

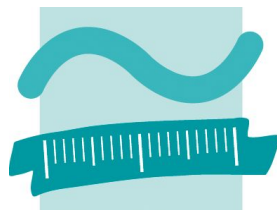
Beuth Hochschule für Technik Berlin
Fachbereich VI
Informatik und Medien
Studiengang Medieninformatik, 8. Semester

Praxisbericht

über das Praktikum in der Zetcom GmbH

von Anita Kusnierz
Matrikelnummer: 837730

Betriebliche Betreuung: Norbert Kanter & Dipl. Inf. (FH) Jörg Kruschinsky



BEUTH HOCHSCHULE
FÜR TECHNIK
BERLIN

University of Applied Sciences



zetcom

Zetcom GmbH
Köpenicker Str. 154A
10997 Berlin

Zeitraum: 15.05.2018 - bis dato

Inhaltsverzeichnis

1. Praktikumsbetrieb	2
1.1 Zetcom Group	2
1.1.1 Unternehmensbereiche	2
1.2 Weg zur Praktikumsstelle	3
2. Tätigkeitsbereiche und Aufgaben	3
2.1 Einarbeitung	3
2.2 Quality Assurance	4
2.2.1 Datenmigration	4
2.2.2 Individuallösungen	11
2.2.3 Sonstige Aufgaben	15
3. Praktikum und Studium	18
4. Bewertung des Praktikums	18

1. Praktikumsbetrieb

1.1 Zetcom Group

Zetcom Group ist ein internationales inhabergeführtes Unternehmen mit Sitz in der Schweiz. Das Unternehmen wurde 1998 in Bern gegründet. Mit seinen mehr als 60 Mitarbeitern entwickelt es Softwareprodukte zur Dokumentation und zum Management von Kulturgütern im Bereich Museen, Sammlungen und Galerien. Die Zetcom Group bietet Museen individuell konzipierte Softwarelösungen an, die sie mit passender IT-Infrastruktur bereitstellt, konfiguriert und wartet. Das Unternehmen unterhält weitere Standorte u.a. in Spanien, in Griechenland, in den USA und sowie ein Projektbüro in Berlin.

Das Unternehmen möchte mit seinen hochwertigen Produkten hochauflösende mehrsprachige Lösungen für das Sammeln und Museumsmanagement liefern.

Das Angebot erstreckt sich von Softwarelösungen für private und institutionelle Sammlungen (*MuseumPlus Classic*) bis zu flexiblen Echtzeit-Museums-Management Anwendungen mit Software as a Service Option.¹ Insgesamt zählen zu den Kunden über 900 renommierten Institutionen auf der ganzen Welt.²

Das Team von Zetcom GmbH umfasst circa 30 Personen und besteht größtenteils aus festangestellten Mitarbeitern. Davon bilden die Hälfte die Projektleitung, die meist aus Mitarbeiter mit Hintergründen aus Kunstgeschichte sind. Circa ein Viertel sind die für Buchhaltung und Geschäftsleitung zuständigen Mitarbeiter und ein Viertel bildet das Entwicklerteam.

1.1.1 Unternehmensbereiche

Die Zetcom Group besteht aus zwei eng zusammen arbeitenden Unternehmen: Zetcom AG und Zetcom GmbH. Die Zetcom GmbH bildet eine wichtige Schnittstelle zwischen dem Hauptstandort in Bern und weiteren Niederlassungen. Die Zetcom AG ist für die Entwicklung und Erstellung der Software zuständig. Durch die Zetcom GmbH erfolgt ein umfangreicher technischer Kundensupport, Entwicklung individueller Fragestellungen der Kunden (Custom Code), Datenmigration, Qualitätssicherung sowie der nationale und internationale Vertrieb der Software. Ferner verzichtet die Unternehmenskultur in Zetcom GmbH auf klassische Abteilungsstrukturen, so dass die einzelnen Bereiche unternehmens- und abteilungsübergreifende Gruppen bilden.

Mein Tätigkeitsbereich ist im Unternehmensbereich der Qualitätssicherung von Datenmigration und Custom Code angesiedelt.

¹ Zetcom GmbH: "Produkte", unter: http://www.zetcom.com/museumplus_de/ (abgerufen am 20.03.2019)

² Zetcom GmbH: "Kunden", unter: <http://www.zetcom.com/kunden/> (abgerufen am 20.03.2019)

1.2 Weg zur Praktikumsstelle

Nachdem ich bei einem kleinen Familienunternehmen zuvor als studentische Hilfskraft gearbeitet habe, suchte ich nach einer neuen beruflichen Herausforderung, die zu meinen Interessenschwerpunkten passte. Es war mir wichtig, eine Tätigkeit zu finden, die auch implizit zu meinem Erststudium der Geisteswissenschaften einen Bezug herstellen kann und mir die Einblicke in den Alltag in einem Softwareunternehmen gibt. Bei meiner Recherche wurde ich auf die Zetcom GmbH aufmerksam und bewarb mich dort als Werkstudentin im Bereich Qualitätssicherung. Nach einem persönlichen Gespräch mit Herrn Kruschinsky, in dem ich meine Bereitschaft für die Zusammenarbeit darlegen konnte, kam es zur Vertragsunterzeichnung.

Es bot sich für mich die Chance, gleich in zwei Unternehmensbereiche einen Einblick zu gewinnen.

2. Tätigkeitsbereiche und Aufgaben

Meine Tätigkeitsbereiche bei Zetcom GmbH beziehen sich auf zwei Hauptaufgabenbereiche. In erster Linie handelte sich um die Mitarbeit in der Qualitätssicherung bei den Datenmigrationen. Die Aufgabe galt als meine Kernaufgabe, wobei die daraus resultierenden Zusammenhänge voraussetzten, dass ich mich selbst mit der Datenmigration tiefer beschäftigen sollte. Der zweite Bereich richtete sich auf die Qualitätssicherung von Individuallösungen, sowie Unterstützung bei der Umsetzung von Funktionalitäten der von Zetcom entwickelten Online Web-Anwendung. Auf diese zwei Themengebiete wird in diesem Abschnitt eingegangen.

2.1 Einarbeitung

Zunächst startete erwartungsgemäß die Einarbeitungsphase, bei der ich mich mit den internen Arbeits-Tools sowie Benutzerhandbüchern befasste, die eine fundierte Informationsquelle über das Produkt lieferten. Ich erhielt neben einer eigenen *E-mail* Adresse alle für meine Arbeit relevanten Tools-Zugänge. Dazu gehören u.a. Zugriffe auf eine aus allen *RIA*-Kunden und Instanzenlinks bestehende Kundendatenbank namens *ZIS*, ein Ticketsystem *OTRS*, eine Anwendung für Projektplanung *TeamGantt* sowie ein Konfigurationstool *ZAC* für *RIA* Anwendungen.

Um am Projekt der Qualitätssicherung mitzuarbeiten, habe ich mich primär mit dem Produkt der Zetcom-Gruppe auseinandergesetzt. Die zwei Produkte, die ich unter die Lupe genommen habe, sind *MuseumPlus Classic* und *MuseumPlus RIA*. Sowohl *MuseumPlus Classic* als auch *MuseumPlus RIA* sind modular aufgebaut und bestehen aus aufeinander abgestimmten Modulen (z.B. *Objekte*, *Adressen*, *Restaurierung*, *Provenienz* etc.). Die Module können einzeln oder zusammen verwendet werden.

Das *MuseumPlus Classic* (1998) ist eine Client-Server-Applikation und baut auf Microsoft Standardprodukten auf. Das Back-end enthält die Tabellen für den Benutzerzugriff, das Front-end enthält die komplette Benutzeroberfläche samt Funktionen.³ Meistens nutzen die kleineren Museen das DBMS ACCESS. Mittlere und größere dagegen verwenden MS SQL Server oder Oracle, wo die IT-Architektur besser ausgebaut ist.

Das *MuseumPlus RIA* (2010) ist das von Zetcom entwickelte Rich Internet Application (RIA) Framework. Dieses Programmiergerüst ist in der objektorientierten Programmiersprache Java EE auf der Basis von einer plugin-basierten Canoo *UltraLightClient* Architektur realisiert.⁴ Es kann individuell konfiguriert und parametrisiert werden. Alle RIA-Anwendungen beruhen auf einer relationalen Datenbank-Architektur. Eine Erweiterung der RIA-Anwendung ist die webbasierte Variante mit *Vaadin* Framework und in HTML5 und CSS3 umgesetzten User Interface. Darüber hinaus bietet Zetcom Group das *SaaS*-Modell (Software as a Service) oder eine lokale Installation von Software über die bereits beim Kunden vorhandene Infrastruktur. Die Cloud ermöglicht es dem Kunden, dass Zetcom für die Wartung der Systemupdates eine tägliche Datensicherung erstellen und für Support-Zwecke zur Verfügung stehen kann.⁵

2.2 Quality Assurance

Die Bereiche, die mit Qualitätsmaßnahmen abgedeckt werden, umfassen das Prüfen von Datenmigrationen und Testen von - in *Groovy* implementierten - Individuallösungen (intern sog. Custom Code), die gemäß den Anforderungen unserer Kunden erstellt wurden. Alle der zu überprüfenden Fälle wurden manuell getestet und in Form einer Dokumentation bzw. Spezifikation, mit Hilfe von für interne Zwecke verständlichem Pseudocode sowie durch Soll-Ist-Vergleich umgesetzt. Meinen Testergebnissen folgten im Anschluss die Feedbackschleifen, um die Fehlerquote durch den menschlichen Faktor zu verringern. Die kontrollierten Ergebnisse wurden in der Regel an zuständige Personen weitergegeben.

2.2.1 Datenmigration

Hintergrund

Die Datenmigrationen im Hause werden mit Hilfe von einem intern entwickelten scripting Tool *VBA RIA-Importer* durchgeführt. Das Tool ermöglicht die Migration von Daten aus *MS Access* in die *RIA* Datenbank. Hierfür lassen sich die einzelnen Schritte der Datenüberführung dem ETL-Prozess (Extract, Transform, Load) zuordnen. Nach der Extraktion der unterschiedlich strukturierten Daten werden sie in ein einheitliches Datenschema transformiert. Dabei wird die Transformation mit Hilfe von Mapping

³<https://support.office.com/de-de/article/aufteilen-einer-access-datenbank-3015ad18-a3a1-4e9c-a7f3-51b1d73498cc>

⁴ Wikipedia: "UltraLightClient", unter: <https://de.wikipedia.org/wiki/UltraLightClient> (abgerufen am 10.04.2019)

⁵ Zetcom GmbH: "Produkte", https://www.zetcom.com/museumplus_de/ (abgerufen am 20.03.2019)

Dokumenten durchgeführt. Anschließend werden die Daten in das Zielsystem geladen.⁶ Eine wichtige Rolle beim Transformationsprozess spielen die sog. Mapping Dokumente, die festlegen, welches Feld im Quellsystem einem Feld im Zielsystem entspricht. Sie werden üblicherweise von den Projektleitern erstellt, die nach Absprache mit Kunden ihre etwaigen Wünsche berücksichtigen. Bevor der Migrationsprozess beginnt, werden alle zu migrierenden Feldinhalte des Quellsystems analysiert und ggf. dem Kunden der Bedarf an Datenbereinigung angeboten. Im Wesentlichen besteht das Mapping aus drei Spalten. Die linke Spalte enthält die von *MS Access* stammenden Quelldaten und die mittlere Spalte legt die Migrationsregeln für die in der rechten Spalte definierten Zielfelder fest. (vgl. Abbildung 2.1 und 2.2).

		Datum:	27.07.2018				
		Version:	04_zetcom				
		kein Korrekturbedarf	noch zu überprüfen	Korrekturbedarf	neu in MuseumPlus	Frage / Änderung	Migration
		Feldtypen MuseumPlus Klassik: S = Textfeld (max. 255 Zeichen), M = Memofeld (große Feldlänge), D = Datumsfelder, L = Feld für Ganzzahlen, F = Feld für Fließkommazahlen					
Nr.	ID	Quelle: Interner Name Tabelle	Quelle: Interner Name Feld	Quelle: Register (Name, Modul)	Quelle: Wiederholgruppe	Quelle: Bezeichnung Feld (Name)	Quelle -> Ziel: Migrationsregel
		Fixe Wortlisten					
682	91,365	Typ	"ObjCategoryVgr"	Typendefinition		Objektyp	WENN TypDstindexL = 21
	91,365	Typ	TypNameUserS	Wortliste fix		Code	
	91,365	Typ	TypAufID	Wortliste fix		Gültig für Sammlung	
	91,365	Typ	TypBeschreibungS	Wortliste fix		Beschreibung	
Flache Felder: Objekte							
	91,375	ObjDaten	ObjID				
	91,375	ObjDaten	ObjAufID				
143	91,375	ObjDaten	ObjTypS	Hauptbereich		Objektyp	als Knoten in ObjCategoryVgr anlegen

Abbildung 2.1: Ein Screenshot des Mappings für das *Objekt* Modul, Ausschnitt einer Quellstruktur in Bezug auf *flache Felder* und *fixe Wortlisten*

Die Quellfelder sowie die Zielfelder sind mit ihren internen Namen versehen, um eine Eindeutigkeit innerhalb des Dokuments herzustellen. Die Struktur des Mappingdokuments wird ebenfalls definiert und in Blöcken aufgeteilt. Dabei bezieht sich eine vor den Blöcken stehende Nummer auf ein konkretes Script, das mit *VBA RIA-Importer* bereitgestellt wird. Zunächst werden im Mapping die *fixen* bzw. *freien Wortlisten* auch genannt *Vocabularies* (Auswahllisten) aufgeführt, im weiteren die sog. *flachen Felder* sowie *Wiederholgruppen* (Gruppen von Feldern) und *Referenzen* (Verknüpfungen zu anderen Modulen).

⁶ Wikipedia: "Migration (Informationstechnik)", unter: [https://de.wikipedia.org/wiki/Migration_\(Informationstechnik\)](https://de.wikipedia.org/wiki/Migration_(Informationstechnik)) (abgerufen am 10.04.2019)

Ziel: Modulname	Ziel: Org. Unit	Ziel: Bereich / Register	Ziel: Wiederholgr uppe	Ziel: Wiederholgr uppe intern	Ziel: Feldbezeichn ung	Ziel: interner Feldname	Ziel: Wortliste / Referenz intern	Datenübernahme: zetcom
Vocabulary								
Vocabulary		Hauptbereich				VocName	ObjCategoryVgr	siehe Screenshot 7
Vocabulary		Hauptbereich	Benennungen		Benennung	VocTerm		
Vocabulary		Hauptbereich			Bereich	Voc_OrgID		
Vocabulary		Hauptbereich			Bemerkungen	vnod_comment_v		
Target Module	Org. Unit			Repeatable Group	Label	Data Binding		
Object						Object_ID		siehe Zeile 11,
Object						Object_OrgID		
Object		Hauptbereich			Objektyp	ObjCategoryVoc	ObjCategoryVgr	Knoten wurden mit ObjCategoryVgr korrekt angelegt, es sind aber die doppelten Knoten im Vocabular vorhanden, siehe Screenshot 7

Abbildung 2.2: Ein Screenshot des Mappings für das *Objekt* Modul, Ausschnitt einer Zielstruktur in Bezug auf flache Felder und fixe Wortlisten

Aufgaben

Zu Beginn meiner Tätigkeit arbeitete mein Team bereits seit über einem Jahr an dem Projekt für *Staatliche Museen zu Berlin* und ich stieg während der Migration ein. Für die Überprüfung der Datenmigration wurden insgesamt vierzehn Excel-Mappingdateien erstellt, wobei jede Datei einem Modul in *RIA*-Anwendung entsprach. Das Vorgehen der Migrationsverifikation bestand grob aus folgenden Schritten:

- Analyse der Daten auf Tabellenebene in *MS Access* mit Hilfe von *Access* Abfragen
- Analyse der Daten direkt in der *RIA*-Anwendung mittels der eingebauten Suchmaske
- Auswertung und Formulierung der Testergebnisse und Dokumentation der Fehlerbilder

Gemäß dem Vorgehen wurden mir die entsprechenden Daten in Form von *MS Access* Tabellen zur Verfügung gestellt. Den Zugriff auf die *RIA* Instanz habe ich über das interne Tool *ZIS* erhalten. Der erste Test betraf in der Regel die quantitative Überprüfung der migrierten Daten innerhalb einzelner Module und einzelner Felder. Im Falle von Modulen habe ich mit Hilfe von einer `SELECT COUNT(*)` Abfrage die Anzahl der Datensätze ermittelt und meine Ergebnisse mit den in der *RIA*-Anwendung verglichen und folglich im Mapping Document vermerkt. Es kam relativ oft vor, dass man bei der quantitativen Verifizierung ggf. eine Bedingung einbeziehen musste. Die Bedingung lautete beispielsweise etwa folgend:

“WENN *ObjAufID* ≠ 138 (Gelöschte Datensätze) für jeden Eintrag mit eigener *OprID* einen Datensatz in Provenienz-Modul anlegen UND zu Objekt-Datensatz referenzieren”.

Mit einer unten gelisteten Abfrage konnte ich direkt die Anzahl der verknüpften *Objekt*-Datensätze abfangen. Obwohl die Anzahl der Datensätze in der Regel stimmig war, habe ich in diesem Fall noch einige Musterdatensätze ausgewählt und manuell überprüft, ob die Referenzen eine korrekte Zuordnung erhielten.

```
SELECT COUNT(OBJPROVENIENZ.OprObjId)
FROM OBJDATEN INNER JOIN OBJPROVENIENZ ON OBJDATEN.ObjId = OBJPROVENIENZ.OprObjId
WHERE OBJDATEN.ObjAufId not like 138;
```

Im weiteren Verlauf unterlagen der Überprüfung die sog. *flachen Felder* und *Wiederholgruppen*. Sowohl die *flachen Felder*, als auch *Wiederholgruppen*, standen meist mit einer *fixen* (unveränderbaren) bzw. *freien* (erweiterbaren) *Wortliste* in Verbindung. Als Beispiel kann ein - in Abbildung 2.1 - dargestelltes *flaches Feld ObjTypS* dienen, das in Beziehung mit einer *fixen Wortliste ObjCategoryVgr* steht. Die Verbindung zu einer Wortliste wurde mit einer entsprechenden Migrationsregel im Mappingdokument vermerkt. Da es sich hier um eine *fixe Wortliste* handelte, konnte man die Datensätze mit **SELECT DISTINCT** im Quellsystem abfragen und für jeden Wortlisteneintrag repräsentativ ein paar Datensätze wählen sowie deren Werte auf Richtigkeit im Zielsystem überprüfen. Im Fall von einer *freien Wortliste* wäre die Anzahl der Datensätze ausschlaggebend und müsste die Überprüfung mehrerer zufällig ausgewählten Datensätze in Anspruch nehmen. Im vorgeführten Beispiel befand sich die vollständige Wortliste *ObjCategoryVgr* in der Quelltable "Typ" und konnte mit einer Bedingung "*WENN TypDstIndexL = 21*" aufgerufen werden (vgl. Abbildung 2.3).

Nach dem Vergleich mit den migrierten Daten in der *RIA*-Anwendung konnte man sofort erkennen, dass es hier zu einer doppelten Übermittlung der Daten gekommen ist (vgl. Abbildung 2.4).

Um den Sachverhalt zu reporten, habe ich in der Regel eine kurze Notiz im Mapping hinterlegt sowie ggf. ein Screenshot beigelegt (vgl. Abbildung 2.4). Folglich sind die weiteren Daten von den in der Quelltable definierten Quellfelder bezüglich der Wortliste wie *TypBeschreibungS* und *TypAufId* ebenfalls korrekt übertragen. Die Werte vom *flachen Feld ObjTypS* wurden auch mit einem korrekten Vocabulareintrag verbunden. Der hier aufgetretene Fehler lag in der Regel an der mehrfachen Anwendung des Scripts auf einem Mappingblock. Dies hatte zur Folge, dass die Vocabulardaten beim ersten Laden nicht entfernt worden sind und deshalb doppelt migriert wurden.

TYPID	TYPMUTAT	TYPMITAR	TYPIDFID	TYPDST	TYPNAMEUSERS	TYPNAMES	TYPNAMEU	TYPBESCH	TYP
35	27.09.2013	EM_BG	1	21	Allgemein	Allgemein		Standardty	
1583	21.07.2014	AMP_SSCh	554	21	Allgemein - AMP		AMP - Allg	Standardty	
545	26.02.2014	GD_MN	1	21	Architektur			Bildarchiv	
375	28.11.2002	Demo	1	21	Archivalie - Akte			Typ für "A	
461	14.02.2003	Demo	1	21	Archivalie - Einzel			Typ für "A	
542	21.05.2003	u_simmons	1	21	Archivalie - Sonsti			Typ für Ar	
376	14.02.2003	Demo	1	21	Archivalie - Vorgar			Typ für "A	
390	07.04.2016	GD_MN	1	21	Audio			Tonaufzeich	
1205	01.10.2015	NG_N_KYZA	2367	21	Autograph/Dokument			NG	
603	01.12.2003	ZET_US	1	21	Behälter			Ägyptische	
546	01.12.2003	ZET_US	164	21	Bildhauerei			Bildarchiv	
1243	28.09.2015	NG_N_KYZA	2367	21	Buch				
1824	28.09.2015	NG_N_KYZA	2367	21	Buch/Portfolio				
392	18.06.2003	Admin	1	21	Druckgrafik			Radierung	
387	01.10.2015	NG_N_KYZA	1	21	Film/Video			Film- und	
318	01.10.2015	NG_N_KYZA	1	21	Fotografie			Fotografis	
543	01.12.2003	ZET_US	164	21	Gebrauchsgegenständ			Bildarchiv	
643	04.02.2004	ZET_US	1	21	Gefäßform			ANT	
1203	12.07.2018	AMP_KF (adm	554	21	Gemälde				
604	24.11.2004	ANT_SB (AD	1	21	Gerät			Ägyptische	
645	04.02.2004	ZET_US	1	21	Glyptik			ANT	
606	01.12.2003	ZET_US	1	21	Grabmaterial			Ägyptische	
1204	27.02.2014	GD_MN	1	21	Grafik				
647	04.02.2004	ZET_US	1	21	Inscript			ANT	
393	01.10.2015	NG_N_KYZA	2367	21	Installation			NG Raumgre	
544	01.12.2003	ZET_US	164	21	Kunstgewerbe			Bildarchiv	
388	24.01.2003	Demo	1	21	Malerei			Tafelmaler	
1863	01.10.2015	NG_N_KYZA	2367	21	Malerei/Gemälde			NG - auch	
803	01.06.2007	GD_X1	1	21	Midas-Objekt				
607	01.12.2003	ZET_US	1	21	Mobiliar			Ägyptische	
605	27.04.2005	ANT_SB (AD	1	21	Modell			Ägyptische	
644	04.02.2004	ZET_US	1	21	Mosaik/Malerei			ANT	
646	04.02.2004	ZET_US	1	21	Münzen			ANT	
623	02.03.2004	ZET_US	1	21	Musikinstrument			Musikinstr	
608	03.04.2013	GD_MN	1	21	Numismatik			Numismatik	
1823	01.10.2015	NG_N_KYZA	2367	21	Objektkunst			NG für ein	
609	01.12.2003	ZET_US	1	21	Organisches Materie			Ägyptische	
1523	18.03.2013	GD_MN	1	21	Ostrakon			Ägyptische	
2125	22.03.2017	NG_N_KYZA	2367	21	Performance			NG für eph	
1244	27.02.2014	GD_MN	1	21	Portfolio				
2124	22.03.2017	NG_N_KYZA	2367	21	Printed Matter			NG für Kün	
610	24.11.2004	ANT_SB (AD	1	21	Schmuck			Ägyptische	
389	18.07.2007	AMP_KK (ad	1	21	Skulptur			Bildhaueri	
1864	22.03.2017	NG_N_KYZA	2367	21	Skulptur/Plastik/Ok			NG Bildhau	

Abbildung 2.3: Ein Screenshot der Access Quelltable *Typ* mit vollständiger Wortliste *ObjCategoryVgr*

Abbildung 2.4: Ansicht einer Wortliste *ObjCategoryVgr* im Zielsystem

Neben der doppelten Migration von Datensätzen sind auch viele Daten nicht vollständig im Zielsystem angekommen. Die Abbildung 2.5 stellt diesen Fehler in Bezug auf *Provenienz*-Modul dar. Laut Mapping sollten in diesem Fall die Werte von zwei Felder

OprReserve01S und *OprBemerkungenM* jeweils vom Typ “Midas” und “Bemerkung intern” in dieselbe *Wiederholgruppe* “Bemerkungen” migriert werden (vgl. Abbildung 2.6 und 2.7).

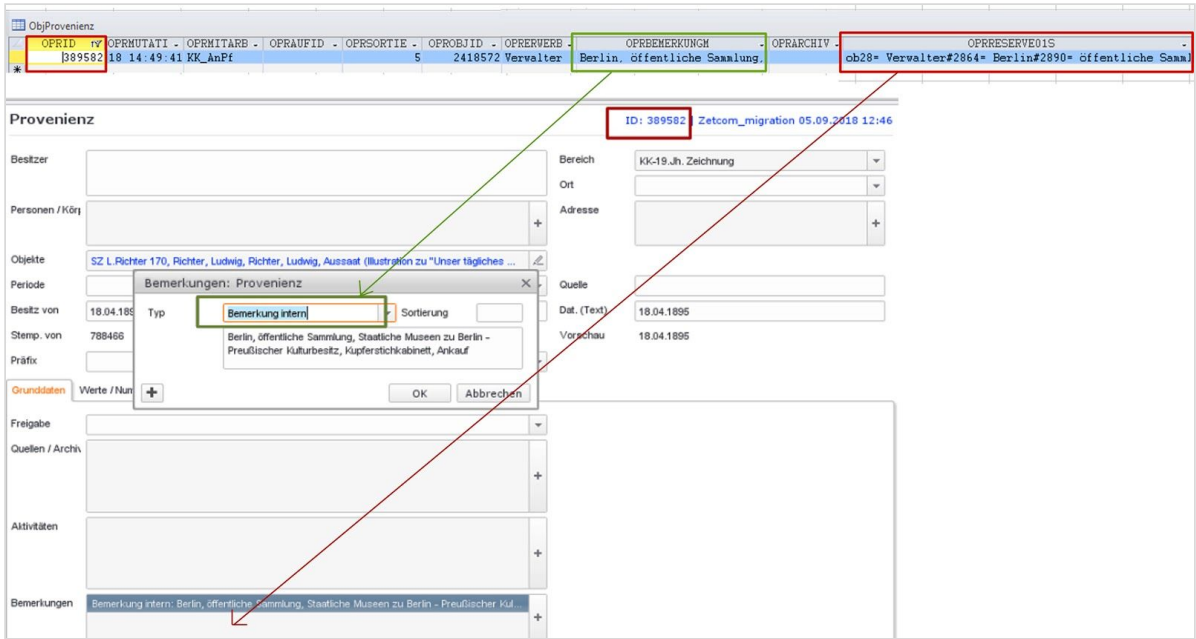


Abbildung 2.5: Der dokumentierte Fehler der Unvollständigkeit im Zielsystem

Darauf weisen die zwei gefüllten Felder *OprReserve01S* und *OprBemerkungenM* in der Abbildung 2.5 hin. Da es sich hier um eine *Wiederholgruppe* handelte und die erweiterte Suche (“Bemerkungen” ist nicht leer) nur einen Eintrag von jeder *Wiederholgruppe* berücksichtigte, ergab die quantitative Überprüfung mittels der erweiterten Suche in der *RIA*-Anwendung eine korrekte Anzahl der Einträge.

Nr.	ID	Quelle: interner Name Tabelle	Quelle: interner Name Feld	Quelle: Register (Modul)	Quelle: Wiederholgruppe	Quelle: Bezeichnung Feld (Name Thesaurus, Wortliste)	Quelle -> Ziel: Migrationsregel
	92,010	ObjProvenienz	OprID				
	92,010	ObjDaten	ObjAufld				
	92,010	ObjProvenienz	"1" OprID				
55	92,010	ObjProvenienz	OprReserve01S	Erwerb./ Inventar	Provenienz: Bemerkungen	Text	
56	92,010	ObjProvenienz	"Midas"	Erwerb./ Inventar	Provenienz: Bemerkungen	Text	
	92,015	ObjProvenienz	OprID				
	92,015	ObjDaten	ObjAufld				
	92,015	ObjProvenienz	"2" OprID				
57	92,015	ObjProvenienz	OprBemerkungenM	Erwerb./ Inventar	Provenienz	Bemerkungen	
58	92,015	ObjProvenienz	"Bemerkung intern"	Erwerb./ Inventar	Provenienz: Bemerkungen	Text	

Abbildung 2.6: Ein Screenshot des Mappings für das *Provenienz*-Modul, Ausschnitt einer Quellstruktur in Bezug auf *Wiederholgruppe* “Bemerkungen”

Ziel: Modulname	Ziel: Org. Unit	Ziel: Bereich / Register	Ziel: Wiederholgrupp e	Ziel: Wiederholgrupp e intern	Ziel: Feldbezeichnung	Ziel: interner Feldname	Ziel: Wortliste / Referenz intern
Ownership				OwnNotesGrp		Ownership_Id	
Ownership				OwnNotesGrp		Ownership_OrgId	
Ownership				OwnNotesGrp		Ownership_ItemNo	
Ownership		Hauptbereich	Bemerkungen	OwnNotesGrp	Bemerkungen	NotesClb	
Ownership		Hauptbereich	Bemerkungen	OwnNotesGrp	Typ	TypeVoc	OwnNotesTypeVgr
Ownership				OwnNotesGrp		Ownership_Id	
Ownership				OwnNotesGrp		Ownership_OrgId	
Ownership				OwnNotesGrp		Ownership_ItemNo	
Ownership		Hauptbereich	Bemerkungen	OwnNotesGrp	Bemerkungen	OwnNotesClb	
Ownership		Hauptbereich	Bemerkungen	OwnNotesGrp	Typ	TypeVoc	OwnNotesTypeVgr

Abbildung 2.7: Ein Screenshot des Mappings für das Provenienz Modul, Ausschnitt einer Quellstruktur in Bezug auf *Wiederholgruppe* “Bemerkungen”

Unabhängig davon lieferte die stichprobenartige Überprüfung einzelner Datensätze eine Inkonsistenz, die sich eben mit fehlenden Einträgen in der *Wiederholgruppe* bewiesen ließ. Lediglich ein Wert aus dem Feld *OprBemerkungenM* wurde migriert. Der Grund dafür lag darin, dass sich die Werte während des Migrationsprozesses überschrieben haben, so dass nur das in einem zweiten Mappingblock definierte Feld - *OprBemerkungenM* - in das Zielsystem übernommen wurde.

Die doppelt und unvollständig migrierten Daten im Zielsystem gehörten zu den meisten Fehlern und waren zugleich die Grundlage der häufigsten Testfälle, die von mir vorgenommen wurden. Die vorgeführten Beispiele veranschaulichen, dass die Migrationsverifikation ein komplexer Prozess ist. Dabei wird es deutlich, dass die Überprüfung von Datenmigrationen hauptsächlich in Hinsicht auf deren Vollständigkeit, Quantität, Richtigkeit und Eindeutigkeit umgesetzt wird.

Schwierigkeiten

Das Testen von Datenmigration war für mich sehr herausfordernd. Im Laufe der Überprüfung ergaben sich viele Schwierigkeiten. Viele davon betrafen das Testverfahren selbst und daraus resultierende Einordnung der Fehlerarten. Die Ergebnisse, die ich oft vorgelegt habe, waren nicht immer nachvollziehbar für meine Kollegen, die die Datenmigration korrigiert haben. Meine Dokumentation vereinfachte zwar den Prozess, leider kamen auch Fehler, die sehr oft in Hinsicht auf den Inhalt von Mappingdokumenten auftraten. Diese enthielten falsch bzw. nicht eindeutige Bedingungen, die anschließend für eine Unklarheit sorgten. Einige Regeln mussten deshalb *ad hoc* umformuliert bzw. entfernt werden. Das bedingt hingegen einen hohen Zeit- und Kommunikationsaufwand.

2.2.2 Individuallösungen

Hintergrund

Die Entwicklung von Individuallösungen bildet ein wichtiger Punkt in Bezug auf *RIA* Anwendungen. *RIA* Framework stellt nicht nur eine grundlegende Funktionalität, Logik und abstrakte Struktur der Anwendung bereit, sondern lässt sich auch exakt an die kundenspezifischen Anforderungen anpassen. Mit sog. *Custom Code* werden eben im Falle spezifischer Kundenwünsche einzelne Funktionalitäten und die Darstellungsart für ein individuelles Projekt vorgenommen. Im Grunde wird mit Hilfe von Individuallösungen die Darstellungsart, Logik sowie Interface neuer Features spezifiziert und implementiert. Ferner wird mit Darstellungsart u.a. Anzeige und Formatierung der Daten sowie Anzeigesteuerung der GUI Komponenten gemeint. Die logischen Zusammenhänge thematisieren CRUD-Operationen, Handeln von Events, sowie benutzerdefinierten Suchen. Bei Interface handelte es sich um den richtigen Umgang mit externen Dateien und um Exportverfahren.

Aufgaben

Meine Aufgabe war es, die bereits implementierten Individuallösungen vornehmlich für das *Staatliche Museen zu Berlin* Projekt zu testen. Die Kommunikation verlief primär über *OTRS*, wo ich meine Testergebnisse direkt im zugewiesenen Ticket dokumentieren konnte. Außerdem kam es häufig vor, dass ich ebenfalls mit *MS Excel* gearbeitet habe, um die Fehler zu dokumentieren.

Die Aufgaben habe ich zwischendurch zugewiesen bekommen und deren Umfang in der Regel klein gehalten wurde, so dass ich mit den Ergebnissen relativ schnell fertig war. Es handelte sich um Funktionalitäten in Hinsicht auf bestimmte Module in der *RIA* Anwendung. So waren es oft die Tests von Mechanismen, die in jeweiligen *Wiederholgruppen* eines bestimmtes Moduls implementiert wurden. Bei den Überprüfungsszenarien wurden meist die relevanten für die Überprüfung Anwendungsbereiche mit ihrem internen Namen versehen sowie mit einer logischen Verknüpfung verbunden. Hierfür eignet sich das Beispiel von zwei Szenarien in Bezug auf *Adresse*-Modul mit Testen vom Mechanismus zur Filterung von *Adress*-Referenzen, was im Grunde die Anzeigesteuerung der GUI Komponenten ausmacht:

- bei *ObjRightsGrp.HolderRef* (Objekte: Rechte: Inhaber) sollen nur Adressen angezeigt werden mit *AdrReferenceFilterGrp.ReferenceFilterVoc = ObjRightsGrp.HolderRef*
- bei *ConParticipantGrp.AddressRef* (Restaurierung: Beteiligte: Adresse) sollen nur Adressen angezeigt werden mit *AdrReferenceFilterGrp.ReferenceFilterVoc = Restaurator*

Ein anderes Szenario in Bezug auf *Registrar*-Modul, deren Struktur auf einer logischen Verknüpfung basierte, wurde etwa folgendermaßen formuliert:

Wenn *ObjIlluminationGrpItem* WENN *StatusVoc = current* and *ObjIlluminationGrpItem* *StatusVoc = historical*
-> wird ein Item mit *StatusVoc = current* im Registrar angelegt

Darüber hinaus gehörten zu den anderen von mir getesteten Funktionalitäten u. a. die Überprüfung von den erweiterten Zeitraumsuchen anhand eines intern konzipierten Schemas sowie Multimedia-Upload nach einer vorgegebenen Matrix.

Die mehr aufwendige und damals noch nicht vollständig spezifizierte Funktionalität, die ich in Bezug auf Testen von Individuallösungen genauer thematisieren möchte, beschäftigt sich mit dem Modul namens *Zitierweise*, das in Verbindung mit dem *Literatur*-Modul steht. Das *Zitierweisen*-Modul ermöglichte die Erstellung einer beliebigen Zitierdefinition, die im Wesentlichen aus selbst definierten Teilen besteht. Diese Teile bilden ein Muster, das die vom *Literatur*-Modul verfügbaren Felder oder Einträge aus einer *Wiederholgruppe*, zusätzliche Präfixe bzw. Suffixe sowie Filterung und Trennzeichen in einer bestimmten Reihenfolge umfassen kann. Eine auf diese Art und Weise erstellte Definition, wird dann automatisch im *Literatur*-Modul eingeblendet. Die Abbildung 2.8 zeigt die Ansicht vom *Zitierweisen*-Modul.

Mein Vorgehen in diesem Fall verlief folgendermaßen: Zunächst habe ich gründlich den Mechanismus zur Ausgabe einer beliebigen Zitierweise im *Literatur*-Modul überprüft. Diese Eigenschaft ließ sich auch direkt im *Zitierweisen*-Modul mit Hilfe vom Feld namens ‘*Test Zitation*’ checken. Das zweite für die Überprüfung kontextrelevante Feld war “*Vorschau*”, wo die angewandten Namensfelder angezeigt wurden. Folglich habe ich die implementierte Funktionalität im *Zitierweisen*-Modul getestet, in dem ich einige Muster eingepflegt habe. In diesem Schritt konnte ich annehmen, wo die Schwerpunkte meiner Testfälle liegen sollten. Im Anschluß habe ich mir einige Testszenarien überlegt und sie erstellt (vgl. Abbildung 2.9). Danach habe ich zu jedem Testfall eine mit Screenshot belegte Notiz angefertigt (vgl. Abbildung 2.10).

Abbildung 2.8: Ansicht vom *Zitierweisen*-Modul

- a) Anlegen eines **flachen Feldes** mit einem **Suffix** ", " (Komma, Leerstelle) (das Feld ist im Literaturdatensatz **vorhanden**)
- b) Anlegen eines **flachen Feldes** mit einem **Suffix** " " (Leerstelle) und **Trenner** als ", " (Kommazeichen) (das Feld ist im Literaturdatensatz **vorhanden**)
- c) Anlegen eines **flachen Feldes** mit einem **Präfix** "Verl." (textuell) und **Suffix** ", " (Komma, Leerstelle) und **Trenner** als Leerstelle (das Feld ist im Literaturdatensatz **vorhanden**)
- d) Anlegen eines **flachen Feldes** mit einem **Suffix** ", " (Komma, Leerstelle) (das Feld ist im Literaturdatensatz **nicht vorhanden**)
- e) Anlegen eines **Wiederholgruppeneintrages** 1 Variante
- f) Anlegen eines **Wiederholgruppeneintrages** 2 Variante
- g) Anlegen eines **Wiederholgruppeneintrages** mit **Gruppentrenner** als Leerstelle und **Gruppensuffix** (Kommazeichen)
- h) Anlegen eines **Wiederholgruppeneintrages** mit **Gruppentrenner** als Kommazeichen/Leerstelle und **Gruppenpräfix** Text, **Sortierung Wert (Typ Knotenauswahl)**
- i) Anlegen eines **Wiederholgruppeneintrages** mit **Gruppensuffix** (Komma, Leerzeichen) (das Feld ist im Literaturdatensatz **nicht vorhanden**)
- j) Anlegen eines **Wiederholgruppeneintrages** mit **Gruppentrenner** als Kommazeichen/Leerstelle und **Gruppenpräfix** Text (das Feld ist im Literaturdatensatz **nicht vorhanden**)
- k) Anlegen eines **Konstanten** mit **Suffix** (Kommazeichen) und **Trenner** als Leerstelle, folgend Anlegen eines **flachen Feldes ohne Präfix, Suffix und Trenner**
- l) Anlegen eines **Konstanten** mit **Suffix** (Semikolon) und **Trenner** als Leerstelle, folgend Anlegen eines **Wiederholfeldes mit Suffix**(Komma, Leerstelle)
- m) Anlegen eines **Konstanten** mit **Suffix** (Semikolon) und **Trenner** als Leerstelle, folgend Anlegen eines **Wiederholfeldes mit Suffix**(Komma, Leerstelle) (Die Wiederholgruppe ist leer)

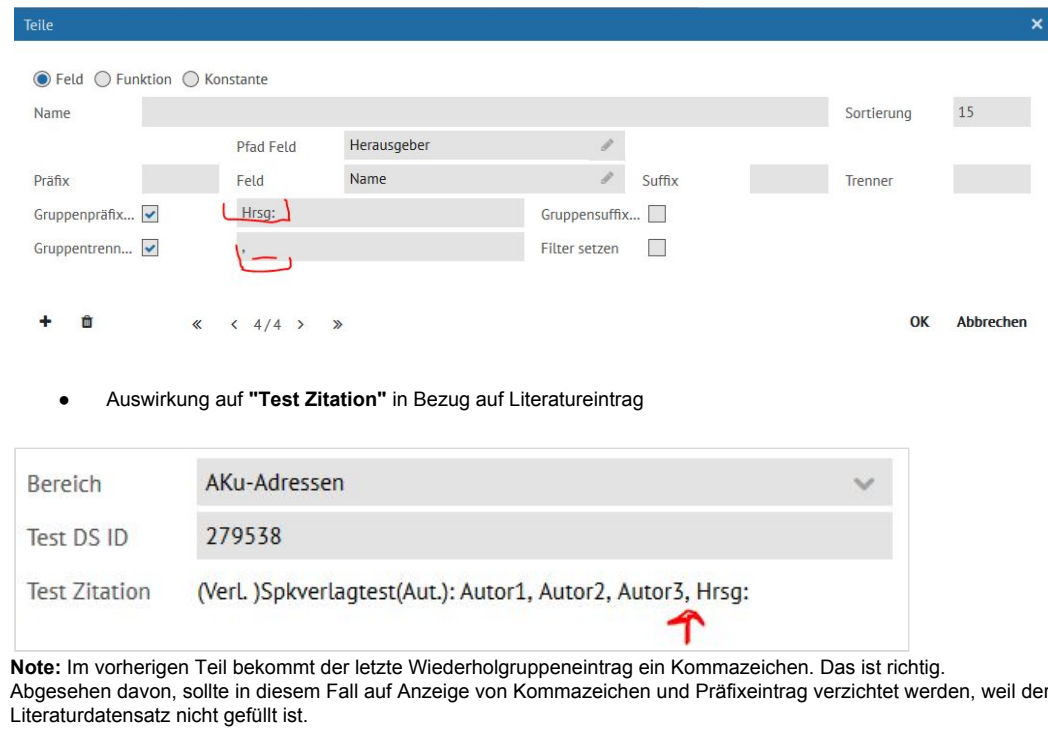
Abbildung 2.9: Übersicht einiger Testfälle in Bezug auf *Zitierweisen*-Modul

- i) Anlegen eines **Wiederholgruppeneintrages** mit **Gruppensuffix** (Komma, Leerzeichen) (das Feld ist im Literaturdatensatz **nicht vorhanden**)

- Auswirkung auf "Test Zitation" in Bezug auf Literatureintrag

Note: es wird ein Kommazeichen angezeigt, obwohl die Wiederholgruppe im Literaturmodul leer ist.

j) Anlegen eines **Wiederholgruppeneintrages** mit **Gruppentrenner** als Kommazeichen/Leerstelle und **Gruppenpräfix** Text (das Feld ist im Literaturdatensatz **nicht vorhanden**)



Teile

☒ Feld ☐ Funktion ☐ Konstante

Name Sortierung 15

Pfad Feld Herausgeber

Präfix Feld Name Suffix Trenner

Gruppenpräfix... ☒ Hrsg: Gruppensuffix... ☐

Gruppentrenn... ☒ Filter setzen ☐

+

• Auswirkung auf "Test Zitation" in Bezug auf Literatureintrag

Bereich

Test DS ID

Test Zitation (Verl.)Spkverlagstest(Aut.): Autor1, Autor2, Autor3, Hrsg:

Note: Im vorherigen Teil bekommt der letzte Wiederholgruppeneintrag ein Kommazeichen. Das ist richtig. Abgesehen davon, sollte in diesem Fall auf Anzeige von Kommazeichen und Präfixeintrag verzichtet werden, weil der Literaturdatensatz nicht gefüllt ist.

Abbildung 2.10: Ausführliche Anzeige vom Testfall i) und j)

4. Zusammengefasst

IST: beim Anlegen eines flachen Feldes wird ein Trenner gesetzt. Der Trenner beeinflusst die Anzeige von Vorschau, nicht die Anzeige von Test Zitation (1b)(1c)

SOLL: Das Setzen von Trenner sollte Auswirkung auf Test "Zitation" Anzeige haben.

IST: Das Feld im Literaturmodul ist leer. Die "Test Zitation" Anzeige zeigt eine Leerstelle mit Trennern und Suffixen an.

SOLL: Die "Test Zitation" Anzeige sollte das Feld mit dem Kommazeichen und eventueller Leerstelle ausblenden (1d) (1i) (1j)

IST: Anlegen eines Wiederholgruppenteiles kann auf zwei Arten erfolgen. Die erste Variante ermöglicht dem Nutzer die Auswahl des PfadFeldes und erneute Auswahl des Feldes.(1 e) Die zweite Variante erlaubt dem Nutzer das gleichzeitige Markieren von Pfadfeld und Feld (führt zur Fehlermeldung).(1f)

SOLL: Anlegen eines Wiederholgruppenteiles optimieren.

1. Dem Nutzer das gleichzeitige Markieren von gewünschten Felder erlauben und Checkboxes mit Gruppensuffix, -präfix und Gruppentrenner einblenden lassen, sowie Korrektur der Fehlermeldung vornehmen.

2. Den Nutzer verhindern, das gleichzeitige Auswählen von Feldern zu erlauben.

Abbildung 2.11: Zusammenfassung der gefundenen Fehler

Die Ergebnisse meiner Arbeit habe ich im letzten Schritt verallgemeinert und prägnanter mit sog. Ist-Soll Vergleich ausgedrückt. Die Abbildung 2.11 zeigt eine von mir verallgemeinernde Zusammenfassung der Fehler. Im Anschluss wurden meine Ergebnisse

kontrolliert bzw. umformuliert, so dass sie an das Entwicklungsteam weitergegeben werden konnten.

2.2.3 Sonstige Aufgaben

Neben meinen Hauptaufgaben hatte ich auch die Möglichkeit, an einer Online Version von *MuseumPlus* mitzuarbeiten. Die *MuseumPlus Online* ist eine Web-Anwendung, wobei im Fokus lediglich eine Sammlung-Darstellung bestehend aus *Objekt*-, *Künstler*- und *Digital Assets*- Modulen steht. Außerdem hat *MuseumPlus Online* den Vorteil, dass es direkt in die Kundenwebsite integriert werden kann.

Meine Aufgabe bestand darin, ein paar Änderungen für einen bereits existierenden Prototypen von *MuseumPlus Online* vorzunehmen. Die Web-Anwendung wurde auf der Basis von HTML5 und CSS3 erstellt und basierte auf einer Java-Bibliothek namens *Thymeleaf*, die im Grunde eine Template Engine für XML, XHTML und HTML5 bereitstellt.⁷ Die Änderungen betrafen hauptsächlich die Optimierung der Schriftgrößen und Abstände sowie Einbinden von Icons-Elementen. Zunächst installierte ich die Java-Entwicklungsumgebung *JetBrains IntelliJ IDEA 3*, *Cygwin* und den Anwendungsserver *GlassFish*. Um eine Verbindung zur Datenbank aufzubauen, habe ich *pgAdmin* installiert.

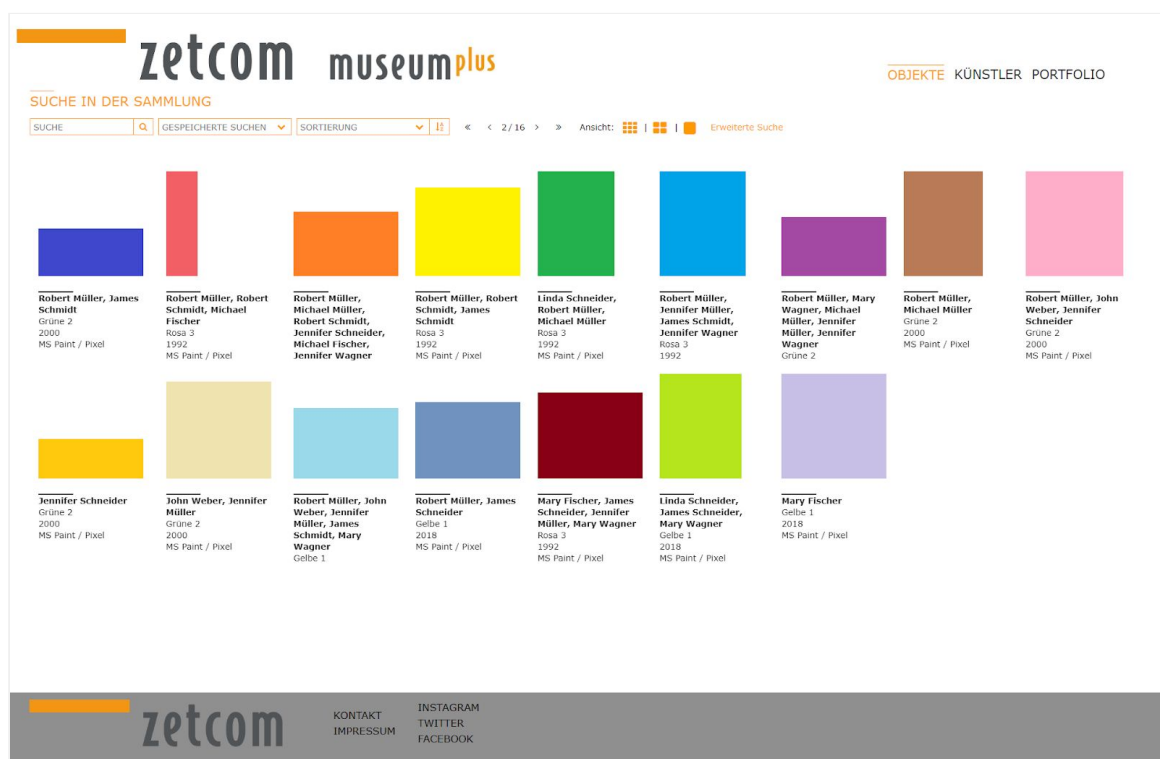


Abbildung 2.12 Die finale Umsetzung der *MuseumPlus Online*

⁷ Wikipedia: "Thymeleaf", unter: <https://en.wikipedia.org/wiki/Thymeleaf> (abgerufen am 10.04.2019)

Daraufhin erhielt ich einen *Gitlab*-Zugang und konnte mit den ersten Aufgaben anfangen. An erster Stelle wurde mir die Ordnerstruktur sowie die Auswirkung auf die einzelnen Komponenten in der Anwendung erklärt.

Die ersten Änderungen waren es, Schriftgrößen und Schriftarten zu ändern sowie das Logo anzupassen. Dazu habe ich mit Hilfe von CSS-Präprozessor *Sass* den Font global gesetzt und Media-Queries eingebunden. Einige Aufgaben wie das Einbinden von Icons, habe ich mit Hilfe von einer schriftartbasierten Icon Sammlung *Font Awesome* umgesetzt.

Die Abbildung 2.13 zeigt einen Ausschnitt von einem Quellcode für ein HTML Template *moduleHeader-de.html*, der für das Einbinden von Icons relevant war.

```
<div class="col-auto mr-2 viewIcons">
  <div class="iconPrefix">Ansicht: </div>    <div class="showLBmedium"
mp:uiobject="ObjOnlineSwitchToLightboxMediumLnk">[LB view small]</div>
  <div class="iconSep"> | </div>
  <div class="showLBlarge" mp:uiobject="ObjOnlineSwitchToLightboxLargeLnk">[LB view big]</div>
  <div class="iconSep"> | </div>
  <div class="showCatView" mp:uiobject="ObjOnlineSwitchToCatalogViewLnk">[CatalogView]</div>
</div>
```

Abbildung 2.13: Ein Ausschnitt von einem Quellcode für ein HTML Template *moduleHeader-de.html*

Um Icons anzuzeigen, musste ich jeweils nach dem Wert des Attributes *mp:uiobject* des dazugehörigen div Elements im internen Konfigurationstool ZAC suchen und eine *print* Ausgabe mit einer ausgesuchten Icon-Art einpflegen, was die Abbildung 2.14 zeigt.

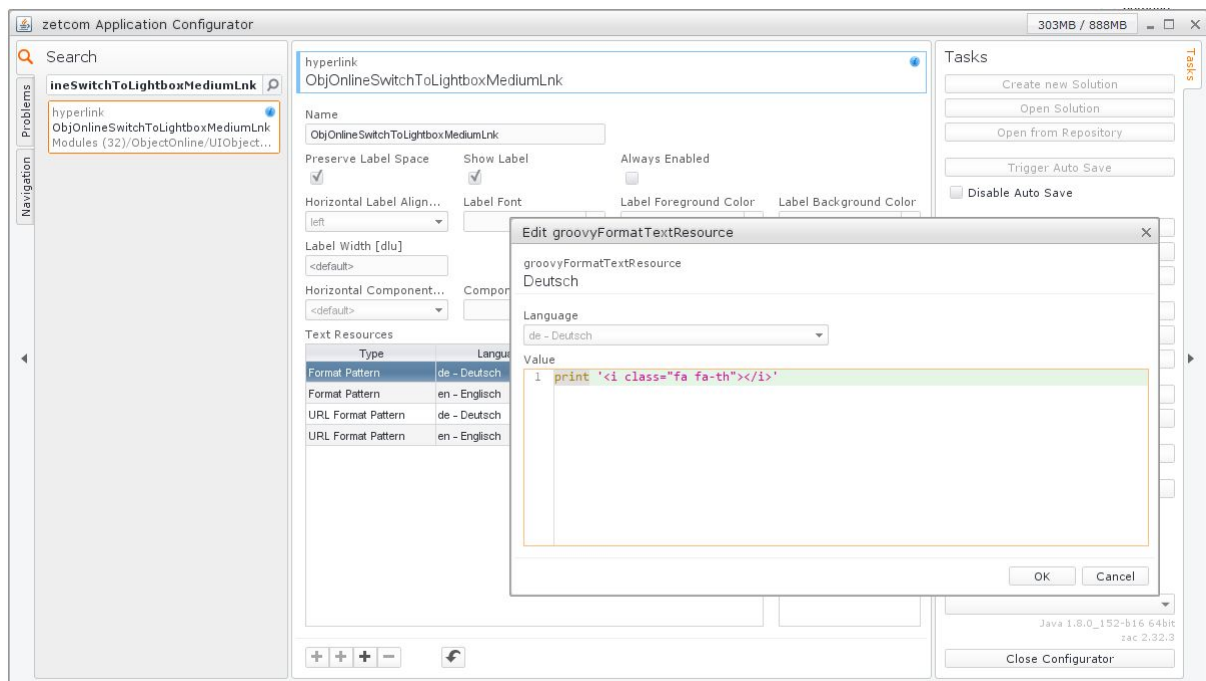


Abbildung 2.14: Ansicht von Konfigurationstool ZAC, Einpflegen einer *Print* Ausgabe für *ObjOnlineSwitchToLightboxMediumLnk*

Nach diesem Schritt konnte man bereits die Icons in der Web-Anwendung erkennen. Für deren Größe und Ausrichtung habe ich ein paar Codezeilen mit CSS3 in einer *header.scss* Datei geschrieben. Ein Auszug aus dem CSS zeigt der folgende Abschnitt.

```
01: /*
02: Placement of links
03: */
04: // adjust for small and middle view
05: .col-auto.mr-2.viewIcons {
06:     margin-right: 15px !important;
07:     float: left;
08:
09:     div {
10:         float: left;
11:     }
12:
13:     div:not(.iconPrefix) div:not(.iconSep) {
14:         font-size: 14px;
15:         line-height: (5em / 3);
16:     }
17:
18:     .showLBmedium,
19:     .showLBlarge,
20:     .showCatView {
21:         padding-left: 6px;
22:     }
23:
24:     .iconPrefix {
25:         padding: 3.5px 10px;
26:     }
27:
28:     .iconSep {
29:         padding: 3.5px 5px;
30:     }
31: }
32: //media queries for large view
33: @media screen and (min-width: 991px) {
34:
35:     .col-auto.mr-2.viewIcons {
36:
37:         div:not(.iconPrefix) div:not(.iconSep) {
38:             font-size: 24px !important;
39:             line-height: (3em / 4) !important;
40:         }
41:         .showLBmedium,
42:         .showLBlarge,
43:         .showCatView {
44:             padding-left: 4px !important;
45:         }
46:     }
47: }
```

Die Abbildung 2.12 zeigt eine finale Umsetzung der *MuseumPlus Online* mit von mir eingeführten Änderungen. Meine Umsetzung habe ich im *Gitlab* veröffentlicht und sie wurden dann im letzten Schritt von der zuständigen Person kontrolliert und zusammengeführt.

3. Praktikum und Studium

Mein Studium hat mir sehr geholfen, meine Aufgaben während des Praktikums zu erfüllen. Die Möglichkeit, an dem großen Projekt bei der Datenmigration für *Staatliche Museen zu Berlin* mitzuarbeiten, eröffnete mir neue Denkhorizonte und zeigte praktische Anwendungsfälle im Arbeitsalltag vom Softwareunternehmen. Besonders die im Studium erworbenen Kenntnisse im Bereich Datenbanken, Web-Engineering und Software-Engineering sowie die erlernten Arbeitsweisen erleichterten mir die erfolgreiche Bewältigung meiner Aufgaben. Die Möglichkeit, bei der Konzeption und Durchführung von Testfällen sowie dem Berichten von Fehlern mitzuarbeiten, schaffte eine gute Grundlage, meine in vielen Aufgaben angewandten Vorgehensweisen fachlich auf der Wahlpflichtveranstaltung “Softwarequalität und -test” zu ergänzen. Diese Kombination kann ich weiterhin für meine Bachelorarbeit einsetzen.

4. Bewertung des Praktikums

Insgesamt ist mein Gesamteindruck sehr positiv. Das Praktikum hat mir geholfen die gelernte Theorie in der Praxis anzuwenden. Im Laufe der Einarbeitungsphase wurde viel Wert auf das Erlernen neuer Inhalte gelegt. Die Zetcom GmbH bot eine angenehme Arbeitsatmosphäre und die Möglichkeit aktiv an den Arbeitsabläufen der Firma teilzunehmen. Die Kollegen waren jederzeit bereit, ihr Wissen mit mir zu teilen. Besonders gefallen hat mir, dass ich großen Freiraum und Vertrauen für Bewältigung meiner Aufgaben hatte. Ich hatte die Möglichkeit eigenständig an den Aufgaben der Qualitätssicherung und Front-end Entwicklung mitzuwirken und konnte dadurch Erfahrung beim manuellen Testen und bei der Programmierarbeit sammeln. Dabei konnte ich beobachten, wie Konzepte, beispielsweise ETL-Prozess, in realen Problemstellungen angewandt werden. Die Vielfalt der Aufgaben ermöglichte es mir, viele neue Fachgebiete kennenzulernen und bildet eine solide Basis für meine beruflichen Ziele.

Abbildungsverzeichnis

Abbildung 2.1: Ein Screenshot des Mappings für das Objekt Modul, Ausschnitt einer Quellstruktur in Bezug auf flache Felder und fixe Wortlisten	5
Abbildung 2.2: Ein Screenshot des Mappings für das Objekt Modul, Ausschnitt einer Zielstruktur in Bezug auf flache Felder und fixe Wortlisten	6
Abbildung 2.3: Ein Screenshot der Access Quelltablette Typ mit vollständiger Wortliste ObjCategoryVgr	8
Abbildung 2.4: Ansicht einer Wortliste ObjCategoryVgr im Zielsystem	8
Abbildung 2.5: Der dokumentierte Fehler der Unvollständigkeit im Zielsystem	9
Abbildung 2.6: Ein Screenshot des Mappings für das Provenienz-Modul, Ausschnitt einer Quellstruktur in Bezug auf Wiederholgruppe "Bemerkungen"	9
Abbildung 2.7: Ein Screenshot des Mappings für das Provenienz Modul, Ausschnitt einer Quellstruktur in Bezug auf Wiederholgruppe "Bemerkungen"	10
Abbildung 2.8: Ansicht vom Zitierweisen-Modul	12
Abbildung 2.9: Übersicht einiger Testfälle in Bezug auf Zitierweisen-Modul	13
Abbildung 2.10: Ausführliche Anzeige vom Testfall i) und j)	14
Abbildung 2.11: Zusammenfassung der gefundenen Fehler	14
Abbildung 2.12 Die finale Umsetzung der MuseumPlus Online	15
Abbildung 2.13: Ein Ausschnitt von einem Quellcode für ein HTML Template moduleHeader-de.html	16
Abbildung 2.14: Ansicht von Konfigurationstool ZAC, Einpflegen einer Print Ausgabe für ObjOnlineSwitchToLightboxMediumLnk	16

Erklärung

Den Praktikumsbericht von Anita Kusnierz habe ich zur Kenntnis genommen.

.....
Ort, Datum

.....
Unterschrift Norbert Kanter

.....
Unterschrift Jörg Kruschinski