

Beuth Hochschule für Technik Berlin
Medieninformatik Online
Bachelorstudiengang am Fachbereich VI

Dokumentation Praxis-Projekt
Sprachspezifische Produktpflege (Rosetta-App)

Salim Oussayfi
3. Juli 2017

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

EINLEITUNG

Die vorliegenden Seiten dienen dazu, mein Praxis-Projekt 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 dazu dient, einen großen Pool an produktspezifischen 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 wird der tägliche Workflow des Auftraggebers optimiert und zeitgleich die Produktionskosten gesenkt.

Die Software sehe ich in seiner Funktion und Nutzen als Analogie zu dem „Stein von Rosette“. Prof. Dr. phil. Wolfgang Schenkel formulierte 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 Priester-synode des Jahres 196 v. Chr., den jugendlichen König Ptolemäus V. Epiphanes für seine dem Volk gewährten Vergünstigungen in verschiedenster Weise zu ehren. ...“

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“, unter: http://archiv.ub.uni-heidelberg.de/propylaeumdok/2886/1/Schenkel_Entzifferung_2012.pdf (abgerufen am 14.5.2017)
Originalveröffentlichung in: Verena M. Lepper und Ingelore Hafemann (Hg.),
Karl Richard Lepsius. Der Begründer der deutschen Ägyptologie (Kaleidogramme 90), Berlin 2012, S. 37–78

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

Aus diesem Grund habe ich die Anwendung „Rosetta-App“ genannt. Der Zusatz „App“ 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 dieser Rosetta-App veranschaulicht, ist chronologisch aufgebaut, d. h. ich beginne mit den allgemeinen Voraussetzungen für das akademische Praxis-Projekt und begebe mich anschließend Schritt für Schritt durch die einzelnen Phasen des Entwicklungsprozesses – angefangen bei der Definition der Unternehmensziele bis hin zur Implementierung der fertigen Anwendung.

Im Anschluss daran werde ich die Dokumentation mit einem Fazit und möglicher Ausblicke zur Weiterentwicklung der Software beenden.

Zum Abschluss dieser Einleitung möchte ich noch auf die beiden ausschlaggebenden Parteien eingehen, die in diesem 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, 65423 Rüsselsheim am Main – im weiteren Verlauf dieser Dokumentation als Opel bezeichnet.

Opel ist hier 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.

GLIEDERUNG/INHALT

1. Aufgabenstellung	S. xx
1.1 Ziele und Anforderungen an das Praxis-Projekt	S. xx
1.2 Benefit Studenten	S. xx
2. Zielsetzung/Unternehmensziele	S. xx
2.1 Zielsetzung	S. xx
2.2 Benefit Unternehmen	S. xx
3. Analyse-Phase	S. xx
3.1 Beschreibung der Ausgangssituation/Ist-Analyse	S. xx
3.2 Vision und Systemidee/Soll-Konzept	S. xx
4. Vorstudie und Marktanalyse	S. xx
4.1 Umweltanalyse	S. xx
4.2 Akteure identifizieren	S. xx
4.3 Umfeld	S. xx
5. Konzept der Qualitätssicherung	S. xx
5.1 Requirements-Engineering-Tool	S. xx
5.2 Iterative Herangehensweise	S. xx
5.3 Retrospektive mit Stakeholdern	S. xx
5.4 Testen	S. xx
6. Systemstruktur	S. xx
6.1 Infrastruktur	S. xx
6.2 Sprache	S. xx
6.3 Frameworks/Libraries	S. xx
6.4 Tools	S. xx
7. Design	S. xx
7.1 UML-Diagramme	S. xx
7.2 Prototyp	S. xx
7.3 Sidemap	S. xx
7.4 Datenstruktur	S. xx
8. Anwendung	S. xx
8.1 Screendesign	S. xx
8.3 ClickDummy InVision	S. xx
8.2 fertige Anwendung	S. xx
8.2 Features	S. xx
9. Ergebnisse und Fazit	S. xx
9.1 Wissensmanagement-Tool	S. xx
9.2 Herausforderungen	S. xx
9.3 Gelerntes	S. xx
10. Ausblick	S. xx
10.1 Skalierbarkeit	S. xx
10.2 Versionierung	S. xx
11. Quellen und Verzeichnisse	S. xx
11.1 Literatur-Quellen	S. xx
11.2 Programm-Verzeichnis	S. xx
11.3 Bild-Nachweise	S. xx
11.4 Abbildungs-Verzeichnis	S. xx
11.5 Tabellen-Verzeichnis	S. xx

1. AUFGABENSTELLUNG

- Ziele und Anforderungen an das Praxis-Projekt

In diesem ersten Abschnitt gehe ich auf die Ziele und Anforderungen des Praxis-Projekts ein.

Im Allgemeinen dient das Praxis-Projekt dazu, das im Studium erlernte sehr praxisnah anzuwenden. Die Technische Hochschule Köln definiert die Ziele des Praxis-Projekts wie folgt: „Im Praxisprojekt sollen die Studierenden Methoden und Techniken, die sie im Studium erlernt haben, in einem realitätsnahen Projekt weitgehend selbstständig anwenden.“

Technische Hochschule Köln: „Praxisprojekte“, unter:
<http://www.f10.th-koeln.de/campus/institute/informatik/studium/praxisprojekte/> (abgerufen am 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, genauer gesagt ein neues Software-System implementiert wird.

Nur mit dem Programmieren irgend einer Anwendung ist es allerdings nicht getan, es sollte ein weites Spektrum 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.

Prof. Dr. Stefan Edlich: „Leitfaden für das Praxisprojekt (Online)“, unter:
<https://prof.beuth-hochschule.de/edlich/praxisprojekt-online/> (abgerufen am 21.5.2017)

Neben den genannten Ziele und der Anforderungen wird auch ein zeitlicher Aufwand für die Bearbeitung des Projektes vorausgesetzt. Ebenfalls im zuvor zitierten „Leitfaden für das Praxisprojekt (Online)“ werden folgende Anforderungen zum Aufwand gennant: „Das Praxisprojekt im 5. Semester wird mit 15 Credits bewertet. Pro Credit werden als durchschnittlicher Arbeitsaufwand für die Studierenden 30 Zeitstunden berechnet. Sie müssen also 450 Stunden für die Bearbeitung des Projektes inkl. Bericht und Präsentation einplanen. Daraus ergibt sich renerisch natürlich auch eine Mindestanzahl von Tagen / Monaten für das Praktikum.“

Prof. Dr. Stefan Edlich: „Leitfaden für das Praxisprojekt (Online)“, unter:
<https://prof.beuth-hochschule.de/edlich/praxisprojekt-online/> (abgerufen am 21.5.2017)

- Benefit Student

Bei der Beschreibung des Benefits für die/den Studierenden möchte ich meine Ausführung einzig auf mich und meine Eindrücke zum Praxis-Semester beziehen und ich bin mir sicher, dass sich meine Ansichten mit denen vieler Kommilitonen decken.

Ich habe das Praxis-Projekt so erlebt, dass ich viele Erfahrungen gewonnen habe, sei es in der Projekt-Arbeit im Allgemeinen oder in der Programmierung im Speziellen.

Ich habe das komplettes Projekt von Anfang an geplant, begleitet und durchgeführt.

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 anwenden zu können.

Ich habe viele Techniken und Methoden aus meiner bisherigen Lehrzeit anwenden können, was das im Einzelnen ist, möchte ich anhand dieser Dokumentation zeigen.

Sicherlich gab es bis jetzt viele interessante und spannende Projekte im Studium, die sowohl im Team als auch als Einzelarbeit zu bearbeiten waren, aber im Umfang an und für sich und Anspruch an der nebenläufigen Organisation des Projekts stellt sich die hier beschriebene Arbeit als eine bedeutende Erfahrung für mich heraus.

2. ZIELSETZUNG/UNTERNEHMENSZIELE

- Zielsetzung

In diesem Abschnitt beschreibe ich die Zielsetzung und die Motivation seitens des Auftraggebers 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>) sollte 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.

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 Urheber des Eintrags angezeigt werden als auch die Nutzer, die den Daten geändert oder gelöscht haben.

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 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 prompt genutzt werden können.

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

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

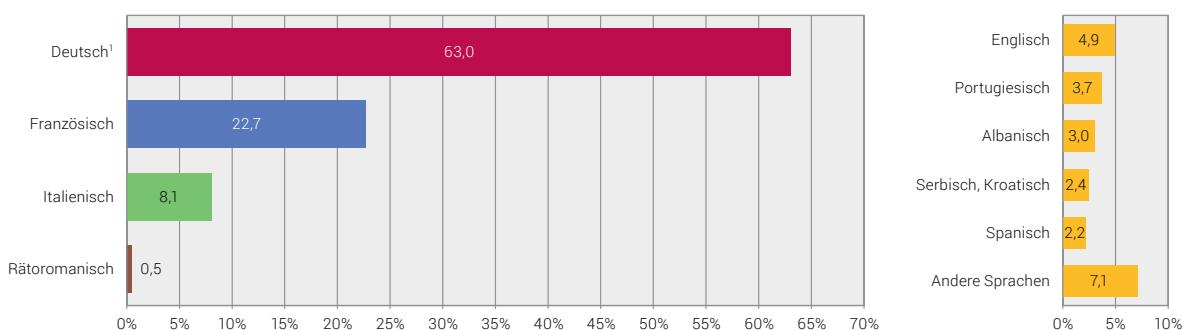
TYPODROM betreut in der Funktion einer Tagesgeschäftsagentur den Kunden Opel.

TYPODROM generiert für Opel Printmedien wie z. B. Kataloge, Anzeigen, POS-Materialien und weitere – auf eine 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 – es sei hier nur erwähnt, dass die Werbematerialien für den gesamten europäischen Raum kreiert und produziert werden.

Für das Praxis-Projekt 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.
Dies sind die Sprachen Französisch, Italienisch und Rätoromanisch.

Die folgende Grafik (Veröffentlicht am 27.03.2017) des Bundesamt für Statistik 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)

© BFS 2017

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

Bundesamt für Statistiken (Schweiz): „Als Hauptsprachen genannte Sprachen, 2015“, unter: <https://www.bfs.admin.ch/bfs/de/home/statistiken/bevoelkerung/sprachen-religionen/sprachen.html> (abgerufen am 27.5.2017)

Opel führt in der Schweiz alle Publikationen in den drei Sprachen Deutsch, Französisch und Italienisch.

Bei der initialen Entwicklung des Systems habe ich zunächst 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 erwähnt, das Projekt unter der Prämisse der Preislisten gestartet habe, werde ich diese Dokumentation auch an diesen Preislisten ausgerichtet weiterführen.

Für jeden Fahrzeugtyp gibt es eine 16 – 28 seitige Preisliste. Jedes dieser Dokumente gibt es in zwei Sprach-Versionen, d.h. es gibt jede Preisliste in Deutsch/Französisch und analog dazu in Deutsch/Italienisch. Je nach Modell-Jahr führt Opel 18 – 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 lässt sich sagen, die Preislisten befassen sich mit allen Fahrzeugteilen, die nicht dem Auto als Ganzes entsprechen.

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

Die folgende Grafik gibt einen Überblick darüber, inwiefern die Abteilungen im Workflow involviert und miteinander verknüpft sind und beschreibt die jeweiligen Stationen.

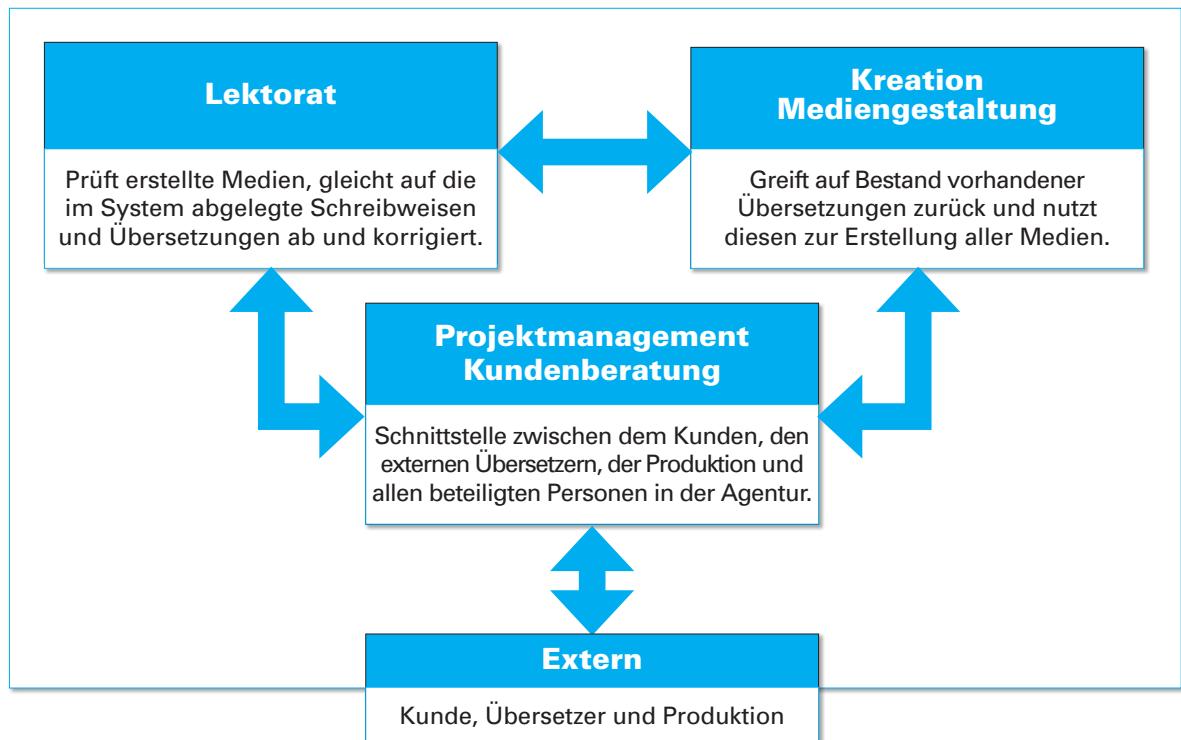


Abb. 2: Grafik veranschaulicht den Agentur internen WorkFlow.

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 bzw. 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 und Redundanzen, zudem ergibt sich hieraus ein erheblicher zeitlicher Mehraufwand.

- Vision und Systemidee/Soll-Konzept

Es sollte 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 Erscheinungszeitpunkt und einer Verlinkung zu dem eigentlichen Dokument. Bei dem hier behandelten Praxis-Projekt 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, des Weiteren 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 sei hier der Google-Übersetzer (<https://translate.google.com>), Linguee (<http://www.linguee.de/>) oder Leo (<http://www.leo.org/>) zu nennen.

Das gilt allerdings nur in einem Bereich, in dem die Anforderungen ausschließlich der Weitergabe von Informationen dienen und eine zuverlässliche Interpretation vorausgesetzt werden darf. Sobald auf professionelle Weise Produkte vertrieben werden, ist diese Herangehensweise aus unternehmerischer Sicht nicht mehr empfehlenswert.

Es sollten ausgebildete Übersetzer, zudem meist „Muttersprachler“ die Texte in die jeweilige Sprache transformieren.

Im Fall des hier behandelten Projektes kommt erschwerend hinzu, dass es sich zum Teil um fachspezifische Bezeichnungen/Wortschöpfungen handelt, die seitens des Kundens 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.)
- FlexFix®-Heckfahrradträger

Es bietet sich also an, diese Produktbezeichnungen, inklusive aller atomaren Sprach-einheiten 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 bedienbare Nutzeroberfläche gewährleistet ist.



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

Front- und Seitenairbag, Beifahrer, manuell deaktivierbar	airbag frontal et airbag latéral du passager avant désactivables manuellement	airbag frontale e laterale passeggero disattivabile manualmente	Serienausstattung	Sicherheit
Post: sum enimus quissum que rem iustisit volupti acerchi et optur. Max Mustermann 08.05.2017	Imodiendi dia consequisci ipsam, essit late volenesti di dolorem poroper feretur laboresse. Max Mustermann 08.05.2017	Kaboritam rem qui quaecum haritasiti officipa. Max Mustermann 08.05.2017	Enthalten in: ADAM (df) (di), AstraST (df) (di), MokkaX (df) (di), InsigniaST (df) (di), InsigniaGS (df) (di), Zafira (df) (di)	
Erstellt von Martina Musterfrau am 03.05.2017 16:31 Uhr	Geändert von Max Mustermann am 05.05.2017 15:13 Uhr		ID 3032	

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, gegeben-falls Vorort beim Kunden auf den Datenbestand zugreifen zu können, was in der Regel auch am Notebook funktioniert, aber hier zur Abdeckung aller Eventualitäten auf dem Mobiltelefon gewehrleistet ist.

5. KONZEPT DER QUALITÄTSSICHERUNG

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

Folgend nenne ich kurz deren jeweilige disjunkten Funktionen im Unternehmen und erläutere, in welcher Relation sie mit dem System stehen werden:

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

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

Nicole R./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 gesammelte Fragen stellen und habe gleichzeitig Feedback und Input/Ideen zur weiteren Herangehensweise erhalten.

Auf diese Art und Weise habe ich gleichzeitig eine Regel des Vorgehenmodells 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 „rueles/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.“

Extreme Programming: A gentle introduction: „The Customer is Always Available“, unter:
<http://www.extremeprogramming.org/rules/customer.html> (abgerufen am 27.5.2017)

Im weiteren Verlauf der Dokumentation werde ich die drei zuvor genannten Kollegen als die Stakeholder des Systems bezeichnen.

Stakeholder sind Personen bzw. Gruppen von Personen mit einem besonderen Interesse am Ergebnis eines Prozesses, wie es auf „agiles-projektmanagement.org“ definiert ist.

Agiles Projektmanagement, Scrum verstehen und erfolgreich anwenden: „Die Rolle der Stakeholder in Scrum“, unter:
<http://agiles-projektmanagement.org/> (abgerufen am 27.5.2017)

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

Extreme Programming: „A gentle introduction“, unter:
<http://www.extremeprogramming.org/> (abgerufen am 27.5.2017)

Im überwiegenden Verlauf der Projekt-Entwicklung habe ich mich allerdings an dem agilen Rahmenwerk Scrum orientiert. <https://www.it-agile.de/wissen/einstieg-und-ueberblick/scrum/>

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.

Scrum: „Definition“, unter: <http://wirtschaftslexikon.gabler.de/Definition/scrum.html> (abgerufen am 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

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, wie es im Gabler Wirtschaftslexikon definiert ist.

Requirements Engineering: „Definition“, unter:
<http://m.wirtschaftslexikon.gabler.de/Definition/requirements-engineering.html> (abgerufen am 27.5.2017)

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

Bleibt man bei dem Vergleich zum Vorgehensmodell Scrum, möchte ich die jeweiligen Requirements gleichsetzen 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 Teilnehmer lediglich der Titel und eine Kurzbeschreibung der jeweiligen Vorgabe eingetragen werden.

Der Nutzerkreis des Requirements-Engineering-Tools war auf vier Personen beschränkt, dies waren die drei Stakeholder und ich.

The screenshot shows a web-based application for managing requirements. At the top, there is a form with fields for 'Author' (dropdown menu), 'Titel' (text input: 'drag&drop'), and 'Beschreibung' (text area). Below the form is a section for 'Verbindlichkeit' (checkboxes: 'muss (Pflicht)', 'soll (Wunsch)', 'kann (Absicht)', 'wird (Vorschlag)'). A 'Submit' button is located at the bottom of the form. Below the form is a table titled 'Requirements Rosetta-App'. The table has columns: 'Verbindlichkeit', 'ID', 'Datum/Uhrzeit', 'Author', 'Titel', and 'Beschreibung'. The rows show various entries with different colors (yellow, red, orange) indicating the level of commitment. The last row (ID 94) is highlighted with a yellow background.

Verbindlichkeit	ID	Datum/Uhrzeit	Author	Titel	Beschreibung
kann	64	21.02.17 19:13:40	Salm	Zwischenablage	Einträge sollen per Klick auf Schaltfläche in die Zwischenablage gespeichert werden können
muss	65	21.02.17 19:59:36	Salm	Suche	Es soll nach Einträgen gesucht werden können
muss	69	26.02.17 11:43:27	Salm	Benutzerverwaltung	Alle Nutzer des Systems haben eigenen Account
muss	77	01.05.17 07:42:29	Thomas	Transparenz	es muss ersichtlich sein, wer Änderungen zu welchem Zeitpunkt vorgenommen hat
muss	81	22.03.17 09:51:40	Nicole	Dateneingabe	Daten müssen automatisch angepflegt werden, nicht手工
muss	89	20.05.17 08:05:29	Nicole	Passwort ändern	Der Nutzer muss sich sein Passwort ändern können
muss	91	20.05.17 07:46:16	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	Salm	Carline/Produktcode	zu jedem Eintrag soll die Carline und der Produktcode ergänzt werden
soll	68	26.02.17 11:46:17	Salm	Autorenvoltständigung	Bei Eingabe in das Suchformular sollen (ab 3 Zeichen) Suchvorschläge angezeigt werden
soll	71	26.02.17 15:09:36	Salm	Einträge formatieren	evtl. zwei voneinander 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 Übersetzer	Bei zwei oder mehreren Übersetzungsvarianten Hervorhebungsmöglichkeit der bevorzugten Übersetzung durch den Übersetzer (Kommentarfunktion)
soll	75	01.06.17 21:17:37	Thomas	Nach Carline suchen?	Man soll die Begrifflichkeiten einer Carline zuweisen können
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	Salm	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	Salm	Interface zur Dateneingabe	- Nutzer soll XML-Daten über eine GU in die Datenbank einfügen können - Nutzer soll aktualisierte PDF in das System laden können
soll	87	19.04.17 15:48:48	Salm	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	Salm	Sprachen	deutsch, italienisch, französisch
wird	78	20.05.17 07:59:45	Salm	Nutzerverwaltung Admin	Der Admin hat vollen Zugriff auf alle Nutzerdaten
wird	94	20.05.17 07:59:52	Salm	Nutzerverwaltung Admin	Der Admin hat volllen Zugriff auf alle Nutzerdaten

Abb. 5: Screenshot des Requirement-Engineering-Tools. Online verfügbar unter: <http://requirements.rosetta-app.de/index.php>

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.

Da es sich bei diesem Tool um einen erforderlichen Nebenschauplatz handelt, möchte ich hier nicht weiter auf die Programmierung eingehen sondern verweise auf den für die Projekt-Entwicklung entscheidenden Inhalt dessen.

Mithilfe der folgenden Tabelle liste ich die einzelnen Vorgaben tabellarisch auf:

Requirements Rosetta-App		
Titel	Beschreibung	erledigt
Sprachen	deutsch, italienisch, französisch	
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	
Auto vervollständigung	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 Uebersetzung durch den Uebersetzer (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 Dateneinpflage	- 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 01.06.2017.

- **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 Vorherigesmodell Scrum.

In einem Sprint wird innerhalb einer festgelegten Dauer von typischerweise zwei bis vier Wochen ein funktionsfähiges Inkrement des Software-Produkts implementiert.

- **Retrospektive Stakeholdern**

In regelmäßigen Meetings mit den Stakeholdern wurde besprochen, was passiert ist. Die implementierten Features wurden getestet, beurteilt und ggf. eine Nachgebesserung in die Wege geleitet.

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 Aplikation. Nachdem ich die jeweiligen Anforderungen in der lokalen Umgebung implementiert habe, habe ich das System synchronisiert, sodass der lokale Status stets mit dem online laufendne 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 dann Teil der Retrospektiven. Seitens der Stakeholder wurde ausschließlich auf der Ebene der Usability bewertet. Mein Aufgabenbereich war/ist es zudem, dass das Programm sich so verhält, wie gewünscht 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 Auto vervollständigung, ein intelligentes Refactoring und vor allem eine komfortable Anbindung an mein für diese Arbeit erstelltes Repository auf github.com bietet.

PHP Storm, JET BRAINS: „Entwicklungsumgebung“, unter: <https://www.jetbrains.com/phpstorm/> (abgerufen am 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.

XAMPP, Apache Friends: „Serverumgebung“, unter: <https://www.apachefriends.org/de/index.html> (abgerufen am 2.5.2017)

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

MariaDB, The MariaDB Foundation: „freies, relationales Open-Source-Datenbankverwaltungssystem“, unter: <https://mariadb.org/> (abgerufen am 28.5.2017)

„MariaDB ist eine Abspaltung (Fork) von MySQL. Diese wurde entwickelt, nachdem Oracle Sun Microsystems im Jahre 2010 übernommen hatte.“
<https://wiki.ubuntuusers.de/MariaDB/>

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 zugehen zu können.

PHP-Handbuch, Funktionsreferenz, Datenbankerweiterungen, Abstraktionsebenen, PDO: „Einführung“, unter: <http://php.net/manual/de/intro.pdo.php> (abgerufen am 28.5.2017)

Daraus ergibt sich, dass man 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 MySQL und MySQLi.

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 für die Erstellung des Systems PHP verwenden.

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

Als Beispiel möchte ich die Verwendung von Klassen nennen, mithilfe derer ich z.B.: 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.

MVC mit PHP: „Grundlagen“, unter:
<http://tutorials.lemme.at/mvc-mit-php/#grundlagen> (abgerufen am 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 darzusetzen, welche dennoch flexibel ist. Die Folge ist eine ausgeglichene Darstellung für den User und gleichzeitig bleibt keinerlei Input im Verbogenen.

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.

Diese Feature erweist sich als besonders praktisch, sobald man bestimmte Daten aus dem System für die aktuelle Arbeit benötigt.

Zum hinzufügen 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.

Bootstrap Multiselect, David Stutz: „BSD 3-Clause License: Copyright (c) 2012 - 2015 David Stutz“, unter: <https://github.com/davidstutz/bootstrap-multiselect/blob/master/dist/js/bootstrap-multiselect.js> (abgerufen am 6.5.2017)

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

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/>“, unter: <http://code.jquery.com/jquery-1.10.2.js> (abgerufen am 6.6.2017)

- Frameworks/Libraries

- Bootstrap

Ich habe das freie CSS-Framework Bootstrap referenziert um so die in diesem Framework vordefinierte Klassen-Hierarchie nutzen zu können.

Überwiegend habe ich den Aufbau der Formulare, Buttons und der Navigation genutzt und diese teilweise individualisiert.

Im Besonderen habe ich mich für dieses Framework entschieden, weil 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.

Bootstrap, Twitter: „HTML, CSS, and JS framework for developing responsive, mobile first projects on the web“, unter: <http://getbootstrap.com/> (abgerufen am 2.5.2017)

- Tools

Neben der eigentlichen Entwicklungsumgebung PHP Storm und der Serverumgebung XAMPP von Apache Friends habe ich die folgenden Tools genutzt, die mich bei der Projekt-Arbeit unterstützen.

- 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.

Github, GitHub, Inc.: „kollaborative Versionsverwaltung“, unter: <https://github.com/> (abgerufen am 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 Projekt-Arbeit ergeben hat.

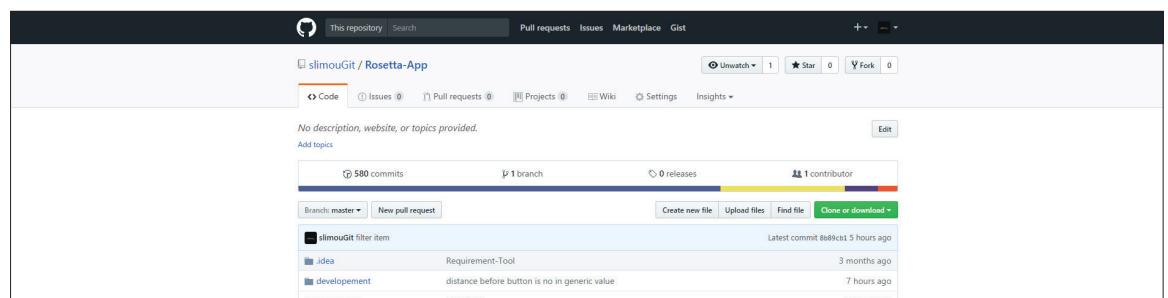


Abb. 6: Screenshot des Repository bei GitHub. Online verfügbar unter: <https://github.com/slimouGit/Rosetta-App.git>

- Trello

Ich habe mit Trello gearbeitet, um den gesamten Workflow zu organisieren.

Ich habe mir Listen zu verschiedenen Teilbereichen der Projekt-Arbeit erstellt und diese jeweils mit einzelnen Karten und Checklisten befüllt.

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

Trello, Trello Inc. (Fog Creek Software): „web-basierte Projektmanagementsoftware“, unter: <https://trello.com/> (abgerufen am 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 ein einfaches, 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.

Astah Community, JUDE (Java and UML Developers' Environment): „UML modellierungs Tool“, unter: <http://astah.net/editions/community> (abgerufen am 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.“

Hochschule Darmstadt: „Die Unified Modeling Language - UML“, unter:
<https://www.fbi.h-da.de/labore/case/uml.html> (abgerufen am 1.6.2017)
 Originalveröffentlichung in: Objektorientierter Softwareentwurf mit UML, Gabriele Bannert

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

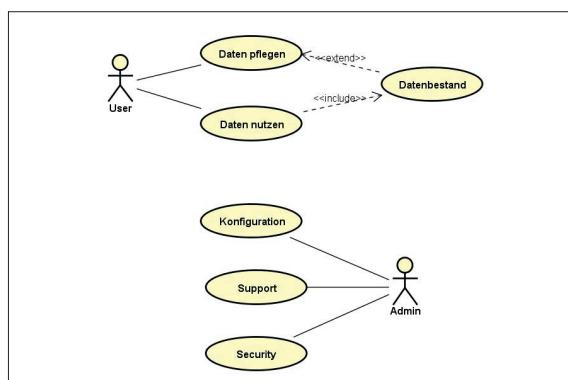


Abb. 8: Anwendungsfall-Diagramm. Online verfügbar unter:
<https://github.com/slimouGit/Rosetta-App/blob/master/developement/documentary/UML/UseCase.jpg>

Deployment

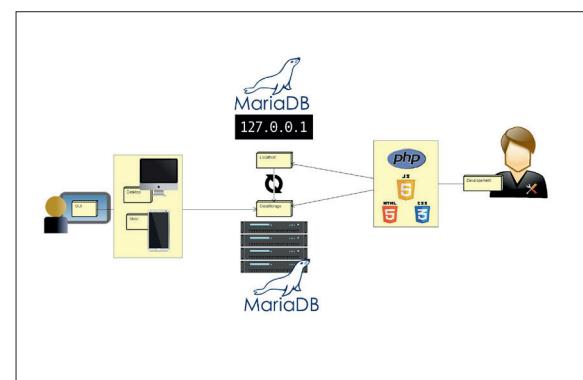


Abb. 9: Anwendungsfall-Diagramm. Online verfügbar unter:
<https://github.com/slimouGit/Rosetta-App/blob/master/developement/documentary/UML/Deployment.jpg>

Bilder: „User“, „Admin“, „Computer“, „Smartphone“: openclipart, „Public Domain Dedication“, unter: <https://openclipart.org> (abgerufen am 1.6.2017)
 Bilder: „PHP“, „JS“, „CSS“, „MariaDB“: seeklogo, „Creative Commons Attribution 3.0“, unter:
<https://seeklogo.com/vector-logo/108600/php> (abgerufen am 1.6.2017)

Component

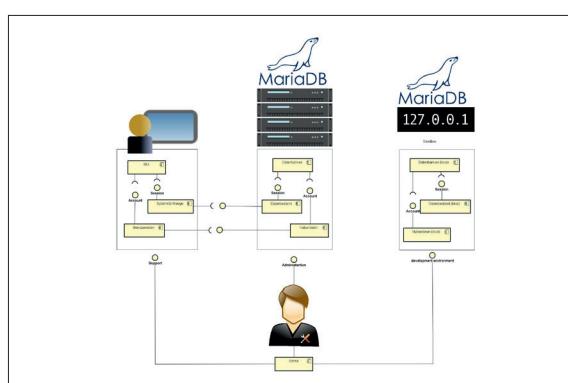


Abb. 10: Komponenten-Diagramm. Online verfügbar unter:
<https://github.com/slimouGit/Rosetta-App/blob/master/developement/documentary/UML/Component.jpg>

Bilder: „User“, „Admin“, „Server“: openclipart, „Public Domain Dedication“, unter: <https://openclipart.org> (abgerufen am 1.6.2017)
 Bilder: „MariaDB“: seeklogo, „Creative Commons Attribution 3.0“, unter:
<https://seeklogo.com/vector-logo/108600/php> (abgerufen am 1.6.2017)

Activity

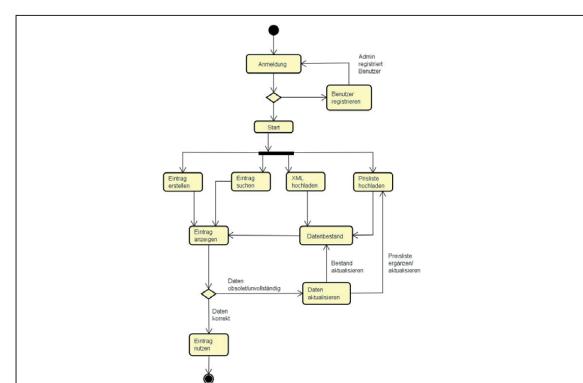


Abb. 11: Aktivitäts-Diagramm. Online verfügbar unter:
<https://github.com/slimouGit/Rosetta-App/blob/master/developement/documentary/UML/Activity.jpg>

- Prototyp

Mit der initialen Idee für mein Projekt und auf Basis des 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/beheben.

Bevor der Prototyp in seinen Funktionen weitestgehend abgeschlossen und funktionsfähig war, habe ich nicht mit der Implementierung des eigentlichen Systems begonnen.

Der Prototyp ist online abrufbar: <http://prototype.rosetta-app.de>

Es sind zwei Beispiel-User registriert, welche verschiedene Rechte habe:

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

Screenshots des initialen Prototyps

The screenshot shows a table with columns: ID, De, Fr, En, Rubrik, Info, Carine, and Edit. The table contains several rows of data, each with a detailed description in French and English, and checkboxes for selection in the last two columns.

Abb. 12: Screenshot: initiale Datenanzeige

The screenshot shows a form with fields for selecting language (Deutsch, Französisch, Italianisch, English) and checkboxes for selecting user roles (General, ADAM, ...). At the bottom are 'Eingehen' and 'Zurücksetzen' buttons.

Abb. 12: Screenshot: initiale Dateneingabe

Screenshots des abgeschlossenen Prototyps

The dashboard has several cards: 'Eintrag suchen' (Search entry), 'Eintrag erstellen' (Create entry), 'Preisliste hochladen' (Upload price list), 'XML hochladen' (Upload XML), 'Nutzer registrieren' (Register user), 'Benutzer verwalten' (Manage users), 'Daten gesamt' (Overall data), and 'Gelöschte Datensätze' (Deleted data).

Abb. 14: Screenshot: Dashboard

The search interface allows filtering by language (Deutsch, Französisch, Italianisch, English) and includes a table with columns: ID, De, Fr, En, Rubrik, InvCode, Carine, and Edit. Below the table is a section for 'Suche' (Search) and 'Ergebnisse' (Results).

Abb. 15: Screenshot: Suche und Datenausgabe

A table with columns: ID, Vorname, Nachname, Email, Authentifikation, Edit, and Passwort. The table lists 14 users, including the administrator and user from the initial prototype.

Abb. 16: Screenshot: Anzeige der Nutzer

The screenshot shows a form for 'Daten ändern' (Change data) with sections for 'Deutsch', 'Französisch', 'Italianisch', 'Rubrik', 'Info', and 'Carine'. It includes checkboxes for roles like 'Kontroll-Pilot Absegeln unter Bedienkreis (nicht mit Ladeneinschaltung)', 'Armeband Fahrerstuhl', 'Zwei-Zonen-Klimaautomatik', etc., and buttons for 'Eingehen' and 'Zurücksetzen'.

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 von Code 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 zum heutigen Tag noch nicht gesondert implementiert sind.

In der folgenden Grafik sind alle Seiten des Systems mit einer jeweiligen Kurzbeschreibung in eine Ordnerstruktur unterteilt.

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

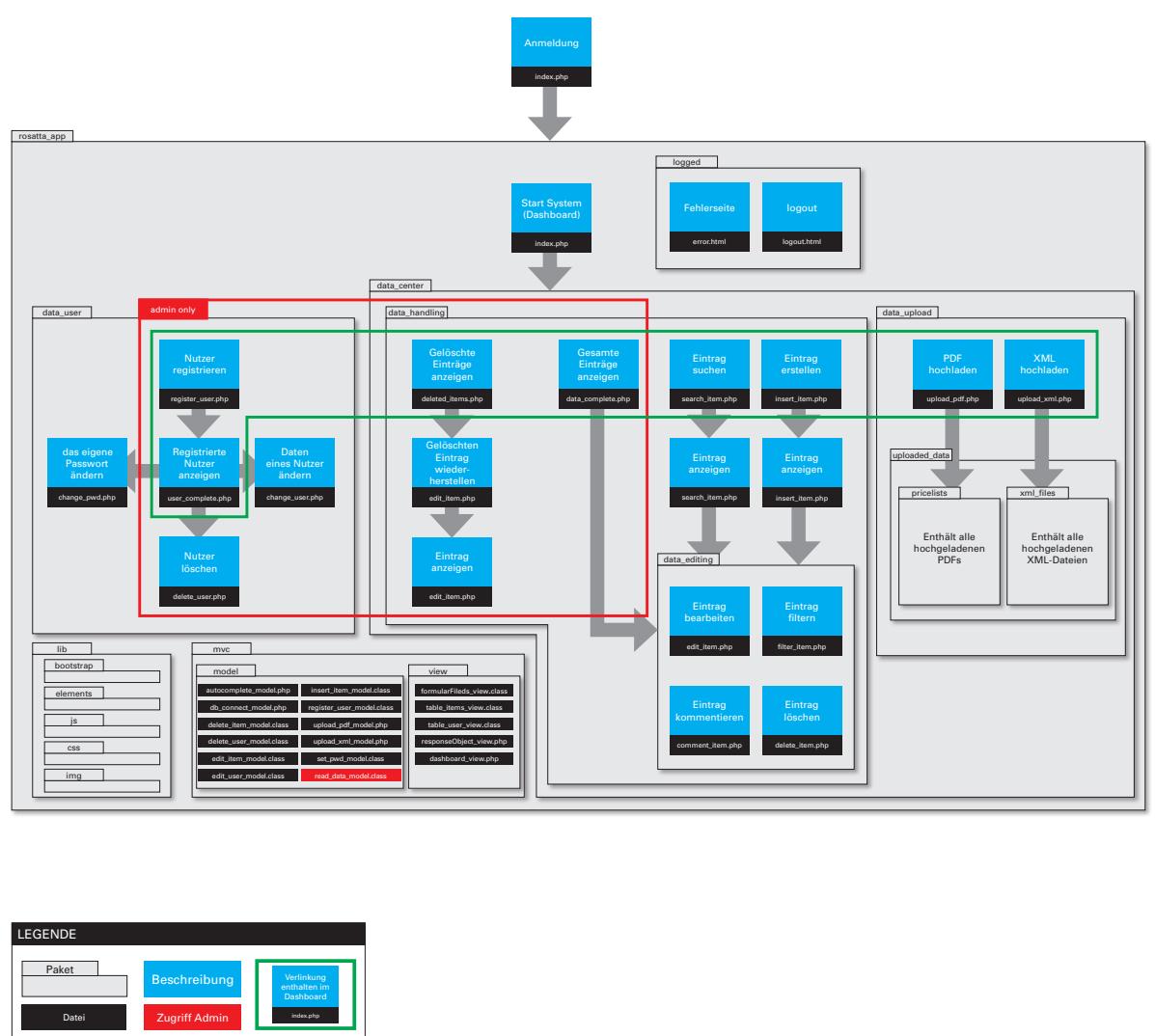


Abb. 18: Sidemap. Online verfügbar unter: https://github.com/slimouGit/Rosetta-App/blob/master/developement/sidemap/sidemap_Rosetta-App.pdf

- Datenstruktur

In den folgenden beiden Tabellen stelle ich die Datenstruktur/das Datenbankschema der Anwendung vor.

Mithilfe 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 Nutzernamen bzw. die Nutze-ID verwendet.

Neben der Bezeichnung der jeweiligen Spalten beinhaltet die folgende Abbildung den Datentyp und eine Kurzbeschreibung. Die in der folgenden Tabelle nicht dargestellte Kollation ist bei Textbasierten Werten immer auf „utf8_general_mysql500_ci“ gestellt. Die jeweiligen Primär-Schlüssel sind in beiden Tabellen „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 das 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 das 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 das 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 Bereich gehe ich auf die schrittweise Umsetzung des Systems ein.
Ich starte mit dem Screendesign und zeige, wie ich einen interaktiven Clickddummy mit Invision umgesetzt habe.

Anschließend stelle ich die fertige Anwendung vor und schließe diesen Abschnitt mit ein paar 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 20px Stege.

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

Bootstrap, Twitter: „Bootstrap grid examples“,
unter: <http://getbootstrap.com/examples/grid/> (abgerufen am 16.5.2017)

The screenshot displays a table-based interface for managing vehicle features. The columns are organized into a 12-column grid. The columns are labeled at the top: Suchen, Erstellen, Preisliste hochladen, Administration, Rubrik, Info/Code, Passwort ändern, and logout. The main content area contains three rows of data, each representing a different feature. Each row has four columns: German, French, Italian, and a summary column. The German and French columns contain detailed descriptions of the feature, while the Italian column provides a simplified version. The summary column includes a list of compatible car models (e.g., ADAM, Astra, MokkaX, Zafira). The interface uses a light blue header and a white background with red borders separating the columns. Action buttons (edit, delete, etc.) are located in the first two columns of each row.

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“, unter:
<http://www.adobe.com/de/creativecloud.html> (abgerufen am 1.6.2017)

Ich habe verschiedene Unterseiten entworfen, teilweise inklusive bestimmter Fallunterscheidungen wie zum Beispiel die Response des Systems nach dem Hochladen einer 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

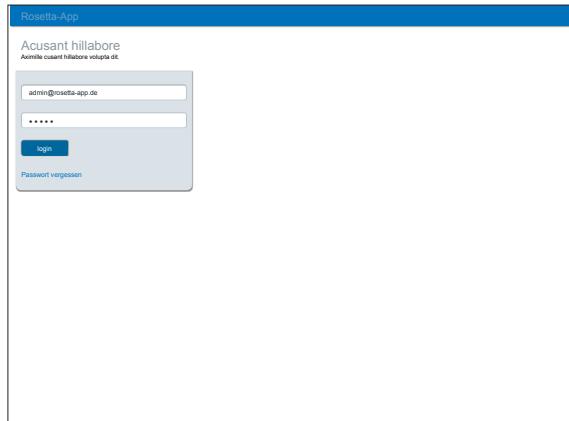


Abb. 20: Screendesign: Login Startseite

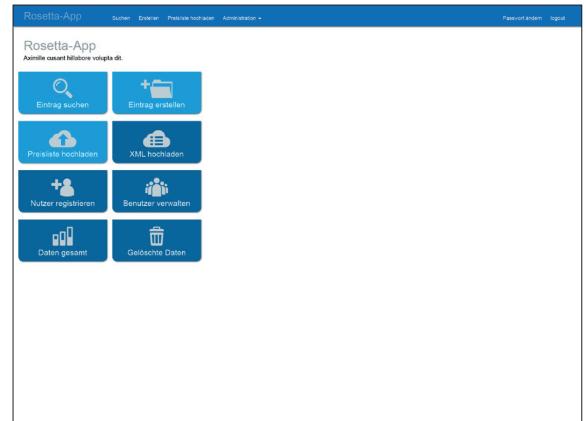


Abb. 21: Screendesign: Dashboard

Bilder/Icons: freepik, „Free vector icons for personal and commercial use“, unter: <http://www.freepik.com/free-icons> (abgerufen am 1.6.2017)
Teilweise nachträglich individualisiert

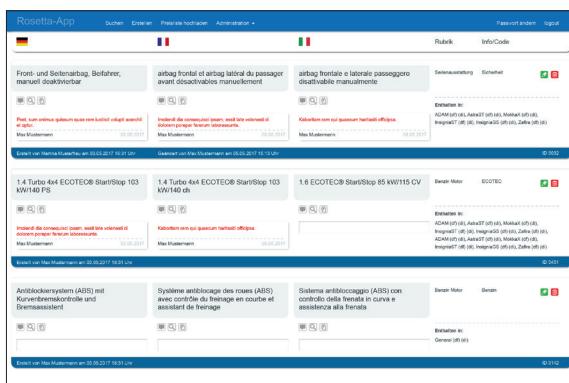


Abb. 22: Screendesign: Anzeige Datensätze

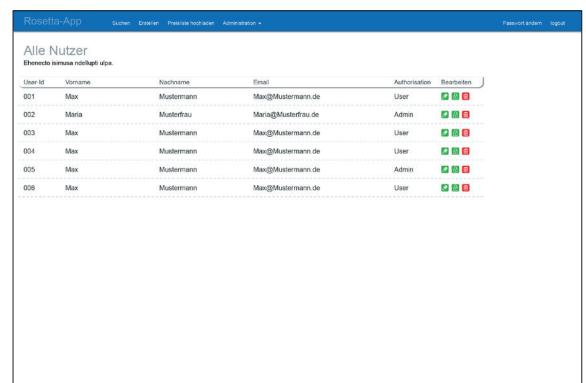


Abb. 23: Screendesign: Anzeige der registrierten Benutzer

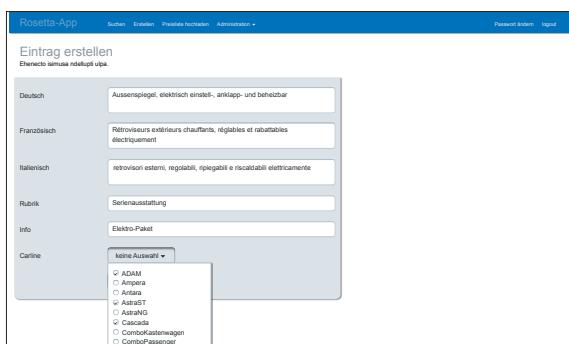


Abb. 24: Screendesign: neuer Eintrag/manuell

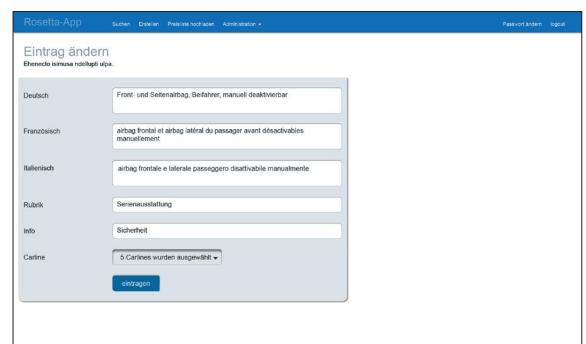


Abb. 25: Screendesign: bestehenden Eintrag bearbeiten

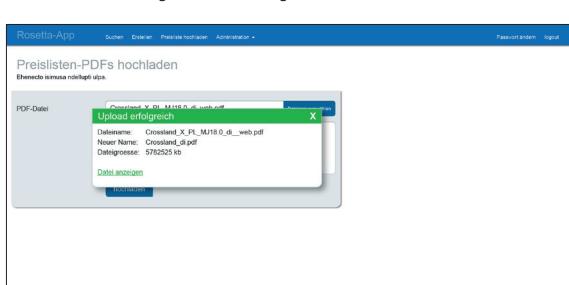


Abb. 26: Screendesign: erfolgreicher Upload eines PDFs

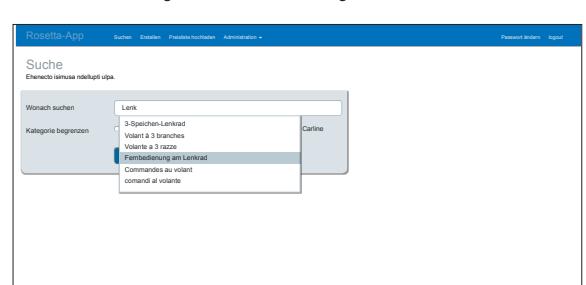


Abb. 27: Screendesign: Auto vervollständigung bei der Suche

- ClickDummy InVision

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

InVision, InVisionApp: „prototyping, collaboration & workflow platform“, unter: <https://www.invisionapp.com/> (abgerufen am 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.

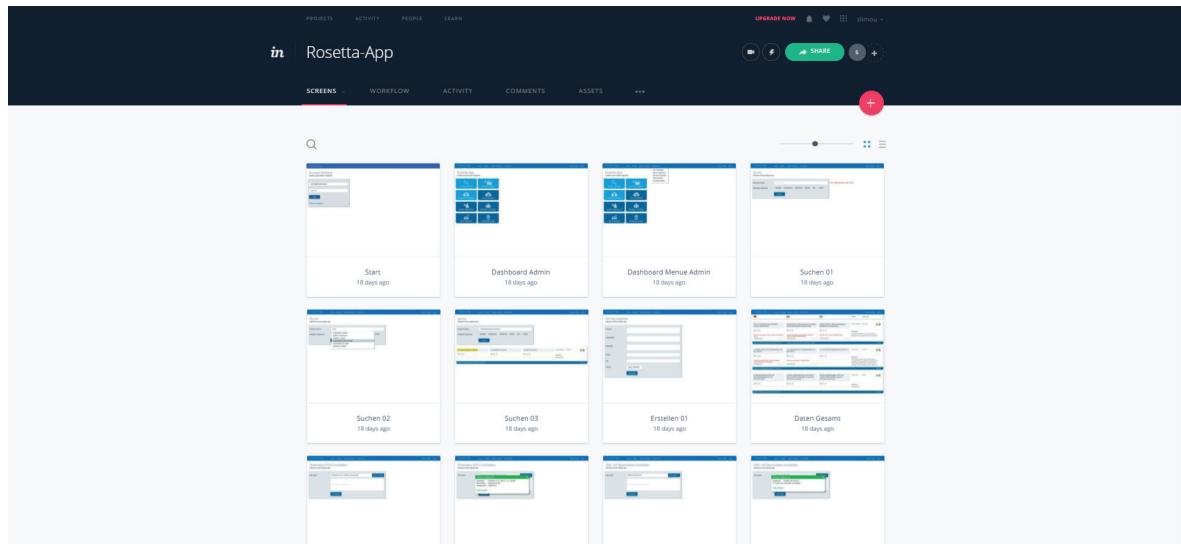


Abb. 28: Screenshot: Sequenzen bei InVisionApp. Online verfügbar unter: <https://projects.invisionapp.com/d/main#/console/10397535/232607158/preview>

- 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 des Unternehmens Apple voll funktionsfähig ist. Das konnte ich bei der Implementierung des Systems beheben und die Anwendung zeigt bei allen gängigen Browsern das gleiche Verhalten.

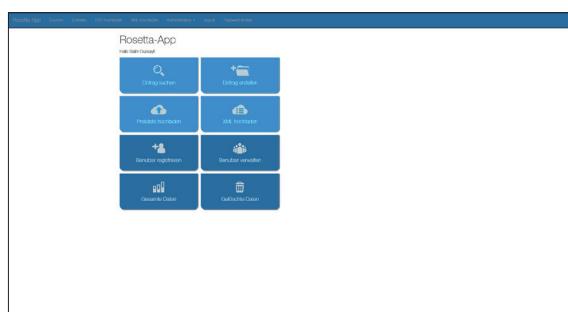


Abb. 29: Screendesign: Dashboard (Stand: 4.6.2017)

This screenshot displays a detailed view of user data within the Rosetta-App. At the top, there is a navigation bar with tabs for 'Dashboard', 'Arbeitsaufgaben', 'Arbeitsvolumen', 'Benutzer', and 'Gesamtaufgaben'. Below this, the main content area is filled with several tables and lists. One prominent table on the right side is titled 'Rosetta App complete data' and contains columns for 'Name', 'Vorname', and 'Ist'. The table lists multiple users with their names and first names, and a small green checkmark icon next to each row. There are also other sections below this table, including a 'User' section and a 'Jobs' section, both displaying similar tabular data. The overall layout is complex and organized to provide a comprehensive overview of the system's data.

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

Der geneigte Leser kann bei belieben die fertige Anwendung testen.
Rosetta-App ist online abrufbar unter www.rosetta-app.de

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

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

Passwort: admin
Passwort: user

- Features

Das 8. Kapitel dieser Dokumentation möchte ich mit einer Auflistung und kurzen Beschreibung einiger nennenswerter Features beenden, welche ich in Rosetta-App implementiert habe.

- Dateneinpfliege

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 text-basierten Austauschformat XML gebraucht gemacht. Es ist aktuell möglich, eine strukturierte XML-Datei aus dem Frontend der Anwendung zu importieren, sodass der Datenbestand mit den dort definierten Informationen gefüllt wird.

```
<?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

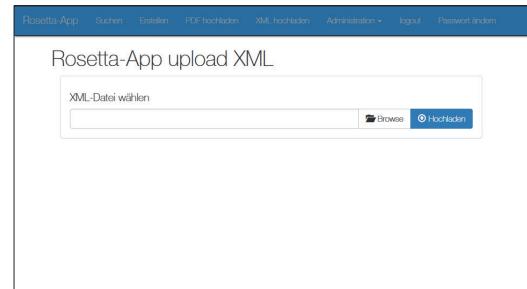


Abb. 32: Screenshot: Interface zum Hochladen einer XML-Datei
(Stand: 4.6.2017)

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 „Items“ 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.

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 Muster lautet wie folgt: „clarline_sprache.pdf“.

- Benutzerverwaltung

Bei der Anwendung ist es nicht vorgesehen, dass ein Benutzer sich selber registrieren kann, dies erfolgt ausschließlich durch einen Nutzer mit Administrator-Rechten.

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 hoch- bzw. runtergestuft werden. Aktuell existieren zwei Hierarchie-Ebenen, und zwar für einen Admin und den „gewöhnlichen“ Nutzer.

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. Das ermöglicht es, bestimmte Datensätze von ähnlichen zu differenzieren bzw. eine bestimmte Schreibweise/Übersetzung zu prorisiere.

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 übergegebenen String enthalten.

- Text in Zwischenablage kopieren

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

Im laufenden Workflow 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 Datenstazes. 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 Wissenmanagement dahingehend definiert, dass sich Wissensmanagement mit dem Erwerb, der Entwicklung, dem Transfer, der Speicherung sowie der Nutzung von Wissen beschäftigt.

Wissensmanagement: „Definition“, unter:
<http://wirtschaftslexikon.gabler.de/Definition/wissensmanagement.html#definition> (abgerufen am 29.5.2017)

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

- Herausforderungen

Sicherlich gab während der Arbeit an dem Projekt zahlreiche unvorhersehbare Hürden und Schwierigkeiten, welche ich zum größten Teil erfolgreich gelöst habe – an dieser Stelle möchte ich allerdings auf zwei ausgewählte Problematiken 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 ist es, dadurch, dass die Einträge in der Datenbank in der Kollation „utf8_general_mysql500_ci“ abgelegt werden.

Des Weiteren besteht dieses Problem weiterhin bei der Suchfunktion insofern, dass die gesuchten Texte im Fund mithilfe der PHP Funktion „preg_replace“ gehighlighted werden. Dies funktioniert zwar 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 „/“ enthält, ist dies der Fall, wird nicht von der Funktion „preg_replace“ gebraucht gemacht, d. h. der Fund wird zum jetzigen Zeitpunkt und in diesem Fall nicht gehightlightet.



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

- XML-Import

Ich hatte Schwierigkeiten beim Hochladen einer XML-Datei über ein Formular und das anschließende Einfügen der Daten in die Datenbank.

Der Grund dafür war die „GENERAL ENTITY Declaration“.

Es war also nicht ohne Weiteres möglich, eine XML-Datei an einem beliebigen Ort im Verzeichnis abzulegen und von dort die Daten in Datenbank zu schreiben. Ich habe es 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.

GENERAL ENTITY Declaration: „ENTITY Declaration“, unter:
http://xmlwriter.net/xml_guide/entity_declaration.shtml (abgerufen am 2.5.2017)

- Gelerntes

Wie auch bei den Ausführungen zu den Herausforderungen werde ich mich bei dem Gelernten auf zwei Punkte konzentrieren, möchte aber nicht unerwähnt lassen, dass sich in der Zeit, die ich für das Projekt aufgebracht habe, zahlreiche Erkenntnisse im Bereich der Software-Entwicklung gewonnen habe.

- Projekt-Arbeit

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 Dokumentieren der ganzen Arbeit.

- PHP-Programmierung

Während des Projekts 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 mit beliebig vielen 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 katalogisierende 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 könnte dies dann auch über verschiedene Tabellen gehandhabt werden, sodass keine vermeidbaren Redundanzen entstünden.

Datenbanken verstehen: „Normalisierung“, unter:
<http://www.datenbanken-verstehen.de/datenmodellierung/normalisierung/>

- Vollautomatische Dateneinpfliege

Die momentane Datenintegration von InDesign über XML nach Rosetta-App ist zwar unkompliziert und es kann ein beliebig großer Bestand an Daten schnell eingepflegt werden, dies bedarf aber dennoch einiger manueller Handlungen.

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

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

Originalveröffentlichung in: Verena M. Lepper und Ingelore Hafemann (Hg.), Karl Richard Lepsius. Der Begründer der deutschen Ägyptologie (Kaleidogramme 90), Berlin 2012, S. 37-78

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

Quelle: BFS – Strukturerhebung (SE) © BFS 2017

<https://www.bfs.admin.ch/bfs/de/home/statistiken/bevoelkerung/sprachen-religionen/sprachen.html>

Regel des Vorgehenmodells XP

<http://www.extremeprogramming.org/rules/customer.html>

Stakeholder

<http://agiles-projektmanagement.org/scrum-stakeholder/>

Scrum

<https://www.it-agile.de/wissen/einstieg-und-ueberblick/scrum>

<http://wirtschaftslexikon.gabler.de/Definition/scrum.html>

Requirements-Engineering-Tool

<http://m.wirtschaftslexikon.gabler.de/Definition/requirements-engineering.html>

MariaDB

<https://wiki.ubuntuusers.de/MariaDB/>

PHP Data Objects

<http://php.net/manual/de/intro pdo.php>

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>

Bootstrap Grid

<http://getbootstrap.com/examples/grid/>

Wissensmanagement-Tool

<http://wirtschaftslexikon.gabler.de/Definition/wissensmanagement.html#definition>

GENERAL ENTITY Declaration

http://xmlwriter.net/xml_guide/entity_declaration.shtml

Programm-Verzeichnis

PHP Storm

...

X

...

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

Abb. 1 Als Hauptsprache genannte Sprachen

S. 6

Abb. 2 ...

S. x

Tabellen-Verzeichnis

Tab. 1 Requirements Rosetta-App

S. 11

Tab. 2 ...

S. x
