged.com".

Figure C.7.: Mockup – List fragments – digitalized

The mockup displays the Unplagged! website interface for creating a new fragment. The top navigation bar includes links for Home, Cases, Blog, Web-board, and About us. Below the navigation is a section titled "New Fragment | create a new fragment in case 'Utenberg'".

Suspect document: A text editor window showing a document named "Utenberg.pdf" on page 1. The text discusses Josef K. being questioned about his actions after a fight. A blue callout bubble highlights the word "mark as plagiarism" and the "type: lorem ipsum" field.

Supposed source: A text editor window showing a document by Kafka on page 8. The text discusses a dog's behavior and a character named Samza. A yellow callout bubble lists detection rules: "comments possible on each line, different colors per user" and "text colors different per plagiarism type". A blue callout bubble points to a "commenter show comment" link.

Comment: A form for adding a comment. It includes fields for "Note" (with a large text area) and "Visibility" (set to "Private/Public/Group"). A yellow callout bubble lists comment states: "new (created but not ready for verification)", "in-progress (can be edited)", "resolved", "feedback", "closed (5 approvals)", and "rejected (5 declined)".

The footer of the application shows the URL "0.2011 - unplagged.com".

Figure C.8.: Mockup – New fragment – digitalized

unplugged.com

Unplugged!

Home Cases Blog Web board About us

Show Fragment 1 fragment in case "Gutenberg"

Suspect document

File: document1.pdf
Page: 1
Line: 34-40

...Als Gregor Samza eines Morgens aus unruhigen Träumen erwachte, fand er sich in seinem Bett zu einem ungeheueren Ungeziefer verwandelt. Und es war ihnen wie eine Bestätigung ihrer neuen Träume und guten Absichten, ...
2 words below and 2 above ...

Supposed source

File: document1.pdf
Page: 1
Line: 34-40

emand musste Josef K. verleumdet haben, denn ohne dass er etwas Böses getan hätte, würde er eines Morgens verhaftet. »Wie ein Handl« sagte er, es war, als sollte die Scham ihn überleben. Als Gregor Samza eines Morgens aus unruhigen Träumen erwachte, fand er sich in seinem Bett zu einem ungeheueren Ungeziefer verwandelt. Und es war ihnen wie eine Bestätigung ihrer neuen Träume und guten Absichten, als am Ziele ihrer Fahrt die Tochter als erste sich erhob und ihren jungen Körper dehnte. »Es ist ein eigenmächtiger Apparat«, sagte der Offizier zu dem Fassungsverlustenden und überblieb mit einem gewissmaßen bewundernden Blick den ihm doch wohlbekannten Apparat.

(+) expand whole page (-) collapse whole page

commenter show facebook-like list with all selected users, comment and date

Ratings

appreciate : 1 rejects : 1

State: Approve

comment ratings that are not the most recent ones of a user are shown greyed out

Comment	User	Date	Rating
guten Absichten, als am Ziele ihrer Fahrt die Tochter als erste sich erhob und ihren jungen Körper dehnte. »Es ist ein eigenmächtiger Apparat«, sagte der Offizier zu dem Fassungsverlustenden und überblieb mit einem gewissmaßen bewundernden Blick den ihm doch wohlbekannten Apparat.	Max Mustermann	2011-11-05	no
Re: guten Absichten, als am Ziele ihrer Fahrt die Tochter als erste sich erhob und ihren jungen Körper dehnte. »Es ist ein eigenmächtiger Apparat«, sagte der Offizier zu dem Fassungsverlustenden und überblieb mit einem gewissmaßen bewundernden Blick den ihm doch wohlbekannten Apparat.	Emmy Watson	2011-11-06	no
Re: guten Absichten, als am Ziele ihrer Fahrt die Tochter als erste sich erhob und ihren jungen Körper dehnte. »Es ist ein eigenmächtiger Apparat«, sagte der Offizier zu dem Fassungsverlustenden und überblieb mit einem gewissmaßen bewundernden Blick den ihm doch wohlbekannten Apparat.	Max Mustermann	2011-11-07	no
guten Absichten, als am Ziele ihrer Fahrt die Tochter als erste sich erhob und ihren jungen Körper dehnte. »Es ist ein eigenmächtiger Apparat«, sagte der Offizier zu dem Fassungsverlustenden und überblieb mit einem gewissmaßen bewundernden Blick den ihm doch wohlbekannten Apparat.	Emmy Watson	2011-11-08	no
guten Absichten, als am Ziele ihrer Fahrt die Tochter als erste sich erhob und ihren jungen Körper dehnte. »Es ist ein eigenmächtiger Apparat«, sagte der Offizier zu dem Fassungsverlustenden und überblieb mit einem gewissmaßen bewundernden Blick den ihm doch wohlbekannten Apparat.	Max Mustermann	2011-10-27	yes

Figure C.9.: Mockup – Show fragment for approval – digitalized

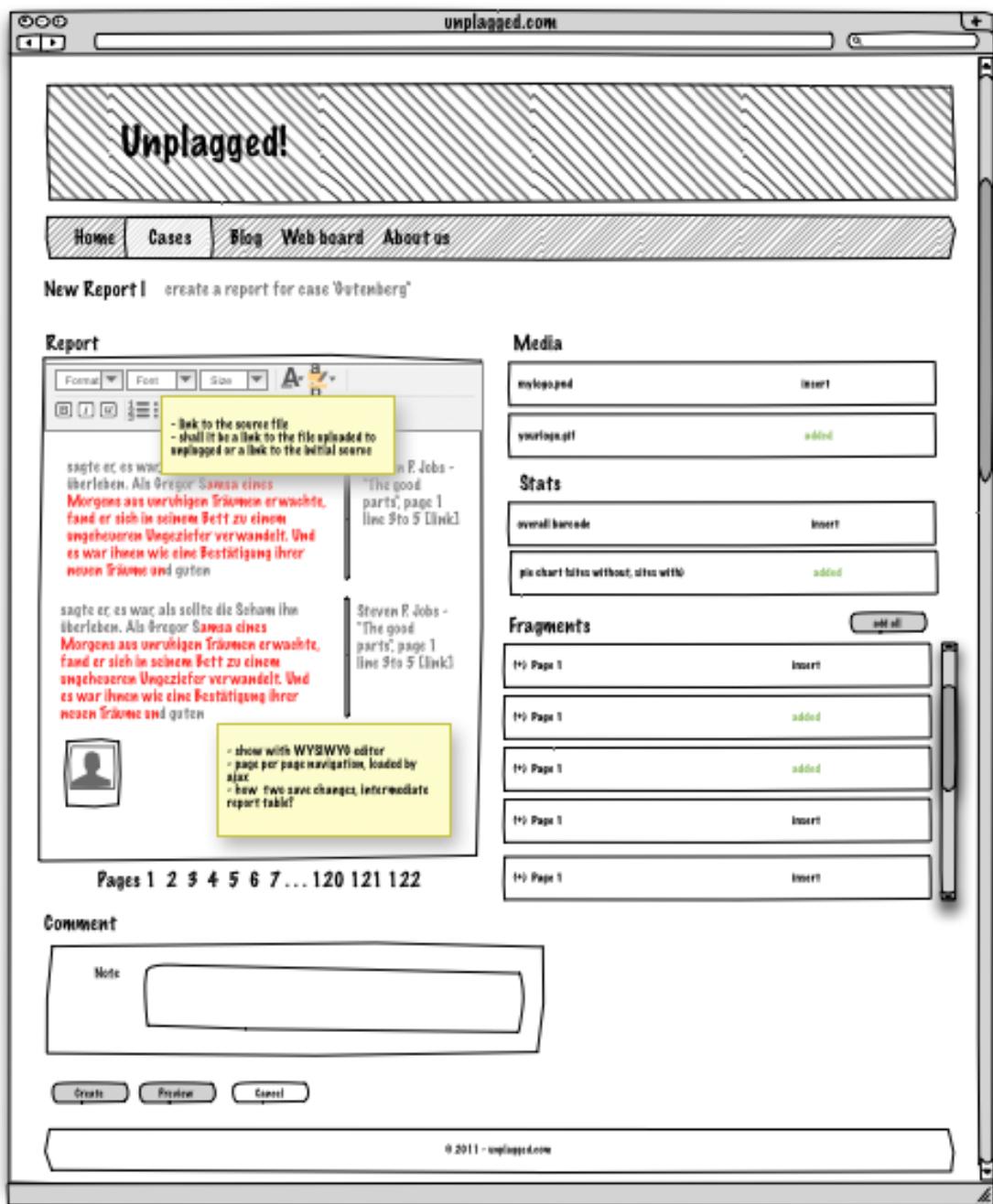


Figure C.10.: Mockup – New report – digitalized