

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 gewährleisten.

Es galt, ein System zu entwickeln, mithilfe dessen der tägliche workflow optimiert und zugleich die Kosten reduziert werden können.

Das System dient dazu, 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 Übersetzungen abgelegt sind.

Das System ist in seiner Funktion und Nutzen als Analogie zu dem „Stein von Rosette“ (Rosetta-Stein) zu verstehen – aus diesem Grund habe ich die Anwendung „Rosetta-App“ genannt. Der Zusatz „App“ ist dadurch begründet, da dieser „neuzeitige Stein“ ausschließlich in einer digitalen Form besteht und dynamisch genutzt und beliebig erweitert werden kann.

Statt der drei Sprachen Altgriechisch, Demotisch und der Hieroglyphen behandelt die Rosetta-App folgende drei Sprachen europäischen Ursprungs: Deutsch, Französisch und Italienisch.

Diese Dokumentation ist chronologisch aufgebaut, d.h. ich beginne mit den allgemeinen Voraussetzungen für das akademische Praxis-Projekt und gebe 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 beenden.

Zum Abschluss dieser Einleitung möchte noch auf die beiden ausschlaggebenden Parteien eingehen, die in diesem Projekt 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 des Projekts und gleichzeitig mein Arbeitgeber.

Die zweite Partei wird gebildet durch die Adam Opel GmbH, Bahnhofsviertel, 60423 Rüsselsheim am Main – im weiteren Verlauf dieser Dokumentation als Opel bezeichnet. TYPODROM betreut Opel als einen seiner Kunden als Tagesgeschäftsagentur.

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 Voraussetzungen 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
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. Implementierung/Phasen	S. xx
8.1 Timeline	S. xx
9. Anwendung	S. xx
9.1 Screendesign/fertige Anwendung	S. xx
9.2 Features	S. xx
9.3 ClickDummy InVision	S. xx
10. Ergebnisse und Fazit	S. xx
10.1 Wissensmanagement-Tool	S. xx
10.2 Gelerntes	S. xx
11. Ausblick	S. xx
11.1 was kann man machen	S. xx
11.2 Aussicht	S. xx
12. Literatur, Quellen, Bildnachweise	S. xx

1. AUFGABENSTELLUNG

– Voraussetzungen Praxis-Projekt

Praktische Aufgabe ...

Analyse, Entwicklung, Implementierung ...

– Benefit Student

Erfahrungen in Projektarbeit gewinnen ...

2. ZIELSETZUNG/UNTERNEHMENSZIELE

– Zielsetzung

Ziel war es, das vorhandene Wissen an einem zentralen Ort zu speichern und unternehmensweit zur Verfügung zu stellen.

Mithilfe des Systems wird das kollaborative Arbeiten gefördert, denn jeder Mitarbeiter kann bei Bedarf die zentral abgelegten Daten für seine Zwecke nutzen und gleichzeitig den Datenbestand erweitern, bzw. aktualisieren.

Durch genügend Transparenz ist es immer ersichtlich und nachvollziehbar, welcher Anwender einzelne Datensätze kreiert, manipuliert, kommentiert oder gelöscht hat. Zu jedem Beitrag wird sowohl Urheber des Datensatzes angezeigt als auch die Nutzer, die den Datensatz geändert, gelöscht oder Kommentiert haben.

Jede Manipulation wird mit dem jeweiligen Datum und Uhrzeit gespeichert und im System angezeigt.

Die zentrale Datenspeicherung hat zur Folge, dass Redundanzen in der Datenbeschaffung und Speicherung vermindert bzw. vermieden werden.

– Benefit Unternehmen

Für die Firma 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 in einem ersten Schritt prüfen, ob ein gewünschter Datensatz bereits existiert, bevor er eine Übersetzung 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 Opel als einen seiner Kunden als Tagesgeschäftsagentur und erstellt für diesen Kunden Printmedien wie z. B. Kataloge, Anzeigen, POS-Materialien und vieles mehr. Diese Werbematerialien werden für den gesamten europäischen Raum kreiert und produziert.

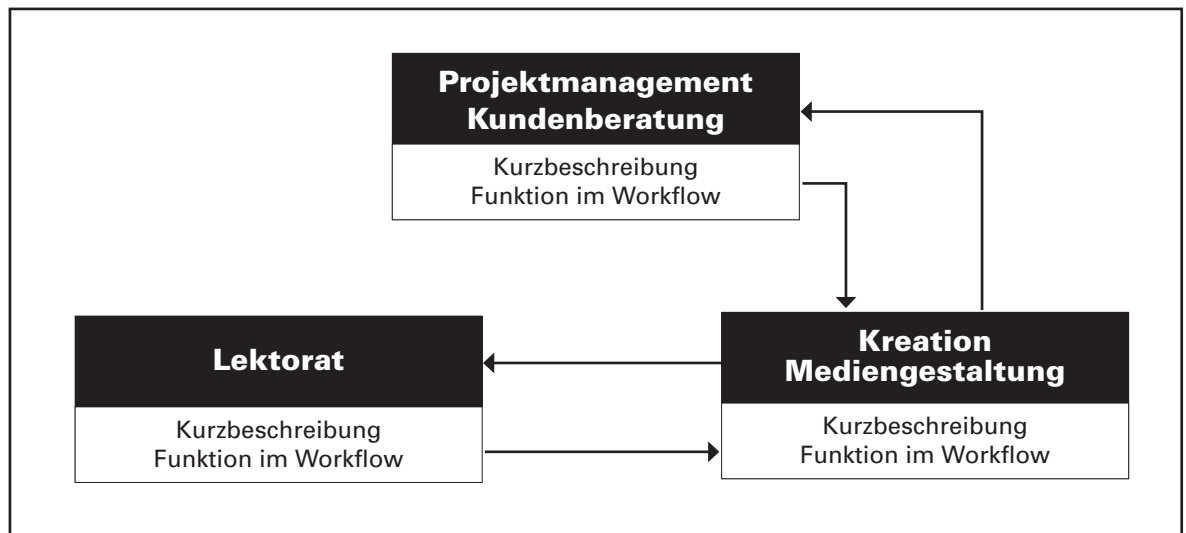
Für das Praxis-Projekt habe ich mich zunächst auf den Schweizer Markt und die dort vertriebenen Preislisten konzentriert – im Kapitel Ausblicke gehe ich auf eine mögliche Skalierbarkeit ein.

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

Bei TYPODROM werden die unterschiedlichen Publikationen abteilungsübergreifend im Tagesgeschäft erstellt. Diese Abteilungen sind das Projektmanagement und die Kundenberatung, die Kreation und die Mediengestaltung, und das Lektorat.

Die Auflistung der verschiedenen Abteilungen soll einen Überblick über die Phasen geben, die ein Printprodukt innerhalb der Agentur durchläuft.

Zur Veranschaulichung dient die folgende Grafik:



Es entstehen täglich neue Print-Produkte, die ähnliche Verkaufsobjekte auf verschiedene Art und Weise abbilden. Diese Objekte 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. Dadurch entstehen unnötige Kosten und Redundanzen, zudem ergibt sich hieraus ein erheblicher zeitlicher Mehraufwand.

– **Vision und Systemidee/Soll-Konzept**

Es soll ein Software-System entwickelt werden, das alle Produkte des Kunden beinhaltet und jeweils die korrekten Übersetzungen und Schreibweisen liefert.

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, Verlinkung zu dem eigentlichen Objekt, etc.

Alle Einträge sind im View editierbar und können auf einfache Art und Weise in die Zwischenablage kopiert bzw. kommentiert werden.

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.

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 oder Linguee 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 nicht mehr gerechtfertigt, bzw. rechtlich erlaubt**.

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

Im Fall des hier behandelten Projektes kommt erschwerend hinzu, dass es sich zu einem großen Teil um fachspezifische **Bezeichnungen/Wortschöpfungen** handelt, die seitens des Kunden entwickelt und verwendet werden.

Es bietet sich also 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 diesen Kunden arbeiten. Im Einzelnen sind dies Mitarbeiter aus der Kreation/Medien-gestaltung, 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 konventionsgemäße, selbsterklärende Nutzeroberfläche** gewährleistet ist.

– **Umfeld**

Das System ist primär als browserseitige Anwendung konzipiert, eine mobile Verwendung ist auch möglich, dient allerdings nur dem Zweck, gegebenenfalls 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 gewährleistet ist.

5. KONZEPT DER QUALITÄTSSICHERUNG

Zur Unterstützung 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, wie sie mit dem System in Berührung kommen werden:

Thomas Schäfer, Mediengestaltung: kann auf den Bestand vorhandener Übersetzungen zugreifen und diese ad hoc zur Erstellung aller Medien nutzen

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

Nicole Rocac, Kundenberatung: Schnittstelle zwischen dem Kunden, den externen Übersetzern 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.

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

Die im vorherigen Abschnitt beschriebene Vorgehensweise glich der eines **Vorgehensmodells** wie z. B. SCRUM.

Auf drei Schnittmengen eines 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**

Zur Festlegung der 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, sind die jeweiligen Requirements gleichzusetzen mit **Backlogs**, welche in gemeinsamer Absprache in einem folgenden Sprint abgearbeitet und implementiert wurden.

Das Tool habe ich in PHP geschrieben, die Einträge liegen in einer MySQL-Datenbank ab. Einzelne Funktionen wie das farbige Hinterlegen der Verbindlichkeiten oder das animierte Ein- und Ausblenden des Formulars sind mit JavaScript implementiert.

Formular ausblenden

Autor:

Titel:

Beschreibung:

Verbindlichkeit: ☐ muss (Pflicht) ☐ soll (Wunsch) ☐ kann (Möglichkeit) ☐ wird (Erfolgung)

Suchen

Requirements Rosetta-App

Verbindlichkeit	ID	Datum/Uhrzeit	Autor	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
soll	66	19.02.17 15:50:45	Salim	Sprachen	deutsch, italienisch, französisch
soll	67	21.02.17 19:42:16	Salim	Carline/Produktcode	zu jedem Eintrag soll die Carline und der Produktcode ergänzt werden
soll	68	26.02.17 11:46:17	Salim	Autovervollständigung	Bei Eingabe in das Suchformular sollen (ab 3 Zeichen) Suchvorschläge angezeigt werden
muss	69	26.02.17 11:43:27	Salim	Benutzerverwaltung	Alle Nutzer des Systems haben eigenen Account
soll	71	26.02.17 15:09:36	Salim	Einträge formatieren	evtl. zwei verschiedene Felder pro Sprache => 1. Headline (bold), 2. Content (auch mit Bullets)
soll	73	21.03.17 09:55:07	Claudio	Häufigkeit des Suchbegriffes	es soll angezeigt werden, wie oft ein Begriff in einem Dokument vorkommt
soll	74	20.05.17 07:43:48	Claudio	Hervorhebung durch Übersetzer	Bei zwei oder mehreren Übersetzungsvarianten Hervorhebungsmöglichkeit der bevorzugten Übersetzung durch den Übersetzer (Kommentarfunktion)
soll	75	20.05.17 07:44:39	Thomas	Nach Carline suchen?	Muss man die Begrifflichkeiten einer Carline zuweisen
soll	76	27.02.17 14:40:12	Thomas	hervorgehobenes Suchwort	gesuchtes Wort soll in Ergebnisanzeige hervorgehoben angezeigt werden
muss	77	01.05.17 07:47:22	Thomas	Transparenz	es muss ersichtlich sein, wer Änderungen zu welchem Zeitpunkt vorgenommen hat
soll	78	20.05.17 07:59:46	Salim	Nutzerverwaltung Admin	Der Admin hat vollen Zugriff auf alle Nutzerdaten
soll	80	20.05.17 07:45:03	Claudio	Kommentarfunktion	Einträge sollen kommentiert werden können
muss	81	22.03.17 09:51:40	Nicole	Dateneinpflege	Daten müssen automatisiert eingepflegt werden, nicht händisch
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
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	94	20.05.17 07:50:52	Salim	Nutzerverwaltung Admin	Der Admin hat vollen Zugriff auf alle Nutzerdaten
soll	96	21.05.17 09:58:37	Claudio	Häufigkeit des Suchbegriffes	es soll angezeigt werden, wie oft ein Begriff in einem Dokument vorkommt

https://github.com/slimouGit/Rosetta-App/blob/master/sidemap/sidemap_Rosetta-App.pdf

Die folgende Tabelle listet 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	
Dateneinpflege	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?	Muss man die Begrifflichkeiten einer Carline zuweisen	
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 Dateneinpflege	- 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	
Filterfunktion	Es soll nach jedem gefundenen Begriff gefiltert werden können	

Tabelle Stand 20.05.2017

- **iterative Herangehensweise**

Requirements = Backlog in Sprite ...

- **Retrospektive Stakeholdern**

Retrospektive nach Iteration ...

- Beurteilung von Implementiertem ...
 - Festlegung neuer ToDos für den nächsten Sprite ...
-

6. SYSTEMSTRUKTUR

– **Infrastruktur**

Ich habe das System mithilfe der Entwicklungsumgebung **PHP Storm des Unternehmens JET BRAINS** entwickelt.

Als Serverumgebung habe ich die freie Software **XAMPP des Entwicklers Apache Friends** genutzt.

Die Daten liegen zudem auf einem Server und können online abgefragt und gepflegt werden.

Als relationales Datenbank-System habe ich MySQL verwendet, die Anbindung läuft über **PHP Data Objects**, d. h. es kann einfach auf ein alternatives relationales Datenbank umgezogen werden.

– **Sprache**

– PHP

Da das System eine typische Web-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.

– JavaScript

Viele Funktionen habe ich mithilfe von JavaScript umgesetzt:

- Copy to clipboard (`document.execCommand(„copy“);`)

- flexible textarea (`auto_grow(element)`)

- ...

– **Frameworks/Libraries**

– Bootstrap

Bootstrap **CSS/Multiselect**

– jQuery

Text

– **Tools**

Github

Zur Versionsverwaltung, damit ich nicht auf die Verwendung eines einzigen Arbeitsplatzes während der Projektarbeit beschränkt bin und zum einfachen Austausch mit dem Betreuer der Hochschule habe ich **Git** verwenden, d. h. Github in Verbindung mit PHP-Storm.

<https://github.com/slimouGit/Rosetta-App.git>

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

Ich habe ein einfaches, unabhängig laufendes **Requirements Engineering Tool** aufgesetzt, mithilfe dessen die Anforderungen an das System definiert werden.

<http://ret.rosetta-app.de/>

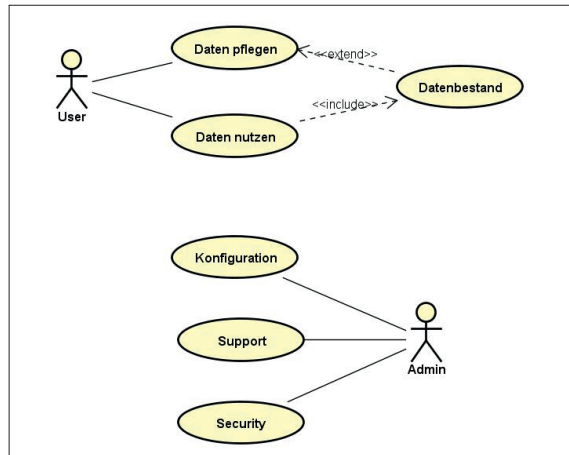
7. DESIGN

- UML-Diagramme

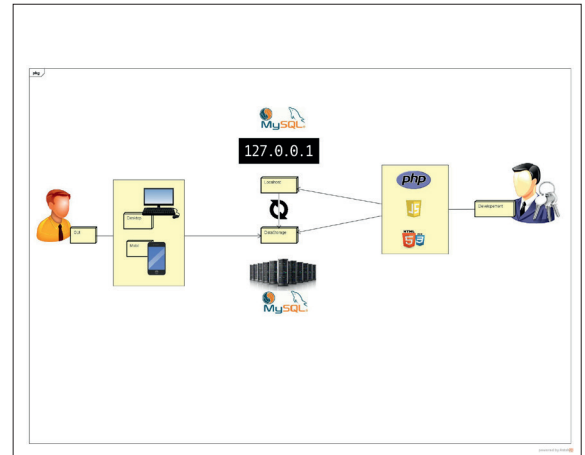
Die folgenden vier Diagramme ...

Mithilfe von **Astah Communication**

Use-Case Verteilungs-Diagramm

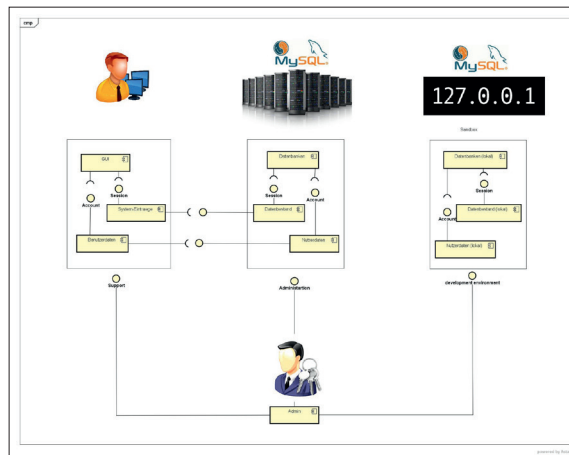


<https://github.com/slimouGit/Rosetta-App/blob/master/development/documentary/UML/UseCase.jpg>



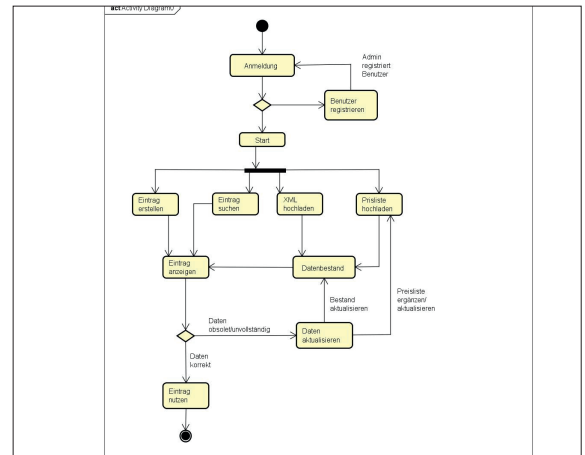
<https://github.com/slimouGit/Rosetta-App/blob/master/development/documentary/UML/Deployment.jpg>

Komponenten-Diagramm



<https://github.com/slimouGit/Rosetta-App/blob/master/development/documentary/UML/Component.jpg>

Aktivitäts-Diagramm



<https://github.com/slimouGit/Rosetta-App/blob/master/development/documentary/UML/Activity.jpg>

Bildquellen, alternative Bilder für User und Admin

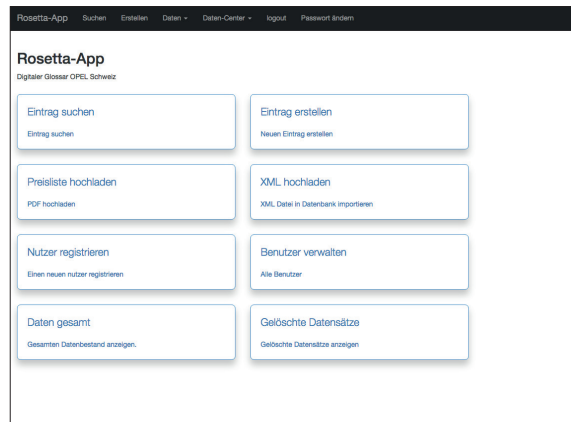
– Prototyp

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

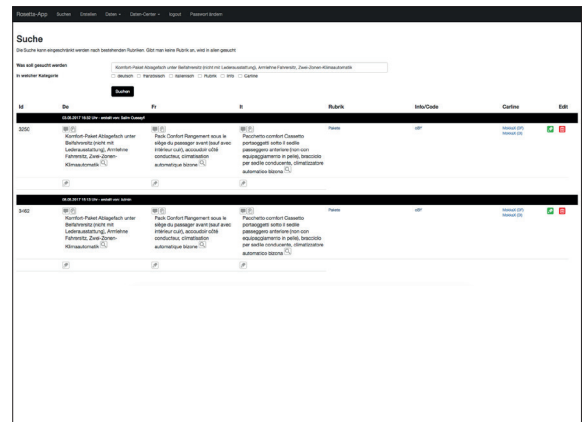
Während dieses Prozesses entstanden laufend neue Ideen, auch durch die beteiligten Personen.

Gleichzeitig tauchten Probleme und Fragen auf, sei es in der Programmierung, der Infrastruktur oder darin, die bestmögliche Nutzung des Systems zu ermöglichen.

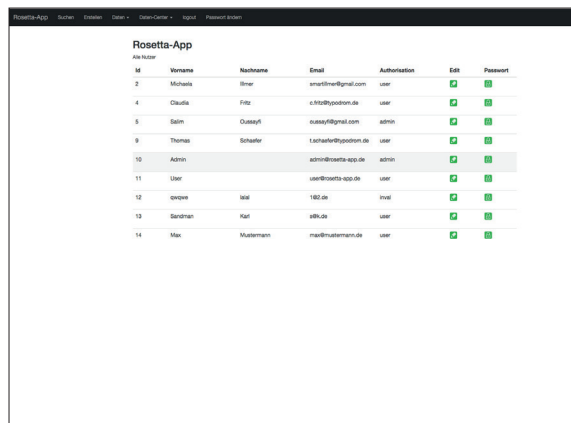
Meine Vorgehensweise, die iterative Entwicklung eines Prototyps vorab, ermöglichte es mir, eben diese Problematiken früh zu erkennen und zu behandeln/beheben.



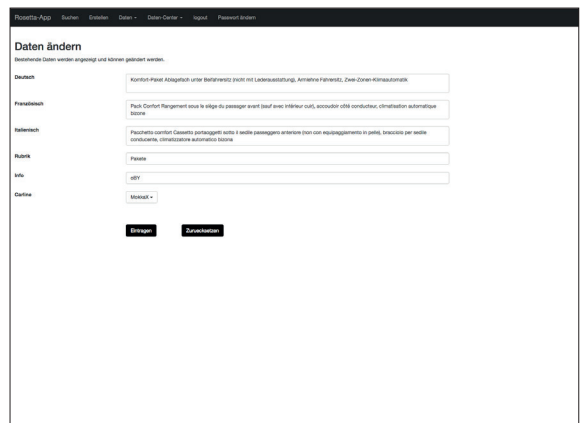
Bildunterschrift



Bildunterschrift



Bildunterschrift



Bildunterschrift

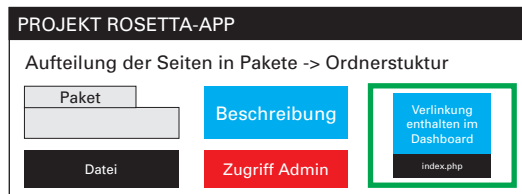
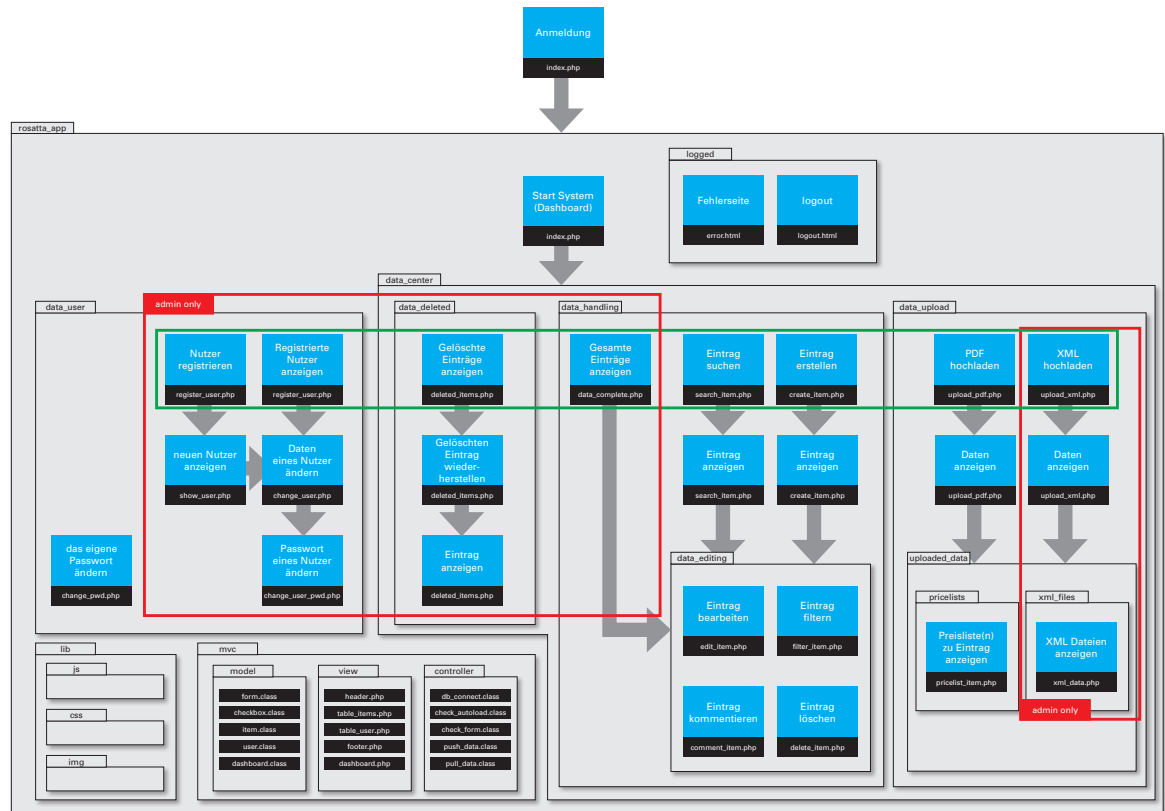
- Sitemap

(nach funktionierenden Prototyp Planung der Struktur)

Pakete (namespaces)

Abhängigkeiten

MVC-Pattern



https://github.com/slimouGit/Rosetta-App/blob/master/developement/sitemap/sitemap_Rosetta-App.pdf

- Verzeichnisstruktur

Folgender Verzeichnisstruktur ...

VERZEICHNIS-STRUKTUR

```

root_directory
- rosetta_app
- logged
- data_center
  - data_handling
  - data_editing
- data_upload
  - uploaded_data
    - pricelists
    - xml_files
  - data_deleted
- data_user
- mvc
  - model
  - view
  - controller
- lib
  - bootstrap
  - js
  - css
  - img
    
```

– 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 normalisierten 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 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_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 sinnngemäße 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)

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 (openssl_random_pseudo_bytes)
password_date	timestamp	Zeitpunkt der Passwortanfrage

8. IMPLEMENTIERUNG/PHASEN

– Timeline

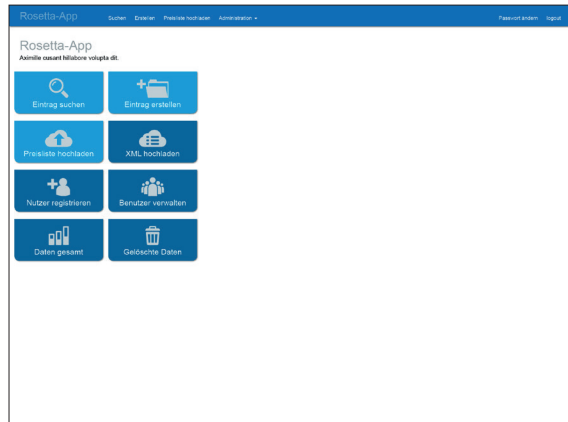
Github/evtl. Grafik mit Meilensteinen/Planungsphase, Entwurfsphase, Implementierungsphase, Einführungsphase, Wartungsphase ...

9. ANWENDUNG

– Screenshots

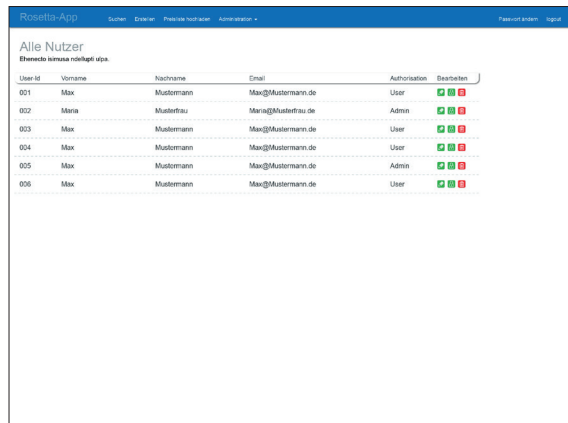
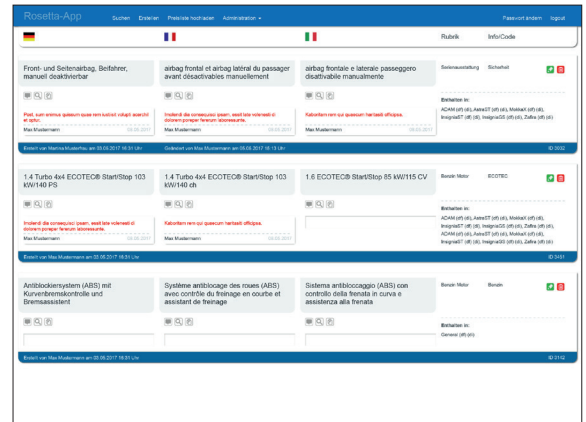
Screendesign mit **Adobe InDesign/Photoshop und Illustrator**

Screendesign/fertige Anwendung ...



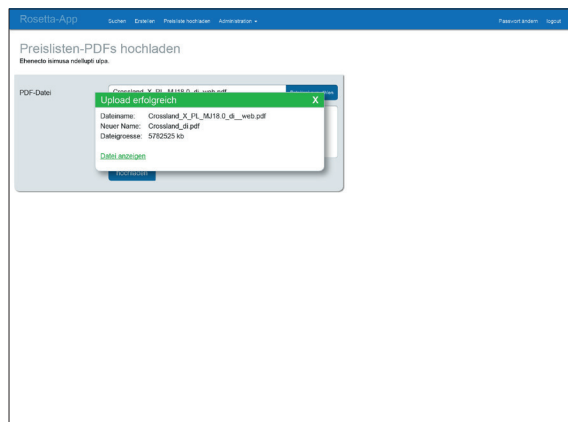
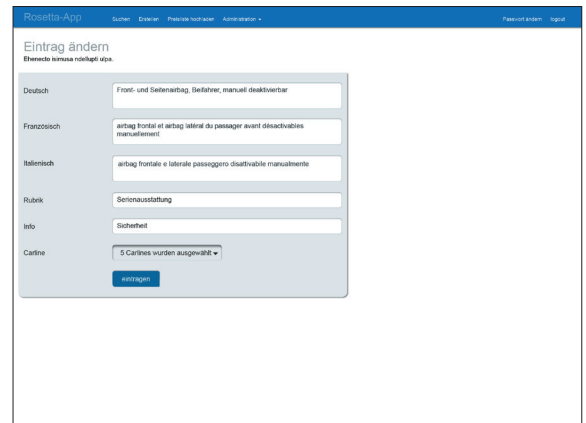
Bildunterschrift

Bildunterschrift



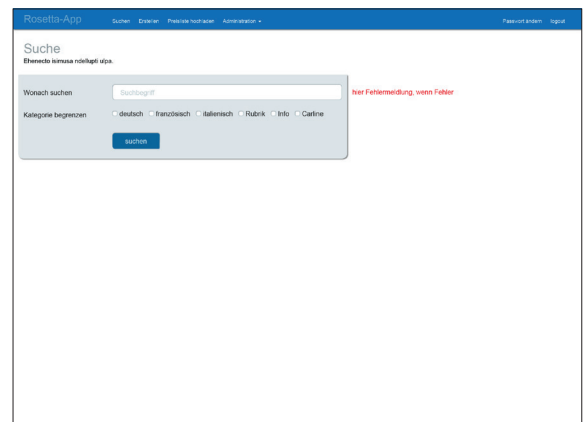
Bildunterschrift

Bildunterschrift



Bildunterschrift

Bildunterschrift



Bildquellen

- **Features**
Zwischenablage (JavaScript), Teextarea scalable ...
-

- **ClickDummy InVison**
<https://projects.invisionapp.com/d/main#/console/10397535/232607158/preview>
-

10. ERGEBNISSE UND FAZIT

- **Wissensmanagement-Tool**
... ein spezialisiertes und vollwertiges Wissensmanagement-Tool aus ...
-

- **Gelerntes**
Projekt nach Vorgaben der Stakeholder/PHP–Kenntnisse erweitert/...
-

11. AUSBLICK

- **was kann man machen**
Versionierung möglich ...
-

- **Aussicht**
System ist skalierbar (gesamter europäischer Markt)/generisch d.h. auf andere Projekte anpassbar ...
-

12. LITERATUR/QUELLEN/BILDNACHWEISE

- **Text**
Text ...
-

