Beuth Hochschule für Technik Berlin
Medieninformatik Online
Bachelorstudiengang am Fachbereich VI

Dokumentation Praxisprojekt

Sprachspezifische Produktpflege (Rosetta-App)

Salim Oussayfi (797754)
Antoninusstraße 64
60439 Frankfurt am Main
oussayfi@gmail.com

eingereicht am 3. Juli 2017

Gutachter/Betreuer:
Prof. Dr. Stefan Edlich
Beuth Hochschule für Technik Berlin
FB VI / Labor Online Learning
Luxemburger Str. 10
13353 Berlin

GLIEDERUNG/INHALT

EINLEITUNG

Die vorliegenden Seiten dienen dazu, mein Praxisprojekt aus dem Sommersemester 2017 zu dokumentieren und dem Leser einen Einblick in alle Bereiche des bis heute abgeschlossenen Software-Entwicklungsprozesses zu geben.

Es galt, ein System zu entwickeln, das in der Lage ist, einen großen Pool produktspezifischer Bezeichnungen in mehreren Sprachen zu katalogisieren und einen dynamischen Zugriff auf die Daten zu gewährleisten. Es handelt sich demzufolge um einen digitalen Glossar, in dem große Mengen an Daten bezüglich Schreibweise und der jeweiligen Übersetzung abgespeichert sind. Mithilfe des Systems sollte der tägliche Workflow des Auftraggebers optimiert und zeitgleich die Produktionskosten gesenkt werden.

Die Software sehe ich in seiner Funktion und Nutzen als Analogie zu dem „Stein von Rosette". Wolfgang Schenkel formuliert in „Die Entzifferung der Hieroglyphen und Karl Richard Lepsius" folgende Passage:

> Im Juli 1799 wurde bei Schanzarbeiten der französischen Armee nahe der Mündung eines der Nilarme in das Mittelmeer, bei dem Dorf ar Rashid, der berühmte »Stein von Rosette« gefunden. Auf diesem »Stein« steht in der Form eines Dekrets der Beschluss einer Priestersynode des Jahres 196 v. Chr., den jugendlichen König Ptolemäus V. Epiphanes für seine dem Volk gewährten Vergünstigungen in verschiedenster Weise zu ehren.

Weiter unten heißt es schließlich:

> Niedergeschrieben worden ist der Text dann tatsächlich auf Ägyptisch in zwei Versionen, zuerst in traditionellem Ägyptisch mit Monumental-Hieroglyphen, danach in einem jüngeren, demotischen Ägyptisch mit demotischer Kursive, schließlich in griechischer Sprache mit griechischen Buchstaben.
>
> Prof. Dr. phil. Wolfgang Schenkel, Die Entzifferung der Hieroglyphen und Karl Richard Lepsius,
> [online] http://archiv.ub.uni-heidelberg.de/propylaeumdok/2886/1/Schenkel_Entzifferung_2012.pdf [14.5.2017]

Mithilfe des Steins von Rosette bestand die Möglichkeit, die Hieroglyphen zu entschlüsseln, da sinngemäße Texte sowohl in Hieroglyphen als auch in bekannten griechischen Buchstaben auf eben diesen Stein geschrieben stehen.

Aus diesem Grund habe ich die Anwendung „Rosetta-App" genannt. Der Zusatz „App" (Applikation) ist dadurch begründet, dass dieser „neuzeitige Stein" ausschließlich in einer digitalen Form besteht und dynamisch genutzt und beliebig erweitert werden kann.

Diese Dokumentation, welche die Entwicklung der Rosetta-App veranschaulichen soll, ist chronologisch aufgebaut. Ich beginne mit den allgemeinen Voraussetzungen für das akademische Praxisprojekt und leite anschließend Schritt für Schritt durch die einzelnen Phasen des Entwicklungsprozesses: von der Definition der Unternehmensziele bis zur Implementierung der fertigen Anwendung.Im Anschluss daran beende ich die Dokumentation mit einem Fazit und möglicher Ausblicke zur Weiterentwicklung der Software.

Zum Abschluss dieser Einleitung möchte ich noch auf die beiden Parteien eingehen, die in dieses Projekt wesentlich involviert sind:

Das ist zum Einen die Typodrom WERBEAGENTUR GmbH, Radilostraße 43, 60489 Frankfurt am Main, im weiteren Verlauf dieser Dokumentation als Typodrom bezeichnet. Typodrom ist der Auftraggeber des Projekts und gleichzeitig mein Arbeitgeber.

Die zweite Partei wird gebildet durch die Adam Opel GmbH, Bahnhofsplatz 1, 65423 Rüsselsheim am Main – im weiteren Verlauf dieser Dokumentation als Opel bezeichnet. Opel ist nicht der direkter Auftraggeber des Projekts sondern das System dient zur Optimierung des Workflows auf Seiten von Typodrom bei der Bearbeitung von Aufträgen durch Opel.

1. AUFGABENSTELLUNG

– Ziele und Anforderungen des Praxisprojekts

In diesem ersten Abschnitt gehe ich auf die Ziele und auf die Anforderungen des Praxisprojekts ein.

Im Allgemeinen dient das Praxisprojekt dazu, das im Studium Erlernte sehr praxisnah anzuwenden. Die Technische Hochschule Köln definiert die Ziele des Praxisprojekts wie folgt:

> Im Praxisprojekt sollen die Studierenden Methoden und Techniken, die sie im Studium erlernt haben, in einem realitätsnahen Projekt weitgehend selbstständig anwenden.
>
> Vgl. Technische Hochschule Köln, Praxisprojekte, [online] http://www.f10.th-koeln.de/campus/institute/informatik/studium/praxisprojekte/ [21.5.2017]

Das Verfolgen dieser Ziele ist sicherlich von Studiengang zu Studiengang auf unterschiedliche Weise zu erreichen. Speziell im Studium der Medieninformatik liegt es nahe, ein Projekt zu wählen, in dem eine digitale Anwendung umzusetzen ist, bzw. ein neues Software-System implementiert wird.
Allein mit dem Programmieren irgend einer Anwendung ist dieses Ziel nicht erfüllt, es soll ein weites Spektrum an Techniken bei der Entwicklung angewendet werden, d. h. es sollten Bereiche aus der Analyse, Entwicklung, Implementierung und der Medien berücksichtigt werden, wie es Prof. Dr. Edlich der Beuth-Hochschule für Technik Berlin im „Leitfaden für das Praxisprojekt (Online)“ erläutert.

Vgl. Prof. Dr. Stefan Edlich, Leitfaden für das Praxisprojekt (Online), [online] https://prof.beuth-hochschule.de/edlich/praxisprojekt-online/ [21.5.2017]

Neben der genannten Ziele und der Anforderungen wird auch ein zeitlicher Aufwand für die Bearbeitung des Projektes vorausgesetzt. Dieser wird im oben zitierten Leitfaden mit einer Dauer von mindestens 450 Stunden vorgegeben.

– Benefit Student

Bei der Beschreibung des Benefits für den Studierenden[1] möchte ich meine Ausführung einzig auf mich und meine Eindrücke zum Praxisprojekt beziehen. Ich bin mir sicher, dass sich meine Ansichten mit denen vieler Kommilitonen decken.

Das Praxisprojekt habe ich so erlebt, dass ich viele Erfahrungen gewonnen habe, sei es in der Projektarbeit im Allgemeinen oder in der Programmierung im Speziellen.

Ich habe das komplette Projekt von Beginn an geplant, begleitet und geleitet.

In meinem beruflichen Alltag habe ich nicht viele Schnittmengen mit den Themengebieten aus dem Studienalltag. Gerade deswegen hat sich die Projektarbeit als eine wertvolle Erfahrung herausgestellt, wenn es darum geht, das im Studium Gelernte in einem konkreten Projekt umsetzen zu können.

Ich habe viele Techniken und Methoden aus meinem bisherigen Studium anwenden können. Anhand dieser Dokumentation möchte ich diese im Einzelnen veranschaulichen.

Sicherlich gab es viele interessante und spannende Projekte in meinem bisherigen Studienverlauf, die sowohl im Team als auch als Einzelarbeit zu bearbeiten waren. Im Umfang und Anspruch an der nebenläufigen Organisation des Projekts stellt e sich die hier beschriebene Arbeit als eine bedeutende Erfahrung für mich heraus.

[1] Ausschließlich zum Zwecke der besseren Lesbarkeit verzichte ich in diesem Dokument auf die unterschiedliche Geschlechterschreibweise.

2. ZIELSETZUNG/UNTERNEHMENSZIELE

– Zielsetzung

In diesem Abschnitt beschreibe ich die Zielsetzung und die Motivation Typodroms für das in der vorliegenden Dokumentation beschriebene System.

Das primäre Ziel war es, vorhandenes Wissen dynamisch an nur einem zentralen Ort zu speichern und unternehmensweit zur Verfügung zu stellen.

Ähnlich wie bei einem webbasierten Wissensmanagement-System wie z. B. Confluence der Firma Atlassian (https://de.atlassian.com/software/confluence) soll auch bei Rosetta-App (http://rosetta-app.de) das kollaborative Arbeiten ermöglicht und gefördert werden.

Jeder Mitarbeiter soll bei Bedarf die zentral abgelegten Daten für seine Zwecke nutzen und gleichzeitig den Datenbestand erweitern, bzw. aktualisieren können.

Das System soll einfach zu handhaben und durch eine intuitive Nutzung zu bedienen sein. Der Funktionsumfang soll ausschließlich das hier dokumentierte Projekt behandeln.

Es galt, durch genügend Transparenz immer ersichtlich und nachvollziehbar sein zu lassen, welcher Anwender einzelne Datensätze kreiert, manipuliert, kommentiert oder gelöscht hat.

Zu jedem Beitrag soll demzufolge sowohl der Urheber des Eintrags angezeigt werden als auch jeder Nutzer, der den Datensatz geändert oder gelöscht hat.

Durch eine Kommentarfunktion soll es möglich sein, die Relevanz mancher Einträge gegenüber anderen zu differenzieren, bzw. weiterführende Informationen zu jedem Eintrag hinterlegen zu können.

Zusätzlich soll jede Manipulation der Daten mit dem jeweiligen Datum und der Uhrzeit gespeichert und im System angezeigt werden.

Die zentrale Datenspeicherung hat zur Folge, dass Redundanzen in der Datenbeschaffung und Speicherung reduziert bzw. vermieden werden, da bereits vorhandenes Wissen für jeden Nutzer einsehbar ist und demzufolge nicht erneut beschafft werden muss.

– Benefit Unternehmen

Für Typodrom entsteht mithilfe des Systems ein Mehrwert unter anderem dadurch, dass der Workflow wesentlich optimiert wird.

Arbeits-Unterbrechungen werden minimiert, da alle nötigen Informationen zur Verfügung stehen und promt genutzt werden können.

Jeder Mitarbeiter kann für sich selbst prüfen, ob eine gewünschte Übersetzung bereits existiert, bevor er sie in Auftrag gibt oder geben lässt.

Somit werden unnötige Kosten vermieden, da die zu übersetzenden Produktbeschreibungen nur einmal übersetzt werden müssen bzw. übersetzt worden sind.

Durch die Kommentar-Funktion können zudem sich ähnelnde Übersetzungen leicht differenziert bzw. priorisiert werden.

3. ANALYSE–PHASE

– Beschreibung der Ausgangssituation/Ist–Analyse

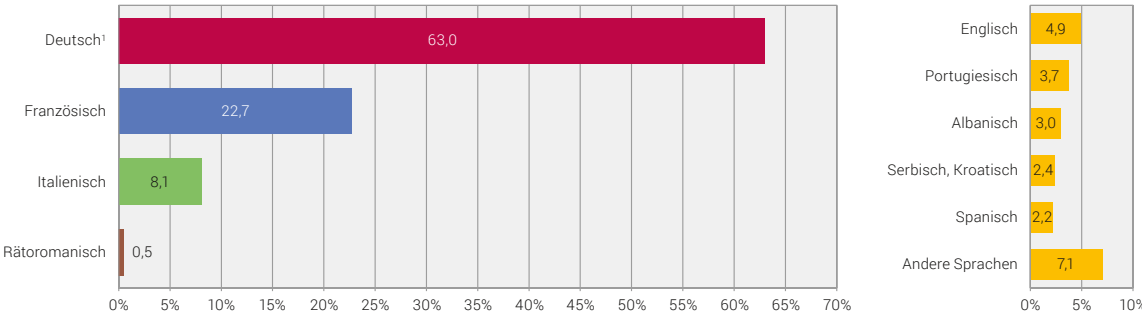
Typodrom betreut in der Funktion einer Tagesgeschäftsagentur den Kunden Opel.

Für Opel generiert Typodrom Printmedien wie z. B. Kataloge, Anzeigen, POS-Materialien und weitere Werbematerialien[2] für den gesamten europäischen Raum kreiert und produziert werden.

Für das Praxisprojekt habe ich mich zunächst auf den Schweizer Markt konzentriert. Im Kapitel Ausblicke gehe ich auf eine mögliche Skalierbarkeit des Systems ein.

In der Schweiz sind neben Deutsch weitere Landessprachen vertreten, die Sprachen Französisch, Italienisch und Rätoromanisch.

Die folgende Grafik des Bundesamt für Statistik (veröffentlicht am 27.3.2017) stellt aus der Strukturerhebung des Jahres 2015 die Gewichtung der gesprochenen Sprachen für den Zeitraum 2015 in der Schweiz dar.



[1] oder Schweizerdeutsch

Ständige Wohnbevölkerung, die in Privathaushalten lebt. Die Befragten konnten mehrere Sprachen angeben.

Quelle: BFS – Strukturerhebung (SE)

Abb. 1: Grafik zeigt die Hauptsprachen in der Schweiz.

Vgl. Bundesamt für Statistiken (Schweiz), Als Hauptsprachen genannte Sprachen (2015) [online] https://www.bfs.admin.ch/bfs/de/home/statistiken/bevoelkerung/sprachen-religionen/sprachen.html [27.5.2017]

Opel führt in der Schweiz alle Publikationen in den drei Sprachen Deutsch, Französisch und Italienisch und vertreibt diese je nach Sprachgewichtung im jeweiligen Kanton.

Bei der initialen Entwicklung des Systems habe ich die Opel Preislisten berücksichtigt. Auch hier ist man zukünftig nicht auf das Medium „Preisliste“ beschränkt und kann analog zur Wahl der Sprachen auch im Bereich des Mediums und unabhängig eines solchen das System beliebig skalieren.

Da ich, wie zuvor erwähnt, das Projekt unter der Prämisse der Preislisten gestartet habe, werde ich diese Dokumentation auch an diesen Preislisten ausgerichtet weiterführen.

Für jeden Fahrzeugtyp wird eine 16- bis 28-seitige Preisliste erstellt, jeweils in zwei Sprachversionen, d. h. es liegt am Ende jede Preisliste in Deutsch/Französisch und analog dazu in Deutsch/Italienisch vor. Je nach Modell-Jahr führt Opel 18 bis 22 Fahrzeugtypen.

In den Preislisten werden Preise und Daten zu den verfügbaren Motoren, der Serien- und Sonderausstattung und zu jeglichem erwerbbaren Zubehör kommuniziert. Kurz gefasst: Die Preislisten befassen sich mit allen Fahrzeugteilen, die nicht dem Auto als Ganzes entsprechen.

[2] Von einer Auflistung des gesamten Spektrums der verkaufsfördernden Publikationen, die Typodrom im Auftrag von Opel entwickelt, sehe ich an dieser Stelle ab, da diese Information nicht zur Beschreibung meines Projekts erforderlich ist

Bei Typodrom werden die Preislisten abteilungsübergreifend im Tagesgeschäft erstellt. Diese Abteilungen sind das Projektmanagement bzw. die Kundenberatung, die Kreation zusammen mit der Mediengestaltung und das Lektorat.

In der folgenden Grafik ist dargestellt, welche Abteilungen in den Workflow involviert und wie sie miteinander verknüpft sind.

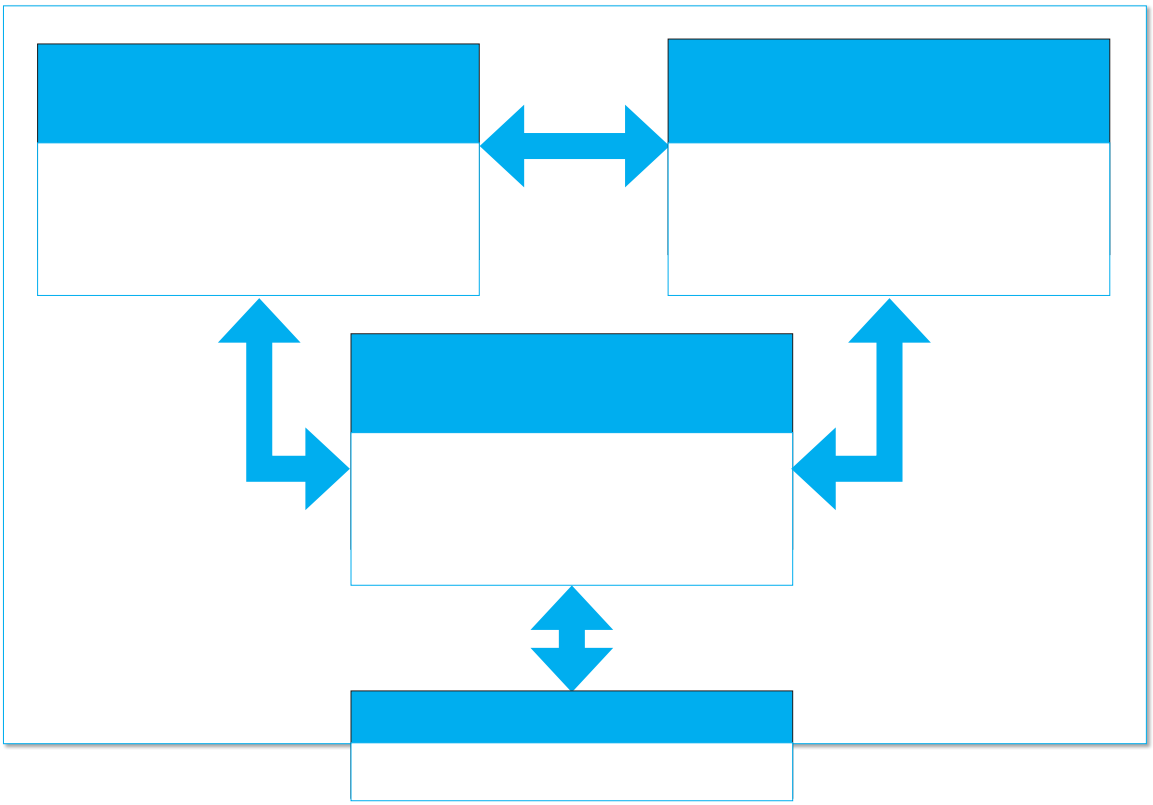


Abb. 2: Grafik mit Veranschaulichung des agenturinternen WorkFlows.

Durch die Zusammenarbeit dieser Entitäten entstehen periodisch neue Preislisten und mit jeder neuen Preisliste werden auch neue Produkte vermarktet bzw. wird eine Fahrzeugausstattung mit dem Zubehör aus einer anderen Flotte erweitert. Diese Produkte sind sehr fachspezifisch und beruhen teilweise auf sprachgebundenen Eigennamen und sprachspezifischen Wortschöpfungen.

Bei der Erstellung neuer Publikationen in den drei Sprachen wird regelmäßig auf externe Ressourcen zugegriffen, um die Produktbeschreibungen in die jeweils benötigte Sprache zu übersetzen. Obwohl diese Daten zum Teil bereits verfügbar aber nicht optimal abrufbar sind. Dadurch entstehen unnötige Kosten, zudem ergibt sich hieraus ein erheblicher zeitlicher Mehraufwand.

– Vision und Systemidee/Soll–Konzept

Es soll ein Software-System entwickelt werden, das alle Produktbezeichnungen von Opel beinhaltet und jeweils die korrekten Übersetzungen und Schreibweisen enthält.

Einzelne Übersetzungen können durch eine Kommentar-Funktion leicht verifiziert und somit von ähnlichen, obsoleten Schreibweisen differenziert werden.

Zudem kann eine Referenz auf die verwendete Publikation gegeben werden mit all ihren Attributen wie z. B. Erscheinungszeitpunkt und einer Verlinkung zu dem eigentlichen Dokument. Bei dem hier behandelten Praxisprojekt wird zu jedem Eintrag automatisch ein Link generiert, über den sich die jeweilige Preisliste als PDF aufrufen lässt.

Alle Einträge sind im View editierbar und können auf einfache Art und Weise in die Zwischenablage kopiert bzw. kommentiert oder gefiltert werden. Durch die Möglichkeit, den Eintrag in die Zwischenablage kopieren zu können, wird das System auf komfortable Art und Weise in den Workflow integriert.

Das System kann von jedem registrierten Mitarbeiter für seine Arbeit genutzt und je nach Berechtigung auch gepflegt werden.

Das System ist skalierbar, d. h., es ist möglich, beliebig viele Sprachen zu integrieren, desweiteren ist das System autark, was das zu behandelnde Medium betrifft.

4. VORSTUDIE UND MARKTANALYSE

– Umweltanalyse

Sicherlich besteht heutzutage die Möglichkeit, sich mithilfe von online zur Verfügung stehenden Übersetzungstools ganze Textabschnitte in jede beliebige Sprache übersetzen zu lassen und damit seine Anliegen dem Empfänger verständlich zu machen. Als Beispiel seien an dieser Stelle der Google-Übersetzer (https://translate.google.com), Linguee (http://www.linguee.de/) und Leo (http://www.leo.org/) genannt. Dies gilt allerdings nur in einem Bereich, in dem die Anforderungen ausschließlich der Weitergabe von Informationen dienen und eine zuverlässige Interpretation vorausgesetzt werden darf.

Sobald auf professionelle Weise Produkte vertrieben werden, ist diese Herangehensweise aus unternehmerischer Sicht nicht mehr empfehlenswert. Es bedarf deshalb ausgebildeter Übersetzer, möglichst Muttersprachler, die die Texte in die jeweilige Sprache transformieren.

Im Fall des hier behandelten Projektes besteht darüber hinaus die Herausforderung darin, dass es sich zum Teil um fachspezifische Bezeichnungen/Wortschöpfungen handelt, die seitens des Kundes entwickelt und verwendet werden. Dies sind beispielsweise:

- Schlüsselloses Schliess- und Startsystem «Open & Start»
- OPC High-Performance-Sitze mit Gütesiegel AGR (Aktion Gesunder Rücken e.V.)

Es bietet sich daher an, diese Produktbezeichnungen, inklusive aller atomaren Spracheinheiten und ganzer zusammengehöriger Textabschnitte zu speichern und zentral zur Verfügung zu stellen.

– Akteure identifizieren

Die Nutzer des Systems sind unternehmensweit vertreten, d. h. alle Mitarbeiter, die für den Kunden Opel arbeiten. Im Einzelnen sind dies Mitarbeiter aus der Kreation/ Mediengestaltung, des Projekt-Managements/der Kundenberatung und aus dem Lektorat.

Da die Nutzer des Systems in heterogenen Bereichen tätig sind und unterschiedliche Affinitäten aufweisen, ist dementsprechend für die Bedienung des Systems keinerlei oder nur wenig Schulungsbedarf erforderlich, was durch eine übersichtliche und intuitiv zu bedienende Nutzeroberfläche gewährleistet ist.

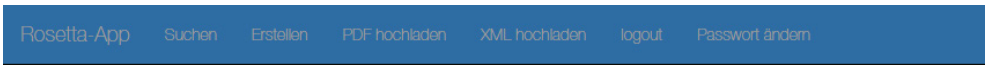


Abb. 3: Navigation mit den sechs Menüpunkten: „Suchen“, „Erstellen“, „PDF hochladen“, „XML hochladen“, „logout“ und „Passwort ändern“.

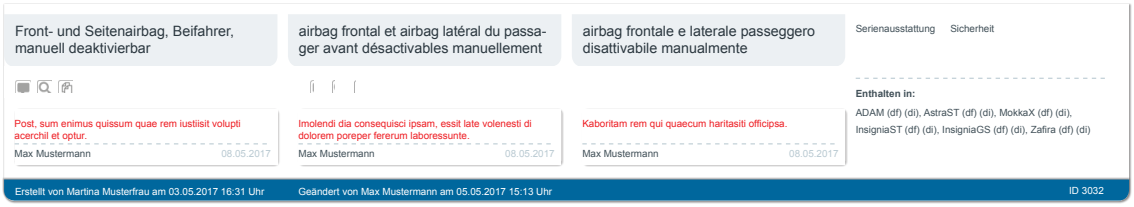


Abb. 4: Darstellung eines Datensatzes mit drei Icons pro Sprache zum „Kommentieren“, „Filtern“ und „Kopieren in die Zwischenablage“. Zu jedem Datensatz sind in der rechten Spalte die beiden Icons zum „Bearbeiten“ und „Löschen“ untergebracht.

– Umfeld

Das System ist primär als browserseitige Anwendung konzipiert. Die Nutzung auf einem mobilen Gerät ist auch möglich, dient allerdings nur dem Zweck, gegebenfalls vorort bei dem Kunden auf den Datenbestand zugreifen zu können. In der Regel funktioniert dies auch auf einem Notebook, soll aber zur Abdeckung aller Eventualitäten ebenso auf weiteren mobilen Endgeräten wie einem Smartphone gewährleistet sein.

5. KONZEPT DER QUALITÄTSSICHERUNG

Zur Unterstützung in der Planung habe ich drei Kollegen gewinnen können, welche für den Kunden Opel arbeiten und somit die spätere Nutzergruppe repräsentieren.

Folgend nenne ich in Kurzform ihre jeweiligen disjunkten Funktionen im Unternehmen und erläutere, in welcher Relation sie mit dem System stehen werden:[3]

Thomas/Mediengestaltung: Kann auf den Bestand vorhandener Übersetzungen zugreifen und diese ad hoc zur Erstellung aller Medien nutzen.

Claudia/Lektorat: Prüft die erstellten Medien, gleicht auf die im System abgelegten Schreibweisen und Übersetzungen ab und korrigiert diese ggf. mithilfe des vorhandenen Bestands.

Nicole/Kundenberatung: Schnittstelle zwischen dem Kunden, den externen Übersetzern, der Produktion und allen beteiligten Personen in der Agentur

In regelmäßigen Besprechungen habe ich diesen drei Kollegen den aktuellen Stand des Projekts präsentiert, konnte ihnen gesammelte Fragen stellen und habe zeitgleich von ihnen Feedback und Input/Ideen zur weiteren Herangehensweise erhalten.

[3] Aus Gründen des Datenschutzes werden die Kollegen nur mit ihren Vornamen genannt.

Auf diese Art und Weise habe ich gleichzeitig eine Regel des Vorgehensmodells XP befolgt, indem ich die späteren Nutzer, hier als Auftraggeber zu sehen, eng in den Entwicklungsprozess eingebunden habe. Auf http://www.extremeprogramming.org/ wird unter dem Punkt „rules/coding“ Folgendes genannt:

> The customer is always Available. One of the few requirements of extreme programming (XP) is to have the customer available. Not only to help the development team, but to be a part of it as well.
>
> Vgl. Extreme Programming, A gentle introduction, The Customer is Always Available, [online] http://www.extremeprogramming.org/rules/customer.html [27.5.2017]

Im weiteren Verlauf der Dokumentation werde ich die drei zuvor genannten Kollegen als die Stakeholder des Systems bezeichnen. Der Begriff „Stakeholder“ wird auf „agiles-projektmanagement.org“ wie folgt definiert:

> Stakeholder (engl. „Teilhaber“) sind Personen bzw. Gruppen von Personen mit einem besonderen Interesse am Ergebnis eines (wirtschaftlichen) Prozesses. Als interne und externe Anspruchsgruppe sind sie entweder direkt oder doch zumindest indirekt von den jeweiligen unternehmerischen Tätigkeiten betroffen. Klassischerweise versteht man unter Stakeholdern u. a. Eigentümer, Mitarbeiter, Aktionäre, Kunden, Partner, Händler und Lieferanten.
>
> Vgl. Agiles Projektmanagement, Scrum verstehen und erfolgreich anwenden, Die Rolle der Stakeholder in Scrum, [online] http://agiles-projektmanagement.org/scrum-stakeholder/ [27.5.2017]

Wie zuvor erwähnt, wies meine im vorherigen Abschnitt beschriebene Vorgehensweise Parallelen zu der eines Vorgehensmodells auf – respektive zum Vorgehensmodell Extreme Programming.

Vgl. Extreme Programming, A gentle introduction, [online] http://www.extremeprogramming.org/ [27.5.2017]

Im überwiegenden Verlauf der Projekt-Entwicklung habe ich mich allerdings an dem agilen Rahmenwerk Scrum orientiert.

Vgl. Scrum: IT-agile, DIE EXPERTEN FÜR AGILE SOFTWAREENTWICKLUNG, [online] https://www.it-agile.de/wissen/einstieg-und-ueberblick/scrum/ [29.5.2017]

In diesem Vorgehensmodell geht man davon aus, dass Softwareprojekte nicht im Voraus detailliert planbar sind. Aus diesem Grund erfolgt die Planung nach dem Prinzip der schrittweisen Verfeinerung, wobei die Entwicklung des Systems durch das Team nahezu gleichberechtigt erfolgt.

Vgl. Scrum: Definition, [online] http://wirtschaftslexikon.gabler.de/Definition/scrum.html [29.5.2017]

Auf drei Schnittmengen dieses Vorgehensmodells und der hier dokumentierten Software-Implementierung gehe ich im Folgenden ein. Es handelt sich hierbei um die Definition der Backlogs, die iterative Vorgehensweise innerhalb zeitlich definierter Sprints und die regelmäßigen Retrospektiven mit den Stakeholdern.

– Requirements–Engineering–Tool

Eine m.E. sehr präzise Definition für das Requirements-Engineering findet sich im Gabler Wirtschaftslexikon.

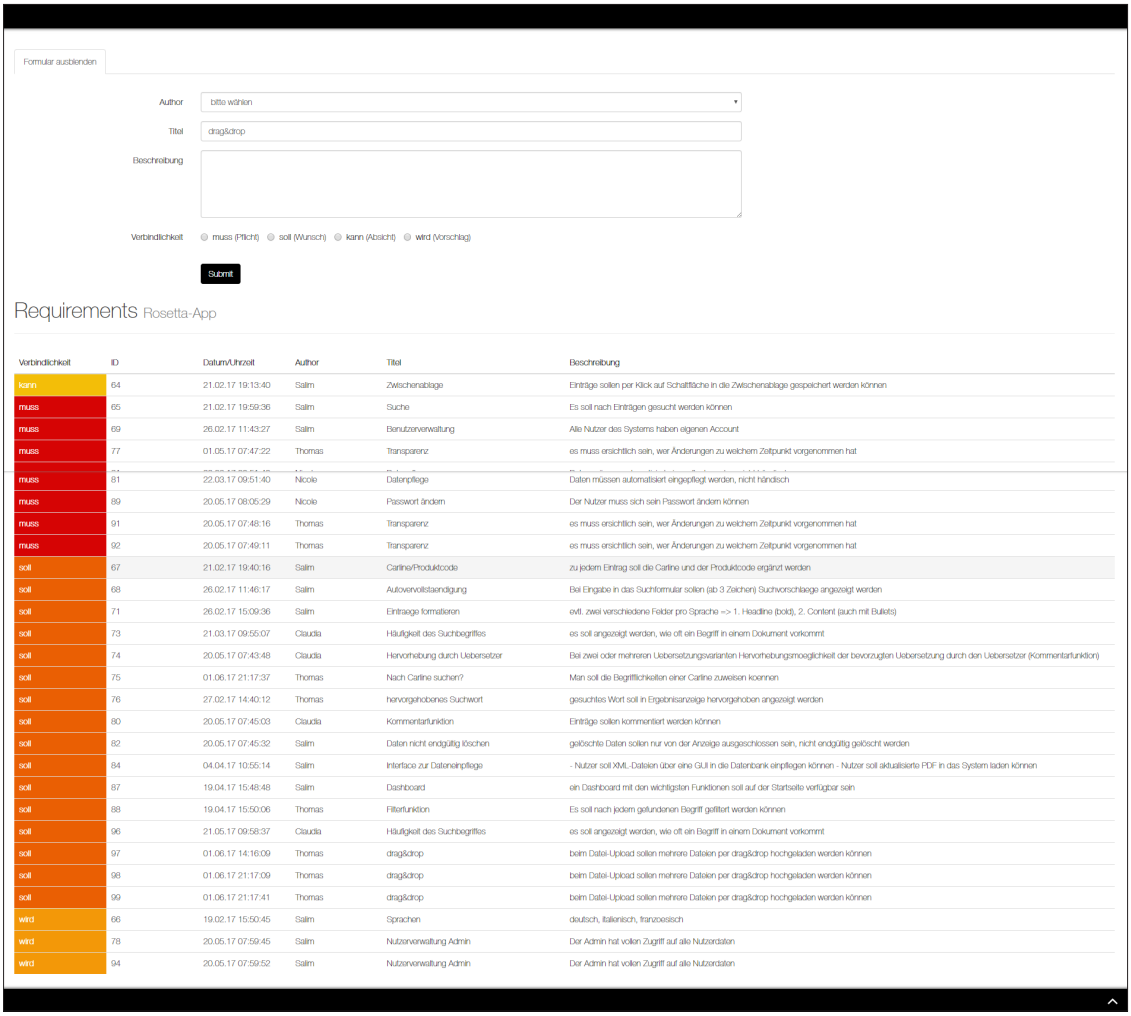
> Ein Requirements-Engineering dient zur Ermittlung, Beschreibung, Analyse und Gewichtung der Anforderungen. Die Festlegung der Requirements erfolgt in einer möglichst exakten und operationalen Form, um eine qualitative Verbesserung der Anforderungsdefinition und eine Reduktion der Fehler zu erreichen.
> Vgl. Requirements Engineering, Definition, [online] http://m.wirtschaftslexikon.gabler.de/Definition/requirements-engineering.html [27.5.2017]

Zur Festlegung eben dieser Vorgaben habe ich ein eigenes Tool geschrieben, mithilfe dessen die Stakeholder und ich einzelne Vorgaben definieren konnten.

Bei dem Vergleich zum Vorgehensmodell Scrum bleibend, setze ich die jeweiligen Requirements gleich mit einem Product Backlog, in welchem die zu entwickelnden Features des Systems priorisiert werden und in gemeinsamer Absprache und unter Berücksichtigung des Nutzens und der Notwendigkeit in einem folgenden Sprint abgearbeitet und implementiert werden.

Das Tool erzwingt eine möglichst einfache Definition der Vorgaben. Es sollen neben der Priorisierung durch den Autor lediglich der Titel und eine Kurzbeschreibung der jeweiligen Vorgabe eingetragen werden.

Der Nutzerkreis des Requirements–Engineering–Tools war für mein Projekt auf vier Personen begrenzt, auf die drei Stakeholder und mich.



Verbindlichkeit	ID	Datum/Uhrzeit	Author	Titel	Beschreibung
kann	64	21.02.17 19:13:40	Salim	Zwischenablage	Einträge sollen per Klick auf Schaltfläche in die Zwischenablage gespeichert werden können
muss	65	21.02.17 19:59:36	Salim	Suche	Es soll nach Einträgen gesucht werden können
muss	69	26.02.17 11:43:27	Salim	Benutzerverwaltung	Alle Nutzer des Systems haben eigenen Account
muss	77	01.05.17 07:47:22	Thomas	Transparenz	es muss ersichtlich sein, wer Änderungen zu welchem Zeitpunkt vorgenommen hat
muss	81	22.03.17 09:51:40	Nicole	Datenpflege	Daten müssen automatisiert eingepflegt werden, nicht händisch
muss	89	20.05.17 08:05:29	Nicole	Passwort ändern	Der Nutzer muss sich sein Passwort ändern können
muss	91	20.05.17 07:48:16	Thomas	Transparenz	es muss ersichtlich sein, wer Änderungen zu welchem Zeitpunkt vorgenommen hat
muss	92	20.05.17 07:49:11	Thomas	Transparenz	es muss ersichtlich sein, wer Änderungen zu welchem Zeitpunkt vorgenommen hat
soll	67	21.02.17 19:40:16	Salim	Carline/Produktcode	zu jedem Eintrag soll die Carline und der Produktcode ergänzt werden
soll	68	26.02.17 11:46:17	Salim	Autovervollstaendigung	Bei Eingabe in das Suchformular sollen (ab 3 Zeichen) Suchvorschlaege angezeigt werden
soll	71	26.02.17 15:09:36	Salim	Eintraege formatieren	evtl. zwei verschiedene Felder pro Sprache => 1. Headline (bold), 2. Content (auch mit Bullets)
soll	73	21.03.17 09:55:07	Claudia	Häufigkeit des Suchbegriffes	es soll angezeigt werden, wie oft ein Begriff in einem Dokument vorkommt
soll	74	20.05.17 07:43:48	Claudia	Hervorhebung durch Uebersetzer	Bei zwei oder mehreren Uebersetzungsvarianten Hervorhebungsmoeglichkeit der bevorzugten Uebersetzung durch den Uebersetzer (Kommentarfunktion)
soll	75	01.06.17 21:17:37	Thomas	Nach Carline suchen?	Man soll die Begrifflichkeiten einer Carline zuweisen koennen
soll	76	27.02.17 14:40:12	Thomas	hervorgehobenes Suchwort	gesuchtes Wort soll in Ergebnisanzeige hervorgehoben angezeigt werden
soll	80	20.05.17 07:45:03	Claudia	Kommentarfunktion	Einträge sollen kommentiert werden können
soll	82	20.05.17 07:45:32	Salim	Daten nicht endgültig löschen	gelöschte Daten sollen nur von der Anzeige ausgeschlossen sein, nicht endgültig gelöscht werden
soll	84	04.04.17 10:55:14	Salim	Interface zur Dateneinpflege	- Nutzer soll XML-Dateien über eine GUI in die Datenbank einpflegen können - Nutzer soll aktualisierte PDF in das System laden können
soll	87	19.04.17 15:48:48	Salim	Dashboard	ein Dashboard mit den wichtigsten Funktionen soll auf der Startseite verfügbar sein
soll	88	19.04.17 15:50:06	Thomas	Filterfunktion	Es soll nach jedem gefundenen Begriff gefiltert werden können
soll	96	21.05.17 09:58:37	Claudia	Häufigkeit des Suchbegriffes	es soll angezeigt werden, wie oft ein Begriff in einem Dokument vorkommt
soll	97	01.06.17 14:16:09	Thomas	drag&drop	beim Datei-Upload sollen mehrere Dateien per drag&drop hochgeladen werden können
soll	98	01.06.17 21:17:09	Thomas	drag&drop	beim Datei-Upload sollen mehrere Dateien per drag&drop hochgeladen werden können
soll	99	01.06.17 21:17:41	Thomas	drag&drop	beim Datei-Upload sollen mehrere Dateien per drag&drop hochgeladen werden können
wird	66	19.02.17 15:50:45	Salim	Sprachen	deutsch, italienisch, franzoesisch
wird	78	20.05.17 07:59:45	Salim	Nutzerverwaltung Admin	Der Admin hat vollen Zugriff auf alle Nutzerdaten
wird	94	20.05.17 07:59:52	Salim	Nutzerverwaltung Admin	Der Admin hat vollen Zugriff auf alle Nutzerdaten

Abb. 5: Screenshot des Requirement-Engineering-Tools. Online verfügbar unter: http://requirements.rosetta-app.de/index.php (Stand: 28.6.2017)

Dieses Werkzeug habe ich in PHP geschrieben, die Einträge liegen in einer MariaDB-Datenbank ab. Einzelne Funktionen wie das farbige Hinterlegen der Verbindlichkeiten oder das animierte Ein- und Ausblenden des Formulars sind mithilfe von JavaScript implementiert.

Es handelt sich bei diesem Tool um einen erforderlichen Nebenschauplatz. Deshalb gehe ich an dieser Stelle nicht näher auf die Programmierung dieses Tools ein, sondern verweise auf den für die Projekt-Entwicklung entscheidenden Inhalt.

Mithilfe der folgenden Tabelle liste ich die einzelnen Vorgaben auf:

Requirements Rosetta-App		
Titel	Beschreibung	erledigt
Sprachen	deutsch, italienisch, franzoesisch	
Zwischenablage	Einträge sollen per Klick auf Schaltfläche in die Zwischenablage gespeichert werden können	
Suche	Es soll nach Einträgen gesucht werden können	
Benutzerverwaltung	Alle Nutzer des Systems haben eigenen Account	
Datenpflege	Daten müssen automatisiert eingepflegt werden, nicht händisch	
Transparenz	es muss ersichtlich sein, wer Änderungen zu welchem Zeitpunkt vorgenommen hat	
Nutzerverwaltung Admin	Der Admin hat vollen Zugriff auf alle Nutzerdaten	
Passwort ändern	Der Nutzer muss sich sein Passwort ändern können	
Carline/Produktcode	zu jedem Eintrag soll die Carline und der Produktcode ergänzt werden	
Autovervollständigung	Bei Eingabe in das Suchformular sollen (ab 3 Zeichen) Suchvorschläge angezeigt werden	
Einträge formatieren	evtl. zwei verschiedene Felder pro Sprache => 1. Headline (bold), 2. Content (auch mit Bullets)	
Häufigkeit des Suchbegriffes	es soll angezeigt werden, wie oft ein Begriff in einem Dokument vorkommt	
Hervorhebung durch Übersetzer	Bei zwei oder mehreren Übersetzungsvarianten Hervorhebungsmöglichkeit der bevorzugten Übersetzung durch den Übersetzer (Kommentarfunktion)	
Nach Carline suchen?	Man soll die Begrifflichkeiten einer Carline zuweisen können.	
hervorgehobenes Suchwort	gesuchtes Wort soll in Ergebnisanzeige hervorgehoben angezeigt werden	
Kommentarfunktion	Einträge sollen kommentiert werden können	
Daten nicht endgültig löschen	gelöschte Daten sollen nur von der Anzeige ausgeschlossen sein, nicht endgültig gelöscht werden	
Interface zur Datenpflege	- Nutzer soll XML-Dateien über eine GUI in die Datenbank einpflegen können - Nutzer soll aktualisierte PDF in das System laden können	
Dashboard	ein Dashboard mit den wichtigsten Funktionen soll auf der Startseite verfügbar sein	
drag&drop	beim Datei-Upload sollen mehrere Dateien per drag&drop hochgeladen werden können	
Filterfunktion	Es soll nach jedem gefundenen Begriff gefiltert werden können	

Tab. 1: Tabelle stellt die Anforderungen dar. In der rechten Spalte ist farblich gekennzeichnet, welche dieser Anforderungen umgesetzt sind. (Stand: 1.6.2017)

– iterative Herangehensweise

Es wurde im Team festgelegt, welche dieser Anforderungen im Einzelnen in einem nächsten Schritt bearbeitet werden sollten. Diese iterative Herangehensweise lässt sich gut vergleichen mit einem Sprint aus dem Vorgehensmodell Scrum. In einem solchen Sprint soll innerhalb einer festgelegten Dauer von typischerweise zwei bis vier Wochen ein funktionsfähiges Inkrement des Software-Produkts implementiert werden.

– Retrospektive Stakeholdern

In regelmäßigen Meetings mit den Stakeholdern wurde besprochen, was umgesetzt wurde. Die implementierten Features wurden getestet, beurteilt und ggf. eine Nachbesserung bzw. Anpassung in Auftrag gegeben.

Diese Methode möchte ich im Ansatz mit der einer Retrospektive vergleichen.

– Testen

Zum Abschluss des Kapitels „Konzept der Qualitätssicherung“ und nachdem ich erläutert habe, wie die drei Stakeholder und ich die einzelnen Anforderungen für das System bestimmt und ich diese auf iterative Weise implementiert habe, möchte ich wie zuvor unter dem Punkt Retrospektive angedeutet, auf den Aspekt des Testens eingehen.

Das Testen war und ist ein fortlaufender Prozess. Die drei Stakeholder haben einen permanenten Zugriff auf die online laufende Applikation. Nachdem ich die jeweiligen Anforderungen in der lokalen Umgebung implementiert habe, habe ich das System zeitnah synchronisiert, sodass der lokale Status stets mit dem online laufenden äquivalent ist.

Auf diese Weise konnte das System unabhängig von Zeit, Ort und innerhalb sehr kurzer Intervalle kontinuierlich getestet werden. Die intensive Besprechung/ Evaluation der Ergebnisse war im Regelfall Teil der Retrospektiven, ggf. auch mal in zwischenzeitlichen Besprechungen mit einzelnen Personen dieser Gruppe.

Seitens der Stakeholder wurde ausschließlich auf der Ebene der Usability bewertet. Mein Aufgabenbereich war/ist es zudem, dass das Programm sich wie gewünscht verhält und dass der Programm-Code optimal abläuft.

6. SYSTEMSTRUKTUR

– Infrastruktur

Das System habe ich mithilfe der integrierten Entwicklungsumgebung PHP Storm des Unternehmens JET BRAINS entwickelt. Ich habe mich für diese IDE entschieden, weil PHP Storm eine gute Autovervollständigung, ein intelligentes Refactoring und vor allem eine komfortable Anbindung an mein für diese Arbeit erstelltes Repository auf github.com bietet.

Vgl. PHP Storm, JET BRAINS: Entwicklungsumgebung, [online] https://www.jetbrains.com/phpstorm/ [2.5.2017]

Als Serverumgebung habe ich die freie Software XAMPP des Entwicklers Apache Friends mit dem integrierten phpMyAdmin-Interface genutzt. Dadurch konnte ich das System auf einem lokalen Server entwickeln und testen.

Vgl. XAMPP, Apache Friends: Serverumgebung, [online] https://www.apachefriends.org/de/index.html [2.5.2017]

Als relationales Datenbank-System habe ich die Open Source-Datenbank MariaDB genutzt.

Vgl. MariaDB, The MariaDB Foundation: freies, relationales Open-Source-Datenbankverwaltungssystem, [online] https://mariadb.org/ [28.5.2017]

Die Kommunikation mit der Datenbank habe ich mittels PHP Data Objects realisiert. Wie von PHP.net beschrieben, stellt diese PHP Data Object-Erweiterung (PDO) eine leichte, konsistente Schnittstelle bereit, um mit PHP auf Datenbanken zuzugreifen. Dadurch, dass der Datenzugriff über eine Abstraktionsschicht läuft, ist es möglich, mittels PDO auf jedes beliebige Datenbanksystem mit den selben Funktionen zugreifen zu können.

Vgl. PHP-Handbuch, Funktionsreferenz, Datenbankerweiterungen, Abstraktionsebenen, PDO, Einführung, [online] http://php.net/manual/de/intro.pdo.php [28.5.2017]

Daraus ergibt sich, dass das System nicht an die jetzige Datenbank gebunden ist und der aktuelle Code bereits für alternative Datenbanken gültig konfiguriert wurde. Doch der entscheidende Grund für die Arbeit mit PDO war für mich die Aktualität dieser Technik im Vergleich zu MySQLi und insbesondere zu MySQL.

Online läuft das System auf einem Server der STRATO AG. Analog zur lokalen Umgebung ist auch dieser Server mit phpMyAdmin und MariaDB ausgestattet, wodurch ein Umzug von lokal nach online relativ unkompliziert ist, es muss lediglich die Datenbankanbindung konfiguriert werden.

– Sprache

- **PHP**

  Da das System eine typische Web 2.0-Anwendung ist, der Nutzer über den Browser auf den dynamisch erzeugten Content zugreift und bidirektional den Datenbestand nutzt, habe ich PHP für die Erstellung des Systems verwendet.

  Ich habe bei der Programmierung teilweise auf objektorientierte Features von PHP zurückgegriffen, um dadurch einen schlankeren Code erzeugen zu können.

  Zum Beispiel habe ich bei der Verwendung von Klassen, mithilfe derer ich u. a. das Gerüst von Formular-Feldern einmal in einer Klasse definiert habe und im Produktions-Code über nur ein kurzen Aufruf mit Parameterübergabe darstellen konnte. Dies kommt der Modellierung des Views in einem MVC-Pattern sehr nahe.

  Das Model in diesem Pattern stelle ich in Form der verschiedenen Datenbank-Anbindungen dar, über die der Datenbestand manipuliert, bzw. ausgegeben wird.

  Der eigentliche Controller, der die Kommunikation zwischen View und Model regelt und die eigentliche Anwendung steuert, ist zum aktuellen Stand der Dokumentation noch nicht implementiert, vielmehr sind zur Zeit Funktionen des Controllers teilweise sowohl im View als auch im Model integriert.

  Um ein echtes MVC-Pattern zu implementieren, müsste man diese Abhängigkeiten dementsprechend anpassen.

  Vgl. MVC mit PHP, Grundlagen, [online] http://tutorials.lemme.at/mvc-mit-php/#grundlagen [6.6.2017]

- **JavaScript**

  Neben der Programmierung mit PHP habe ich teilweise auf JavaScript zurückgegriffen und auch das Framework jQuery genutzt.

  Mithilfe dieser Skriptsprache habe ich Funktionen implementiert wie z. B. das automatische Skalieren der Textfelder, sobald ein Übertrag an Inhalt entsteht. Dies ermöglicht es, eine feste Größe des Textfeldes darzustellen, welche dennoch flexibel ist. Die Folge ist eine ausgeglichene Darstellung für den User und gleichzeitig bleibt keinerlei Input im Verborgenen.

Eine weiteres nützliches Feature ist die Funktion „copyToClipboard()“, welche den generischen Parameter „e“ entgegen nimmt, dem Body-Element im DOM ein temporäres Input-Field hinzufügt, dieses Feld mit dem zu kopierenden Wert belegt, sodass man den gewünschten Text mithilfe des Aufrufs „document.execCommand(„copy“)“ in die Zwischenablage kopieren kann. Hierzu ist für jede Textstelle, die kopiert werden kann, eine eindeutige ID nötig, dies habe ich wiederum mittels PHP realisiert. Dieses Feature erweist sich als besonders praktisch, sobald man bestimmte Daten aus dem System für die aktuelle Arbeit benötigt.

Zum Auswählen der Carlines habe ich ein Multiselect-Formular genutzt, welches sich in einem eleganten Dropdown-Menü verbirgt. Hierzu habe ich auf das unter der BSD-Lizenz frei verwendbare Script „Bootstrap-Multiselect“ zurückgegriffen.

Vgl. Bootstrap Multiselect, David Stutz: BSD 3-Clause License: Copyright (c) 2012 - 2015 David Stutz, [online] https://github.com/davidstutz/bootstrap-multiselect/blob/master/dist/js/bootstrap-multiselect.js [6.5.2017]

Die Autocomplete Funktion habe ich mithilfe der unter der MIT-Lizenz stehenden freien jQuery JavaScript Library verwirklicht.

Vgl. Autocomplete, jQuery JavaScript Library v1.10.2: Copyright 2005, 2013 jQuery Foundation, Inc. and other contributors, Released under the MIT license, http://jquery.org/license, [online] http://code.jquery.com/jquery-1.10.2.js [6.6.2017]

– Frameworks

- **Bootstrap**

  Ich habe das freie CSS-Framework Bootstrap referenziert um so die in diesem Framework vordefinierte Klassen-Hierarchie nutzen zu können. Bootstrap habe ich für den Aufbau der Formulare, Buttons und der Navigation genutzt und diese teilweise individualisiert.

  Im Besonderen habe ich mich allerdings für dieses Framework entschieden, da es ein 12-Grid-System unterstützt, welches ich beim Screendesign angewandt habe. Ein weiterer ausschlaggebender Grund für die Nutzung von Bootstrap ist die bereits implementierte responsive Darstellung, d.h. ich musste nicht eigene MediaQuerries für die unterschiedlichsten Endgeräte definieren.

  Vgl. Bootstrap, Twitter: HTML, CSS, and JS framework for developing responsive, mobile first projects on the web, [online] http://getbootstrap.com/ [2.5.2017]

– Tools

Neben der eigentlichen Entwicklungsumgebung PHP Storm und der Server-Umgebung XAMPP von Apache Friends habe ich zur Unterstützung meiner Projektarbeit die nachfolgende Tools genutzt.

- **Github**
  Zur Versionsverwaltung und damit ich nicht auf die Verwendung eines einzigen Arbeitsplatzes während der Projektarbeit beschränkt bin, habe ich Git verwendet, d. h. Github in Verbindung mit PHP-Storm.

  Vgl. Github, GitHub, Inc., kollaborative Versionsverwaltung, [online] https://github.com/ [2.5.2017]

  Das dortige Repository dient zudem zum einfachen Austausch mit dem Betreuer der Hochschule und als Beleg des erbrachten Aufwands, der sich bei der Projektarbeit ergeben hat.

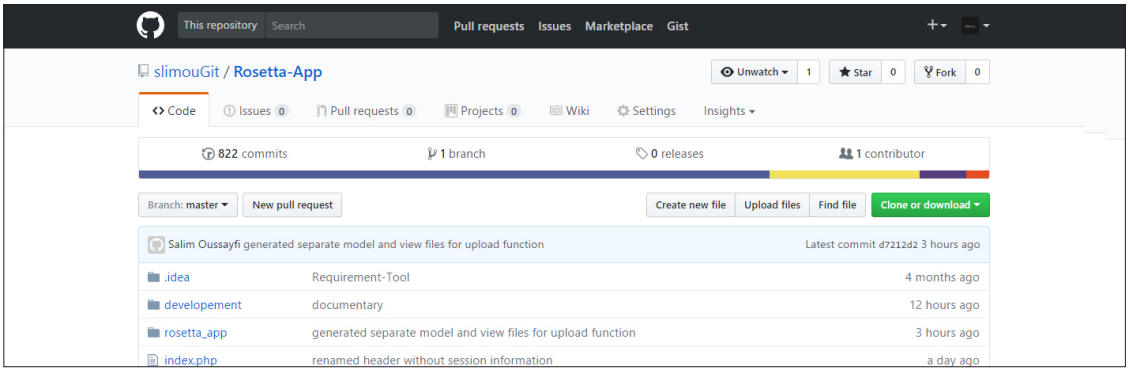


Abb. 6: Screenshot des Repository bei GitHub. (Stand: 26.6.2017) Online verfügbar unter: https://github.com/slimouGit/Rosetta-App.git

- **Trello**
  Um den gesamten Workflow zu organisieren, habe ich mit Trello gearbeitet.

  Ich habe mir Listen zu verschiedenen Teilbereichen der Projektarbeit erstellt und in diesen jeweils einzelne Karten und Checklisten erstellt.

  Dadurch hatte ich stets den vollen Umfang im Blick und konnte meine Arbeit optimal strukturiert planen.

  Vgl. Trello, Trello Inc. (Fog Creek Software), web-basierte Projektmanagementsoftware, [online] https://trello.com/ [2.5.2017]

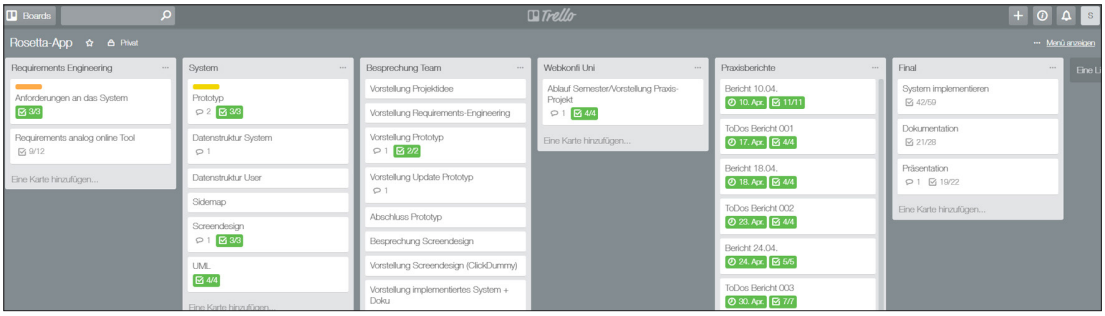


Abb. 7: Screenshot Trello: Board für Projekt mit verschiedenen Listen und Karten innerhalb dieser Listen (Stand: 3.6.2017)

- **Requirements Engineering**
  Wie bereits unter dem Punkt „Konzept der Qualitätssicherung“ erwähnt, habe ich ein unabhängig laufendes Requirements-Engineering-Tool aufgesetzt, mithilfe dessen die Anforderungen an das System definiert wurden.
  Requirement-Engineering-Tools ist online verfügbar unter: http://requirements.rosetta-app.de/index.php

- **UML**
  Zur Modellierung der UML-Diagramme habe ich das für akademische (nicht kommerzielle) Zwecke freie Tool Astah Community genutzt.
  Vgl. Astah Community, JUDE (Java and UML Developers' Environment), UML Modellierungstool, [online] http://astah.net/editions/community [2.5.2017]

7. DESIGN

– UML–Diagramme

Zur Modellierung und Dokumentation bestimmter Abläufe in der Software habe ich Unified Modeling Language (UML) verwendet.

Auf der Internetpräsenz der Hochschule Darmstadt wird der Begriff UML wie folgt definiert:

> Die UML ist eine Modellierungssprache, also eine Sprache zur Beschreibung von Software-Systemen. Sie bietet eine einheitliche Notation, die für viele Anwendungsgebiete nutzbar ist. Sie enthält Diagramme und Prosa-Beschreibungsformen. Mit Hilfe der UML können statische, dynamische und Implementierungsaspekte von Softwaresystemen beschrieben werden. Die UML ist damit zur Zeit die umfassendste Sprache und Notation zur Spezifikation, Konstruktion Visualisierung und Dokumentation von Modellen für die Softwareentwicklung.
>
> Vgl. Hochschule Darmstadt, Die Unified Modeling Language – UML, [online] https://www.fbi.h-da.de/labore/case/uml.html [1.6.2017]

Ich habe vier Diagramme modelliert. Dies sind im Einzelnen die beiden Verhaltensdiagramme „Use-Case“ und „Activity“ und die beiden Diagramme „Deployment“ und „Component“ aus der Gruppe der Architekturdiagramme.

Use–Case

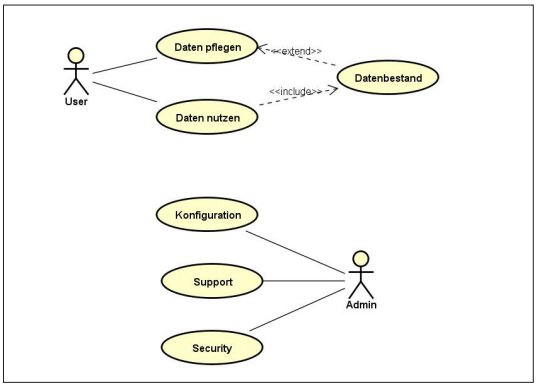


Abb. 8: Anwendungsfall-Diagramm. Online verfügbar unter: https://github.com/slimouGit/Rosetta-App/blob/master/developement/documentary/UML/UseCase.jpg

Deployment

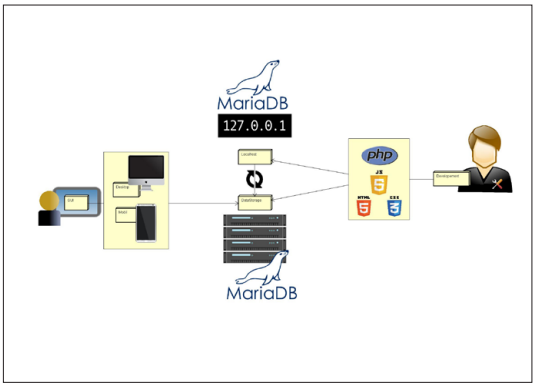


Abb. 9: Verteilungs-Diagramm. Online verfügbar unter: https://github.com/slimouGit/Rosetta-App/blob/master/developement/documentary/UML/Deployment.jpg

Bilder: „User“, „Admin“, „Computer“, „Smartphone“, openclipart, Public Domain Dedication, [online] https://openclipart.org [1.6.2017]
Bilder: „PHP“, „JS“, „CSS“, „MariaDB“, seeklogo, Creative Commons Attribution 3.0, [online] https://seeklogo.com/vector-logo/108600/php [1.6.2017]

Component

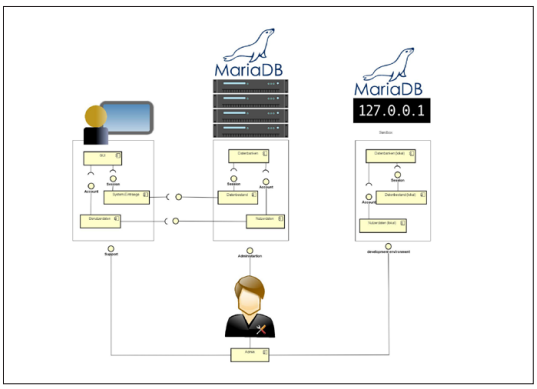


Abb. 10: Komponenten-Diagramm. Online verfügbar unter: https://github.com/slimouGit/Rosetta-App/blob/master/developement/documentary/UML/Component.jpg

Bilder: „User“, „Admin“, „Server“, openclipart, Public Domain Dedication, [online] https://openclipart.org [1.6.2017]
Bilder: „MariaDB“, seeklogo, Creative Commons Attribution 3.0, [online] https://seeklogo.com/vector-logo/108600/php [1.6.2017]

Activity

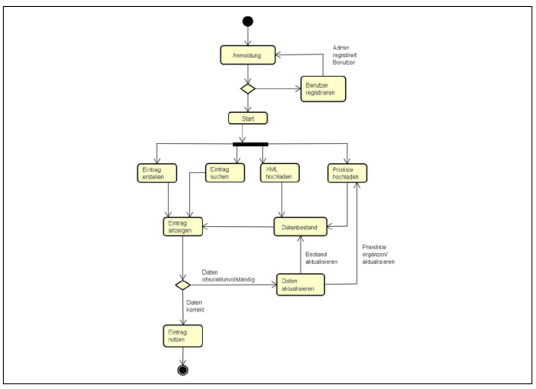


Abb. 11: Aktivitäts-Diagramm. Online verfügbar unter: https://github.com/slimouGit/Rosetta-App/blob/master/developement/documentary/UML/Activity.jpg

– Prototyp

Mit der initialen Idee für mein Projekt und auf Basis des von mir eingereichten Exposés habe ich meine Arbeit damit begonnen, einen Prototypen zu entwickeln.

Während dieses Prozesses entstanden laufend neue Ideen, auch durch die beteiligten Stakeholder. Gleichzeitig tauchten Probleme und Fragen auf, sei es in der Programmierung, der Infrastruktur oder darin, die bestmögliche Nutzung des Systems zu ermöglichen.

Meine Vorgehensweise, die iterative Entwicklung eines Prototyps vorab, ermöglichte es mir, eben diese Problematiken früh zu erkennen und zu behandeln/zu beheben. Erst, als dieser Prototyp in seinen Funktionen weitestgehend abgeschlossen und funktionsfähig war, habe ich mit der Implementierung des eigentlichen Systems begonnen.

Der Prototyp ist unter http://prototype.rosetta-app.de abrufbar.

Es sind zwei Beispiel-Nutzer registriert, welche verschiedene Rechte haben:

Administrator:	admin@rosetta-app.de	Passwort: admin
User:	user@rosetta-app.de	Passwort: user

Screenshots des initialen Prototyps

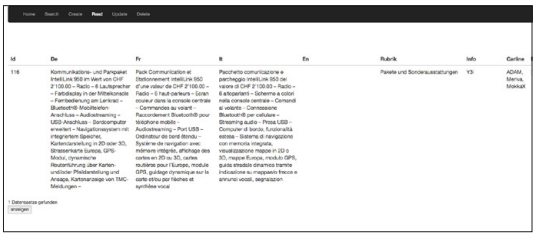


Abb. 12: Screenshot: initiale Datenanzeige

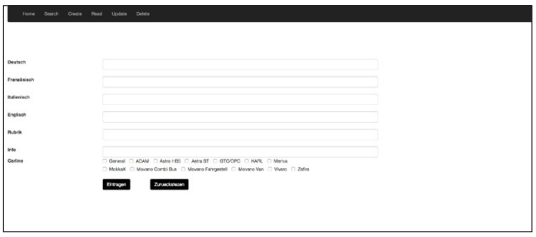


Abb. 13: Screenshot: initiale Dateneingabe

Screenshots des abgeschlossenen Prototyps

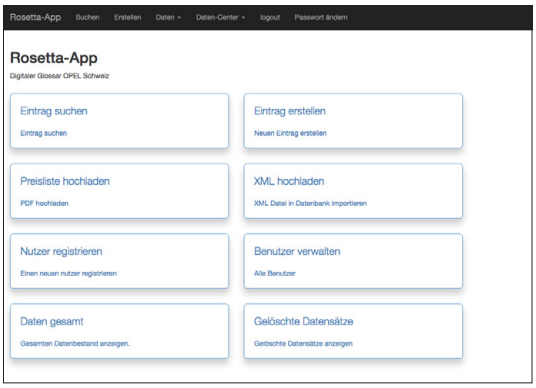


Abb. 14: Screenshot: Dashboard

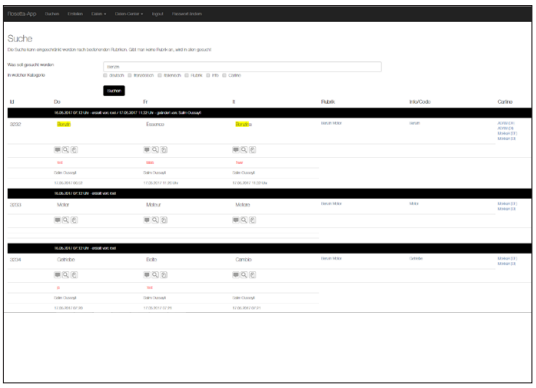


Abb. 15: Screenshot: Suche und Datenausgabe

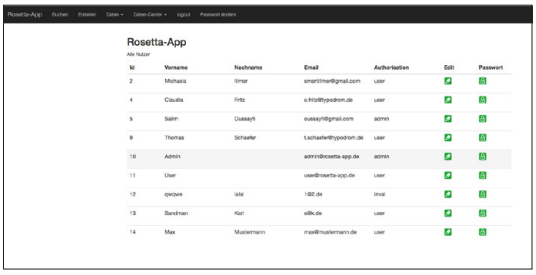


Abb. 16: Screenshot: Anzeige der Nutzer

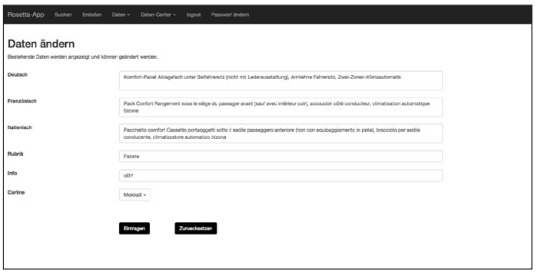


Abb. 17: Screenshot: Daten ändern

– Sidemap

Nachdem die Implementierung des Prototyps abgeschlossen war, habe ich zu meiner Orientierung und besseren Übersicht und für eine sinnvollen Gliederung des Codes eine Sidemap entwickelt.

Ich habe eine Trennung des Codes vorgenommen, d.h. der Code ist gegliedert in Kommunikation mit der Datenbank, dem Gerüst für die darzustellenden Daten und den Seiten, die die eigentlichen Daten anzeigen. Diese Herangehensweise entspricht teilweise der eines MVC-Patterns – „teilweise“ begründet sich daraus, dass die Funktionen des Controllers bis zum heutigen Tag noch nicht gesondert implementiert sind.

In der folgenden Grafik sind alle Seiten des Systems mit einer jeweiligen Kurzbeschreibung in eine sinngemäße und aufgabenrelevante Struktur unterteilt.

Im linken, unteren Bereich dieser Grafik befinden sich die beiden Ordner, „lib“ und „mvc“. Der Ordner „lib“ enthält das Bootstrap-Framework, die Elemente wie Footer und Header, die JavaScript Dateien, die CSS-Datei und alle Bilder. Der Ordner „mvc“ enthält aktuell alle Dokumente und Klassen des Models und des Views in separaten Ordnern.

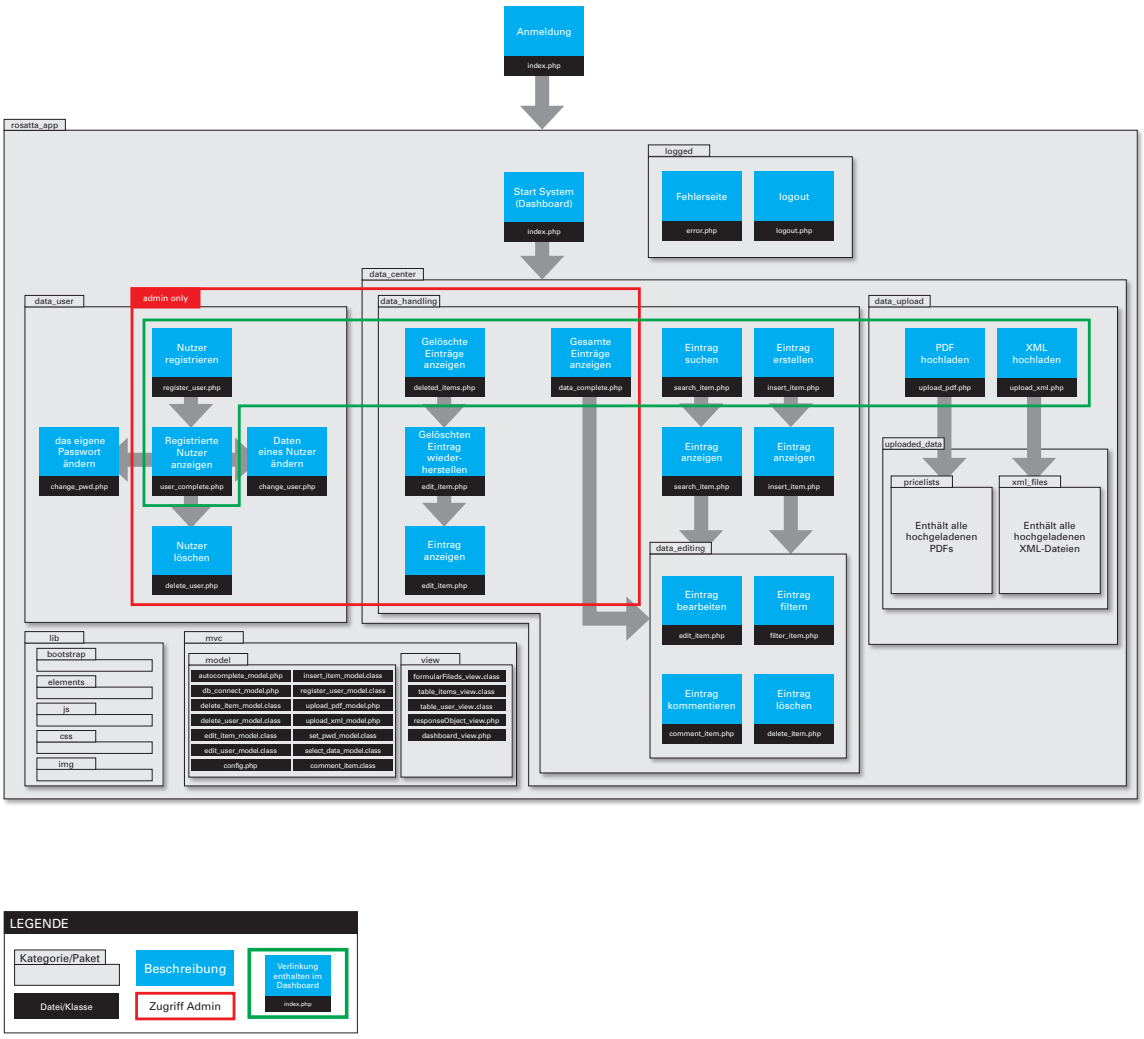


Abb. 18: Sidemap. Online verfügbar unter: https://github.com/slimouGit/Rosetta-App/blob/master/developement/sidemap/sidemap_Rosetta-App.pdf

– Datenstruktur

In den folgenden zwei Tabellen stelle ich die Datenstruktur/das Datenbankschema der Anwendung vor. Mittels der beiden Tabellen „rosetta_data“ und „rosetta_user“ ist es möglich, ein weitreichendes Spektrum an Relationen darzustellen und umfangreiche Operationen im System zu gewährleisten. Die Datenbank-Tabellen sind in einer redundanzfreien Form gehalten.
Die Beziehung beider Tabellen wird über den jeweiligen Nutzer hergestellt, der einen Tupel generiert, manipuliert oder löscht. Als eindeutige Kennung wird der vollständige Nutzername bzw. die Nutzer-ID verwendet.
Neben der Bezeichnung der jeweiligen Spalten beinhaltet die folgende Abbildung den Datentyp und eine Kurzbeschreibung. Die in den Tabellen nicht dargestellte Kollation ist bei den Textbasierten Werten „varchar“ und „text“ immer „utf8_general_mysql500_ci“. Die jeweiligen Primär-Schlüssel sind in beidenTabellen „item_id“, bzw. „user_id“.

Zur besseren Übersicht habe ich die Tabellen jeweils in vier sinngemäße Kategorien eingeteilt. In der Tabelle „rosetta_data“ sind dies beispielhaft: „Kennung“, „Daten“, „Zusatzinformationen“ und „create/update/delete Operationen“.

rosetta_data		
Spaltenbezeichnung	Typ	Beschreibung
Kennung		
token	varchar	Eindeutige Kennung erhält jeder Datensatz bei Initialisierung
data_id	int	ID des Datensatzes
state	varchar	Initial ist das Feld auf „active“, wird der Datensatz gelöscht, wechselt der Wert auf „deleted“
Daten		
item_de	text	Text Deutsch
item_de_comment	varchar	Kommentar Deutsch
user_de_comment	varchar	Von wem ist der Kommentar
date_de_comment	varchar	Datum des Kommentars (hier als String)
item_fr	text	Text Französisch
item_fr_comment	varchar	Kommentar Französisch
user_fr_comment	varchar	Von wem ist der Kommentar
date_fr_comment	varchar	Datum des Kommentars (hier als String)
item_it	text	Text Italienisch
item_it_comment	varchar	Kommentar Italienisch
user_it_comment	varchar	Von wem ist der Kommentar
date_it_comment	varchar	Datum des Kommentars (hier als String)
Zusatzinformation		
category	varchar	Rubrik des Objekts
info	varchar	Info bzw. Objekt-Code
carline	varchar	Enthalten in welchen Carlines (= Verlinkung zu PDFs)
create/update/delete Operationen		
user_create	varchar	Wer hat den Datensatz erstellt (forename surname)
date_create	timestamp	Wann wurde der Datensatz erstellt
user_update	varchar	Wer hat den Datensatz aktualisiert (forename surname)
date_update	timestamp	Wann wurde der Datensatz aktualisiert
user_delete	varchar	Wer hat den Datensatz gelöscht (forename surname)
date_delete	varchar	Wann wurde der Datensatz gelöscht (hier als String)

Tab. 2: Tabelle stellt Datenstruktur der Datenbank-Tabelle „rosetta_data“ dar.

rosetta_users		
Spaltenbezeichnung	Typ	Beschreibung
Kennung		
user_id	int	ID des Users
authorization	varchar	Rechte User -> Differenzierung user/admin
Daten		
forename	varchar	Vorname
surname	varchar	Nachname
email	varchar	Email des Users
password	varchar	Password (php password_hash)
create/update Benutzer-Operationen		
create_user	timestamp	Erstellzeitpunkt User
update_user	timestamp	Änderungszeitpunkt User
Passwort ändern		
password_code	varchar	Temporäre Kennung (Sicherheitsaspekt, wenn Passwort neu angefordert wird)
password_date	timestamp	Zeitpunkt der Passwortanfrage

Tab. 3: Tabelle stellt Datenstruktur der Datenbank-Tabelle „rosetta_users“ dar.

8. ANWENDUNG

In diesem Kapitel gehe ich auf die schrittweise Umsetzung des Systems ein. Ich starte mit dem Screendesign und zeige, wie ich einen interaktiven Clickkdummy mit Invision umgesetzt habe. Anschließend stelle ich die fertige Anwendung vor und schließe diesen Abschnitt mit der Vorstellung einiger ausgewählter Features des Tools ab.

– Screendesign

Für das Screendesign habe ich mit dem 960 Grid System gearbeitet. Dieses Raster ist insgesamt 960px breit, hat 12 Spalten mit jeweils 60px. Jede Spalte hat 10px Randabstand nach rechts und links, also Stege in einer Breite von 20px.

Ich habe mich für dieses gängige und flexible Spaltenlayout entschieden, da ich das System unter Anwendung des Bootstrap-Frameworks verwirklicht habe und auch Bootstrap mit dieser Spalteneinteilung arbeitet.

Vgl. Bootstrap, Twitter, Bootstrap grid examples, [online] http://getbootstrap.com/examples/grid/ [16.5.2017]

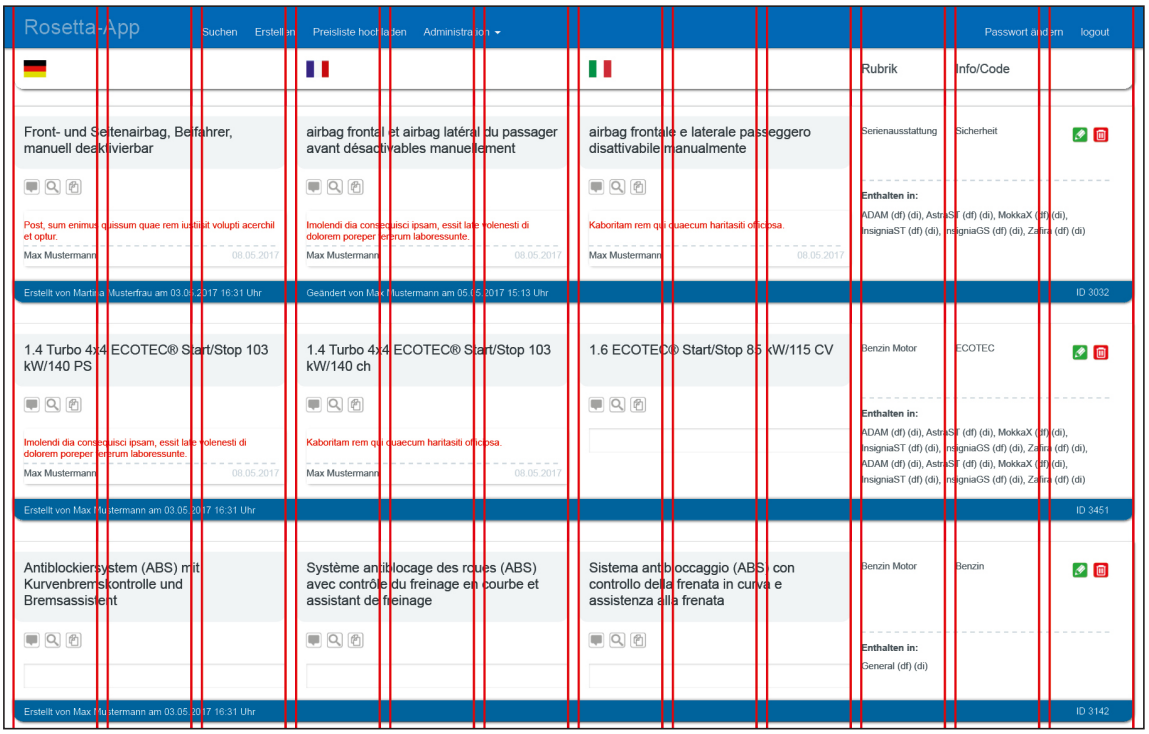


Abb. 19: Screendesign im Spaltenlayout.

Das Screendesign habe ich mit den Programmen Photoshop, Illustrator und InDesign der Adobe Creative Cloud (Studentenlizenz) entwickelt.

Adobe Creative Cloud, Studentenlizenz, [online] http://www.adobe.com/de/creativecloud.html [1.6.2017]

Ich habe verschiedene Unterseiten entworfen, teilweise inklusive bestimmter Fallunterscheidungen wie zum Beispiel die Response des Systems nach dem Hochladen einer XML-Datei. Auf diese Weise konnte ich das fertige Screendesign zum Aufbau der späteren CSS-Datei nutzen, da ich alle nötigen Informationen im Bereich des Designs zur Verfügung hatte.

Die folgenden Screenshots stellen einen Teil der entworfenen Seiten dar. Das komplette Design ist in Form eines PDFs in meinem Repository auf Github verfügbar: https://github.com/slimouGit/Rosetta-App/blob/master/developement/screendesign/screendesign_Rosetta-App.pdf (Stand: 28.6.2017)

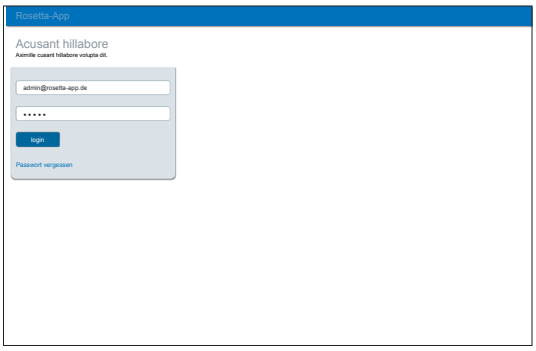


Abb. 20: Screendesign: Login Startseite

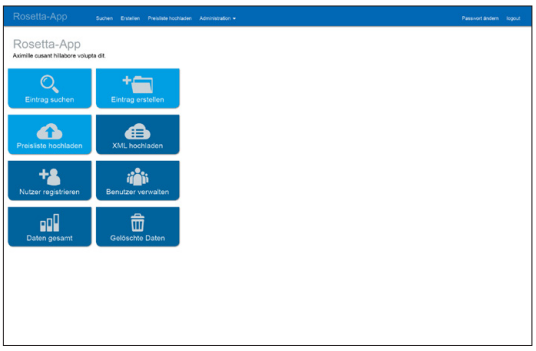


Abb. 21: Screendesign: Dashboard

Bilder/Icons: freepik, Free vector icons for personal and commercial use, [online] http://www.freepik.com/free-icons [1.6.2017]
Teilweise nachträglich indivisualisiert

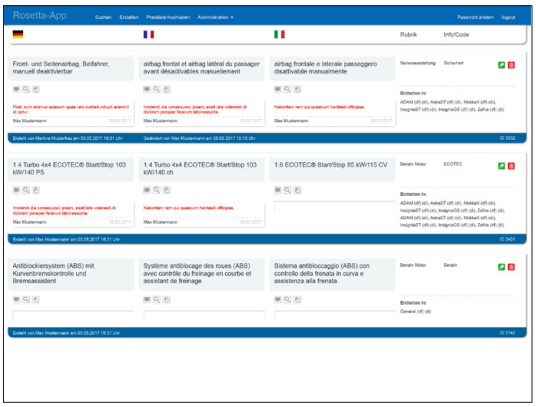


Abb. 22: Screendesign: Anzeige Datensätze

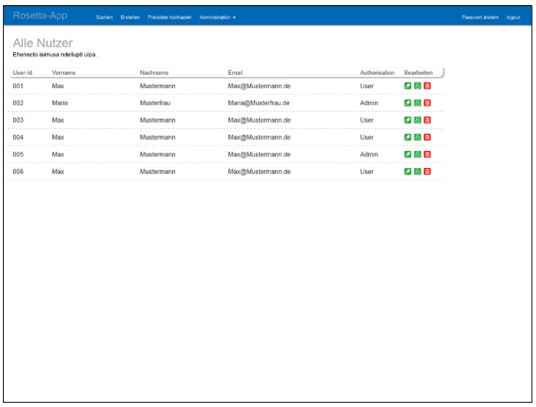


Abb. 23: Screendesign: Anzeige der registrierten Benutzer

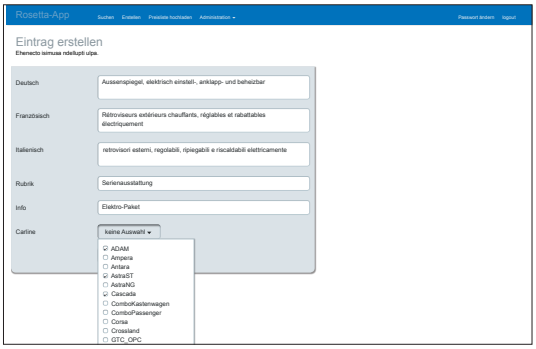


Abb. 24: Screendesign: neuer Eintrag/manuell

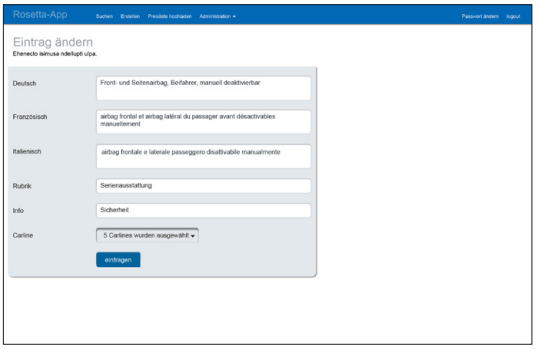


Abb. 25: Screendesign: bestehenden Eintrag bearbeiten

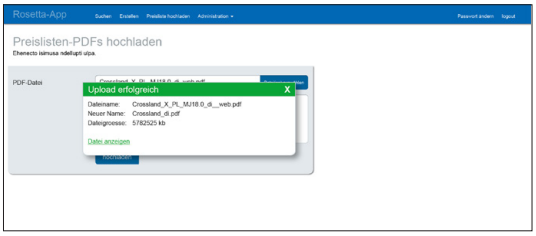


Abb. 26: Screendesign: erfolgreicher Upload eines PDFs

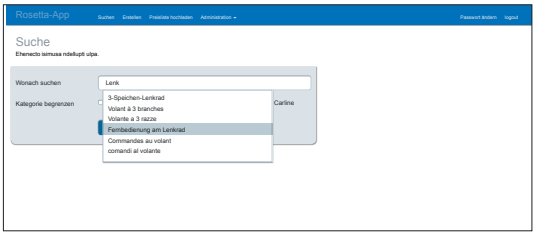


Abb. 27: Screendesign: Autovervollständigung bei der Suche

Das Design beinhaltet nicht die in einem ausgereiften Corporate Design verfügbaren Angaben zu Farbwerten, zum Raster usw. Ich habe im Rahmen dieser Projektarbeit kein CD entwickelt, vielmehr habe ich die für die Implementierung nötigen Informationen den einzelnen Screens entnommen.

– ClickDummy InVison

Sobald das Screendesign abgeschlossen war, konnte ich mittels der einzelnen Screens und InVision einen interaktiven Prototypen erstellen.

Vgl. InVision, InVisionApp, prototyping, collaboration & workflow platform, [online] https://www.invisionapp.com/ [19.5.2017]

Mithilfe dieses Prototyps war es mir möglich, das Projekt in seinem späteren Erscheinungsbild zu präsentieren. Es wird die Anwendung mit seinen Funktionen aus der Sicht eines Nutzers mit allen Rechten simuliert.

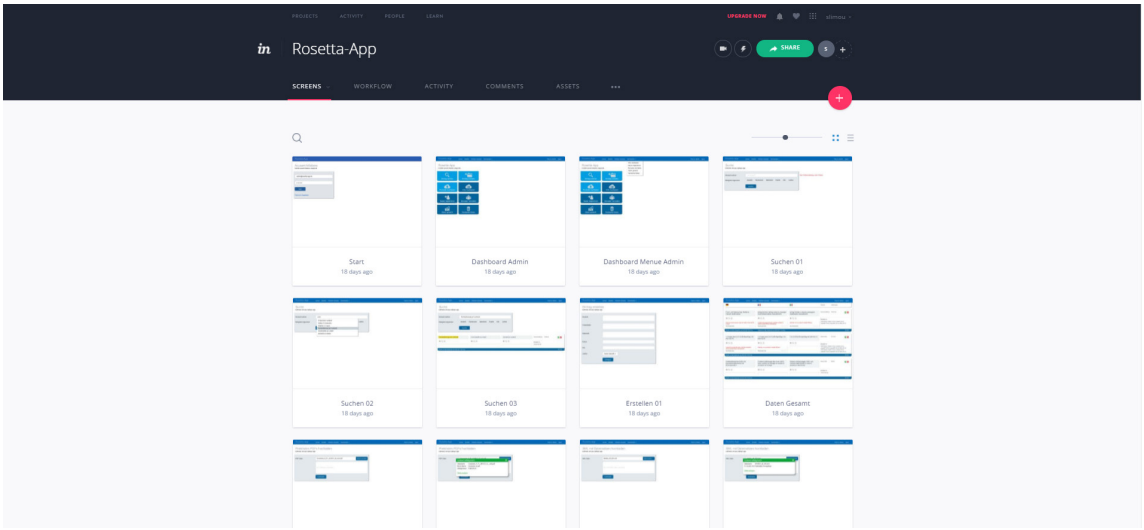


Abb. 28: Screenshot: Sequenzen bei InVisionApp.

– Fertige Anwendung

Orientierend am Aufbau des Prototyps und dem fertigen Screendesign konnte ich mit der Implementierung des Systems starten.

Hierbei galt es für mich vor allem einen „sauberen“ Code zu generieren. Ein Problem beim ursprünglichen Prototyp war es, dass dieser nur für die beiden Browser Google Chrome und Safari voll funktionsfähig ist. Das konnte ich bei der Implementierung des Systems beheben. Die Anwendung zeigt jetzt bei allen gängigen Browsern das gleiche Verhalten.

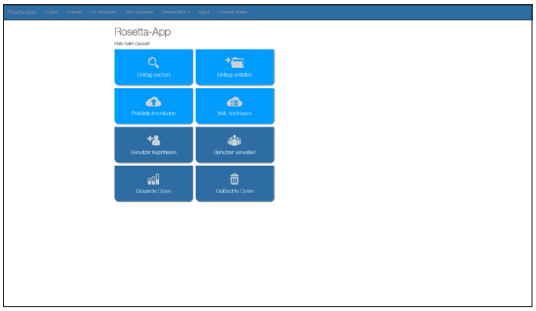


Abb. 29: Implementiertes System: Dashboard (Stand: 4.6.2017)

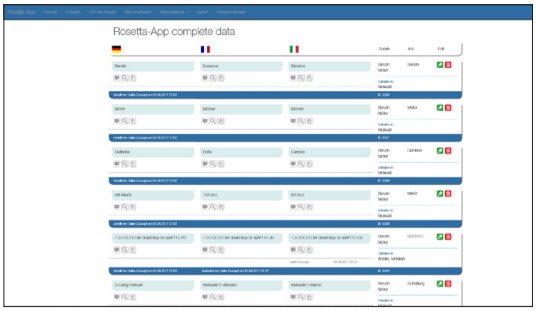


Abb.30: Implementiertes System: Anzeige der gesamten Daten (Stand: 4.6.2017)

Der geneigte Leser kann nach Belieben die fertige Anwendung testen.

Rosetta-App ist online abrufbar unter www.rosetta-app.de

Es sind zwei Beispiel-Nutzer registriert, welche verschiedene Rechte haben:

Administrator:	admin@rosetta-app.de	Passwort: admin
User:	user@rosetta-app.de	Passwort: user

– Features

Zum Abschluss dieses Kapitels stelle ich nachfolgend ausgewählte Features, die ich in der Rosetta-App implementiert habe, in Kurzform vor.

- **Datenpflege**
  Eine entscheidende Frage in der Entwicklungsarbeit war es, wie der große Bestand an Daten möglichst fehlerfrei in das System eingepflegt werden kann.

  Ich habe zu diesem Zweck von dem textbasierten Austauschformat XML Gebrauch gemacht. Es ist aktuell möglich, eine strukturierte XML-Datei aus dem Frontend der Anwendung zu importieren, sodass der Datenbestand mit den dort definierten Informationen erweitert wird.

```xml
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>

<items>
    <item>
        <item_de>Adaptives Bremslicht</item_de>
        <item_fr>Feux de stop adaptatifs</item_fr>
        <item_it>Luce di stop adattiva</item_it>
        <category>Serienausstattung</category>
        <info>Sicherheit</info>
        <carline>AstraST</carline>
    </item>
</items>
```

Abb. 31: XML-Struktur mit Beispiel-Datensatz

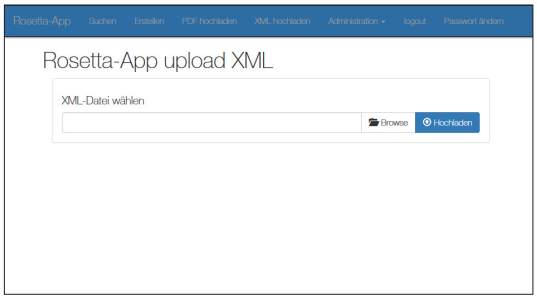


Abb. 32: Screenshot: Interface zum Hochladen einer XML-Datei (Stand: 4.6.2017)

Ein wesentlicher Vorteil dieser Herangehensweise ist es, dass diese XML-Dateien direkt aus Adobe InDesign exportiert werden können. InDesign ist das Layout und Satzprogramm, mit dem alle Publikationen erstellt werden. Auf diese Weise können beliebig viele Items per Knopfdruck in Rosetta-App integriert werden. Der Begriff „Item“ steht im aktuellen Fall für eine abgeschlossene Dateneinheit im XML-Baum.

- **PDF hochladen**
  Aufgrund dessen, dass nahezu jeder Datensatz seinen Ursprung in einem vorhandenen Dokument hat, bzw. dort in einem bestimmten Kontext steht, ist es möglich, zu jedem Eintrag PDF Dokumente zu referenzieren bzw. hochzuladen.

  Dies ist aktuell so gelöst, dass zu jedem Eintrag die Carline gewählt wird, bei der der Eintrag verwendet wurde. Diese Carlines werden angezeigt und bilden Links zu den eigentlichen Dokumenten.

  Wird keine Carline angegeben, wird auf ein Gesamt-Dokument verlinkt, welches die Seiten aller verfügbaren Preislisten beinhaltet.

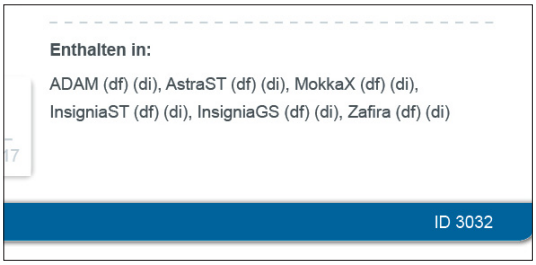


Abb. 33: Screenshot: Pro Carline wird jeweils ein Link zur Deutsch-Französischen und Deutsch-Italienischen Preisliste generiert.

Die Generierung des Links erfolgt automatisch, d. h. es ist nicht nötig, die Preisliste nach festgelegtem Muster benannt hochzuladen. Rosetta-App erkennt eigenständig den Fahrzeugnamen und die verwendete Sprache und nennt das Dokument entsprechend um, das Ziel-Muster lautet wie folgt: carline_language.pdf

- **Benutzerverwaltung**
  Für die Anwendung ist nicht vorgesehen, dass ein Benutzer sich selbst registriert, dies erfolgt ausschließlich durch einen Nutzer mit Administrator-Rechten (Admin).

  Ein Admin hat auch im laufenden System Zugriff auf die Daten eines jeden Nutzers, ausgenommen auf das Passwort, dennoch kann der Admin das Passwort eines Nutzers neu setzen. Jeder beliebige Nutzer kann zu jeder Zeit im Sinne seiner Berechtigungen herauf- bzw. herab gestuft werden. Aktuell existieren zwei Hierarchie-Ebenen: Eine Admin-Ebene und eine „gewöhnlicher" Nutzer-Ebene.

  Jeder Nutzer hat die Möglichkeit, sein eigenes Passwort bei laufender Session selber zu ändern. Sollte der Nutzer sein Passwort vergessen haben, kann er dies über den persönlichen Kontakt zu einem Administrator neu beantragen oder über die „Passwort vergessen"-Funktion eigenständig organisieren. Hierbei wird dem Nutzer eine Email an die von ihm angegebene Email-Adresse gesendet, sofern diese im System existiert. Als Sicherheitsaspekt wird bei der Anforderung eines neuen Passworts ein „zufällig" generierter String in die Spalte „password_code" des Tupels des Nutzers eingetragen. Ruft der Nutzer den ihm per Email gesendeten Link auf, erfolgt zunächst ein Abgleich dieses Codes mit dem über die GET-Methode übergebenen Code. Sofern beide Codes identisch sind, kann der Nutzer sich ein neues Passwort eintragen.

Abb. 34: Interface zum beantragen eines neuen Passworts

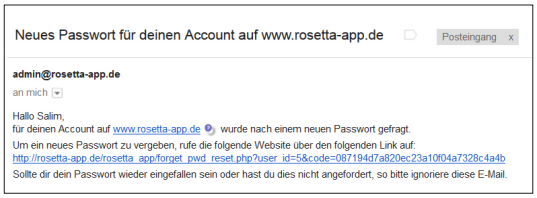


Abb. 35: Email mit Link zu Formular zum setzen des Passworts.

- **Datensatz kommentieren**
  Jeder Datensatz kann kommentiert werden. Dies ermöglicht es, bestimmte Datensätze von ähnlichen zu differenzieren bzw. eine bestimmte Schreibweise/Übersetzung zu priorisieren. Der Urheber und Zeitpunkt des Kommentars wird im System angezeigt.

- **Daten filtern**
  Jeder angezeigte Datensatz kann per Klick auf das jeweilige Icon gefiltert werden. Es wird eine Anfrage an die Datenbank gestartet und alle Daten ausgegeben, die ebenfalls den übergegeben String enthalten.

- **Text in Zwischenablage kopieren**
  Als besonders nützlich stellt sich heraus, den Text eines bestimmten Datensatzes auf einfache Weise in die Zwischenablage kopieren zu können.

  Im laufenden Arbeitsprozess kann so die gewünschte Übersetzung/Formulierung z. B. direkt für die Verwendung in Adobe InDesign genutzt werden.

  Jeder Spalte im Datensatz wird via PHP eine eindeutige ID mitgegeben, welche dazu dient, mittels Klick auf den Verweis/das Icon die JavaScript-Funktion „copyToClipboard()" aufzurufen und somit den gewünschten Text in die Zwischenablage zu kopieren.

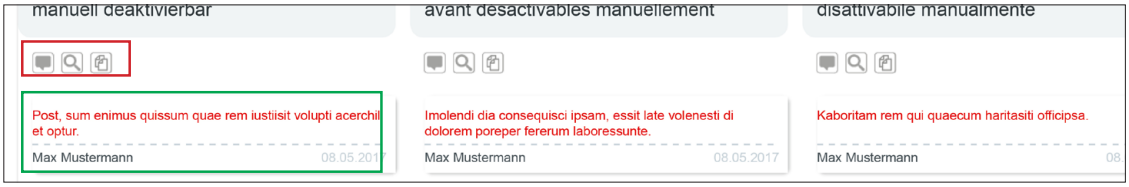


Abb. 36: Darstellung der Icons zum Kommentieren, Filtern und Kopieren in die Zwischenablage, hier rot umrandet von links nach rechts am Beispiel des Deutschen Datensatzes. Beispiel eines Kommentars mit Angabe des Urhebers und des Datums, hier grün umrandet.

9. ERGEBNISSE UND FAZIT

– Wissensmanagement-Tool

Ich setze an dieser Stelle das fertige Tool gleich mit einem vollwertigem und spezialisiertem Wissensmanagement-Tool. Im Gabler Wirtschaftslexikon wird der Begriff Wissensmanagement dahingehend definiert, dass sich Wissensmanagement mit dem Erwerb, der Entwicklung, dem Transfer, der Speicherung sowie der Nutzung von Wissen beschäftigt.

Vgl. Wissensmanagement, Definition, [online] http://wirtschaftslexikon.gabler.de/Definition/wissensmanagement.html#definition [29.5.2017]

In Anlehnung an SABIO des Hamburger Software-Unternehmens (www.sabio.de/) und dem Wissensmanagement-Tool Confluence des Australischen Software-Entwicklers Atlassian (https://de.atlassian.com/software/confluence) lässt sich auch Rosetta-App als ein Webbasiertes Wissensmanagement-Tool bezeichnen.

– Herausforderungen

Während der Arbeit an dem Projekt gab es zahlreiche unvorhersehbare Hürden und Schwierigkeiten, welche ich zum größten Teil erfolgreich lösen konnte. An dieser Stelle möchte ich auf zwei ausgewählte Probleme eingehen.

- **UTF Codierung**
  Ein Problem hat sich bei der Zeichen-Codierung der Deutschen Umlaute und bestimmter Zeichen der Französischen Sprache ergeben. Diese Zeichen müssen parallel im Frontend eingegeben/dargestellt, in der Datenbank abgelegt und wieder ausgegeben werden können. Gelöst wurde es dadurch, dass die Einträge in der Datenbank in der Kollation „utf8_general_mysql500_ci“ abgelegt werden und die PDO-Datenbank mittels „charset=utf8“ umgesetzt wird.

  Dennoch besteht dieses Problem bei der Suchfunktion weiterhin, insofern dass die gesuchten Texte im Fund mithilfe der PHP Funktion „preg_replace“ gehighlighted werden. Dies funktioniert bei Umlauten und allen anderen Zeichen problemlos. Das einzige Zeichen, wodurch ein Fehler geworfen wird, ist der Slash. Das habe ich aktuell so gelöst, dass zunächst geprüft wird, ob der gesuchte Begriff einen Slash enthält. Ist dies der Fall, wird nicht von der Funktion „preg_replace“ gebrauch gemacht, d. h. der Fund wird zum jetzigen Zeitpunkt und in diesem Fall nicht gehighlightet.

Abb. 37: Beispiel highlighting des gesuchten Begriffs im Fund.

- **XML-Import**
  Hier bestand die Schwierigkeit im Hochladen einer XML-Datei über ein Formular und dem anschließenden Einfügen der Daten in die Datenbank.

  Der Grund dafür war die „GENERAL ENTITY Declaration“. Es war deshalb nicht ohne Weiteres möglich, eine XML-Datei an einem beliebigen Ort im Verzeichnis abzulegen und von dort die Daten in die Datenbank zu schreiben. Ich habe das Problem so gelöst, dass die XML-Datei in das selbe Verzeichnis wie das Script geladen wird und von dort aus auch die Daten importiert werden. Anschließend wird die XML-Datei in das gewünschte Verzeichnis kopiert und die ursprünglich hochgeladene Datei gelöscht.

  Vgl. GENERAL ENTITY Declaration, ENTITY Declaration, [online] http://xmlwriter.net/xml_guide/entity_declaration.shtml [2.5.2017]

– Erfahrungen

Wie bei den Ausführungen zu den Herausforderungen werde ich mich bei dem Gelernten auf zwei Punkte konzentrieren.

- **Projektarbeit**
  Das Projekt im Ganzen hat sich als sehr umfangreich und lehrreich ausgezeichnet. Es galt nicht nur, die Anwendung zu programmieren, sondern es mussten viele weitere Aspekte berücksichtigt werden, sei es die Analyse und Planung im Vorfeld, die Zusammenarbeit mit den Stakeholdern, das Entwickeln des UX/UI-Designs und das parallele Dokumentieren der gesamten Arbeit.

- **PHP-Programmierung**
  Während der Projektarbeit konnte ich meine Kenntnisse in der Programmierung mit PHP auffrischen bzw. erweitern. Ich habe mich mit dem Klassenkonzept und dem Referenzieren einzelner Funktionen innerhalb dieser Klassen befasst, wodurch ich einen schlankeren Code schreiben konnte. Es war mir möglich, im Gegensatz zu dem Programm-Gerüst des Prototyps, welches ich lediglich zum Testen der Funktionalitäten geschrieben habe, einen sauberen Code zu generieren.

10. AUSBLICK

– Skalierbarkeit

Das System ist in seiner Funktion beliebig skalierbar, d. h. in seiner aktuellen Form kann es durch beliebig viele Sprachen erweitert werden und somit für den gesamten Europäischen Markt verwendet werden.Unabhängig davon ist Rosetta-App eine vielseitig anwendbare Software in einer flexiblen Form, d. h. es kann für ein weites Spektrum an Anforderungen genutzt und dahingehend transformiert werden. Dies könnten beispielsweise jegliche katalogisierte Datenbestände sein, welche auch Abbildungen enthalten.

– Versionierung

Analog zu der Technik, dass Datensätze nicht wirklich gelöscht werden, sondern lediglich mit einem anderen Wert in der Spalte „state“ belegt und demzufolge einfach nicht mehr angezeigt werden, kann man die jeweiligen Tupel mit einem Eintrag zu Versionierungszwecken versehen, sodass es möglich ist, zu jedem Zeitpunkt auf die unterschiedlichen Versionen einer Produktbezeichnung zurückzugreifen und diese ggf. wieder zu aktivieren.

Zum Zweck der Normalisierung sollte dies über verschiedene Tabellen gehandhabt werden, sodass keine vermeidbaren Redundanzen entstehen.

Vgl. Datenbanken verstehen, Normalisierung, [online] http://www.datenbanken-verstehen.de/datenmodellierung/normalisierung/ [8.6.2017]

– Vollautomatische Datenpflege

Die momentane Datenintegration von InDesign über XML nach Rosetta-App ist zwar unkompliziert und es kann ein beliebig großer Bestand an Daten schnell integriert werden. Dies bedarf dennoch einiger manueller Handlungen durch den Nutzer und eine wohlgeformte und genau strukturierte XML-Datei.

Wünschenswert ist eine automatische Synchronisation der Daten zwischen dem Satz- und Layout-Programm InDesign und einer nächsten Version der Rosetta-App als vollwertiges Redaktionssystem.

– Anpassungen aktuelles System

Neben der eben genannten Möglichkeiten, das bestehende System zu erweitern oder gar auf ein anderes Anwendungsgebiet wie das aktuell behandelte zu transformieren, besteht sicherlich in mancher Hinsicht „hier und da“ auch in der momentan laufenden Anwendung Bedarf an Anpassungen.

Diese dieses 10. Kapitel abschließende Punkte gliedere ich in zwei Kategorien: Zum Einen Aspekte aus Sicht der Usability und zum Anderen aus Sicht der Programmierung.

- **Usability**
  Ein nützliches Feature ist es, sich als Nutzer des Systems einen bestimmten Begriff direkt in einem Dokument anzeigen zu lassen und so den gesamten Kontext betrachten zu können. Aktuell ist dies auch möglich, indem der Anwender sich das zu jedem Eintrag verlinkte PDF der Preisliste öffnet und den Begriff dort über die Suchfunktion des Browsers anwählt. Eine m.E. weitaus komfortablere Herangehensweise wäre, wenn dieser Ablauf automatisiert wird. Dies könnte auf folgende Art umgesetzt werden: Der Nutzer wählt ein zu jedem Eintrag verlinktes PDF aus. Das PDF öffnet sich in dem aktuellen oder in einem neuen Tab und springt direkt zu der Fundstelle, welche gehighlightet wird.

  Ein weiterer Punkt ist, die Häufigkeit eines Begriffes/einer Formulierung innerhalb eines Dokuments anzeigen lassen zu können. Dies ist z. B. wichtig zur Argumentation bei dem Kunden, wenn erklärt werden soll, warum eine bestimmte Übersetzung einer anderen vorzuziehen ist.

  Als letzten Punkt im Bereich der Usability führe ich das Hochladen von PDF- bzw. XML-Dokumenten auf. Bislang ist es möglich die Dokumente einzeln hochzuladen, wesentlich effektiver wäre es, z. B. via drag&drop mehere Dokumente gleichzeitig in das System einspeisen zu können.

- **Programmierung**
  Bei der Programmierung habe ich mich dem Aufbau eines MVC-Patterns genähert, aber nicht vollends umgesetzt. Die unabhängige Funktion eines Controllers ist zum heutigen Tag nicht implementiert. Auch, wenn das System so abläuft, wie gewünscht, ist es mir dennoch ein Anliegen, mich in die Programmierung eines „sauberen" MVC-Patterns einzuarbeiten und dies anwenden zu können. Die Motivation, die dahinter steckt, ist, dass ich gängige Techniken sicher beherrschen möchte. Auch in Anbetracht dessen, dass ich zukünftig in einem studiumsrelevanten Gebiet beruflich tätig zu sein möchte.

  „‚PDF automatisch umbenennen' optimieren" ist mein letzter Punkt, den ich erwähnen möchte. Die Technik, die ich zu diesem Zweck angewandt habe, funktioniert und die PDF-Daten werden beim Hochladen innerhalb eines bestimmten Musters unbenannt. Dies wird zur Zeit bedingt durch gewisse Gegebenheiten der vorhandenen Namensgebung des Dokuments, das hochgeladen werden soll und erfolgt mithilfe einer imperativen Programmierung.

  Ein sehr reizvoller Gedanke ist für mich der Bereich der Artificial Intelligence, respektive des Machine Learnings, welche zur Lösung solcher Probleme anwendet werden könnte.

In den oben genannten Punkten spiegeln sich einige Anreize, die sich für mich in der Projektarbeit ergeben haben und, welche ich in einem fortlaufenden und von der universitären Projektarbeit unabhängigen Entwicklungs-Prozess umsetzen möchte, sei es speziell im aktuellen Fall oder in einem folgenden Projekt.

– Weiterentwicklung

Zum Abschluss dieser Dokumentation sei nochmals erwähnt, dass ich das beschriebene System bis heute auf einem lokalen Server an meinem Arbeitsplatz und zuhause entwickelt habe. Zur Ansicht und zum Testen stand und steht Rosetta-App dem ausgewählten Kreis aus den Stakeholder und mir auf meinen privat angemieteten Server zur Verfügung.

Auf diese Weise hatte und hat gleichzeitig der Betreuer des Praxisprojekts jederzeit Zugriff auf das online laufende System.

Diesen aktuellen Stand möchte ich zudem zur Bewertung des Praxisprojektes geltend machen, d. h. der eingereichte Code ist mit dem unter rosetta-app.de laufenden identisch.

Im weiteren Lebenszyklus des Systems, auch im Hinblick auf die kontinuierliche Pflege der kompletten Daten und der Nutzerkonten, sollte das System auf den unternehmensinternen Server von Typodrom installiert werden.

11. QUELLEN UND VERZEICHNISSE

Literatur-Verzeichnis

Stein von Rosette
Prof. Dr. phil. Wolfgang Schenkel formulierte in „Die Entzifferung der Hieroglyphen und Karl Richard Lepsius“
http://archiv.ub.uni-heidelberg.de/propylaeumdok/2886/1/Schenkel_Entzifferung_2012.pdf

Ziele und Anforderungen an das Praxis–Projekt
http://www.f10.th-koeln.de/campus/institute/informatik/studium/praxisprojekte/
https://prof.beuth-hochschule.de/edlich/praxisprojekt-online/

Als Hauptsprachen genannte Sprachen in der Schweiz
https://www.bfs.admin.ch/bfs/de/home/statistiken/bevoelkerung/sprachen-religionen/sprachen.html

Regel des Vorgehensmodells XP
http://www.extremeprogramming.org/rules/customer.html

Stakeholder
http://agiles-projektmanagement.org/scrum-stakeholder/

Extreme Programming
http://www.extremeprogramming.org/

Scrum
https://www.it-agile.de/wissen/einstieg-und-ueberblick/scrum/
http://wirtschaftslexikon.gabler.de/Definition/scrum.html

Requirements–Engineering
http://m.wirtschaftslexikon.gabler.de/Definition/requirements-engineering.html

PHP Data Objects
http://php.net/manual/de/intro.pdo.php

PHP MVC
http://tutorials.lemme.at/mvc-mit-php/#grundlagen

Bootstrap-Multiselect
https://github.com/davidstutz/bootstrap-multiselect/blob/master/dist/js/bootstrap-multiselect.js,
BSD 3-Clause License: Copyright (c) 2012 - 2015 David Stutz

Autocomplete
http://code.jquery.com/jquery-1.10.2.js, Copyright 2005, 2013 jQuery Foundation, Inc. and other contributors,
Released under the MIT license, http://jquery.org/license

UML
https://www.fbi.h-da.de/labore/case/uml.html

Wissensmanagement-Tool
http://wirtschaftslexikon.gabler.de/Definition/wissensmanagement.html#definition

GENERAL ENTITY Declaration
http://xmlwriter.net/xml_guide/entity_declaration.shtml

Datenbanken verstehen, Normalisierung
http://www.datenbanken-verstehen.de/datenmodellierung/normalisierung/

Programm-Verzeichnis

PHP Storm (JET BRAINS)

XAMMP (Apache Friends)

MariaDB (The MariaDB Foundation)

Github (Github Inc.)

Trello (Trello Inc.)

Astah Community (JUDE)

Adobe InDesign, Illustrator, Photoshop (Adobe Creative Cloud)

InVision (InVision App)

Bild-Nachweise

freepik, http://www.freepik.com/free-icons
Free vector icons for personal and commercial use

openclipart, https://openclipart.org
Public Domain Dedication

seeklogo, https://seeklogo.com/vector-logo/108600/php
Creative Commons Attribution 3.0

Abbildungs-Verzeichnis

Tabellen-Verzeichnis