HSR
HOCHSCHULE FÜR TECHNIK
RAPPERSWIL

FHO Fachhochschule Ostschweiz

foryouandyourcustomers AG

eShop/PIM Prototyping

Semesterarbeit

Abteilung Informatik
Hochschule für Technik Rapperswil

Herbstsemester 2012

Autor(en):	Patrick Zenhäusern
	Simon Stähli
Betreuer:	Dr. Daniel Keller
Projektpartner:	Dr. Daniel Keller, foryouandyourcustomers AG
Experte:	Dr. Daniel Keller
Gegenleser:	-

HSR
HOCHSCHULE FÜR TECHNIK
RAPPERSWIL
FHO Fachhochschule Ostschweiz

Studienarbeit: SA eShop/PIM Prototyping

Management Summary

Simon Stähli (sstaehli@hsr.ch)
Patrick Zenhäusern (pzenhaeu@hsr.ch)

Inhalt

Inhalt 2
1. Einführung 3
1.1 Zweck 3
1.2 Gültigkeitsbereich 3
1.3 Referenzen 3
2. Ausgangslage 4
3. Projektdaten 4
4. Vorgehen 4
5. Technologie 4
6. Ergebnisse 4

1. Einführung

1.1 Zweck

Das Management Summary soll aussenstehenden Personen in kurzer und knapper Form die wichtigsten Daten über dieses Projekt sowie die dabei entstandene Arbeit liefern.

1.2 Gültigkeitsbereich

Gültig während des Modules „Studienarbeit Informatik“ im Zeitraum des Semesters HS 12.

1.3 Referenzen

Redmine (http://152.96.56.49/redmine/projects/pim)
Glossar (http://152.96.56.49/redmine/projects/pim/wiki/Glossar)

2. Ausgangslage

Im Auftrag der Firma foryouandyourcustomers AG, welche PIM (Product Information Management) Systeme für grössere Firmen konzipiert und realisiert, sollte eine Software für das Prototyping von Navigationsstrukturen für e-Shops und PIM-Systeme entwickelt werden. Den Kunden soll eine benutzerfreundliche Möglichkeit geboten werden, ihre zukünftigen Navigationsstrukturen zu entwickeln, zu testen und zu vergleichen.

3. Projektdaten

Auftraggeber:	foryouandyourcustomers AG
Projektbetreuer:	Dr. Daniel Keller
Projektmitglieder:	Patrick Zenhäusern
	Simon Stähli
Projektdauer:	14 Wochen (21.09.2012 - 21.12.2012)
Projektumfang:	476h
Projektmodell:	Light-RUP

4. Vorgehen

Bei diesem Projekt wurde nach einer vereinfachten Variante von RUP vorgegangen.
In einem ersten Schritt wurden die Anforderungen analysiert und ausgearbeitet. Mögliche Funktionalitäten wurden mittels UseCases festgehalten, sowie ein erster einfacher Prototyp erstellt um die Technologischen-Möglichkeiten zu belegen.
Am Ende der Elaboration-Phase stand der gesamte Projektplan fest und es war bereits ein Prototyp erarbeitet bei dem die wichtigsten Kernfunktionalitäten möglich waren. Diese zeigten auch einen Querschnitt durch die gesamte Architektur auf. In den Construction-Phasen wurden danach iterativ die restlichen Anforderungen, implementiert. Durch Meetings mit dem Auftraggeber wurden jedoch auch während des Projektes noch fortlaufend neue, zum Teil auch grössere, Anforderungen in den Projektplan aufgenommen und später erfolgreich in die Applikation eingebaut.

5. Technologie

Die Applikation wurde als Web-Applikation entworfen und auf der Basis von ASP.NET realisiert. Im Hintergrund greift die Applikation auf eine MySQL Datenbank zu um Daten persistent zu speichern. Um den Benutzern eine möglichst dynamische und benutzerfreundliche Web-Applikation zu bieten, wurden diverse Funktionalitäten, beispielsweise Drag & Drop, mittels JavaScript implementiert.

6. Ergebnisse

Als Ergebnis dieses Projektes steht eine betriebsbereite Web-Applikation, welche Produktmanagern von e-Shops oder PIM-Systemen das Erstellen, Editieren und Testen von Navigationshierarchien auf eine leichte Weise ermöglicht. Die Applikation ist Multi-User fähig, wobei über einen Administratoren-Account neue Benutzer erstellt und verwaltet werden können. Die Kernfunktionalität stellt hierbei klar das Erstellen und Editieren von Navigationshierarchien mittels JavaScript (Client-Seitig) und ASP.NET bzw. C# (Server-Seitig) dar. Um es Produktmanagern zu erlauben, diese Arbeit möglichst effektiv zu erledigen, stehen ihnen weitere Funktionalitäten wie das Rückgängig machen von Arbeitsschritten, das Vergleichen von Navigationen, Statistiken oder auch das Importieren und Exportieren von Navigationshierarchien aus Excel-Dateien zur Verfügung.

HSR
HOCHSCHULE FÜR TECHNIK
RAPPERSWIL
FHO Fachhochschule Ostschweiz

Studienarbeit: SA eShop/PIM Prototyping

Analyse von Navigationsstrukturen

Simon Stähli (sstaehli@hsr.ch)
Patrick Zenhäusern (pzenhaeu@hsr.ch)

Inhalt

Inhalt 2
1. Analyse von Navigationsstrukturen in bekannten Online-Shops 3
1.1 Digitec.ch 3
1.2 Digitec.ch Beta 3
1.3 Coop.ch 3
1.4 Leshop.ch 3
1.5 Opo.ch 4
1.6 Liebherr.com 4
1.7 Amazon.com 4
1.8 Swisscom.ch 4
2. Fazit 4

1. Analyse von Navigationsstrukturen in bekannten Online-Shops

1.1 Digitec.ch

Top Level Elemente	20
Navigationstiefe	3
Filterkriterien	Je nach Kategorie 4 – 10
Mehrfachfilterung	möglich
Artikel ohne Filterkriterien	Teilweise mehrere hundert, fast immer über 50
Sortierung	ja
Persönlicher Eindruck	Sehr gut, Top Level Kategorisierung könnte jedoch besser aufgebaut werden. Ein gesuchter Artikel könnte teilweise mehreren Top Level-Kategorien zugeordnet werden.

1.2 Digitec.ch Beta

Top Level Elemente	12
Navigationstiefe	2
Filterkriterien	Je nach Kategorie 4 – 10
Mehrfachfilterung	möglich
Artikel ohne Filterkriterien	Teilweise mehrere hundert, fast immer über 50
Sortierung	ja
Persönlicher Eindruck	Bessere Top Level Strukturierung als beim alten Shop. Gute Idee, aber schlechte Umsetzung. Wirkt unübersichtlicher als der alte Shop.

1.3 Coop.ch

Top Level Elemente	6
Navigationstiefe	Meist 3, bei einzelnen Kategorien mehr
Filterkriterien	Durchgehend maximal 3
Mehrfachfilterung	Nur ein Filterkriterium ist möglich
Artikel ohne Filterkriterien	Teilweise mehrere hundert, fast immer über 50
Sortierung	ja
Persönlicher Eindruck	Übersichtlich, leider fehlt die Möglichkeit um mehrere Filterkriterien gleichzeitig zu setzen

1.4 Leshop.ch

Top Level Elemente	35
Navigationstiefe	3-4 (4te kategorie bereits sehr fein kategorisiert)
Filterkriterien	Keine, in der Navigations-Struktur als unterstes Level eingebaut
Mehrfachfilterung	-
Artikel ohne Filterkriterien	Meist unter 20 pro Kategorie
Sortierung	ja
Persönlicher Eindruck	Durch zuviele Top Level elmente wirkt die seite unübersichtlich

1.5 Opo.ch

Top Level Elemente	3
Navigationstiefe	4-5
Filterkriterien	Versteckt möglich
Mehrfachfilterung	-
Artikel ohne Filterkriterien	Teils bis 100
Sortierung	Nicht vorhanden
Persönlicher Eindruck	Sehr verwirrender ablauf der Navigationsstruktur. Teilweise zwei Klicks nötig, jedoch immer noch in der selben Kategorie

1.6 Liebherr.com

Top Level Elemente	9
Navigationstiefe	3
Filterkriterien	Nicht möglich
Mehrfachfilterung	-
Artikel ohne Filterkriterien	Bis zu 10
Sortierung	nein
Persönlicher Eindruck	Etwas verwirrend. Hauptkategorien nur von der Startseite sichtbar. Nicht die komplette Hirarchie ist sichtbar, was verwirrend sein kann wenn man in einer Subkategorie ist.

1.7 Amazon.com

Top Level Elemente	15
Navigationstiefe	3
Filterkriterien	Teilweise möglich
Mehrfachfilterung	ja
Artikel ohne Filterkriterien	Sehr viele
Sortierung	ja
Persönlicher Eindruck	Nicht konsistent zwischen den verschiedenen Top Level-Kategorien. Wirkt verwirrend und unübersichtlich.

1.8 Swisscom.ch

Top Level Elemente	Ca. 5
Navigationstiefe	2
Filterkriterien	ja
Mehrfachfilterung	ja
Artikel ohne Filterkriterien	Bis 30
Sortierung	ja
Persönlicher Eindruck	Online-Shop ist nicht abgegrenzt von der restlichen Website, kann verwirrend sein. Shop ansich ist jedoch gut bedienbar und übersichtlich.

2. Fazit

Digitec und Swisscom sind in unseren Augen die übersichtlichsten Online-Shops. Auch diese beiden sind aber noch verbesserbar. Es wundert uns ein wenig, dass keiner der Online-Shops alle Kriterien eines gut bedienbaren Online-Shops umgesetzt hat, obwohl alle Online-Shops relativ neu wirken.

HSR
HOCHSCHULE FÜR TECHNIK
RAPPERSWIL
FHO Fachhochschule Ostschweiz

Studienarbeit: SA eShop/PIM Prototyping

Projektplan

Simon Stähli (sstaehli@hsr.ch)
Patrick Zenhäusern (pzenhaeu@hsr.ch)

Inhalt

Inhalt 2
1. Einführung 3
1.1 Zweck 3
1.2 Gültigkeitsbereich 3
1.3 Referenzen 3
2. Projekt Übersicht 4
2.1 Zweck und Ziel 4
3. Projektorganisation 4
3.1 Organisationsstruktur 4
3.1.1 Kommunikation 4
3.2 Externe Schnittstellen 4
4. Management Abläufe 4
4.1 Vorgehensweise 4
4.2 Zeitbudget 5
4.3 Zeitliche Planung 5
4.3.1 Phasen / Iterationen 5
4.4 Besprechungen 6
5. Risikomanagement 6
5.1 Risiken 6
5.2 Umgang mit Risiken 6
6. Infrastruktur 6
6.1 Zentraler Server 6
6.2 Entwicklungsumgebung 6
6.3 Betriebssystem 7
6.4 Dokumentationswerkzeug 7
7. Qualitätsmassnahmen 7
7.1 Dokumentation 7
7.2 Projektmanagement 7
7.3 Entwicklung 7
7.3.1 Vorgehen 8
7.3.2 Code Reviews 8
7.4 Coding Standards & Tools 8
7.5 Testen 8
7.5.1 Code Coverage 8
7.5.2 Usability-Tests 9

1. Einführung

1.1 Zweck

Dieses Dokument enthält den Projektplan der Studienarbeit eShop/PIM Prototyping. Darin enthalten sind Informationen zu folgenden Punkten:

- Gruppenmitglieder und deren Rolle während der Projektentwicklung
- Beschreibung des Projektes sowie dessen Main- und Advanced-Features
- Zeitliche Planung
- Festlegung der Risiken
- Festlegung der Qualitätsmassnahmen

1.2 Gültigkeitsbereich

Gültig während des Modules „Studienarbeit Informatik" im Zeitraum des Semesters HS 12.

1.3 Referenzen

Redmine (http://152.96.56.49/redmine/projects/pim)
Glossar (http://152.96.56.49/redmine/projects/pim/wiki/Glossar)

2. Projekt Übersicht

Es soll ein webbasiertes Tool zum Verwalten & Ausprobieren von Navigationsstrukturen entwickelt werden. Dabei wird ASP.NET verwendet. Mittels des Tools soll es für verschiedene Benutzergruppen (Kunden), möglich sein, Navigationen zu importieren (aus einem Excel), zu erstellen, zu verändern und schlussendlich zu testen. Der Kunde soll so erkennen können, ob eine Navigation sinnvoll & kundenfreundlich aufgebaut ist. Dabei geht es vor allem um die Navigationstiefe bzw. die Navigationshierarchie. Der Kunde sollte durch das Tool beispielweise erkennen können, dass die Navigation zu kompliziert ist, weil zu viele „Klicks" gebraucht werden um eine bestimmte Kategorie zu erreichen oder weil die Kategorie zu „versteckt" (in der falschen Hierarchie) ist.

2.1 Zweck und Ziel

- Das wichtigste Ziel bei dem Projekt ist es eine funktionierende, durchdachte und somit gut designte Software-Lösung abzuliefern. Der Fokus liegt dabei auf einer sauberen Programmierung und einer vollständigen Umsetzung der Anforderungen.
- Erstellen eines webbasierten Tools, welches mit einfachen Mitteln (Drag & Drop, Inline Editing) erlaubt eine Navigationsstruktur zu erstellen und zu verändern.
- Anwenden des bisher gelernten Wissens/Fähigkeiten, mit Schwerpunkt auf den Modulen Software-Engineering 1 und 2, Programmieren 1 und 2, sowie User-Interfaces 1 und 2.

3. Projektorganisation

Das Projektteam besteht aus zwei Mitgliedern und einem Betreuer.

3.1 Organisationsstruktur

Da das Team aus nur zwei Personen besteht, sehen wir keine Notwendigkeit Rollen zu verteilen. Wir sehen uns beide als gleichwertig an und werden auch die Arbeit gleichmässig aufteilen.
Wir werden uns jeweils gegenseitig informieren und „briefen", so dass wir beide über das gesamte Wissen verfügen.

3.1.1 Kommunikation

Die Hauptkommunikationskanäle sind:

- Chat (Facebook, WhatsApp)
- Redmine
- Face-To-Face

Das Team trifft sich jeden Donnerstag um ca. 12.00 Uhr bis ca. 17.00 Uhr, sowie jeden Freitag um ca. 12.00 Uhr bis ca. 17.00 Uhr an der HSR um am Projekt zu arbeiten und allfällige Schwierigkeiten zu behandeln. Ausnahmen hierbei treten vor allem in der Construction-Phase auf. Es kann vorkommen, dass zeitweise von zu Hause aus programmiert wird.

3.2 Externe Schnittstellen

Als Ansprechperson ist **Dr. Daniel Keller (d1keller@hsr.ch)** für dieses Projekt zuständig.

4. Management Abläufe

4.1 Vorgehensweise

Es wird nach einer vereinfachten Form des Rational Unified Process vorgegangen. Details zu den einzelnen Phasen und Iterationen befinden sich in diesem Kapitel unter dem Abschnitt Phasen / Iterationen.

4.2 Zeitbudget

Wir gehen davon aus das jedes Projektmitglied im Durchschnitt 17 Stunden pro Woche aufwenden muss. Über die insgesamt 14 Wochen ergibt dies insgesamt 476 Personen-Stunden. Die Stunden errechnen sich an der Anzahl der Credits für das Modul „Studienarbeit Informatik" (8 Credits) und dem Zeitaufwand pro Credit (ca. 30h). Die Stunden sind jedoch nur eine Richtzeit sollten die Projektteilnehmer die geforderte Aufgabe in kürzerer Zeit lösen, so muss nicht durch sinnlose Zusatzarbeit das Zeitbudget komplett ausgeschöpft werden.
Das Projekt läuft vom 21.09.2012 bis spätestens 21.12.2012 17:00 Uhr und wird termingerecht beendet. Optionale Features können bei Bedarf und bei genügend Zeit umgesetzt werden. Der Fokus liegt ganz klar darin, die Basisversion fertig zu stellen.

4.3 Zeitliche Planung

Die Zeitliche Planung kann komplett in Redmine verfolgt werden, alle Phasen sind dort aufgelistet:
http://152.96.56.49/redmine/projects/pim/roadmap
Zusätzlich befindet sich im Ordner „08. Projektmonitoring & Schlussbericht" eine Zeitauswertung, sowie eine Auswertung von allen Arbeitspaketen.

4.3.1 Phasen / Iterationen

4.3.1.1 Arbeitspakete / Phasenplanung: Redmine

Da wir mit Redmine arbeiten sind alle Arbeitspakete und Phasen direkt dort ersichtlich.
Für den Betreuer ist ein Konto eingerichtet:

Benutzername	Passwort	URL
d1keller	<Herr Keller bekannt>	http://152.96.56.49/redmine/projects/pim

4.3.1.2 Inception

http://152.96.56.49/redmine/versions/1

4.3.1.3 Elaboration E1

http://152.96.56.49/redmine/versions/2

4.3.1.4 Elaboration E2

http://152.96.56.49/redmine/versions/3

4.3.1.5 Construction C1

http://152.96.56.49/redmine/versions/4

4.3.1.6 Construction C2

http://152.96.56.49/redmine/versions/5

4.3.1.7 Construction C3

http://152.96.56.49/redmine/versions/8

4.3.1.8 Construction C4

http://152.96.56.49/redmine/versions/9

4.3.1.9 Transition

http://152.96.56.49/redmine/versions/6

4.4 Besprechungen

Besprechungen/Arbeiten des Teams finden in der Regel jeweils freitags oder donnerstags Nachmittag in der Mensa der HSR statt.
Anwesend sind alle Teammitglieder und der Betreuer.
In späteren Phasen/Iterationen (ab Construction) können bei gutem Vorankommen und wenig Fragen die Meetings auch nur im Zwei-Wochen Rhythmus stattfinden. Ziel der Besprechungen ist es, den aktuellen Stand des Projektes zu protokollieren, Aufgaben zu verteilen und auf allfällige Probleme oder Fehler hinzuweisen. Ausserdem werden an den Meetings weitere Schritte in der Projektentwicklung geplant und festgehalten. Dazu wird jeweils ein Sitzungsprotokoll erstellt (direkt in Redmine)

5. Risikomanagement

5.1 Risiken

Siehe Datei TechnischeRisiken.pdf.

5.2 Umgang mit Risiken

Da wir eine sehr geringe Anzahl an projektspezifische Risiken haben, haben wir nur sehr kleine, generelle Risiken, wie zum Beispiel Krankheit bzw. Totalausfall eines Teammitgliedes. Diese Risiken haben dafür aber ein sehr starkes Schadenspotenzial. Eine (nicht andauernde) Krankheit kann durch Zusatzaufwand des zweiten Teammitgliedes und unserer eingeplanten Reservezeit (C4) vorgebeugt werden. Der Totalausfall eines Teammitgliedes hingegen kann sicherlich nicht komplett kompensiert werden, würde dieser Fall eintreten so müsste das Projekt entweder abgebrochen, verlängert oder mit weniger Anforderungen versehen werden. Da dieses Risiko jedoch sehr tief ist, wird kein detaillierter Plan dafür aufgestellt.

Die technischen Risiken zeichnen sich hauptsächlich dadurch aus, dass die Einarbeitung oder Umsetzung der Materie für einen der Teammitglieder unterschätzt wird und sich dies in einem erhöhten Zeitbedarf niederschlägt. Tiefere Anforderungen an die technische Umsetzung können nicht gemacht werden, da sie als Bestandteil unserer Anforderung fest an das fertige Produkt gebunden sind. Spätestens bei End of Elaboration sollte aber dieses Risiko eliminiert sein, da bis zu diesem Zeitpunkt die Einarbeitung abgeschlossen sein sollte. Falls nicht müsste über einen Projektabbruch bzw. eine Planänderung nachgedacht werden.

6. Infrastruktur

Die Teammitglieder arbeiten alle mit ihren persönlichen Laptops.

6.1 Zentraler Server

Dem Team steht ein Ubuntu 64bit-Server zur Verfügung. Dieser kann über die IP http://152.96.56.49 erreicht werden oder per SSH.
Jedes Projektmitglied hat einen persönlichen Account, welcher dem HSR Kürzel entspricht.

6.2 Entwicklungsumgebung

Für die Programmentwicklung wird mit Microsoft Visual Studio 2012 gearbeitet.

Es werden bei allen Visual Studios die Standardeinstellungen beibehalten, so dass überall die gleichen Formatierungsregeln etc. verwendet werden.

6.3 Betriebssystem

Die Teammitglieder verwenden beide Microsoft **Windows 7 64bit.**
Weil mit asp.NET programmiert wird und das Produkt später deshalb so oder so auf einem Windows Server mit IIS laufen wird, ist das Betriebssystem für die Lauffähigkeit des Produktes nicht weiter relevant.

6.4 Dokumentationswerkzeug

Die Dokumentation erfolgt mithilfe von Microsoft Office 2010 und Redmine.

7. Qualitätsmassnahmen

Massnahme	Ziel	Zeitraum
GoogleDrive-Ordner für Diagramme / Bilder / PDFs	Verfügbar für Alle, automatisches Backup	Aufsetzung zu Beginn von E1 (SW 02) Verwendung während dem gesamten Projekt
Verwendung von Git als Revision Control System	Aktueller Projektstatus und Versionshistorie für jeden zugänglich	Aufsetzung zu Beginn von E1 (SW 02) Verwendung während dem gesamten Projekt
Automatisiertes Backup von Redmine, Git Repository, sowie den MySQL Datenbanken via Unix-Script	Verhinderung von Datenverlust bei Crash des Servers.	Aufsetzung zu Beginn von E1 (SW 02) Verwendung während dem gesamten Projekt
Parallele Unit-Test Erstellung während Entwicklung	Fehlerminimierung und verbessertes Code Design	Während der gesamten Entwicklungsphase
Regelmässige Meetings mit dem Betreuer	Verhinderung von Entwicklung in falsche Richtung / Weiterhilfe bei Problemen	Während der gesamten Projektphase

7.1 Dokumentation

Die Dokumentation befindet sich in einem GoogleDrive (Cloud), so dass diese von überall her erreichbar ist. Die Teammitglieder besitzen alle einen persönlichen Account.

7.2 Projektmanagement

Für das Projektmanagement wird Redmine auf einem virtuellen Ubuntu 64 Bit Server verwendet.
Login für Betreuer/:

Benutzername	Passwort	URL
d1keller	<Herr Keller bekannt>	http://152.96.56.49/redmine/projects/pim

7.3 Entwicklung

Auf dem zentralen Server ist ein Git-Repository eingerichtet. Da die Versionierungsmethode Git ist, müssen keine speziellen Richtlinien für das „committen“ festgelegt werden. Code wird jedoch nur auf das zentrale Git-Repo gepusht, wenn er funktionsfähig ist. In Verbindung mit Redmine, werden in den Commits darauf hingewiesen, an welchen Tickets gearbeitet und welche davon beendet wurden.

7.3.1 Vorgehen

Zur Umsetzung unseres Projektes verwenden wir das iterative Vorgehensmodell RUP (Rational Unified Process). Dabei wird das Projekt in vier Phasen (Inception, Elaboration, Construction, Transition) aufgeteilt, wobei die einzelnen Phasen mehrere Iterationen haben können. Eine Iteration wird jeweils mit einem Meilenstein abgeschlossen.
Genaueres zu den einzelnen Phasen und Iterationen kann im Kapitel Phasen / Iterationen nachgelesen werden.

7.3.2 Code Reviews

Wir werden während des Projektes drei bis vier Code Reviews durchführen. Dabei werden die Teammitglieder zusammen, direkt an einem Laptop, wichtige Codeteile anschauen (es werden also nur Code-Reviews gemacht). Ein spezifisches Protokoll wird dafür nicht geführt, allfällige Notizen werden direkt im entsprechenden Ticket eingetragen. Kleine Änderungen werden direkt während dem Review durchgeführt. Fallen bei dem Review grössere Änderungen an, werden aus diesem direkt Tickets erstellt und einem der Teammitglieder zu gewiesen.
Die Reviews sollen jeweils ziemlich Ende einer Construction-Iteration stattfinden und werden jeweils ca. 1.5 – 2h dauern.

7.4 Coding Standards & Tools

- Um einen hohen Standard für unseren Code zu gewährleisten, verwenden wir das Tool "ReSharper" (http://www.jetbrains.com/resharper/). Das Tool ist weit verbreitet und wird auch in der MsTech Vorlesung an der HSR empfohlen - von dieser kann man auch eine Lizenz beziehen (somit fallen auch keine zusätzlichen Kosten an).
  Das Tool bietet unter anderem folgende für uns nützliche / relevanten Funktionen:
  - Code Analyse ("Style-Checking etc.")
  - Refactoring Tools (diese sind im normalen VisualStudio 2010 so gut wie nicht vorhanden)
  - Code Cleanup (Automatische Formatierung des Codes, konsistente Verwendung von Modifiern, etc.)
- Für Metriken wird das Tool SourceMonitor (http://www.campwoodsw.com/sourcemonitor.html) verwendet. Es ist eines der wenigen, etablierten, Free-Tools welche für .net existieren (zumindest nach unserer Kenntnis).
  Das Tool kann die meisten gängigen Metriken wie z.B. LOC, Komplexität, Tiefe etc. auslesen und ausgeben. Während der Entwicklungszeit werden die Metriken in regelmässigen Abständen (Wöchentlich) angeschaut und gegebenenfalls Änderungen im Code vorgenommen. Am Ende des Projektes werden die "Abschluss-Metriken" dokumentiert.
- Für die Code-Coverage verwenden wir die interne Test-Coverage, die VisualStudio mit der Version 2012 mit sich brachte.

7.5 Testen

7.5.1 Code Coverage

Da es sich bei der Hauptfunktionalität dieser Applikation um die Bedienung einer Web-Oberfläche handelt, wird eine eher tiefere Code Coverage von ca. 80% angepeilt. Soviel wie möglich / sinnvoll soll mit Unit Tests abgedeckt werden. Für die Coverage wird nur Data, sowie der Business-Layer beachtet, der Presentation-Layer wird aussen vor gelassen, da dieser hauptsächlich GUI-Code (HTML, etc.) enthält. Der restliche Teil der Software soll durch Usability-, sowie Systemtests abgedeckt werden. Bei den System Tests wird jeweils ein Testprotokoll geführt, auf welchem aufgelistet ist was, wie, unter welcher Erwartung und mit welchem Ergebnis getestet wurde.

7.5.2 Usability-Tests

Der Schwerpunkt der in diesem Projekt entwickelten Software liegt bei der direkten Bedienung der Oberfläche durch einen Benutzer. Um eine gute Benutzbarkeit zu gewährleisten werden Usability-Tests direkt mit dem Kunden durchgeführt.

Risikomanagement

Projekt: SA eShop/PIM Prototyping
Erstellt am: 25.09.2012
Autor: Patrick Zenhäusern / Simon Stähli
Gewichteter Schaden: 5.1

Nr.	Titel	Beschreibung	max. Schaden [h]	Eintrittswahrscheinlichkeit	Gewichteter Schaden	Vorbeugung	Verhalten beim Eintreten
R1	Lokaler Datenverlust	Verlust der Daten einer ganzen Woche, eines einzelnen Teammitgliedes.	16	10%	1.6	Häufige Pushes auf den Server. Wenig lokale Daten.	Verlorene Arbeit wiederholen.
R2	Server-Datenverlust	Ausfall des virtuellen Servers mit Datenverlust auf dem Server.	8	5%	0.4	Daten sind lokal sowie auf dem Server vorhanden. Sicherung wird jeden Tag durch geführt.	Wiederherstellen des Servers mit lokalen Daten und Daten von Backup.
R3	Patrick Zenhäusern findet sich nicht mit asp.NET zurecht (da noch nie benutzt)	Einarbeitung in asp.net ist viel schwerer anzuwenden als gedacht, die Lernfortschritte sind zu langsam	30	5%	1.5	Simon Stähli schult P. Zenhäusern gut und schnell. P. Zenhäusern liest sich im Internet ein.	Teilweise Kompensation durch andere Teammitglieder.
R4	JavaScript bietet keine Cross-Browser Garantie	JavaScript Lösungen bzw. Libraries haben zum Teil verschiedenes oder Fehl-Verhalten in unterschiedlichen Browsern (IE, Opera, Firefox,..)	16	10%	1.6	Gründliche Analyse von JavaScript Libraries im Voraus	im ersten Schritt müssen Workarounds für das auftretende Problem online gesucht werden. Ansonsten allenfalls eine andere Library evaluieren
me			**70**		**5.1**		

HSR
HOCHSCHULE FÜR TECHNIK
RAPPERSWIL
FHO Fachhochschule Ostschweiz

Studienarbeit: SA eShop/PIM Prototyping

Anforderungsspezifikation

Simon Stähli (sstaehli@hsr.ch)
Patrick Zenhäusern (pzenhaeu@hsr.ch)

Inhalt

Inhalt 2
1. Einführung 4
1.1 Zweck 4
1.2 Gültigkeitsbereich 4
1.3 Referenzen 4
2. Allgemeine Beschreibung 5
2.1 Produktperspektive 5
2.2 Produkt Funktion 5
2.3 Benutzer Charakteristik 6
2.4 Einschränkungen 6
2.5 Annahmen 6
2.6 Abhängigkeiten 6
3. Wireframes 7
4. UseCase Diagramm 8
4.1 Faktoren 9
4.2 Stakeholder 9
4.3 Beschreibungen (Brief) 9
4.3.1 UC01: Benutzer verwalten(CRUD) 9
4.3.2 UC02: Navigation erstellen 9
4.3.3 UC03: Excel Import 9
4.3.4 UC04: Navigation verwalten 9
4.3.5 UC05: Filter Kriterien mit Navigationspunkt verknüpfen 9
UC06: Filterkriterien verwalten 9
4.3.6 UC07: Navigation testen 10
4.4 Beschreibungen (Fully Dressed) 10
4.4.1 UC01: Benutzer verwalten(CRUD) 10
4.4.2 UC02: Navigation erstellen 11
4.4.3 UC03: Excel Import 11
4.4.4 UC04: Navigation verwalten 12
4.4.5 UC05: Filter Kriterien mit Navigationspunkt verknüpfen 13
4.4.6 UC06: Filterkriterien verwalten 13
4.4.7 UC07: Navigation testen 14
5. Nichtfunktionale Anforderungen (gestützt auf ISO/IEC 25000) 14
5.1 Leistung 14
5.2 Menge 14
5.3 Qualitätsmerkmale 15
5.3.1 Funktionalität (Functionality) 15

5.3.2 Zuverlässigkeit (Reliability) 15
5.3.3 Benutzbarkeit (Usability) 15
5.3.4 Effizienz (Efficiency) 16
5.3.5 Änderbarkeit (Maintainability) 16
5.3.6 Übertragbarkeit (Portability) 16
5.4 Randbedingungen 17

1. Einführung

1.1 Zweck

Beinhaltet die Anforderungsspezifikation für das Projekt eShop / PIM Prototyping im Rahmen der Studienarbeit HS 12.

1.2 Gültigkeitsbereich

Gültig während des Modules „Studienarbeit Informatik" im Zeitraum des Semesters HS 12.

1.3 Referenzen

Redmine (http://152.96.56.49/redmine/projects/pim)
Glossar (http://152.96.56.49/redmine/projects/pim/wiki/Glossar)

2. Allgemeine Beschreibung

2.1 Produktperspektive

Eine wichtige Aufgabe jedes Produktmanagers ist es, zu definieren wie die Navigationsstruktur bei Online-Shops gestaltet werden soll. Die Navigationsstruktur eines Produkt Management Systems ist ein zentraler Aspekt in Bezug auf die Benutzbarkeit des Online-Shops. Kunden möchten ein Produkt bzw. eine Produktgruppe schnell und bequem über die hierarchische Navigation finden.
Um den Produktmanagern diese Aufgabe zu erleichtern, möchte unser Auftraggeber, foryouandyourcustomers, diesen ein Tool zur Verfügung stellen mit welchem Navigationsstrukturen ausprobiert und verändert werden können. Aktuell wird die Navigationsstruktur jeweils in einem Excel-File festgelegt. Das effektive Ausprobieren der Navigation ist hier jedoch nicht möglich, sowie auch das Bearbeiten und Verändern der Struktur stellt sich als umständlich dar.
Mit der Entwicklung eines solchen Tools, soll dem Produktmanager hiermit ein einfaches aber effektives Hilfsmittel bei der Festlegung von Navigationsstrukturen zur Seite gestellt werden. Es soll somit bereits im Voraus evaluiert werden können, ob eine Navigationsstruktur tauglich ist oder noch verbessert werden muss.

2.2 Produkt Funktion

Das Produkt stellt folgende Funktionen zur Verfügung:

- Webbasiert
- Login pro Kunde
- Daten sollen persistent in einer Datenbank gespeichert werden
- Kundenverwaltung für Administratoren
    - Accounts erstellen / bearbeiten / löschen
- Kunden können:
    - Eine oder mehrere Navigationen erstellen und bearbeiten
        - Die „Grund-Navigation“ kann durch einen Import von einem Excel erstellt werden
        - Die maximale Navigationstiefe beträgt Fünf
        - Die Navigation kann jederzeit bearbeitet werden, dabei können per Drag und Drop Navigationen neu sortiert oder verschoben werden, bestehende Texte abgeändert und neue Navigationspunkte eingefügt werden
        - Ein Kunde kann mehrere Navigationsstrukturen besitzen
    - Filter Kriterien hinzufügen und bearbeiten
        - Filter Kriterien können jeweils nur auf unterster Navigationstiefe hinzugefügt werden
        - Es können beliebig viele Kriterien hinzugefügt werden
    - Pro Navigation soll die Möglichkeit bestehen Statistische-Werte einzusehen
        - Anzahl Menü-Einträge insgesamt / pro Ebene
        - minimale / maximale / durchschnittliche Tiefe des Baumes

Optionale Features

- Es können zum Testen und Vergleichen mehrere Navigationen nebeneinander dargestellt werden (Ausnutzung von grossen Bildschirmen)
- Undo-Funktion
- Referenzen von Navigationspunkten
- Excel-Export

2.3 Benutzer Charakteristik

Zielgruppe des Produktes sind Kunden von unseren Auftraggebern (foryouandyourcustomers). Die Kunden sind zumeist Produktmanager oder Personen mit einer ähnlichen Funktion. Bei der Benutzergruppe kann somit davon ausgegangen werden, dass diese grundsätzliche Erfahrungen in Bezug auf Webapplikationen haben. Im speziellen mit der Bedienung, wie in unserem Falle, Online-Shops bzw. dessen Navigationsstruktur.

2.4 Einschränkungen

Die grösste Einschränkung des Produktes ist, dass die Navigation nur zum Ausprobieren ist und in keinem direkten Zusammenhang mit einem Online-Shop System, SAP, etc. steht. Es liegt also keine tiefer liegende Funktionalität hinter der Navigation.
Damit das Produkt benutzt werden kann muss ein Webbrowser installiert sein, JavaScript vorhanden und aktiviert sein und eine Internetverbindung bestehen. Das Produkt wird nicht auf mobilen Geräten eingesetzt, Benutzer bedienen die Applikation jeweils über einen Computer und besitzen Monitore mit einer Auflösung von 1280x1024 Pixel oder höher.

2.5 Annahmen

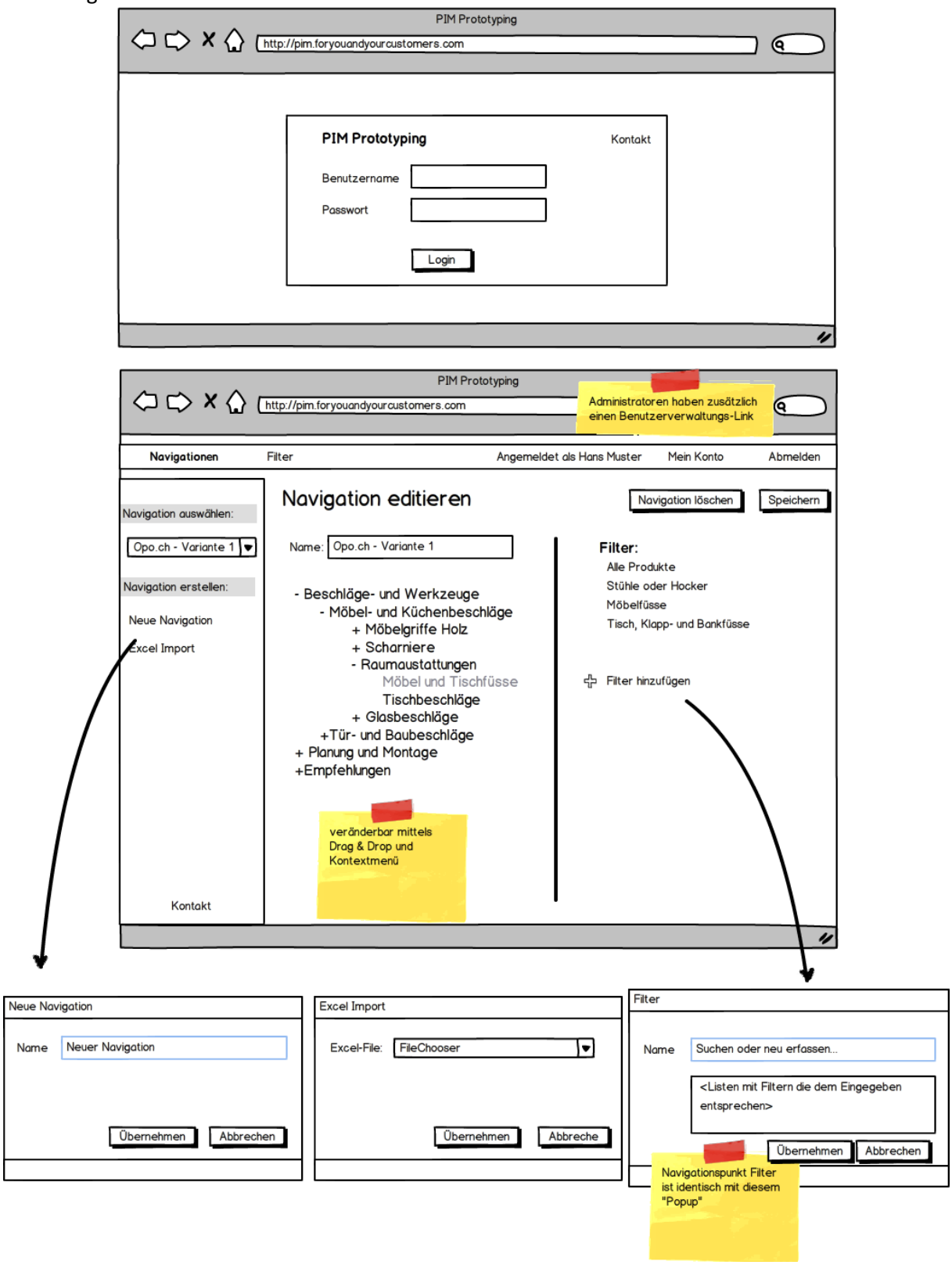
Für das Projekt wurden keine Annahmen getroffen.

2.6 Abhängigkeiten

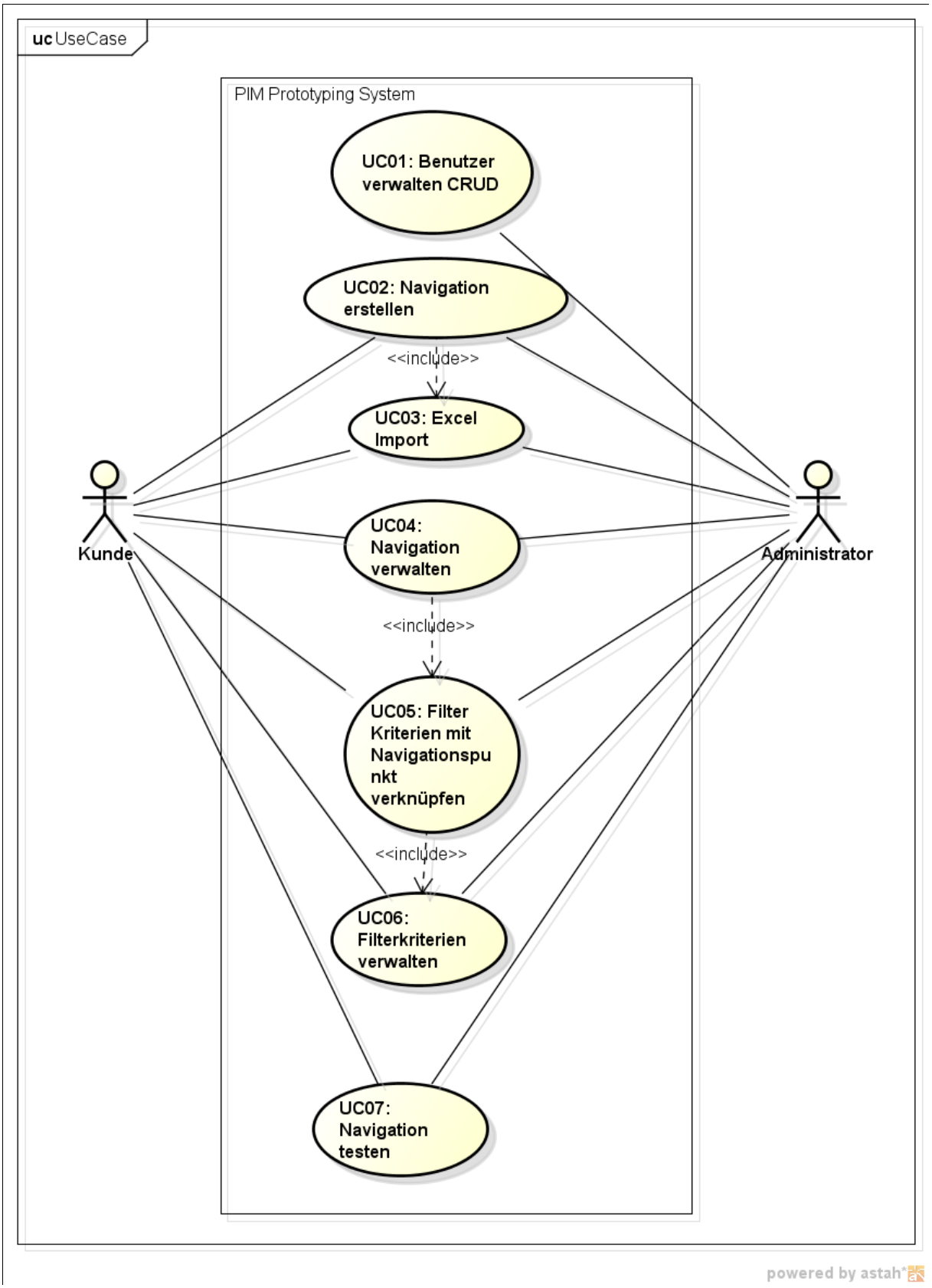
JavaScript wird benötigt. Auf dem Server auf dem das Produkt gehostet wird muss ein funktionstüchtiger IIS Server (Version 7.0+) vorhanden sein.

3. Wireframes

Um eine Vorstellung des Aussehens des Produktes zu bekommen wurden mit Balsamiq die wichtigsten Masken als Wireframes erstellt. Natürlich kann die konkrete Umsetzung noch geringe Änderungen enthalten.



4. UseCase Diagramm



4.1 Faktoren

Kunde: Will auf einfache Art und Weise Navigationen erstellen, verwalten und testen können.
Administrator: Will Kunden erstellen und verwalten können. Er will natürlich jederzeit die Möglichkeit haben die Navigationen von allen Kunden zu verwalten. Damit er Ihnen bei allfälligen Problemen helfen kann.

4.2 Stakeholder

Auftraggeber (foryouandyourcustomers): Will seinen „Kunden" die Augen öffnen. Durch das Produkt sollen diese selber erkennen, dass von Ihnen erstellte Navigationen (bzw. Navigationstiefen) nicht benutzerfreundlich sind. Das Produkt dient unserem Auftraggeber also dazu, dass er weniger Überzeugungsarbeit beim Kunden leisten muss und von Anfang an ein vom Kunden akzeptiertes System entwickeln kann.

4.3 Beschreibungen (Brief)

4.3.1 UC01: Benutzer verwalten(CRUD)

Alle Kunden die das System benutze möchten, besitzen ein eigenes Konto im System. Ein Konto kann von einem Administrator angelegt, verändert oder auch gelöscht werden.

4.3.2 UC02: Navigation erstellen

Ein Kunde hat die Möglichkeit im System eine neue Navigation zu erstellen. Navigationen können mit einem Namen im System persistent gespeichert werden. Zu einer Navigation gehört jeweils auch der gesamte Navigationsstruktur-Baum welcher auch in diesem Schritt erfasst werden kann.

4.3.3 UC03: Excel Import

Um den Kunden die Möglichkeit zu bieten bestehende Navigationsstrukturen, welche zuvor in einem Excel-File erfasst wurden, bequem im System einzulesen, soll ein Excel-Import zur Seite stehen. Hier kann ein Excel-File eingelesen und somit die gesamte Navigation unter einem Namen im System gespeichert werden. Das einzulesende Excel-File muss natürlich nach bestimmten Vorgaben ausgebaut sein.

4.3.4 UC04: Navigation verwalten

Bereits zuvor im System erfasst Navigationen eines Kunden sollen von diesem auch in einem späteren Schritt wieder eingesehen werden können. Der Kunde kann vorhandene Navigationen jederzeit bearbeiten. Er hat die Möglichkeit einzelne Menüpunkte direkt inline um zu benennen, sowie die komplette Struktur bzw. Hierarchie der Navigation per Drag & Drop zu ändern. Der Kunde kann auch einzelne Menüpunkte oder die komplette Navigation löschen. Nach dem Bearbeiten wird die Navigation je nach Wunsch des Kunden persistent abgespeichert oder auch nicht.

4.3.5 UC05: Filter Kriterien mit Navigationspunkt verknüpfen

Im System sollen einzelnen Navigationspunkten einer Navigation jeweils auch noch zusätzlich Filter-Kriterien hinterlegt werden können. Damit kann der Kunde wie bei einem echten Shop auf der „untersten Ebene" schlussendlich noch Filterkriterien angezeigt bekommen.

UC06: Filterkriterien verwalten

Filterkriterien sollen zentral an einer Stelle verwaltet werden können. Der Kunde sollte hier die Möglichkeit haben neue Filter-Kriterien zu erstellen, bestehende verändern oder aber auch löschen können.

4.3.6 UC07: Navigation testen

Der Kunde kann seine erstellten Navigationen auch testen, d.h. einzelne Punkte anklicken bzw. aufklappen etc. und diese so auf ihre Tauglichkeit prüfen. Konkret ist dieser UseCase nur eine vereinfachte Form des UC04. Denn beide UseCases finden am gleichen Ort statt und von diesem UseCase kann auch direkt in UC04 gewechselt werden.

4.4 Beschreibungen (Fully Dressed)

4.4.1 UC01: Benutzer verwalten(CRUD)

Primary Actor	Administrator
Stakeholders & Interests	Administrator: Will Benutzer möglichst einfach und schnell verwalten können.
Preconditions	-
Postconditions	Benutzer ist gemäss den Wünschen / Änderungen des Administrators abgespeichert.
Main Success Scenario	1. Administrator loggt sich ein 2. Administrator begibt sich in die Benutzerverwaltung 3. Administrator legt einen neuen Benutzer an 4. Administrator ordnet Benutzer einer Gruppe zu 5. Administrator füllt die restlichen Benötigen Informationen für Benutzer aus 6. Administrator speichert den neuen Benutzer Wiederhole Schritt 2 – 5 bis Administrator alle gewünschten Benutzer erstellt hat.
Extensions	3a: 1. Administrator will einen Benutzer löschen 2. Administrator wählt aus einer Liste von Benutzern den entsprechenden Account aus und kann ihn nach einer Bestätigungsmeldung löschen. 3b: Administrator will einen Benutzer verändern. Er wählt diesen dazu aus einer Liste von vorhandenen Benutzern aus und fährt dann weiter mit Schritt 4. 6a: Administrator will Benutzer nicht speichern und bricht das erfassen ab. Der Benutzer ist nicht gespeichert. *a: Internetverbindung bricht ab: 1. Benutzer ist nicht gespeichert 2. Falls Internetseite noch offen ist und Session nicht abgelaufen kann nach dem Wiederherstellen der Verbindung gespeichert werden

4.4.2 UC02: Navigation erstellen

Primary Actor	Benutzer
Stakeholders & Interests	Benutzer: Will Navigation möglichst einfach und schnell erstellen können.
Preconditions	-
Postconditions	Navigation ist gemäss den Wünschen / Änderungen des Benutzers erstellt.
Main Success Scenario	1. Benutzer loggt sich ein 2. Benutzer wählt „Neue Navigation erstellen“ 3. Benutzer gibt einen Namen für die neue Navigation ein 4. Benutzer schliesst Erstellung mittels „Speichern“ ab. Benutzer befindet sich dann bei UC04.
Extensions	4a: Benutzer will Navigation doch nicht erstellen und bricht ab. *a: Internetverbindung bricht ab: 1. Navigation ist nicht erstellt 2. Falls Internetseite noch offen ist und Session nicht abgelaufen kann nach dem Wiederherstellen der Verbindung erstellt werden (mittels Klick auf Speichern)

4.4.3 UC03: Excel Import

Primary Actor	Benutzer
Stakeholders & Interests	Benutzer: Will Navigation möglichst einfach und schnell erstellen können, dazu will er ein existierendes Excel-File als Vorlage verwenden.
Preconditions	Benutzer hat ein Excel-File gemäss vorgegebener Struktur.
Postconditions	Navigation ist gemäss den Wünschen / Änderungen des Benutzers erstellt.
Main Success Scenario	1. Benutzer loggt sich ein 2. Benutzer wählt „Neue Navigation erstellen“ 3. Benutzer wählt die Option „Excel-Import“ aus 4. Benutzer wählt sein Excel-File aus 5. Benutzer startet Import 6. Benutzer gibt einen Namen für die neue Navigation ein 7. Benutzer schliesst Erstellung mittels „Speichern“ ab. Benutzer befindet sich dann bei UC04 (im Unterschied zu UC02 existiert jetzt aber bereits eine Navigationsstruktur, nämlich diese aus dem Excel-File)
Extensions	5a: Excel-File ist einem ungültigen Format. Dem Benutzer wird eine entsprechende Fehlermeldung und ein Hinweis auf das korrekte Format ausgegeben. Benutzer befindet sich wieder beim Punkt 2. 7a: Benutzer will Navigation doch nicht erstellen und bricht ab. *a: Internetverbindung bricht ab: 1. Navigation ist nicht erstellt 2. Falls Internetseite noch offen ist und Session nicht abgelaufen kann nach dem Wiederherstellen der Verbindung erstellt werden (mittels Klick auf Speichern)

4.4.4 UC04: Navigation verwalten

Primary Actor	Kunde
Stakeholders & Interests	Kunde: Will seine Navigation möglichst einfach und schnell bearbeiten können. Er will schnell Resultate sehen. Die Navigation soll für ihn übersichtlich und schön dargestellt sein.
Preconditions	UC02 oder UC03 erfüllt
Postconditions	Navigation ist gemäss den Wünschen / Änderungen des Kunden abgespeichert.
Main Success Scenario	1. Kunde loggt sich ein 2. Kunde wählt eine seiner Navigationen aus 3. Kunde verändert Namen eines Navigationspunktes. Wiederhole Schritt 3 bis Kunde Navigation gemäss seinen Vorstellungen geändert hat 4. Kunde Speichert die Navigation
Extensions	3a: Kunde verschiebt Navigationspunkte an beliebigen Ort (auch innerhalb der Hierarchie) mittels Drag & Drop und kann so Reihenfolge, so wie die Hierarchie ändern. 3b. Kunde möchte einen neuen Navigationspunkt hinzufügen. Er kann diesen an einer beliebigen Stelle in der Navigation einfügen. Kunde befindet sich bei Schritt 3. 3c: Kunde möchte den Navigationspunkt löschen. Es erscheint eine Bestätigungsmeldung ob wirklich gelöscht werden soll. 4a Kunde bricht ab nach einer Bestätigung ist die Navigation wieder auf dem gleichen Zustand wie vor der Bearbeitung *a: Internetverbindung bricht ab: 1. Navigation ist nicht gespeichert 2. Falls Internetseite noch offen ist und Session nicht abgelaufen kann nach dem Wiederherstellen der Verbindung gespeichert werden

4.4.5 UC05: Filter Kriterien mit Navigationspunkt verknüpfen

Primary Actor	Kunde
Stakeholders & Interests	Kunde: Will auf der untersten Ebene der Navigation Filterkriterien zuordnen können.
Preconditions	UC02 oder UC03 erfüllt und es existiert mindestens ein Navigationspunkt.
Postconditions	Filterkriterien sind gemäss den Vorstellungen des Kunden den Navigationspunkten zugeordnet.
Main Success Scenario	1. Kunde loggt sich ein 2. Kunde wählt eine seiner Navigationen aus 3. Kunde wählt einen Navigationspunkt auf tiefster Ebene aus 4. Kunde wählt mittels einer Suche oder Liste ein Filterkriterium aus, dass er diesem Punkt zuordnen will. Kunde widerholt Schritt 4 bis er dem Navigationspunkt alle gewünschten Kriterien hinzugefügt hat. Kunde wiederholt Schritte 3 – 4 bis er allen seinen Navigationspunkten die gewünschten Kriterien hinzugefügt hat. 5. Kunde speichert die Navigation ab
Extensions	4a: Das gewünschte Filterkriterium existiert noch nicht. Kunde erfasst dieses und fügt es danach dem Navigationspunkt hinzu. Dann gleicher Ablauf wie sonst. 5a Kunde bricht ab nach einer Bestätigung ist die Navigation wieder auf dem gleichen Zustand wie vor der Bearbeitung *a: Internetverbindung bricht ab: 1. Navigation ist nicht gespeichert 2. Falls Internetseite noch offen ist und Session nicht abgelaufen kann nach dem Wiederherstellen der Verbindung gespeichert werden

4.4.6 UC06: Filterkriterien verwalten

Primary Actor	Kunde
Stakeholders & Interests	Kunde: Will an einem Ort zentral alle Filterkriterien schnell und einfach bearbeiten können.
Preconditions	-
Postconditions	Filterkriterien sind gemäss den Vorstellungen des Kunden erfasst.
Main Success Scenario	1. Kunde loggt sich ein 2. Kunde begibt sich ins „Filterkriterien verwalten“ Menü 3. Kunde fügt ein neues Filterkriterium hinzu Kunde wiederholt schritt 3 bis alle gewünschten Kriterien erfasst sind. 4. Kunde speichert ab, die Filterkriterien sind persistent gespeichert
Extensions	3a: Kunde wählt ein bestehendes Filterkriterium aus und ändert dessen Bezeichnung. 3b: Kunde wählt ein bestehendes Filterkriterium aus und löscht dieses nach einer entsprechenden Bestätigung. *a: Internetverbindung bricht ab: 1. Filterkriterien sind nicht gespeichert 2. Falls Internetseite noch offen ist und Session nicht abgelaufen kann nach dem Wiederherstellen der Verbindung gespeichert werden

4.4.7 UC07: Navigation testen

Primary Actor	Kunde
Stakeholders & Interests	Kunde: Will seine Navigationen testen um zu prüfen ob diese für ein produktives System „tauglich“ sind.
Preconditions	UC02 oder UC03 erfüllt und mindestens ein Navigationspunkt erfasst.
Postconditions	-
Main Success Scenario	1. Kunde loggt sich ein 2. Kunde wählt eine seiner Navigationen aus 3. Kunde klappt Navigationsstufen auf und zu und macht sich Gedanken ob die Navigation so in Ordnung ist. *Kunde kann von diesem UC jederzeit zu UC04 springen, da die komplette Verarbeitung „inline“ stattfindet.*
Extensions	-

5. Nichtfunktionale Anforderungen (gestützt auf ISO/IEC 25000)

ISO 25000 ist seit 2005 der Nachfolger von ISO 9126.
Offizielle Seite:
http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=35683
Wikipedia:
http://de.wikipedia.org/wiki/ISO/IEC_25000

5.1 Leistung

- Da das Produkt webbasiert sein wird, ist es nicht einfach eine Vorhersage auf die Leistung zu machen, da die Antwortzeiten sehr abhängig von der Geschwindigkeit bzw. Qualität der Bandbreite des Benutzers ist. Wenn wir davon ausgehen, dass der Benutzer eine funktionierende Verbindung hat, wird die Reaktionszeit unserer Webseite nicht langsamer als andere Webseiten sein, die der Kunde besucht.
- Weil keine allzu grosse Rechenleistungen anfallen und vieles auf der Clientseite abläuft (JavaScript) gehen wir davon aus, dass die Applikation von der Geschwindigkeit nicht merkbar langsamer als eine lokal installierte Vergleichbare Applikation sein wird

5.2 Menge

- Wir können davon ausgehen, dass das Produkt nicht auf grosse Mengen gleichzeitiger Benutzer optimiert werden muss. Die Sicherstellung von Multi-/Paralleler- Benutzung wird aber dennoch sichergestellt werden (mittels Sessions / Cookies etc.)
- Laut unserem Auftraggeber sollte von folgenden Richtwerten für erfasste Daten ausgegangen werden:
    - Ca. 10 User
        - Pro User maximal 10 Navigationen
            - pro Navigation maximal 400 Menü-Einträge
            - maximal 1000 Filter-Kriterien Attribute insgesamt

5.3 Qualitätsmerkmale

5.3.1 Funktionalität (Functionality)

5.3.1.1 Interoperabilität

- Da die Software webbasiert sein wird. Werden alle gängigen Betriebssysteme unterstützt. Wichtig dabei ist nur, dass ein aktueller Browser sowie JavaScript vorhanden ist.
- Mobilegeräte werden NICHT unterstützt, da darauf Drag & Drop im Browser nicht oder nur eingeschränkt funktioniert.

5.3.1.2 Sicherheit

- Durch ein Login wird sichergestellt, dass die jeweiligen Navigationen nur von berechtigten Personen angesehen werden kann
- Um die Privatsphäre der Benutzer zu gewährleisten werden die Passwörter als md5-Hash verschlüsselt abgespeichert

5.3.1.3 Angemessenheit

- Durch Drag & Drop wird sichergestellt, dass die Umstrukturierung / Sortierung der Navigation einfach und der Aufgabe angemessen ist

5.3.2 Zuverlässigkeit (Reliability)

5.3.2.1 Fehlertoleranz

- Sollte ein Fehler entstehen, wird dieser dem Benutzer nicht einfach weiter gereicht, sondern intern behandelt. Falls der Fehler behandelt wurde aber der reibungslose Ablauf nicht sichergestellt werden kann wird die wird der Grund des Fehlers plus eine Erklärung für nicht Informatik-Fachkundige ausgegeben.

5.3.2.2 Reife

- Durch saubere Programmierung und gutes Testing soll erreicht werden, dass das Produkt durch den Benutzer selten versagt. Wenn immer möglich sollen hilfreiche Fehlermeldungen ausgegeben werden.

5.3.3 Benutzbarkeit (Usability)

5.3.3.1 Verständlichkeit

- Das Produkt sollte ziemlich selbsterklärend sein Um dies zu erreichen werden treffende Namen, Beschreibungen und Icons gewählt.
- Benutzer sollen sich jederzeit im Klaren sein was sie gerade machen und was Aktionen für Konsequenzen mit sich bringen.
- Bei Punkten welche doch etwas komplexer sind wird jeweils eine Hilfe angeboten.

5.3.3.2 Erlernbarkeit

- Dadurch dass die im Produkt verwendeten Technologien stark an Standards orientiert sind, sollte es für Nutzer ein leichtes sein mit dem Produkt zu arbeiten. Ein Benutzer sollte innerhalb von ca. 5 Minuten das Produkt bedienen können und sich darin zu Recht finden.

5.3.3.3 Bedienbarkeit

- Das Produkt soll mittels einer normalen Maus und einer Tastatur bedienbar sein.
  Es wird jedoch nicht möglich sein, dass Produkt nur per Maus oder nur per Tastatur zu bedienen.

5.3.4 Effizienz (Efficiency)

5.3.4.1 Zeitverhalten

- Der Benutzer soll kaum merken, dass er im Web arbeitet die Reaktionszeiten sollten so angemessen sein, dass eine flüssige Bedienung für den Benutzer besteht
- Wartezeiten treten jeweils nur beim Speichern und dem Login auf. Das Verschieben und ändern von Navigationspunkten wird Clientseitig mittels JavaScript abgehandelt. Request werden dabei oft im Hintergrund (für den Benutzer nicht sichtbar) durchgeführt.

5.3.4.2 Verbrauchsverhalten

- Das Produkt wird sehr wenig Ressourcen (CPU / Grafik / Speicher) verbrauchen, da die verwendeten Funktionen keine komplexen Berechnungen oder grosse Datenmengen enthalten

5.3.5 Änderbarkeit (Maintainability)

5.3.5.1 Analysierbarkeit

- Da eine saubere Trennung von Businesslogik, Presentation, Application, etc. (mittels Schichtenmodell) angepeilt wird, wird das Finden von Fehlern stark erleichtert

5.3.5.2 Modifizierbarkeit

- Durch das Schichten Modell und der Einhaltung des „Open-Closed Principles“ wird es mit wenig Aufwand verbunden sein Erweiterungen und Änderungen an der Software zu machen

5.3.5.3 Stabilität

- Der grösste Störfaktor ist der Ausfall der Internetverbindung, sollte dies passieren werden entweder die Daten vom Server neu geladen oder aber (falls der Benutzer die Seite noch nicht selbst neu geladen hat) die Daten noch einmal versucht zu speichern

5.3.5.4 Testbarkeit

- Da das Produkt eine überschaubare Anzahl Funktionen und somit auch von möglichen UseCases etc. hat wird ein sauberes Testing möglich sein (zumindest für die CRUD Operationen)

5.3.6 Übertragbarkeit (Portability)

5.3.6.1 Anpassbarkeit

- Dadurch, dass das Produkt web fähig ist sollte es auf allen Betriebssystemen die einen aktuellen Webbrowser und JavaScript haben funktionieren

5.3.6.2 Austauschbarkeit

- Der Server kann natürlich nicht einfach von einem IIS auf durch einen Apache ausgetauscht werden, da das Ganze mit asp.NET (Microsoft Technologie) programmiert wird.
- Das Produkt kann aber relativ leicht auf einen neuen Server übertragen werden, dies ist durch relative Pfade in den Programmen und durch Config-Files sicher gestellt

5.3.6.3 Installierbarkeit

- Da das Produkt im Browser läuft muss nichts installiert werden

5.4 Randbedingungen

- Benutze Programmiersprache: C# (asp.NET), JavaScript, HTML
- Benötigter Server: IIS (Version 7.0+)
- Als Übertragungsprotokoll wird HTTP verwendet

HSR
HOCHSCHULE FÜR TECHNIK
RAPPERSWIL
FHO Fachhochschule Ostschweiz

Studienarbeit: SA eShop/PIM Prototyping

Domainanalyse

Simon Stähli (sstaehli@hsr.ch)
Patrick Zenhäusern (pzenhaeu@hsr.ch)

Inhalt

Inhalt 2
1. Einführung 3
1.1 Zweck 3
1.2 Gültigkeitsbereich 3
1.3 Referenzen 3
2. Domain Modell 4

1. Einführung

1.1 Zweck

Beinhaltet die Domain Analyse für das Projekt eShop / PIM Prototyping im Rahmen der Studienarbeit HS 12.

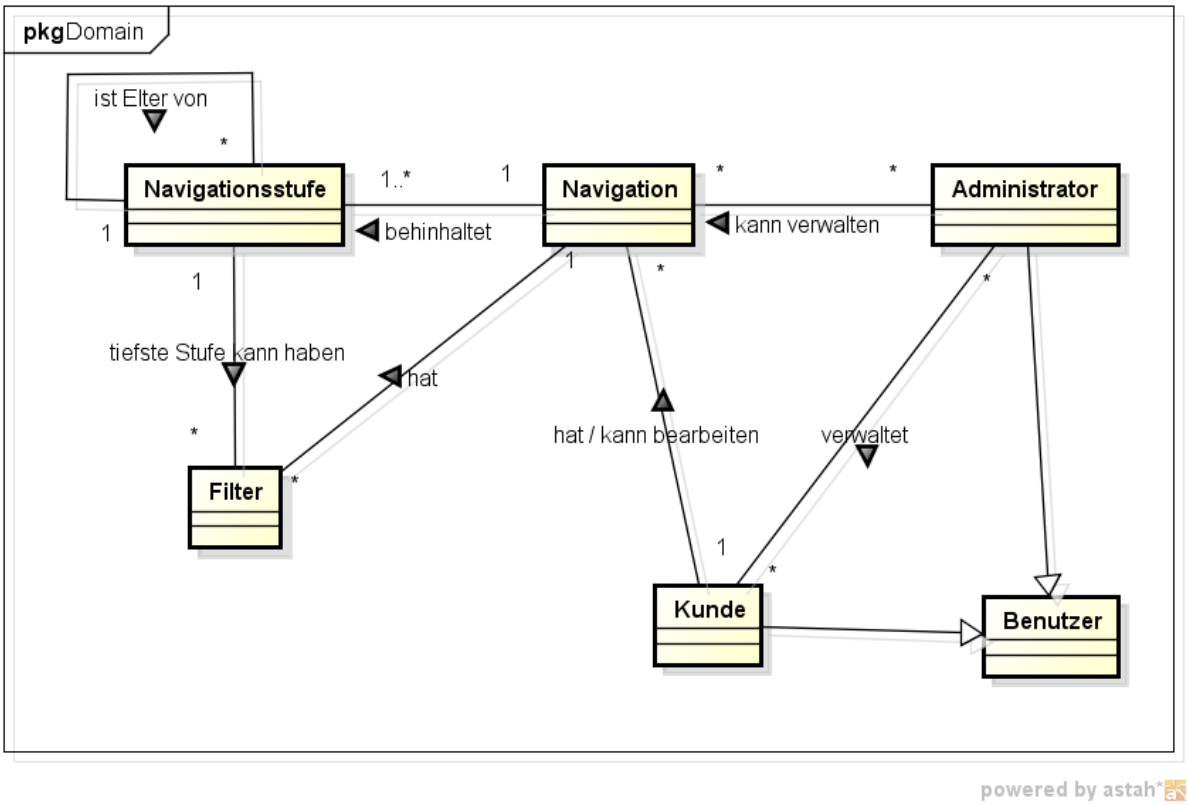
1.2 Gültigkeitsbereich

Gültig während des Modules „Studienarbeit Informatik“ im Zeitraum des Semesters HS 12.

1.3 Referenzen

Redmine (http://152.96.56.49/redmine/projects/pim)
Glossar (http://152.96.56.49/redmine/projects/pim/wiki/Glossar)

2. Domain Modell



Beim Domainmodell wurde versucht, dies so einfach zu machen, dass keine Erklärung nötig ist bzw. diese in einem kurzen Abschnitt beschrieben werden kann:
Das System wird von Kunden benutzt, welche Navigationen erstellen können. Die Navigationen erhalten jeweils verschiedene Navigationsstufen, mindestens jedoch eine erste Navigationsstufe. Des Weiteren können der letzten Navigationsstufe jeweils Filter angehängt werden.
Zusätzlich existieren Administratoren welche Kunden, sowie alle Navigationen verwalten können.

HSR
HOCHSCHULE FÜR TECHNIK
RAPPERSWIL
FHO Fachhochschule Ostschweiz

Studienarbeit: SA eShop/PIM Prototyping

Software Architektur

Simon Stähli (sstaehli@hsr.ch)
Patrick Zenhäusern (pzenhaeu@hsr.ch)

Inhalt

Inhalt 2
1. Einführung 3
1.1 Zweck 3
1.2 Gültigkeitsbereich 3
1.3 Referenzen 3
2. Systemübersicht 4
3. Architektonische Ziele & Einschränkungen 5
3.1 Ziele 5
3.2 Entwicklungstools 5
4. Logische Architektur 6
4.1 Presentation –Layer 6
4.2 Business-Logic-Layer 6
4.3 Der Data-Access-Layer 6
4.3.1 Klassenstruktur auf dem „Data“ Layer 7
4.4 Funktionalität/Klassen: Navigation Verwalten 9
4.4.1 Allgemeines 9
4.4.2 Klassendiagramm 9
4.4.3 Funktionsprinzip 10
4.4.4 Code Teil: JavaScript / jsTree 10
4.4.5 Klasse: ViewNavigation 10
4.4.6 Klasse: NavigationManagerController 10
4.4.7 Klassen: Holder 10
4.4.8 Klasse: Wrapper 11
4.4.9 Klasse: UndoStack 11
4.4.10 Klasse: ReturnObject 11
4.4.11 Funktionsprinzip: Referenzen 11
5. Datenspeicherung 12
6. Fehlerbehandlung und Logging 12
7. Security 13
8. Grössen und Leistung 13

1. Einführung

1.1 Zweck

Beinhaltet die Software Architektur für das Projekt eShop / PIM Prototyping im Rahmen der Studienarbeit HS 12.

1.2 Gültigkeitsbereich

Gültig während des Modules „Studienarbeit Informatik“ im Zeitraum des Semesters HS 12.

1.3 Referenzen

Redmine (http://152.96.56.49/redmine/projects/pim)
Glossar (http://152.96.56.49/redmine/projects/pim/wiki/Glossar)

2. Systemübersicht

Die Applikation wird als Web-Anwendung realisiert und ist somit eine Client-Server Applikation. Die Clients stellen in diesem Falle Kunden dar, welche über das Internet mittels lokalem Web-Browser auf die Applikation zugreifen. Die Web-Applikation selbst wird auf einem Mircosoft IIS-Server laufen. Um die Daten persistent zu speichern, wird eine MySQL Datenbank eingesetzt.

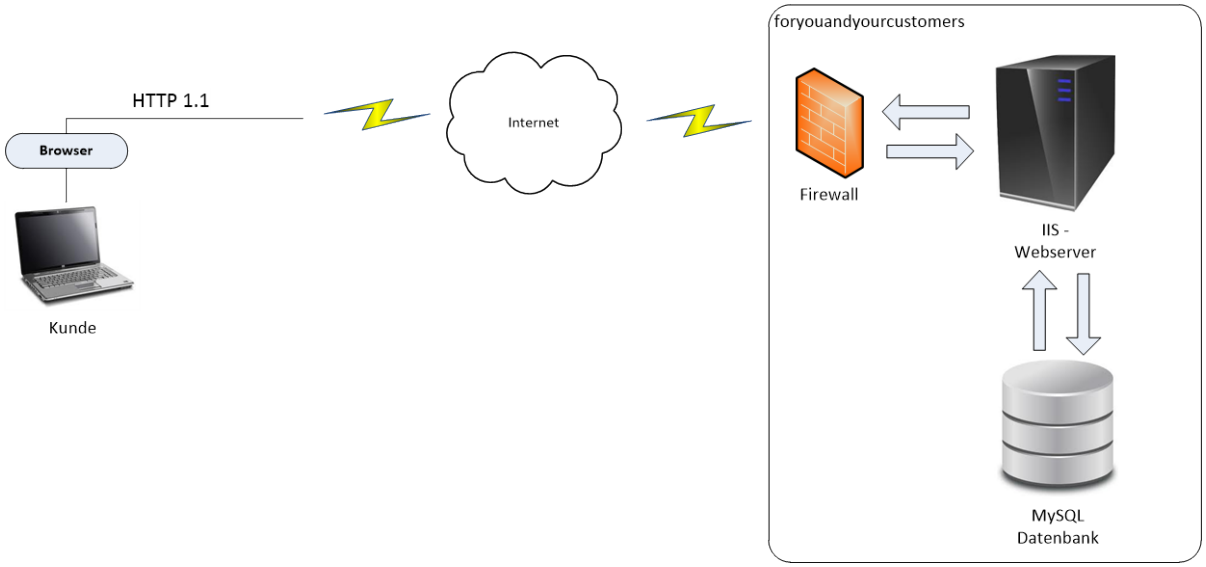


Abbildung 1 - Übersicht der Architektur

3. Architektonische Ziele & Einschränkungen

3.1 Ziele

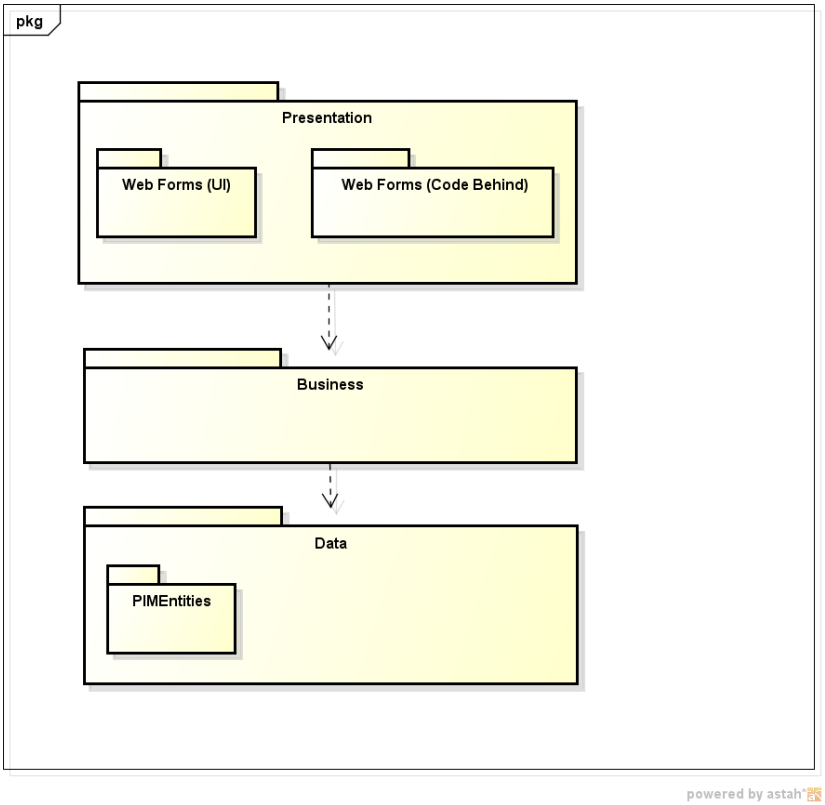
Da die Applikation Kunden ohne jeglichen Installationsaufwand zur Verfügung gestellt werden soll, wird diese als Web-Applikation entwickelt. Somit braucht der Kunde nur einen Webbrowser, welcher beim heutigen Stand der Technik auf eigentlich allen Computern bereits verfügbar ist. Die Bedienung der Applikation sollte möglichst einfach und intuitiv ausfallen.
Die Architektur soll so gewählt werden, dass immer einer klare Trennung von Backend (Business) und Frontend (Presentation) stattfindet. Durch Unit Tests soll die Korrektheit sichergestellt werden. In diesem Dokument werden die wichtigsten Klassen beschrieben, im Code selbst befinden sich keine (oder nur selten) Kommentare zu den Funktionalitäten. Da jedoch sehr auf saubere Programmierung (insbesondere ohne „Smells“) geachtet wird, sollten Kommentare auch nicht nötig sein.

3.2 Entwicklungstools

Astah	Dieses Tool bietet die Möglichkeit diverse Diagramme, unter anderem UML und ERD, zu erstellen.
Mircosoft Visual Studio 2012	Die gesamte Entwicklung der Applikation wird über diese Entwicklungs-IDE erledigt.
Redmine	Dient zur Hilfe bei der Planung und Unterhaltung des Projektes. Des Weiteren kann das Zeit-Management über dieses Tool abgewickelt werden.
GIT	Als Versionsverwaltung des Codes wird GIT eingesetzt.
Firebug	Plugin für Firefox wesentliche Unterstützung bei der Entwicklung von JavaScript Code.

4. Logische Architektur

Die Applikation wird mit einer klassischen 3-Layer-Architektur implementiert. Dabei handelt es sich um die Layers „Presentation“, „Business-Logic“ sowie „Data-Access“.



4.1 Presentation –Layer

Der Presentation-Layer beinhaltet alles, was für die Darstellung sowie Interaktion mit dem Benutzer benötigt wird. In diesem Layer wird ASP.NET eingesetzt. Durch den Einsatz von ASP.NET Web Forms wird eine Trennung von Markup und Code angestrebt.
Um dem Benutzer eine möglichst einfache und intuitive Bedienung zu gewähren, wird ein Drag & Drop fähiger Navigationsbaum mittels JavaScript implementiert. Hierbei wird die JavaScript auf die JavaScript Bibliothek jQuery und dessen Plugin jsTree zurückgegriffen. Auch die meisten Anfragen an das Backend werden mittels jQuery und Ajax im Hintergrund realisiert.

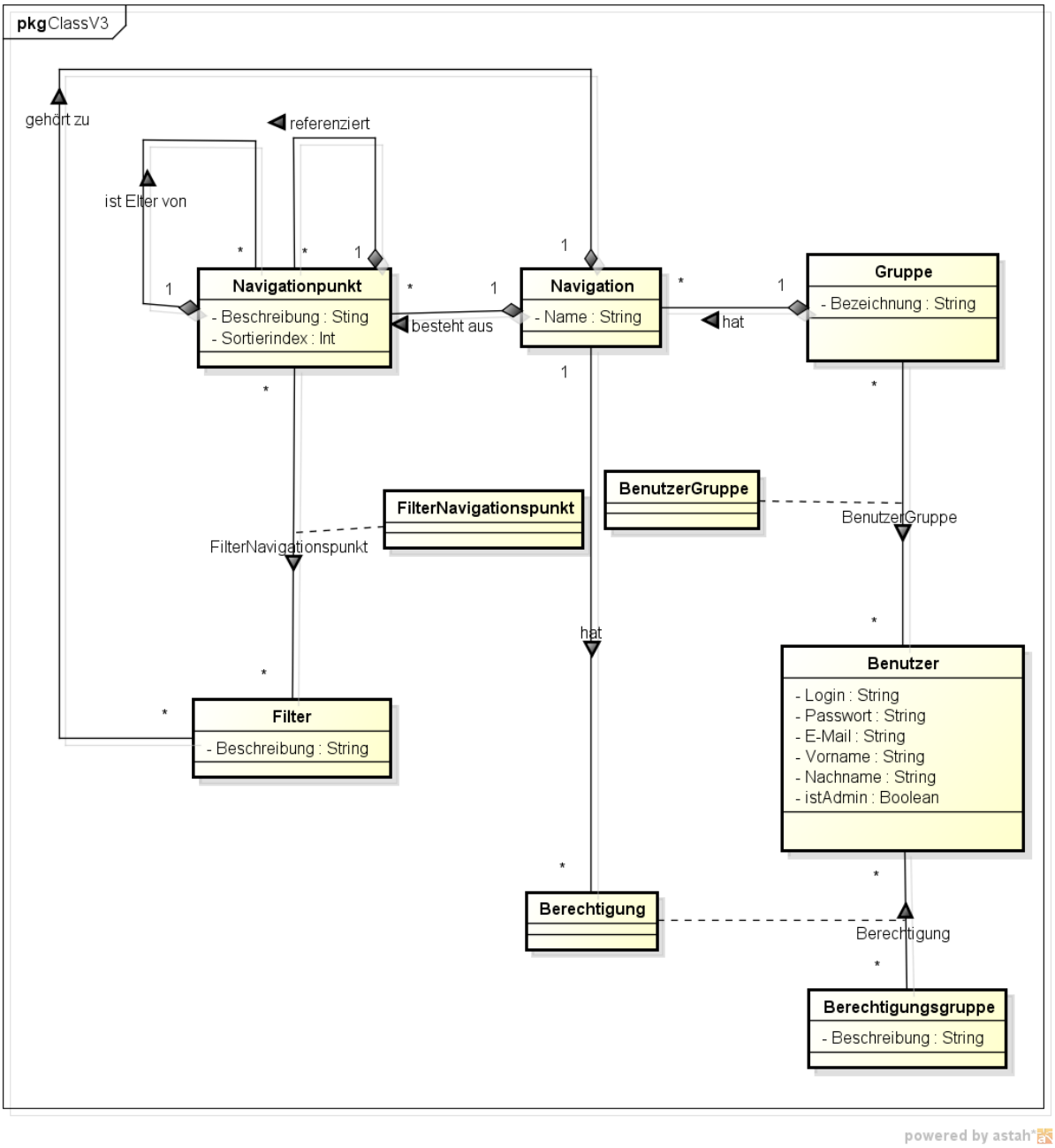
4.2 Business-Logic-Layer

Der Business-Logic-Layer stellt alle Funktionalitäten zur Verfügung welche vom Presentation-Layer benötigt werden um Daten zu laden, editieren, speichern oder auch validieren. In diesem Layer wird ausschliesslich C# als Programmiersprache verwendet. Er stellt auch die Kommunikation zwischen Presentation-Layer und Data-Access Layer dar.

4.3 Der Data-Access-Layer

Jeglicher Zugriff auf Elemente in der Datenbank wird über den Data-Access-Layer geregelt. Dieser Layer wird mithilfe des ADO.NET Entity Frameworks implementiert. Das Datenmodell wurde anhand des Datenbankmodells erstellt. Sämtliche Zugriffe auf die Datenbank werden ohne Ausnahme über das Entity Framework Datenmodell ausgeführt. Dies bringt die üblichen Vorteile die ein OR-Mapper mit sich bringt.

4.3.1 Klassenstruktur auf dem „Data“ Layer



Das Klassendiagramm oben ist ausschliesslich für alle Klassen, welche auf dem Data-Layer bestehen. Diese Klassen sind (fast) 1:1 Mappings der Datenbanktabellen. Das Mapping wird dabei vom Ado.Net Framework vorgenommen und verwaltet. Dieses stellt dann Entitäten zur Verfügung, mit welchen im Business Layer gearbeitet wird. Durch diese Klassen wird die grundsätzliche und wichtigste Funktionalität des Produktes beschrieben. Diese Klassen sollten kaum mehr verändert werden, da Sie mit der Datenbank verknüpft sind und Änderungen somit hohe Änderungsarbeit in der Datenbank mit sich bringen.

4.3.1.1 Erklärung zum Diagramm

Das Diagramm ist in den meisten Fällen selbst erklärend. Zu erwähnen gilt es, das ein spezielles Augenmerk darauf gelegt wurde, dass das Produkt leicht erweiterbar sein soll. In unserem Produkt würde theoretisch auch ein einfacheres Klassendiagramm gänzlich ohne (Berechtigungs)Gruppen genügen, da eine solche Einschränkung jede Erweiterung jedoch sehr mühsam machen würde und der Mehraufwand für die zusätzlichen Klassen gering ist werden diese trotzdem implementiert.

4.3.1.2 Benutzergruppe

Ein Benutzer kann jeweils Mitglied von mehreren Benutzergruppen sein. Dadurch ist es möglich, dass ein Benutzer z.B. bei zwei Kundenfirmen arbeiten kann, in unserem Produkt aber trotzdem nur ein Account benötigt, welcher dann entsprechend den beiden Gruppen zugewiesen wird.
Wie auch zu sehen ist, sind Navigationen jeweils direkt an die Benutzergruppen gehängt und ab diesem Punkt existieren auch nur noch Kompositionen. Konkret heisst dies: Wird eine Benutzergruppe gelöscht, werden alle dazugehörigen Navigationen, sowie deren Navigationspunkte und Filter gelöscht. Die Benutzergruppe stellt somit eine sehr wichtige Klasse da.

4.3.1.3 Berechtigungsgruppe

Ein Benutzer kann Mitglied von mehreren Berechtigungsgruppen sein. So ist es möglich, dass z.B. die Berechtigungsgruppen „Produkt Manager“, „Verkäufer“ und „Read Only“ eingeführt werden. Die Benutzer können dann jeweils pro Navigation einer oder mehreren Gruppen zugeordnet werden (so könnten auch Berechtigungen kombiniert werden).
Zusätzlich existiert direkt in der Benutzerklasse noch das Attribut „istAdmin“. Diese einzelne Berechtigung ist stärker als alle anderen. Sie dient primär dazu, dass man Administratoren nicht für jede Navigation extra alle Rechte zuteilen muss.
Das Berechtigungssystem wird in der ersten Version nicht so komplex umgesetzt, es wird jeweils nur Administratoren und Kunden geben. Wie jedoch bereits angemerkt, wird durch diese Struktur die Erweiterbarkeit nicht eingeschränkt.

4.3.1.4 Navigationspunkt

4.3.1.4.1 Navigationspunkte Kinder

Wie zu sehen ist, kann ein Navigationspunkt endlos viele Kind-Navigationspunkte haben und diese wiederum endlos viele Kindeskinder-Navigationspunkte etc. Die Kinder werden im Produkt hierarchisch dargestellt, in der Navigation kann man sich das so vorstellen, dass es einige Navigationspunkte auf Stufe 0 gibt – ohne Eltern – diese haben dann Kinder, welche auf Stufe 1 sind, diese können wiederum Kinder haben, welche dann auf der Stufe 2 sind etc. Ein Kind kann nicht ohne seinen Elternpunkt überleben, d.h. wird dieser gelöscht so werden auch die Kinder entfernt.

4.3.1.4.2 Navigationspunkte Referenzen

Erst im Verlaufe der Produktentwicklung kam der Wunsch auf, dass ein Navigationspunkt auch referenziert werden kann. Referenziert bedeutet, dass von einem bestehenden Navigationspunkt eine „Referenz-Kopie“ erstellt wird, dadurch entsteht ein neuer identischer Navigationspunkt, dieser hat eine Referenz auf den kopierten Navigationspunkt. Alle Änderungen, wie z.B. das Ändern des Namens oder das Hinzufügen bzw. Entfernen von Filterkriterien werden dann jeweils beim Originalpunkt und bei allen Referenzen durchgeführt. Änderungen können jeweils entweder beim Originalpunkt oder auch beim Referenzpunkt durchgeführt werden, die Änderungen werden in beiden Fällen auf alle Referenzen bzw. Originale weiter geleitet. Die Einzige Ausnahme hierbei stellt natürlich das Löschen dar: Löscht man einen Referenzpunkt so hat dies keinen Einfluss auf den Originalpunkt, umgekehrt jedoch werden auch alle Referenzen gelöscht.
Referenzen sind also vergleichbar mit „Verknüpfungen“ oder „Links (ln)“ (auf Linux).

4.3.1.5 Filter

Die Filterklasse hat zwei Beziehungen, zum einen hat jeder Filter immer eine Beziehung zu genau einer Navigation, wird diese Navigation gelöscht verschwinden auch alle Filter. Zum anderen kann ein Filter mehreren Navigationspunkten zugewiesen werden, diese Beziehung wird in einer Beziehungstabelle gespeichert. Es sind beide Beziehungen zwingend nötig, denn nur durch die direkte Beziehung zur Navigation können alle zur Verfügung stehenden Filter die zu dieser Navigation

gehören (egal ob sie bereits einem Navigationspunkt zugewiesen wurden oder nicht) angezeigt werden.

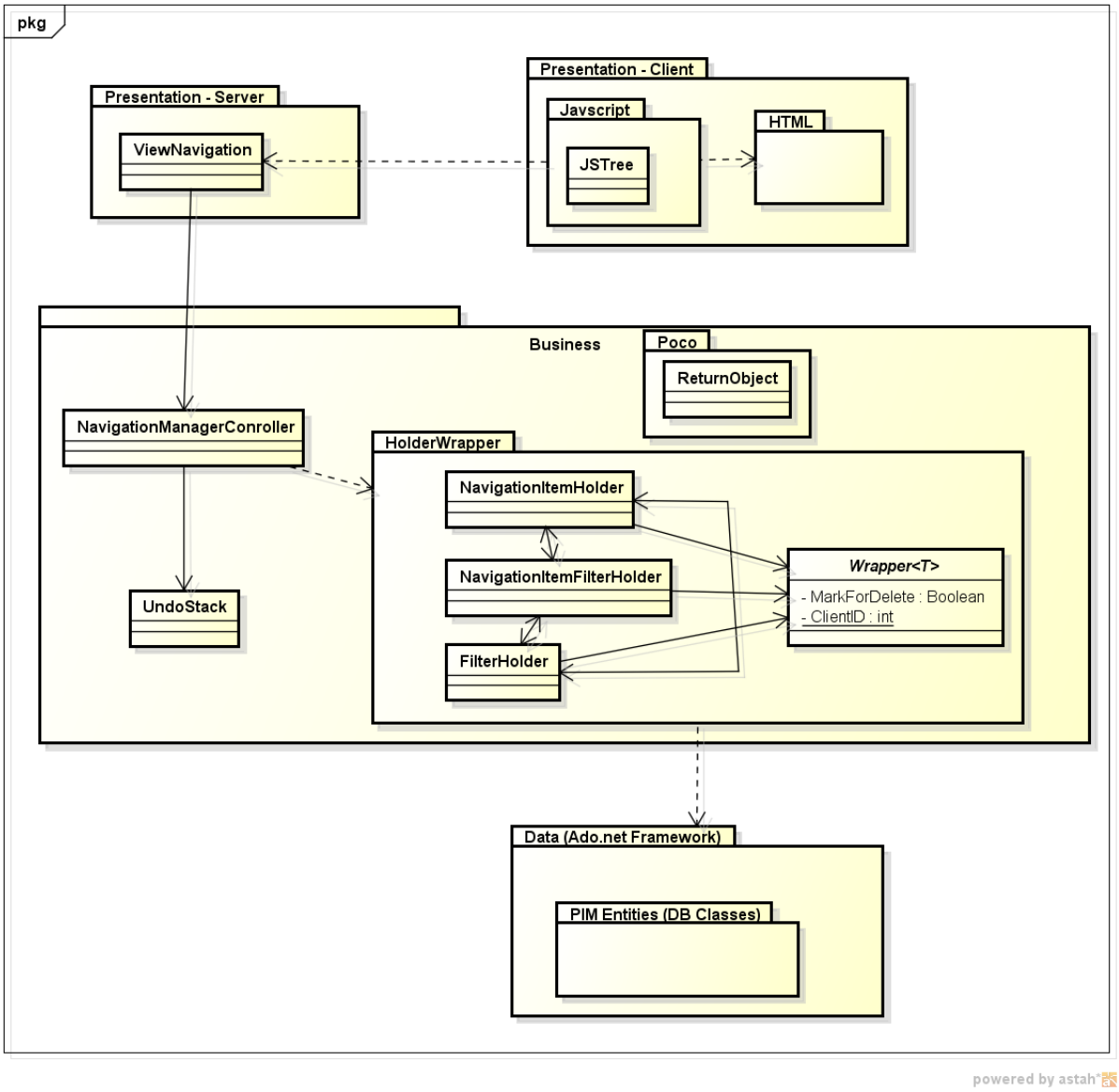
4.4 Funktionalität/Klassen: Navigation Verwalten

4.4.1 Allgemeines

Im Gegensatz zu den Klassen auf dem Daten-Layer werden in diesem Kapitel nicht alle Klassen im Detail beschrieben. Es wird lediglich auf die Wichtigsten unter Ihnen eingegangen, dabei soll vor allem aufgezeigt werden, welche Klassen was für eine Verantwortung/Aufgabe wahrnehmen und wie diese miteinander kommunizieren.

4.4.2 Klassendiagramm

Navigationen zu verwalten ist die wichtigste und die meist verwendete Aufgabe in unserem Produkt. In diesem Kapitel wird deshalb auf die dafür relevanten Klassen eingegangen.



4.4.3 Funktionsprinzip

Jede vom Benutzer vorgenommene Manipulation (z.B. das Erstellen einen Navigationspunktes oder das Verschieben eines Teilbaumes) wird zuerst Clientseitig mittels JavaScript durchgeführt, dann wird der Request an den Server durchgeführt. Je nachdem ob die Abfrage erfolgreich war oder nicht, wird die Manipulation Clientseitig wieder verworfen oder behalten. Konkret bedeutet das also, dass für eine kurze Zeitdauer Clientseitig ein anderer Zustand als Serverseitig bestehen kann, durch gute Rückgabewerte bzw. Erfolg-/ Fehlermeldungen wird jedoch in jedem Fall immer sichergestellt, dass Client und Server identisch sind.

4.4.4 Code Teil: JavaScript / jsTree

Die JavaScript / jsTree Klassen bzw. Files (da diese nicht wirklich objektorientiert sind) befinden sich beim Client im Browser. Sie sind für die Clientseitige Manipulation der Navigation zuständig. Auch werden (fast) alle Anfragen mittels jQuery bzw. Ajax gemacht, diese sind in allen Fällen immer an eine Klasse im Präsentation-Layer gerichtet. Im Falle der Navigationsbearbeitung werden diese fast immer an die Klasse ViewNavigation ausgeführt. Durch das Plugin „jsTree“ wird die ganze Baummanipulation Clientseitig stark erleichtert, so mussten alle Baumfunktionen wie z.B. „Drag & Drop“ nicht selber implementiert werden. Eigene Arbeit musste hauptsächlich in die Kommunikation mit dem Server (und die Manipulation Server-seitig) gesteckt werden. Eine grosse Ausnahme stellen hierbei die Filterkriterien, sowie die Referenzen dar, diese wurden durch das jsTree Plugin nicht abgedeckt und mussten selber implementiert werden. Dies wurde so umgesetzt, dass der jsTree selber nichts von Referenzen und Filterkriterien weiss, alle Navigationspunkte werden als gleich angesehen.

4.4.5 Klasse: ViewNavigation

Die Klasse ViewNavigation kann man sich als eine Art FrontEnd-Controller vorstellen. Sie nimmt Anfragen vom Client (JS) entgegen, prüft hauptsächliche ob eine gültige Session besteht (Benutzer ist eingeloggt, es wurde eine Navigation ausgewählt) und leitet die Anfrage dann an den Business-Layer bzw. die Klasse NavigationManagerController weiter.

4.4.6 Klasse: NavigationManagerController

Diese Klasse stellt das Gegenstück zur Klasse ViewNavigation dar, sie ist sozusagen der Back-End Controller. Sie hält Referenzen zu allen Holder-Klassen und leitet die Anfragen an die korrekte Methode von diesen weiter. Zudem wird bei jeder Anfrage eine neue „UndoList“ erzeugt, d.h. pro Anfrage gibt es immer genau eine UndoList, diese kann beliebig viele Undo-Funktionen enthalten.

4.4.7 Klassen: Holder

Die Holder-Klassen kann man als eine Art Liste von Navigationspunkten bzw. Filterkriterien sehen. Sie enthalten die konkrete Implementation der Methoden welcher zur Manipulation der Navigation gebraucht werden. Zudem enthält jeder Holder eine Liste mit Wrapper-Objekten welche er verwaltet, im Falle vom NavigationItemHolder ist das z.B. eine Liste mit allen Navigationspunkten die es in dieser Navigation gibt. Alle Holder kennen Sich gegenseitig und sind somit auch voneinander abhängig, dies ist nötig, da einige Operationen oft Elemente von allen Holdern benötigen. Die Holder sind auch dafür verantwortlich, dass bei jeder Manipulation die entsprechende „Gegen Manipulation“ (realisiert mittels „Delegate“ (=Funktionen die in eine Lambda-Variable gespeichert und zu einem späteren Zeitpunkt ausgeführt werden)) in die momentan aktive UndoList gespeichert wird. Damit ist sichergestellt, dass jede Anfrage rückgängig gemacht werden kann.

4.4.8 Klasse: Wrapper

Die Wrapperklasse ist generisch. Jeder Holder hat eine Liste von Wrappern für einen bestimmten Typ, z.B. hat der NavigationItemHolder eine Liste mit Wrappern des Typs „navigationitem“.Er stellt einen Zwischenschritt vom Business-Layer zum Data-Layer dar. Er erhält eine statische Counter Variable, mit Hilfe dieser Variable vergibt er für jedes Objekt des Data-Layers eine eindeutige ID. Diese ID ist essentiell. Sie wird gebraucht um Objekte eindeutig identifizieren zu können, Client- sowie Serverseitig. Sie kann mit einem Primärschlüssel in der der Datenbank verglichen werden, nur dass sie nicht nur pro Tabelle eindeutig ist, sondern für die ganze Applikation. Der Grund weshalb nicht die Primärschlüssel der Tabellen als IDs verwendet werden ist ganz einfach: Neu angelegte Objekte sind noch nicht in der Datenbank vorhanden (erst nach dem Speichern!) und haben somit noch gar keinen Primärschlüssel.
Weiter enthält er für jedes Objekt des Daten-Layers das Attribut „markForDelete“, dieses Attribut wird auf „true“ gesetzt, sobald ein Navigationspunkt gelöscht wird. Man mag sich fragen: Wieso wird das Objekt nicht einfach direkt gelöscht? Der Grund dafür ist die Undo-Funktion. Ist ein Objekt erst einmal ganz gelöscht ist es (fast) nicht mehr möglich es identisch wieder herzustellen, eine Variable wieder auf „false“ zu setzen stellt jedoch kein Problem dar. Alle Objekte werden erst beim Speichern endgültig gelöscht oder persistiert.

4.4.9 Klasse: UndoStack

Der UndoStack ist ziemlich simpel aufgebaut: Pro Anfrage gibt es jeweils eine Liste mit beliebig vielen „Undo Einträgen“, diese Einträge werden von den Holdern abgesetzt und als „Actions“ in einer Liste gespeichert. Der UndStack enthält somit einen Stack (bzw. eine Liste) von Listen mit Actions, wobei jede dieser Listen für genau eine Anfrage steht. Soll nun eine Aktion rückgängig gemacht werden (dieser Request kommt immer vom Client), so wird die oberste Liste vom Stack genommen und alle Actions die sich darin befinden ausgeführt. Die Actions sind dabei meist nichts weiter als die Umkehrfunktionen der zuvor ausgeführten Funktionen (z.B. Erstellen -> Löschen).
Sobald der momentane Stand gespeichert wird („Benutzer drückt „Speichern““) so wird der UndoStack komplett geleert, dies hat den Grund, weil nicht alle gespeicherte Aktionen rückgängig gemacht werden können (z.B. das Löschen eins Objektes).

4.4.10 Klasse: ReturnObject

Das ReturnObject ist ein etwas spezielles Konstrukt es wird von den Holdern mit allen nötigen Informationen gefüllt, welche der Client für eine korrekte Auswertung der Antwort braucht. Fast jede Anfrage, die ein Client macht, erhält als Antwort immer eine Liste von ReturnObject’s. Wobei ein ReturnObject jeweils für einen Navigationspunkt steht und z.B. dessen ID, sein Eltern-ID, seine Original-ID usw. enthalten kann. Diese Informationen sind vor allem bei komplexen Funktionen wie z.B. Copy & Copy Reference notwendig, nur durch die Auswertung dieser Informationen kann sich der Client später korrekt verhalten.
Tritt während der Manipulation ein Fehler auf oder wird eine nicht erlaubt Aktion durchgeführt, so wird entweder eine leere Liste (dann erscheint eine Standardfehlermeldung) oder aber eine Liste mit nur einem ReturnObject, welches das Attribut „isError=true“ enthält und eine entsprechende Fehlermeldung, zurückgegeben.

4.4.11 Funktionsprinzip: Referenzen

Auf die Referenzen soll hier speziell eingegangen werden, da diese das wohl komplizierteste Konstrukt in der Software darstellen.

4.4.11.1 Client

Wie oben schon erwähnt weiss der jsTree (Clientseite) nichts von Referenzen, für ihn ist jeder Navigationspunkt normal und wird dementsprechend auch immer gleich behandelt. Wird vom

Benutzer der Befehl abgegeben „Kopiere Referenz von Punkt X“, so wird an den Server im Prinzip ein Move Request mit dem Parameter „isReference = true“ versendet. Anhand der Antwort des Servers weiss der jsTree dann was er machen muss (d.h. dass er kein Move sondern nur ein Copy ausführen soll). Besteht der Referenzpunkt dann und wir dieser oder sein Originalpunkt verändert, so werden Clientseitig zuerst normal die Anfragen durchgeführt, erst Anhand der Antwort des Servers weiss der jsTree dann was er noch zusätzlichen machen muss (z.B. wird bei einem „Rename“ zuerst nur der eigene Punkt umbenennt und erst durch die Antwort des Servers (bzw. den darin enthaltenen IDs) weiss der jsTree dann welche Punkte er auch noch umbenennen muss). Hier stellt sich vielleicht die Frage: „Warum so kompliziert? Warum dem jsTree nicht einfach bekannt geben, dass es Referenzen gibt?“ – Der Grund ist, dass es weit über den Rahmen der Semesterarbeit gestiegen wäre das komplette jsTree Plugin (welches komplex ist) umzuschreiben.

4.4.11.2 Server

Der Frontend-, sowie der Backendkontroller wissen im Prinzip - wie der Client - auch nichts über Referenzen, sie leiten die Anfragen einfach an die korrekte Methode weiter. Erst in den Holdern werden die Referenzen relevant: Jede Manipulationsmethode der Holder prüft in einem ersten Schritt immer ob es sich bei dem übergebenen Navigationspunkt (fortan als NP bezeichnet) um eine Referenz handelt, dabei wird immer rekursiv vorgegangen: Falls der NP eine Referenz ist, wird dieser durch den von ihr referenzierten, (scheinbaren) Original-NP ersetzt, da dieser NP jedoch wiederum eine Referenz sein kann, wird dies so oft wiederholt bis es sich schlussendlich um den wirklichen Original-NP handelt.
Als Beispiel: NP 12 kann eine Referenz von NP 6 sein. NP 6 kann aber wiederum eine Referenz von NP 3 sein. Würde eine Methode für NP 12 aufgerufen, so würde dieser zuerst durch NP 6 ersetzt und dann durch NP 3. NP 3 ist in diesem Falle somit der Original-NP.
Die verlangte Manipulation wird dann zuerst auf dem Original-NP durchgeführt und danach bei allen referenzierten NP‘s (wobei natürlich auch hier wieder rekursiv vorgegangen werden muss). Alle davon betroffen NP’s bzw. dessen IDs werden in ReturnObject’s gespeichert, der Liste mit Rückgabeobjekten angehängt und dann dem Client zurückgegeben.

5. Datenspeicherung

Die Daten werden in einer MySQL Datenbank abgespeichert.
Dabei wurden folgende relevanten Eigenschaften verwendet:

- **Tabellenformat** InnoDB (damit Beziehungen inklusive kaskadiertem Löschen etc. verwendet werden können)
- **Kollation**: utf8 damit (fast) beliebige Zeichensätze verwendet werden können

Ein ERM/ERP Diagramm brauchen wir nicht, da dies 1:1 dem Klassendiagramm entspricht. Das komplette SQL um die Datenbank zu erzeugen befindet sich im gleichen Ordner wie dieses Dokument.

6. Fehlerbehandlung und Logging

Um Fehler Global in der Applikation zu loggen, wird die Apache log4net Bibliothek verwendet. Es werden grundsätzliche alle auftretenden Exceptions global in der Applikation abgefangen und in einem Logfile geloggt. Dem User wird bei Fehlern jeweils eine einfache Fehlerseite, mit kurzem Hinweis des aufgetretenen Fehlers angezeigt. Dieser beinhaltet bewusst, aus Sicherheitsgründen, keine zu spezifischen Meldungen. Falls gewisse Fehler eine spezielle Fehlerbehandlung erfordern, wurden diese entsprechend abgefangen und behandelt. Doch auch in diesem Fall wird ein Eintrag im Logfile erstellt.

Bei fehlgeschlagenen Ajax-Anfragen aufgrund eines Timeouts wird dies dem Benutzer entsprechend mitgeteilt, hierbei werden jedoch keine Log-Einträge generiert, da diese Art von Fehlern auf dem Server gar nicht ersichtlich sind.

7. Security

Um eine gewisse Sicherheit zu gewährleisten werden Passwörter in der Datenbank in einem SHA-256 Hash abgelegt. Um jedoch Datensicherheit zu gewährleisten, müsste der Server entsprechend konfiguriert werden, dass sämtliche Kommunikation über das HTTPS-Protokoll abläuft

8. Grössen und Leistung

Durch die verwendeten Technologien existieren grundsätzlich keine wirklichen Einschränkungen bzw. wenn überhaupt die normalen Einschränkungen die MySQL hat. Jedoch sind diese von keiner nennenswerten Relevanz für das Produkt und werden deshalb aussen vor gelassen.

HSR
HOCHSCHULE FÜR TECHNIK
RAPPERSWIL
FHO Fachhochschule Ostschweiz

Studienarbeit: SA eShop/PIM Prototyping

Qualitätssicherung

Simon Stähli (sstaehli@hsr.ch)
Patrick Zenhäusern (pzenhaeu@hsr.ch)

Inhalt

Inhalt 2
1. Einführung 4
2. Code Reviews 5
2.1 End Code-Review mit Betreuer 5
2.1.1 Umsetzung 5
3. Usability-Test 6
3.1 Planung 6
3.1.1 Testsubjekte 6
3.1.2 Durchführungsort 6
3.1.3 Durchführungszeitraum 6
3.1.4 Einschränkungen 6
3.1.5 Ablauf 6
3.2 Auswertung 7
3.2.1 Notizen und Anmerkungen aus dem Test 7
3.2.2 Auswertung Testperson A 7
3.2.3 Auswertung Testperson B 8
4. Unit Tests 10
4.1 Einführung 10
4.2 Testabdeckung 10
5. Systemtests 11
5.1 Testumgebung 11
5.2 Navigation verwalten 11
5.2.1 Videos 11
5.2.2 Abgedeckte UseCases 11
5.2.3 1. Test - Grundfunktionalität 1 (Erstellen, Umbenennen, Löschen) 11
5.2.4 2. Test - Grundfunktionalität 2 (Filterkriterien) 13
5.2.5 3. Test - Grundfunktionalität 3 (Nicht-Manipulative Operationen) 14
5.2.6 4. Test - Kopieren 16
5.2.7 5. Test - Verschieben (Ausschneiden) 18
5.2.8 6. Test - Referenzen 19
5.2.9 7. Test - Undo-Funktion 23
5.2.10 8. Test - Navigationen vergleichen 25
5.2.11 9. Test - Excel Export 26
5.3 Neue Navigation erstellen 28
5.3.1 Abgedeckte UseCases 28
5.3.2 1. Test - Navigation per Excel Importieren 28
5.4 Filter Verwaltung 29

5.4.1 Abgedeckte UseCases 29
5.4.2 1. Test - Filterverwaltung Anzeige, Hinzufügen, Umbenennen, Löschen 29
5.5 Benutzerverwaltung 30
5.5.1 Abgedeckte UseCases 30
5.5.2 1. Test - Benutzerverwaltung Neue Kundengruppe 30
5.5.3 2. Test - Benutzerverwaltung Neuer Benutzer 30
5.5.4 3. Test - Benutzerverwaltung Benutzer Verändern 31
5.5.5 4. Test - Benutzerverwaltung Benutzer Löschen 31
5.6 Mein Konto 32
5.6.1 Abgedeckte UseCases 32
5.6.2 1. Test - Passwort ändern 32
6. Source Code Analyse 33
6.1 Struktur 33
6.2 Kennzahlen 33
6.3 Code Metriken 34
6.3.1 Cyclomatic Complexity 34
6.3.2 Maintainability Index 34
6.3.3 Kiviat-Metric Graph 35

1. Einführung

In diesem Dokument wird auf die verschiedenen Aspekte eingegangen, welche bei diesem Projekt zum Einsatz kamen um die Qualität des End-Produktes sicher zu stellen.
Über den ganzen Entwicklungszyklus hinweg wurden Unit-Tests für neu implementierte Methoden erstellt um deren Funktionalität zu verifizieren. Jeweils am Ende einer Construction-Iteration wurde ein Code-Review durchgeführt um allfällige Fragen oder kleinere Unstimmigkeiten zu bereinigen. Um das Ziel, die Entwicklung einer benutzerfreundlichen Applikation, zu erreichen, wurden auch Usability-Tests bei einem potentiellen Kunden durchgeführt und deren Feedback in den weiteren Entwicklungsprozess miteinbezogen.
Gegen Ende des Projektes, als alle Funktionalitäten in der Applikation implementiert waren, wurden des Weiteren Systemtests durchgeführt und entsprechend dokumentiert.
Nachfolgend wird einzeln und detaillierter auf die genannten Aspekte eingegangen.

2. Code Reviews

Insgesamt wurden vier Code-Reviews, meist am Ende einer Construction-Phase, durchgeführt. Da jedoch auch während den Construction-Phasen eine rege Kommunikation zwischen den beiden Team-Mitgliedern herrschte, konnten bereits viele Fragen und Unklarheiten im Voraus geklärt werden und die Reviews fielen kurz aus. Ein kleines Protokoll wurde jeweils im verwendeten Projekt-Management-Tool Redmine erstellte und wird somit hier aussen vorgelassen. Für grössere entstandene Arbeiten wurden sogleich Tickets eröffnet.

2.1 End Code-Review mit Betreuer

Zum Ende des Projektes, nach Abschluss der vierten und somit letzten Construction-Phase, wurde noch ein Code-Review mit Herrn Keller über die ganze Applikation hinweg durchgeführt.

Durchführungs-Datum: 13.12.2012, Mitte Transition
Teilnehmer: Daniel Keller
Patrick Zenhäusern
Simon Stähli

Allgemein Anmerkungen:

- Magic Numbers, in diesem Projekt häufig der Wert -1, sollten generell als Konstanten definiert werden. Bspw. UNDEFINED_ID
- Zeilenlänge sollte nicht zu lange sein, so dass auch auf Laptop-Screens entwickelt werden kann. Ein Wert von 110 Zeichen sei hierbei optimal
- Mehrfache Returns in Methoden wenn möglich vermeiden, besser nur ein Return am Ende der Methode
- Bei Parametern Validationen nicht nach einem Blacklist-Verfahren vorgehen, sondern mit positiv formulierter Klausel prüfen ob im jeweiligen Fall die Ausführung der Methode zugelassen werden soll

Klasse Navigationsitemholder.cs

- Methode „Get“ besser als „GetItemById“ benennen
  - Bei kleineren Klassen wäre dies kein Problem, jedoch ist „Get“ in einer grossen Klasse mit verschiedenen Aufgaben nicht immer eindeutig bzw. es kann nicht immer direkt erkannt werden was die Methode macht
- Methode „Restore“ besser als „RestoreDelete“ benennen
- Trennen der Klasse Navigationsitemholder in zwei Klassen:
  - Read-Only Klasse, welche alle nicht manipulativen Methoden wie zum Beispiel „Get“ enthält
  - Manipulative Klasse, welche Methoden enthält die Änderungen an Objekten oder Struktur durchführen

2.1.1 Umsetzung

Nach Absprache mit Herr Keller werden die Anmerkungen im Rahmen der SA nicht mehr umgesetzt. Der Grund dafür ist, dass zum Zeitpunkt des Reviews bereits die Hälfte der Transition-Phase vorbei war und somit nur noch eine Woche für das Projekt blieb. In dieser kurzen Zeit sollten keine Code-Änderungen mehr gemacht werden, da dieses nicht mehr richtig getestet werden können. Bei einer Weiterführung des Projektes müssten jedoch diese Anregungen berücksichtigt werden.

3. Usability-Test

Durch den Usability-Test soll sichergestellt werden, dass das Produkt benutzergerecht entwickelt wurde und von diesen intuitiv verstanden und bedient werden kann. Der Test soll daher auch direkt bei einem jetzigen Kunden von unserem Auftraggeber (foryouandyourcustomers) durchgeführt werden. Die Testperson sollte die Funktion des „Produktmanagers", also der Zielgruppe des Produktes, haben.

3.1 Planung

3.1.1 Testsubjekte

Als Testsubjekt sollten ein oder zwei Produktmanager der Firma „OPO Oeschger" sein.
Der Grund für die Wahl ist:

- Zielgruppe des Produktes
- Bereits Kunde von unserem Auftraggeber
- Ein neuer Webshop ist für in Entwicklung, die Produktmanager sollten somit „voll in der Materie" sein

3.1.2 Durchführungsort

Das Produkt soll direkt in der Firma OPO Oeschger AG, Steinackerstrasse 68, CH-8302 Kloten durchgeführt werden.

3.1.3 Durchführungszeitraum

Der Test soll am 15.11.2012 um 14.00 Uhr durchgeführt werden. Der Zeitraum pro Test beträgt dabei ca. 30 Minuten.

3.1.4 Einschränkungen

- Es muss eine Internetverbindung aufgebaut werden können, da sich die Webapplikation auf einem Server im Internet läuft

3.1.5 Ablauf

1. Begrüssung, Vorstellung der eigenen Person, Erklärung des Projektes
2. Zeigen des Systems anhand einer Musternavigation
3. Aufgaben siehe Excel „Aufgabe Usability Test.xls"
4. Fragen:
    a. Sind die Statistiken sinnvoll und werden diese auf Anhieb verstanden?
    b. Generell, welche Bedienelemente oder auch Abläufe sind nicht klar? Sind beispielsweise die Toolbox Symbole verständlich?
    c. Welche Funktionalität fehlt?
    d. Was würden die Testpersonen daran ändern?

3.2 Auswertung

Der Test wurde mit zwei Personen durchgeführt, diese arbeiten beide als Produktmanager. Der Test wurde direkt am Arbeitsplatz der Testpersonen unter der Leitung von D. Keller (als Moderator) durchgeführt.
Um allfällige Diskriminierung zu vermeiden werden die Testpersonen nicht bei ihren Namen genannt sondern unter Testperson A und Testperson B beschrieben.

3.2.1 Notizen und Anmerkungen aus dem Test

Für beide Tests wurde jeweils eine Liste mit Anmerkungen und Notizen zum Test erstellt. Für die meisten Anmerkungen wurden in Redmine Tickets erstellt, die Ticketnummern sind jeweils in Rot im Format #<Ticket Nummer> dargestellt, wird eine Anmerkung nicht umgesetzt, so ist diese ~~durchgestrichen~~ und mit einem Kommentar in Rot versehen.

3.2.2 Auswertung Testperson A

3.2.2.1 Allgemeines Feedback

Die Testperson A nahm das Produkt als positiv und einfach wahr. Sie fand sich ohne Einführung direkt im Produkt zurecht und konnte die gegebenen Aufgaben ohne Probleme lösen. Auf die Frage, ob Sie sich vorstellen könnte, das Produkt später zu nutzen bekamen wir ein „Ja“.

3.2.2.2 Navigations-Struktur allgemein:

- Umbenennen von Navigationspunkten mit Doppelklick → #132
- Umleiten nach dem eine neue Navigation erstellt wurde → #133
- ~~Neue Navigation evtl. besser auf derselben Seite, wie verwalten~~ → wird nicht umgesetzt, weil die Trennung so wie es ist sauberer ist und einfacher erweitert werden kann
- Erstellen von Verknüpfungen / Referenzen von Navigationspunkten / Teilbäumen, diese sollen dann Kursiv dargestellt werden → #117

3.2.2.3 Statistik

- Maximale Länge der Beschreibungen → #134
- Objekte / Verknüpfungen → #117
- Anzahl Filterkriterien auf welcher Stufe → #135

3.2.2.4 Filter

- Suche für bestehende Filter → #136
- Sortierungsmöglichkeit besser sichtbar machen / Default alphabetisch sortieren → #137
- Abwechselnde Farbe geht verloren nach Sortierung → #138
- Sprache teilweise Englisch statt Deutsch (Edit / dellte) → #139

3.2.2.5 Excel

- Parsen von Komma, Semikolon etc. Abstände, bessere Fehlertoleranz einbauen → #140
- Filter Import, bei verschiedener Anzahl von Stufen (alle in derselben Spalte oder direkt in der nächsten Zelle) → #141

3.2.2.6 Fehler

- „Service unavailable“ nach Erstellen einer neuen Navigation → #142

3.2.2.7 Zusätzliche Ideen:

- ~~Filter Kriterien pro Kunde anstatt pro Navigation~~ → wird nicht umgesetzt weil Filter pro Navigation eine sauberere Trennung bieten, wenn man die Filter einen anderen Navigation will, so kann man anstatt eine neue Navigation zu erstellen, die alte einfach als Kopie speichern, würde man die Kriterien immer pro Kunde speichern so kann es schnell ein grosses Durcheinander geben (z.B. wenn man eine Navigation für einen Zoo und eine für Maschinen hat → passt nicht zusammen)
- ~~Vereinen von Bäumen pro Kunde~~ → wird nicht umgesetzt, weil es ein zu grosser Aufwand wäre, es wird nur das Vergleichen von Bäumen möglich sein (wobei einer der zwei Bäume editiert werden kann)
- Mehrere Benutzer gleichzeitig
  - Sperren von Teilbäumen oder der ganzen Navigation zum Bearbeiten → #143 wird teilweise umgesetzt, die Sperrung/der Hinweis wird pro Navigation geschehen / ausgegeben

3.2.3 Auswertung Testperson B

3.2.3.1 Allgemeines Feedback

Die Testperson nahm das Produkt schlussendlich auch als „positiv“ war, wobei die Testperson B jedoch etwas mehr Mühe hatte sich einzufinden, Ihr war nicht klar, dass das Kontextmenü mittels „Rechtsklick“ hervorgeholt werden konnte. Nach einer kurzen Einführung konnte jedoch auch diese Testperson alle Aufgaben ohne grössere Probleme lösen. Auch Testperson B kann sich vorstellen später mit dem Produkt zu arbeiten.

3.2.3.2 Navigations-Struktur allgemein:

- Baum anfangs nicht sichtbar (vor fertig laden) → #144 wird abgeklärt und ggf. umgesetzt
- Kontext Menü / Aktionen in der Toolbox → #145 wird sogar noch durch Tastenkombinationen (CTRL + S / CTRL + V, etc.) erweitert
- Filter Kriterien hinzufügen Button (Icon) → #146
- Suche/Filtersuche ganzes Wort markieren bei erneutem Klick ins Feld → #147
- Anordnung Icons (Änderungen Verwerfen nach links) → #145
- Input-Felder Schriftart und Feldlänge → #148
- ~~Filterkriterium mit Rechtsklick~~ → wird nicht umgesetzt, da dies die Testperson nur beim ersten Mal nicht gesehen hat, dies einzuführen würde das Konzept unnötig kompliziert machen

3.2.3.3 Filter

- Nach hinzufügen, anzeigen des hinzugefügten Filters / oder auch Meldung → #149

3.2.3.4 Allgemein

- ~~Tabs nicht ersichtlich (Hinweis kam von D. Keller)~~ → wird nicht umgesetzt, es sollen gar keine Tabs sein, sondern Menüpunkte, die Markierung soll nur ein „Is Active“ darstellen, zudem

fanden das beide Testpersonen in Ordnung → allerdings nochmals Rücksprache mit D. Keller machen (Termin: In der Sitzung vom Freitag 23.11.2012)

- Hinweis wenn bereits jemand an der Navigation arbeitet → #143

4. Unit Tests

4.1 Einführung

Eine Vielzahl der Tests greift auf die Datenbank zu, wir sind uns bewusst, dass dies keine Unit-Tests im klassischen Sinne sind. Den Mehraufwand um jedoch das Entity Framework durch ein Mocking-Framework zu mocken hielten wir jedoch bei diesem Anwendungsgebiet für nicht angebracht. Da die Hauptfunktionalität der Applikation auf Daten aus der Datenbank angewiesen ist. Somit stellen unsere Unit-Tests eher sogenannte Integrations-Tests dar.
Die Tests arbeiten jeweils mit einer speziellen Benutze Gruppe „unittest“ und dem Benutzer „unittestuser“. Beides wird automatisch angelegt falls nicht vorhanden, nach den Unit Tests werden jeweils alle erstellten Navigationen wieder gelöscht. Durch dieses Vorgehen wird sichergestellt, dass die anderen Daten nicht beeinflusst werden.
Die Unit Tests decken alle relevanten Funktionen ab, der Fokus liegt dabei klar in der Funktionalität der Navigationsverwaltung. Es wird sowohl getestet ob die Funktionen korrekte Manipulationen an der Navigation vornehmen und auch ob die Daten korrekt in der Datenbank gespeichert werden.

4.2 Testabdeckung

Insgesamt wurden für das System 125 Unit-Tests erstellt. Die Unit-Tests wurden jeweils kontinuierlich während des Entwicklungsprozesses erstellt. Durch die Unit-Tests wird eine Abdeckung von 84% im Business-Layer erreicht. Was die Vorgaben von 80% sogar ein wenig übertrifft. Die anderen beiden Layer (Presentation & Data) beinhalten keine Tests. Der Grund dafür ist, dass zum einen der Data-Layer im Prinzip nur aus dem Ado.Net Entity Framework Mapping der Datenbank besteht – welches automatisch generiert wurde – und zum anderen, dass der Clientseitige Teil im Browser sehr mühsam zum Testen wäre, dieser wird im Systemtest gründlich getestet.

5. Systemtests

Mit den Systemtests sollen komplexere Szenarien bzw. Use-Cases abgedeckt werden. Es soll ausserdem auch aufgezeigt werden was das System kann bzw. was für eine Funktionalität es bietet.

5.1 Testumgebung

Für die Systemtests wurde die Applikation auf einem Remote IIS (Version 8) installiert und konfiguriert. Die Datenbank lief auf einem MySQL Server in der Version 5.5. Client-seitig wurde jeweils Mozilla Firefox in der zu diesem Zeitpunkt aktuellen Version 17.0 getestet.

5.2 Navigation verwalten

Für dieses Kapitel wurden klar am meisten Tests durchgeführt. Da das Testszenario sehr gross und komplex ist, wurden die Tests in mehrere Teile unterteilt, wobei die Reihenfolge indirekt relevant ist. Die Folgetests haben oft eine Ausgangsituation welche durch die Vorhergehenden Tests entsteht. Wenn die Reihenfolge der Tests in diesem Dokument befolgt wird, so können alle Tests am Stück durchgeführt werden. Es kann jedoch natürlich auch ein beliebiger Test gemacht werden, dazu muss einfach zuerst die Navigation in der Ausgangsituation nachgebaut werden.
In den folgenden Tabellen wird aufgelistet was gemacht wurde, das erwartete Ergebnis wird jedoch nicht immer angegeben, bei logischen Testpunkten wird auf ein Ergebnis verzichtet. Es wird jedoch regelmässig mit Screenshots aufgezeigt wie die Navigation und die Statistik zu einem bestimmten Zeitpunkt aussehen sollten.

5.2.1 Videos

Es existieren für fast alle „Navigation verwalten“ Tests Videos welche die Testszenarien 1:1 durch spielen. Die Videos sind vor allem als zusätzliche Hilfestellung & Abwechslung zu den unten aufgelisteten Testprotokollen zu betrachten. Die Videos befinden sich im Ordner „videos“ und haben jeweils die gleiche Bezeichnung wie der zugehörige Test in diesem Dokument.

5.2.2 Abgedeckte UseCases

Implizit abgedeckt:

- UC02: Navigation erstellen
  - Für alle Testszenarien muss eine neue (leere) Navigation erstellt werden, sonst würde keines von Ihnen funktionieren, diese Funktion ist jedoch so trivial, dass sie nicht implizit getestet wird

Vollständig abgedeckt:

- UC04: Navigation verwalten
- UC05: Filter Kriterien mit Navigationspunkt verknüpfen
- UC07: Navigation testen

Teilweise abgedeckt:

- UC06: Filterkriterien verwalten
  - Das Auswählen und das Erstellen von Filterkriterien

5.2.3 1. Test - Grundfunktionalität 1 (Erstellen, Umbenennen, Löschen)

Ausgangssituation:
Leere Navigation namens „Tier-Navigation“, alle Statistiken auf 0.

Nr.	Aktion	Erwartetes Ergebnis	Status
1	Ausgangsituation herstellen: Neue Leere Navigation mit dem Namen „Tier-Navigation“ erstellen	Navigation: Leer Statistik: Leer	✓

2	Neuer Top Level Eintrag mittels Tastenkombination „CTRL+Leertaste“ namens „Säugetiere“		✓
3	Neuer Top Level Eintrag mittels Klick auf das Icon Top namens „Reptilien“		✓
4	Neuer Top Level Eintrag mittels Kontextmenü „Top Level-Element hinzufügen“ namens „Amphibien“		✓
5	Neues Element mit Klick auf „Säugetiere“ und dann „Leertaste“ namens „Pferde“		✓
6	Neues Element mit Klick auf das Icon namens „Tiger“		✓
7	Neues Element mit Klick auf „Pferde“ und dann mittels Kontextmenü „Element hinzufügen“ namens „Nordpolpferde“		✓
8	Navigation speichern mit Klick auf das Icon		✓
9	Öffnen der Statistiken mittels Klick auf das Symbol	Navigation: Säugetiere Pferde Nordpolpferde Tiger Reptilien Amphibien Statistik: **Tiefe** Durchschnittliche Tiefe 1.67 Maximale Tiefe 3 Minimale Tiefe 1 **Länge der Namen** Durschnittliche Länge 8.67 Maximale Länge 13 Minimale Länge 5 **Pro Stufe** Anzahl Navigationspunkte Insgesamt 6 Anzahl Navigationspunkte ohne Referenzen 6 Anzahl Referenzen 0 Anzahl Blätter (End-Navigationspunkte) 4 Anzahl Elemente auf Stufe 0 3 Anzahl Elemente auf Stufe 1 2 Anzahl Elemente auf Stufe 2 1 Durschnittliche Anzahl direkter Kinder für Stufe 0 0.67 Durschnittliche Anzahl direkter Kinder für Stufe 1 0.5 Durchschnittliche Anzahl Kinder für Stufe0 1 Durchschnittliche Anzahl Kinder für Stufe1 0.5 (Alle Filter Statistiken auf 0)	✓
10	Navigation als Kopie Speichern mit Klick auf das Icon namens „Tier-Navigation Vorlage“	Erwartung genau gleiche Navigation wie bei Schritt 9 als Kopie	✓
11	Umbenennen mit Doppelklick auf „Nordpolpferde“ nach „Südpolpferdä“		✓
12	Umbenennen mit Klick auf „Südpolpferdä“ und drücken der Taste „F2“ auf „Südpolpferdö“		✓
13	Umbenennen mit Klick auf „Südpolpferdö“ und Klick auf das Icon auf „Südpolpferdü“		✓
14	Umbenennen mit Klick auf „Südpolpferdü“, dann mittels Kontextmenü „Umbenennen“ auf „Südpolpferde“	Erwartung Navigation gleich wie bei Schritt 9, aber „Nordpolpferde“ heisst jetzt „Südpolpferde“	✓

15	Klick auf „Pferde“ und betätigen der „Delete“ Taste	„Pferde“ und „Südpolpferde“ gelöscht	✓
16	Klick auf „Tiger“ und Klick auf das Icon ⊖		✓
17	Klick auf „Amphibien“ und mittels Kontextmenü „Löschen“ den Punkt löschen		✓
18	Speichern der Navigation mittels Tastenkombination „CTRL+S“	Navigation: Säugetiere Reptilien Statistik: **Tiefe** Durchschnittliche Tiefe 1 Maximale Tiefe 1 Minimale Tiefe 1 **Länge der Namen** Durschnittliche Länge 9.5 Maximale Länge 10 Minimale Länge 9 **Pro Stufe** Anzahl Navigationspunkte Insgesamt 2 Anzahl Navigationspunkte ohne Referenzen 2 Anzahl Referenzen 0 Anzahl Blätter (End-Navigationspunkte) 2 Anzahl Elemente auf Stufe 0 2 (Alle Filter Statistiken auf 0)	✓
19	Logout, Re-Login, Öffnen der Navigation „Tier-Navigation“	Exakt gleiche Navigation und Filter wie bei Punkt 18	✓
20	Löschen der kompletten Navigation mit Klick auf das Icon ✖	Navigation gelöscht	✓

5.2.4 2. Test - Grundfunktionalität 2 (Filterkriterien)

Ausgangssituation:
Wenn der Test vom Kapitel 5.2.3 ausgeführt wurde, so sollte diese Navigation bereits unter dem Namen „Tier-Navigation Vorlage“ vorhanden sein, ansonsten muss diese erstellt werden.

Nr.	Aktion	Erwartetes Ergebnis	Status
1	Ausgangsituation herstellen (Wenn möglich „Tier-Navigation Vorlage“ öffnen)	Säugetiere Pferde Nordpolpferde Tiger Reptilien Amphibien	✓
2	Filter hinzufügen mit Klick auf „Nordpolpferde“ und Eingabe des Textes „Geschlächt“ in das Filterkriteriumfeld, Bestätigung mittels Enter	Meldung: „Sind Sie sicher, dass Sie einen neuen Filter Geschlächt erfassen und hinzufügen möchten?“	✓
3	Meldung mit „Abbrechen“ bestätigen	Filter wurde nicht hinzugefügt	✓
4	Den Inhalt des Textfeldes auf „Geschlecht“ ändern und den Filter mittels Klick auf das Icon ⊕ hinzufügen	Meldung: „Sind Sie sicher, dass Sie einen neuen Filter Geschlecht erfassen und hinzufügen möchten?“	✓
5	Meldung mit „Ok“ bestätigen	Filter hinzugefügt	✓
6	Klick auf „Tiger“ in das Filtertextfeld den Text „Gesch“ eingeben, aus der Liste „Geschlecht“ mit Maustaste anklicken und mit Enter bestätigen	Filter wird ohne Bestätigung hinzugefügt (da bereits erfasst)	✓

7	Direkt weiter schreiben und ein neuer Filter Namens „Geburtsort“ erfassen, Meldung mit „OK“ bestätigen		
8	Klick auf „Nordpolpferde“, in das Filtertextfeld „Ge“ eingeben, mittels Pfeiltasten „Geburtsort“ auswählen und mit Enter bestätigen	Filter wird ohne Bestätigung hinzugefügt (da bereits erfasst)	
9	Filter „Geburtsort“ wieder löschen mittels Klick auf das Icon		
10	Bei „Nordpolpferde“ in das Filtertextfeld „Geschlecht“ eingeben und mit Enter Taste bestätigen	Filter sollte nicht erneut hinzugefügt werden und eine Fehlermeldung erscheinen: „ Der Filter existiert bereits bei diesem Punkt!“	
11	Öffnen der Statistiken mittels Klick auf das Symbol	Navigation: Säugetiere Pferde Südpolpferde Geschlecht Tiger Geschlecht Geburtsort Reptilien Amphibien Statistik:	
12	Navigation mittels Klick auf das Icon speichern		

Tiefe

Durchschnittliche Tiefe	1.67
Maximale Tiefe	3
Minimale Tiefe	1

Länge der Namen

Durschnittliche Länge	8.5
Maximale Länge	12
Minimale Länge	5

Pro Stufe

Anzahl Navigationspunkte Insgesamt	6
Anzahl Navigationspunkte ohne Referenzen	6
Anzahl Referenzen	0
Anzahl Blätter (End-Navigationspunkte)	4
Anzahl Elemente auf Stufe 0	3
Anzahl Elemente auf Stufe 1	2
Anzahl Elemente auf Stufe 2	1
Durschnittliche Anzahl direkter Kinder für Stufe 0	0.67
Durschnittliche Anzahl direkter Kinder für Stufe 1	0.5
Durchschnittliche Anzahl Kinder für Stufe0	1
Durchschnittliche Anzahl Kinder für Stufe1	0.5

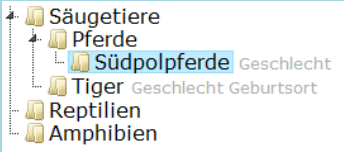
Filter

Anzahl vorhandener Filter	2
Insgeamt zugewiesene Filter	3
Insgeamt zugewiesene Filter ohne Referenzen	3
Insgeamt zugewiesene Filter an Referenzen	0
Anzahl verschiedener zugewiesener Filter	2
Durchschnittliche Anzahl zugewiesene Filter	0.75
Maximale Anzahl zugewiesene Filter	2
Minimale Anzahl zugewiesene Filter	0
Anzahl Filter auf Stufe 0	0
Anzahl Filter auf Stufe 1	2
Anzahl Filter auf Stufe 2	1
Durchschnittliche Anzahl Filter auf Stufe 0	0
Durchschnittliche Anzahl Filter auf Stufe 1	2
Durchschnittliche Anzahl Filter auf Stufe 2	1

5.2.5 3. Test - Grundfunktionalität 3 (Nicht-Manipulative Operationen)

In diesem Kapitel werden nur nicht manipulative Operationen getestet, d.h. Aktionen bei denen die Navigation nicht verändert wird.

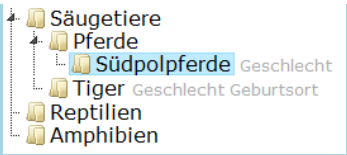
Ausgangssituation:
Wenn der Test vom Kapitel 5.2.4 ausgeführt wurde, so sollte diese Navigation bereits unter dem Namen „Tier-Navigation Vorlage“ vorhanden sein, ansonsten muss diese erstellt werden.

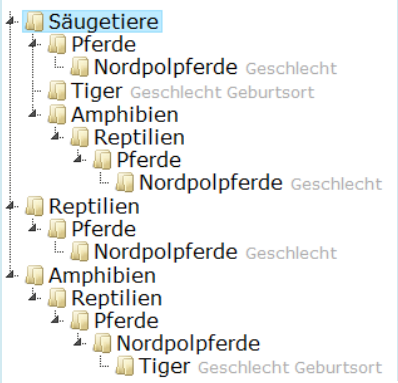
Nr.	Aktion	Erwartetes Ergebnis	Status
1	Ausgangsituation herstellen (Wenn möglich „Tier-Navigation Vorlage“ öffnen)		✓
2	In das Feld „Suchen...“ den Begriff „ie“ eingeben und mit der Enter Taste bestätigen	Säugetiere, Reptilien und Amphibien sind im Baum rot markiert (wurden als Suchergebnis gefunden)	✓
3	Das Suchergebnis mittels Klick auf das Icon wieder löschen	Im Baum ist nichts mehr markiert und das Suchfeld ist wieder leer	✓
4	In das Feld „Suchen...“ den Begriff „nicht“ eingeben und mit der Enter Taste bestätigen	Im Baum ist nichts markiert, da die Suche keine Ergebnisse erzielt hat	✓
5	Auf das Icon klicken	Navigation ist komplett zugeklappt (d.h. nur Säugetiere, Repitlien und Amphibien sichtbar)	✓
6	Auf das Icon klicken	Navigation komplett aufgeklappt	✓
7	Klick auf „Säugetiere“ und einmal die „Pfeil nach Unten“ taste auf der Tastatur drücken	„Pferde“ ist markiert	✓
8	CTRL + „Pfeil nach Links“ auf der Tastatur drücken	Der Subbaum von „Pferde“ wird zu geklappt	✓
9	CTRL + F drücken	Fokus ist im Suchfeld, das Wort „nicht“ ist markiert	✓
10	Auf „Säugetiere“ klicken und Klick auf das Icon machen	Säugetiere und alle darunterliegende Punkte gelöscht	✓
11	Klick auf das Icon	Meldung „Sind Sie sicher, dass Sie ihre Änderungen verwerfen wollen?“	✓
12	Meldung mit „Abbrechen“ bestätigen	Navigation noch immer unverändert	✓
13	Klick auf das Icon	Meldung „Sind Sie sicher, dass Sie ihre Änderungen verwerfen wollen?“	✓
14	Meldung mit „Ok“ bestätigen	Ausgangssituation wieder hergestellt, keien Änderungen wurden gespeichert	✓
15	Auf „Säugetiere“ klicken und Klick auf das Icon machen	Säugetiere und alle darunterliegende Punkte gelöscht	✓
16	Aus dem Navigation Wahl Menü eine beliebig andere Navigation auswählen	Meldung „Achtung! Es bestehen ungspeicherte Änderungen, wenn sie die Navigation jetzt wechseln, gehen diese verloren. Wollen Sie die Navigation wechseln?“	✓
17	Meldung mit „Abbrechen“ bestätigen	Man befindet sich immernoch in der gleichen Navigation, Kategorie „Säugetiere“ nach wie vor gelöscht	✓

18	Aus dem Navigation Wahl Menü eine beliebig andere Navigation auswählen	Meldung „Achtung! Es bestehen ungspeicherte Änderungen, wenn sie die Navigation jetzt wechseln, gehen diese verloren. Wollen Sie die Navigation wechseln?“	✓
19	Meldung mit „Ok“ bestätigen	Man befindet sich in einer anderen Navigation	✓
20	Zurück zur alten Navigation „Tier-Navigation Vorlage“ wechseln	Navigation befindet sich im gleichen Zustand wie bei Schritt 1 (Ausgangsituation)	✓

5.2.6 4. Test - Kopieren

Ausgangssituation:
Wenn der Test vom Kapitel 5.2.3 bzw. 5.2.4 ausgeführt wurde, so sollte diese Navigation bereits unter dem Namen „Tier-Navigation Vorlage“ vorhanden sein, ansonsten muss diese erstellt werden.

Nr.	Aktion	Erwartetes Ergebnis	Status
1	Ausgangsituation herstellen (Wenn möglich „Tier-Navigation Vorlage“ öffnen)		✓
2	Klick auf „Pferde“ und dann Klick auf das Icon		✓
3	Klick auf „Reptilien“ und anschliessend auf das Icon	Teil Baum „Säugetiere“ komplett nach „Reptilien“ kopiert	✓
4	Betätigen der Tastenkombination „CTRL + C“ (direkt beim markierten Punkt „Reptilien“)		✓
5	Klick auf „Amphibien“ und betätigen der Tastenkombination „CTRL+V“	Teil Baum „Reptilien“ komplett nach „Amphibien“ kopiert	✓
6	Beim Menüpunkt „Amphibien“ mittels Kontextmenü „Kopieren“ den Teil Baum kopieren		✓
7	Klick auf „Säugetiere“ und mittels Kontextmenü „Einfügen“	Teilbaum „Amphibien“ komplett nach „Säugetiere“ kopiert	✓

8	Mittels Drag und Drop und gedrückter CTRL Taste „Tiger“ in das letzte Element („Nordpolpferde“) ziehen	Tiger wurde in das letzte Element („Nordpolpferde“) kopiert  **Tiefe** Durchschnittliche Tiefe 4.67 Maximale Tiefe 5 Minimale Tiefe 4 **Länge der Namen** Durschnittliche Länge 8.79 Maximale Länge 13 Minimale Länge 5 **Pro Stufe** Anzahl Navigationspunkte Insgesamt 28 Anzahl Navigationspunkte ohne Referenzen 28 Anzahl Referenzen 0 Anzahl Blätter (End-Navigationspunkte) 10 Anzahl Elemente auf Stufe 0 3 Anzahl Elemente auf Stufe 1 8 Anzahl Elemente auf Stufe 2 8 Anzahl Elemente auf Stufe 3 6 Anzahl Elemente auf Stufe 4 3 Durschnittliche Anzahl direkter Kinder für Stufe 0 2.67 Durschnittliche Anzahl direkter Kinder für Stufe 1 1 Durschnittliche Anzahl direkter Kinder für Stufe 2 0.75 Durschnittliche Anzahl direkter Kinder für Stufe 3 0.5 Durchschnittliche Anzahl Kinder für Stufe0 8.33 Durchschnittliche Anzahl Kinder für Stufe1 2.13 Durchschnittliche Anzahl Kinder für Stufe2 1.13 Durchschnittliche Anzahl Kinder für Stufe3 0.5 **Filter** Anzahl vorhandener Filter 2 Insgeamt zugewiesene Filter 13 Insgeamt zugewiesene Filter ohne Referenzen 13 Insgeamt zugewiesene Filter an Referenzen 0 Anzahl verschiedener zugewiesener Filter 2 Durchschnittliche Anzahl zugewiesene Filter 1.3 Maximale Anzahl zugewiesene Filter 2 Minimale Anzahl zugewiesene Filter 0 Anzahl Filter auf Stufe 0 0 Anzahl Filter auf Stufe 1 2 Anzahl Filter auf Stufe 2 4 Anzahl Filter auf Stufe 3 3 Anzahl Filter auf Stufe 4 4 Durchschnittliche Anzahl Filter auf Stufe 0 0 Durchschnittliche Anzahl Filter auf Stufe 1 2 Durchschnittliche Anzahl Filter auf Stufe 2 1.33 Durchschnittliche Anzahl Filter auf Stufe 3 1 Durchschnittliche Anzahl Filter auf Stufe 4 1.33	✓
9	Klick auf das Icon und Navigation unter dem Namen „Tier-Navigation-Copy-Tests“ speichern		✓
10	Im „Navigation“ Wahl Menü die Navigation „Tier-Navigation-Copy-Tests“ auswählen	Meldung „Achtung! Es bestehen ungspeicherte Änderungen, wenn sie die Navigation jetzt wechseln, gehen diese verloren. Wollen Sie die Navigation wechseln?“	✓
11	Meldung mit „Ok“ bestätigen	Man befindet sich jetzt in der Navigation „Tier-Navigation-Copy-Tests“, Navigation & Statistik komplett identisch mit Schritt 8	✓

12	Navigation mittels Klick auf das Icon löschen	Navigation gelöscht	

5.2.7 5. Test - Verschieben (Ausschneiden)

Ausgangssituation:
Wenn der Test vom Kapitel 5.2.3 bzw. 5.2.4 ausgeführt wurde, so sollte diese Navigation bereits unter dem Namen „Tier-Navigation Vorlage“ vorhanden sein, ansonsten muss diese erstellt werden.

Nr.	Aktion	Erwartetes Ergebnis	Status
1	Ausgangsituation herstellen (Wenn möglich „Tier-Navigation Vorlage“ öffnen)	Säugetiere Pferde Südpolpferde Geschlecht Tiger Geschlecht Geburtsort Reptilien Amphibien	
2	Sortierung ändern: Mittels Drag und Drop „Säugetiere“ nach ganz unten verschieben (nach „Amphibien“)		
3	Sortierung ändern: Mittels Drag und Drop „Reptilien“ und „Amphibien“ vertauschen	Amphibien Reptilien Säugetiere Pferde Nordpolpferde Geschlecht Tiger Geschlecht Geburtsort	
4	Mittels Drag und Drop „Pferde“ nach „Amphibien“ verschieben		
5	Klick auf „Amphibien“ dann Klick auf das Icon		
6	Klick auf „Reptilien“ dann Klick auf das Icon	Reptilien Amphibien Pferde Nordpolpferde Geschlecht Säugetiere Tiger Geschlecht Geburtsort	
7	Klick auf „Tiger“ dann Tastenkombination „CTRL+X“		
8	Klick auf „Pferde“ dann Tastenkombination „CTRL+V“		
9	Klick auf „Reptilien“ und mittels Kontextmenü „Ausschneiden“ wählen		

10	Klick auf „Säugetiere“ und mittels Kontextmenü „Einfügen“ wählen	Säugetiere Reptilien Amphibien Pferde Nordpolpferde Geschlecht Tiger Geschlecht Geburtsort **Tiefe** Durchschnittliche Tiefe 5 Maximale Tiefe 5 Minimale Tiefe 5 **Länge der Namen** Durschnittliche Länge 8.67 Maximale Länge 13 Minimale Länge 5 **Pro Stufe** Anzahl Navigationspunkte Insgesamt 6 Anzahl Navigationspunkte ohne Referenzen 6 Anzahl Referenzen 0 Anzahl Blätter (End-Navigationspunkte) 2 Anzahl Elemente auf Stufe 0 1 Anzahl Elemente auf Stufe 1 1 Anzahl Elemente auf Stufe 2 1 Anzahl Elemente auf Stufe 3 1 Anzahl Elemente auf Stufe 4 2 Durschnittliche Anzahl direkter Kinder für Stufe 0 1 Durschnittliche Anzahl direkter Kinder für Stufe 1 1 Durschnittliche Anzahl direkter Kinder für Stufe 2 1 Durschnittliche Anzahl direkter Kinder für Stufe 3 2 Durchschnittliche Anzahl Kinder für Stufe0 5 Durchschnittliche Anzahl Kinder für Stufe1 4 Durchschnittliche Anzahl Kinder für Stufe2 3 Durchschnittliche Anzahl Kinder für Stufe3 2 **Filter** Anzahl vorhandener Filter 2 Insgeamt zugewiesene Filter 3 Insgeamt zugewiesene Filter ohne Referenzen 3 Insgeamt zugewiesene Filter an Referenzen 0 Anzahl verschiedener zugewiesener Filter 2 Durchschnittliche Anzahl zugewiesene Filter 1.5 Maximale Anzahl zugewiesene Filter 2 Minimale Anzahl zugewiesene Filter 1 Anzahl Filter auf Stufe 0 0 Anzahl Filter auf Stufe 1 0 Anzahl Filter auf Stufe 2 0 Anzahl Filter auf Stufe 3 0 Anzahl Filter auf Stufe 4 3 Durchschnittliche Anzahl Filter auf Stufe 0 0 Durchschnittliche Anzahl Filter auf Stufe 1 0 Durchschnittliche Anzahl Filter auf Stufe 2 0 Durchschnittliche Anzahl Filter auf Stufe 3 0 Durchschnittliche Anzahl Filter auf Stufe 4 1.5	
11	Klick auf das Icon und Navigation unter dem Namen „Tier-Navigation-Move-Tests“ speichern		
12	Im „Navigation“ Wahl Menü die Navigation „Tier-Navigation-Copy-Tests“ auswählen	Meldung „Achtung! Es bestehen ungespeicherte Änderungen, wenn sie die Navigation jetzt wechseln, gehen diese verloren. Wollen Sie die Navigation wechseln?“	
13	Meldung mit „Ok“ bestätigen	Man befindet sich jetzt in der Navigation „Tier-Navigation-Move-Tests“, Navigation & Statistik komplett identisch mit Schritt 10	
14	Navigation mittels Klick auf das Icon löschen	Navigation gelöscht	

5.2.8 6. Test - Referenzen

In diesem Kapitel werden die Referenzen getestet, es wird jedoch darauf verzichtet „Referenzen auf Referenzen“ zu testen, da dies keine direkte Anforderung ist und selten benötigt wird.

Ausgangssituation:

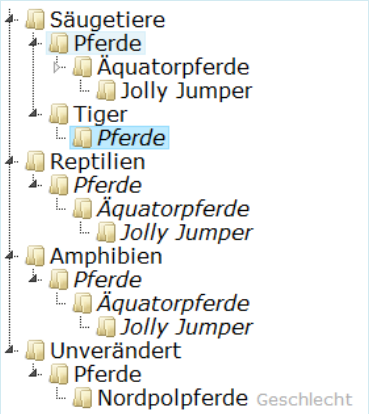
Wenn der Test vom Kapitel 5.2.3 bzw. 5.2.4 ausgeführt wurde, so sollte diese Navigation bereits unter dem Namen „Tier-Navigation Vorlage“ vorhanden sein, ansonsten muss diese erstellt werden.

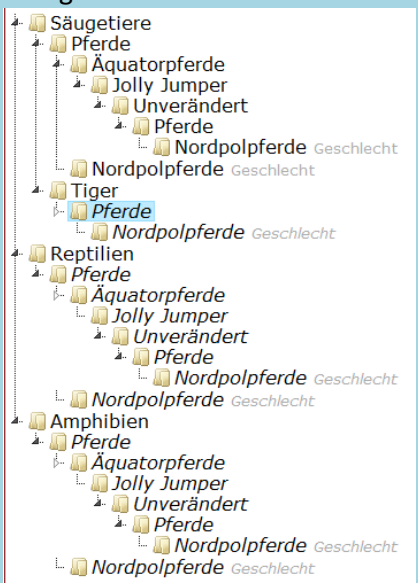
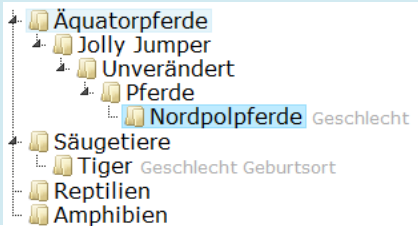
Empfehlung:
Bei diesem Test empfiehlt sich mindestens parallel zu dem Testprotokoll den Video an zu sehen, da dieses Szenario recht komplex und lange ist und im Video noch einige Erklärungen zu gewissen Operationen geliefert werden.

Nr.	Aktion	Erwartetes Ergebnis	Status
1	Ausgangsituation herstellen (Wenn möglich „Tier-Navigation Vorlage“ öffnen)	Säugetiere Pferde Südpolpferde Geschlecht Tiger Geschlecht Geburtsort Reptilien Amphibien	✓
2	Klick auf das Icon und Navigation unter dem Namen „Tier-Navigation-Referenz-Tests“ speichern		✓
3	Im „Navigation“ Wahl Menü die Navigation „Tier-Navigation-Referenz-Tests“ auswählen	Man befindet sich jetzt in der Navigation „Tier-Navigation-Referenz-Tests“, Navigation & Statistik komplett identisch mit Schritt 1	✓
4	Klick auf „Pferde“ und dann Klick auf das Icon		✓
5	Klick auf „Tiger“ und anschliessend auf das Icon	Referenz des Teil Baum „Pferde“ komplett nach „Tiger“ kopiert	✓
6	Klick auf „Pferde“ und betätigen der Tastenkombination „CTRL + R“		✓
7	Klick auf „Reptilien“ und betätigen der Tastenkombination „CTRL + V“	Referenz des Teil Baum „Pferde“ komplett nach „Reptilien“ kopiert	✓
8	Klick auf „Pferde“ und mittels Kontextmenü den Eintrag „Referenz kopieren“ wählen		✓
9	Klick auf „Amphibien“ und mittels Kontextmenü den Eintrag „Einfügen“ wählen	Referenz des Teil Baum „Pferde“ komplett nach „Amphibien“ kopiert Navigation: Säugetiere Pferde Nordpolpferde Geschlecht Tiger *Pferde* *Nordpolpferde* Geschlecht Reptilien *Pferde* *Nordpolpferde* Geschlecht Amphibien *Pferde* *Nordpolpferde* Geschlecht	✓
10	Mittels „CTRL+Leertaste“ ein neuer Punkt Namens „Unverändert“ einfügen		✓
11	Klick auf „Pferde“ und betätigen der Taste „CTRL +C“		✓
12	Klick auf „Unverändert“ und betätigen der Taste „CTRL +V“	Normale Kopie von „Pferde“ erstellt	✓
13	Klick auf das Original „Nordpolpferde“ (bei Säugetieren) dann mittels F2 umbenennen auf „Südpolpferde“	Alle „Nordpolpferde“ ausser diese bei „Unverändert“ wurden auf „Südpolpferde“ umbenannt	✓
14	Klick auf eine Referenz „Südpolpferde“ (kursiv), dann mittels F2 umbenennen auf „Äquatorpferde“	Alle wurden auf „Äquatorpferde“ umbenannt	✓

15	Klick auf das Original „Pferde“ (bei Säugetieren) dann mittels Leertaste ein neues Element namens „Nordpolpferde“ hinzufügen	„Nordpolpferde“ überall ausser bei „Unverändert“ erstellt	✓
16	Klick auf eine Referenz „Äquatorpferde“ (kursiv) dann mittels Leertaste ein neues Element namens „Jolly Jumper“ hinzufügen	„Jolly Jumper“ überall ausser bei „Unverändert“ erstellt Navigation: Säugetiere Pferde Äquatorpferde Jolly Jumper Nordpolpferde Tiger *Pferde* *Äquatorpferde* *Jolly Jumper* *Nordpolpferde* Reptilien *Pferde* *Äquatorpferde* *Jolly Jumper* *Nordpolpferde* Amphibien *Pferde* *Äquatorpferde* *Jolly Jumper* *Nordpolpferde* Unverändert Pferde Nordpolpferde Geschlecht Statistik:	✓
17	Navigation mittels Klick auf das Icon 💾 speichern		✓

Tiefe	
Durchschnittliche Tiefe	4
Maximale Tiefe	5
Minimale Tiefe	3
Länge der Namen	
Durschnittliche Länge	10.39
Maximale Länge	13
Minimale Länge	5
Pro Stufe	
Anzahl Navigationspunkte Insgesamt	23
Anzahl Navigationspunkte ohne Referenzen	11
Anzahl Referenzen	12
Anzahl Blätter (End-Navigationspunkte)	9
Anzahl Elemente auf Stufe 0	4
Anzahl Elemente auf Stufe 1	5
Anzahl Elemente auf Stufe 2	8
Anzahl Elemente auf Stufe 3	5
Anzahl Elemente auf Stufe 4	1
Durschnittliche Anzahl direkter Kinder für Stufe 0	1.25
Durschnittliche Anzahl direkter Kinder für Stufe 1	1.6
Durschnittliche Anzahl direkter Kinder für Stufe 2	0.63
Durschnittliche Anzahl direkter Kinder für Stufe 3	0.2
Durchschnittliche Anzahl Kinder für Stufe0	4.75
Durchschnittliche Anzahl Kinder für Stufe1	2.8
Durchschnittliche Anzahl Kinder für Stufe2	0.75
Durchschnittliche Anzahl Kinder für Stufe3	0.2
Filter	
Anzahl vorhandener Filter	2
Insgeamt zugewiesene Filter	1
Insgeamt zugewiesene Filter ohne Referenzen	1
Insgeamt zugewiesene Filter an Referenzen	0
Anzahl verschiedener zugewiesener Filter	1
Durchschnittliche Anzahl zugewiesene Filter	0.11
Maximale Anzahl zugewiesene Filter	1
Minimale Anzahl zugewiesene Filter	0
Anzahl Filter auf Stufe 0	0
Anzahl Filter auf Stufe 1	0
Anzahl Filter auf Stufe 2	1
Anzahl Filter auf Stufe 3	0
Anzahl Filter auf Stufe 4	0
Durchschnittliche Anzahl Filter auf Stufe 0	0
Durchschnittliche Anzahl Filter auf Stufe 1	0
Durchschnittliche Anzahl Filter auf Stufe 2	0.25
Durchschnittliche Anzahl Filter auf Stufe 3	0
Durchschnittliche Anzahl Filter auf Stufe 4	0

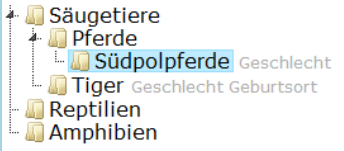
18	Reload der Seite mittels „F5“	Gleiche Navigation wie bei Schritt 16	✓
19	Klick auf das Original „Nordpolpferde“ (bei Säugetieren), dann den Filter „Geburtsort“ hinzufügen	Filter „Geburtsort“ überall ausser bei „Unverändert“ hinzugefügt	✓
20	Klick auf eine Referenz „Jolly Jumper“ (kursiv) dann den Filter „Geschlecht“ hinzufügen	Filter „Geschlecht“ überall ausser bei „Unverändert“ hinzugefügt	✓
21	Klick auf das Original „Nordpolpferde“ (bei Säugetieren), dann den Filter „Geburtsort“ entfernen	Filter „Geburtsort“ überall enfernt	✓
22	Klick auf eine Referenz „Jolly Jumper“ (kursiv) dann den Filter „Geschlecht“ entfernen	Filter „Geschlecht“ überall ausser bei „Unverändert“ entfernt	✓
23	Klick auf das Original „Nordpolpferde“ (bei Säugetieren), mittels Kontextmenü „Löschen“ wählen	„Nordpolpferde“ überall ausser bei „Unverändert“ entfernt	✓
24	Klick auf die erste Referenz „Äquatorpferde“ (kursiv), mittels Kontextmenü „Löschen“ wählen	„Äquatorpferde“ NUR bei dieser Referenz entfernt Navigation: 	✓
25	Navigation mittels Klick auf das Icon speichern		✓
26	Reload der Seite mittels „F5“	Gleiche Navigation wie bei Schritt 24	✓
27	Klick auf „Nordpolpferde“ (bei „Unverändert“) betätigen von „CTRL + C“		✓
28	Klick auf das Original „Pferde“ (bei Säugetieren) betätigen von „CTRL + V“	„Nordpolpferde“ wurde überall (ausser bei „Unverändert“ selbst) hineinkopiert	✓
29	Klick auf eine Referenz „Pferde“ (kursiv) betätigen der Tastenkombination „CTRL + V“	Es wurde nichts eingefügt, es erscheint eine Fehlermeldung: „Kopieren fehlgeschlagen! Man kann in eine Referenz nichts hineinkopieren, kopieren Sie es in das Original.“	✓
30	Mittels „Drag & Drop“ verschieben von „Unverändert“ (ganz unten) in eine Referenz „Pferde“ (kursiv)	Es wurde nichts verschoben, Fehlermeldung: " Verschieben fehlgeschlagen! Man kann in eine Referenz nichts hineinschieben, schieben Sie es in das Original.“	✓

31	Mittels „Drag & Drop“ verschieben von „Unverändert“ (ganz unten) in das Original „Jolly Jumper“ (bei „Säugetieren“)	„Unverändert“ wurde verschoben, ist überall vorhanden Navigation:  	✓
32	Navigation mittels Klick auf das Icon 💾 speichern		✓
33	Reload der Seite mittels „F5“	Gleiche Navigation wie bei Schritt 31	✓
34	Mittels „Drag &Drop“ das Original „Äquatorpferde“ (bei „Säugetieren“) hinausziehen und oberhalb von „Säugetieren“ platzieren	Der Teil Baum „Äquatorpferde“ ist auch bei allen Referenzen entfernt und ist jetzt eigenständig oberhalb von Säugetieren	✓
35	Mittels „Drag & Drop“ das Original „Pferde“ nach „Äquatorpferde“ verschieben	Referenzen verändern sich nicht, nur das Original „Pferde“ befindet sich an einem neuen Ort	✓
36	Klick auf das Original Pferde und mittels des Kontextmenü löschen	Alle „Pferde“ inklusive Subelementen sind gelöscht  	✓
37	Navigation mittels Klick auf das Icon 💾 speichern		✓
38	Reload der Seite mittels „F5“	Gleiche Navigation wie bei Schritt 36	✓
39	Navigation mittels Klick auf das Icon ✖ löschen	Navigation gelöscht	✓

5.2.9 7. Test - Undo-Funktion

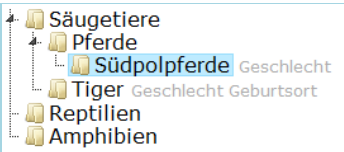
Ausgangssituation:
Wenn der Test vom Kapitel 5.2.3 bzw. 5.2.4 ausgeführt wurde, so sollte diese Navigation bereits unter dem Namen „Tier-Navigation Vorlage“ vorhanden sein, ansonsten muss diese erstellt werden.

Nr.	Aktion	Erwartetes Ergebnis	Status
1	Ausgangsituation herstellen (Wenn möglich „Tier-Navigation Vorlage“ öffnen)	Säugetiere Pferde Südpolpferde Geschlecht Tiger Geschlecht Geburtsort Reptilien Amphibien	✓
2	„Amphibien“ umbenennen auf „Tiere“		✓
3	„Pferde“ löschen		✓
4	Ein neues Top Level Item namens „Vögel“ einfügen		✓
5	Bei „Reptilien“ ein neues Element namens „Schlangen“ hinzufügen	Säugetiere Tiger Geschlecht Geburtsort Reptilien Schlangen Tiere Vögel	✓
6	Bei „Tiger“ den Filter „Geburtsort“ entfernen		✓
7	Bei „Schlangen“ den Filter „Geschlecht“ einfügen		✓
8	„Reptilien“ nach „Tiere“ kopieren		✓
9	„Vögel“ nach „Tiere“ verschieben		✓
10	Eine Referenz von „Säugetiere“ nach „Tiere“ kopieren	Säugetiere Tiger Geschlecht Reptilien Schlangen Geschlecht Tiere Reptilien Schlangen Geschlecht Vögel *Säugetiere* *Tiger Geschlecht*	✓
11	Original „Säugetiere“ umbenennen auf „Säuger“		✓
12	Bei Original „Tiger“ ein neues Element namens „Sibirischer“ hinzufügen		✓
13	Original „Tiger“ löschen		✓
14	Referenz „Säuger“ löschen	Säuger Reptilien Schlangen Geschlecht Tiere Reptilien Schlangen Geschlecht Vögel	✓
15	Rückgängig mittels „CTRL + Z“	Referenz von „Säuger“ wieder vorhanden	✓
16	Rückgängig mittels Klick auf das Icon	Tiger wieder vorhanden (Original, sowie Referenzen)	✓
17	Rückgängig	„Sibirischer“ beim Original, sowie den Referenzen wieder entfernt	✓
18	Rückgängig	„Säuger“ heisst wieder „Säugetiere“	✓
19	Rückgängig	Die Referenz von „Säugetiere“ ist entfernt	✓
20	Rückgängig	„Vögel“ ist wieder ein TopLevel Item	✓

21	Rückgängig	Die Kopie von „Reptilien“ bei „Tiere“ ist gelöscht	✓
22	Rückgängig	Der Filter „Geschlecht“ bei „Schlangen“ ist entfernt	✓
23	Rückgängig	Der Filter „Geburtsort“ bei „Tiger“ ist wieder vorhanden	✓
24	Rückgängig	Das Element „Schlange“ ist entfernt	✓
25	Rückgängig	„Vögel“ ist entfernt	✓
26	Rückgängig	„Pferde“ (inklusive Sub-Elementen) ist wieder da	✓
27	Rückgängig	„Tiere“ heisst wieder „Amphibien“ Ursprüngliche Navigation wiederhergestellt: 	✓
28	Navigation mittels Klick auf das Icon 💾 speichern		✓
29	Reload der Seite mittels „F5“	Gleiche Navigation wie bei Schritt 27	✓

5.2.10 8. Test - Navigationen vergleichen

Ausgangssituation:
Wenn der Test vom Kapitel 5.2.3 bzw. 5.2.4 ausgeführt wurde, so sollte diese Navigation bereits unter dem Namen „Tier-Navigation Vorlage“ vorhanden sein, ansonsten muss diese erstellt werden.
Empfehlung:
Bei diesem Test empfiehlt sich mindestens parallel zu dem Testprotokoll den Video an zu sehen, das Szenario an sich ist zwar nicht komplex, es kann sich jedoch viel besser veranschaulicht werden als in reiner Textform.

Nr.	Aktion	Erwartetes Ergebnis	Status
1	Ausgangsituation herstellen (Wenn möglich „Tier-Navigation Vorlage“ öffnen)		✓
2	Löschen von „Reptilien“		✓
3	Löschen von „Amphibien“		✓
4	Neuer Top Level Eintrag „Vögel“		✓
5	Bei „Vögel“ ein neues Element „Geier“ erstellen		✓
6	Bei „Vögel“ ein neues Element namens „Adler“ erstellen		✓
7	Bei „Adler“ den Filter „Flügelspannweite“ hinzufügen		✓
8	Neuer Top Level Eintrag „Dinosaurier“		✓

9	Klick auf das Icon und Navigation unter dem Namen „Tier-Navigation-Vergleich“ speichern		✓
10	Änderungen verwerfen mittels Klick auf das Icon		✓
11	Mittels des Dropdownmenus „Navigationen zum Vergleichen auswählen“ die Navigation „Tier-Navigation Vorlage“ öffnen	Navigation „Tier-Navigation Vorlage“ öffnet sich als Vergleichs-Navigation	✓
12	Mittels des Dropdownmenus „Navigationen zum Vergleichen auswählen“ die Navigation „Tier-Navigation-Vergleich“ öffnen	Navigation „Tier-Navigation-Vergleich“ öffnet sich als Vergleichs-Navigation	✓
13	Mit Drag & Drop eine der beiden geöffneten Vergleichs Navigationen besser platzieren, so dass sich alle drei Navigationen nicht überlappen bzw. alle sichtbar sind	Alle drei Navigationen sichtbar	✓
14	Klick auf das Icon	Bei allen drei Navigationen werden alle Navigationspunkte angezeigt	✓
15	Klick auf das Icon	Bei allen drei Navigationen werden alle Navigationspunkte zugeklappt	✓
16	Mittels des Suchfeldes nach „Pferde“ suchen	In allen drei Navigationen ist Pferde „rot“ markiert (wurde gefunden)	✓
17	Mittels des Suchfeldes nach „Reptilien“ suchen	In der zu bearbeitenden Navigation, sowie der Vergleichs-Navigation „Tier-Navigation Vorlage“ wird „Reptilien“ gefunden und markiert	✓
18	Mittels des Suchfeldes nach „Adler“ suchen	Nur in der Vergleichs-Navigation „Tier-Navigation-Vergleich“ wird Adler gefunden und markiert	✓
19	Bei der Vergleichs-Navigation „Tier-Navigation Vorlage“ Punkte auf und zu klappen	Punkte werden nur in der Vergleichs-Navigation afu und zu geklappt, in der original Version bleiben diese so wie sie vorher waren	✓
20	Versuchen in einer der Vergleichs-Navigationen einen Punkt zu löschen / ändern / hinzuzufügen	Darf nicht funktionieren	✓
21	In der zu bearbeitenden Navigation den Punkt „Waale“ bei „Säugetiere“ hinzufügen	Punkt „Waale“ wird nur bei der zu bearbeitenden Navigation hinzugefügt und nicht bei den Vergleichs-Navigationen	✓
22	Schliessen der Vergleichs-Navigation „Tier-Navigation Vorlage“		✓
23	Öffnen der Vergleichs-Navigation „Tier-Navigation Vorlage“	Der Punkt „Waale“ ist nicht vorhanden, die Vergleichsnavigation ist immernoch im ursprünglichen Zustand	✓
24	Änderungen verwerfen mittels Klick auf das Icon		✓

5.2.11 9. Test - Excel Export

Ausgangssituation:
Wenn der Test vom Kapitel 5.2.3 bzw. 5.2.4 ausgeführt wurde, so sollte diese Navigation bereits unter dem Namen „Tier-Navigation Vorlage“ vorhanden sein, ansonsten muss diese erstellt werden.

Nr.	Aktion	Erwartetes Ergebnis	Status
1	Ausgangsituation herstellen (Wenn möglich „Tier-Navigation Vorlage“ öffnen)	Säugetiere Pferde Südpolpferde Geschlecht Tiger Geschlecht Geburtsort Reptilien Amphibien	✓
2	Excel Export mittels Klick auf das Icon	Es sollte der Download für die Excel Datei „Export_Tier-Navigation-Vorlage.xls“ gestartet werden	✓
	Excel:		

	A	B	C	D
1	Säugetiere			
2		Pferde		
3			Nordpolpferde	Geschlecht
4		Tiger	Geschlecht, Geburtsort	
5	Reptilien			
6	Amphibien			

3	Referenz von „Amphibien“ in „Säugetiere“ erstellen		✓
4	Excel Export mittels Klick auf das Icon	Es sollte der Download für die Excel Datei „Export_Tier-Navigation-Vorlage.xls“ gestartet werden	✓
	Excel:		

	A	B	C	D
1	Säugetiere			
2		Pferde		
3			Nordpolpferde	Geschlecht
4		Tiger	Geschlecht, Geburtsort	
5		*Amphibien*		
6	Reptilien			
7	Amphibien			

5	Änderungen in der Navigation verwerfen		✓

5.3 Neue Navigation erstellen

5.3.1 Abgedeckte UseCases

Implizit abgedeckt:

- Keine

Vollständig abgedeckt:

- UC03: Excel Import

Teilweise abgedeckt:

- Keine

5.3.2 1. Test - Navigation per Excel Importieren

Ausgangssituation:
Es muss ein Excel File vorhanden sein, welches der vorgegebenen Struktur entspricht. Ein Beispiel-Excel, welches zum Testen importiert wird, ist auf derselben Seite verfügbar. Das Beispiel-Excel muss erst lokal auf dem Computer gespeichert werden um danach wieder importiert werden zu können.

Nr.	Aktion	Erwartetes Ergebnis	Status
1	Öffnen des Moduls „Neue Navigation erstellen“		✓
2	Eintragen eines Namens „Importierte Navigation“ sowie Auswählen des zuvor heruntergeladenen Beispiel Excel-Files.		✓
3	Klicken der Schaltfläche „Neue Navigation anlegen“	Meldung erscheint, dass die Navigation erfolgreich angelegt wurde.	✓
4		Importierte Navigation wird sogleich angezeigt. Hauptmenüpunkt 1 Unterkategorie 1 Produktegruppe A Filter A Filter B Filter C Produktegruppe B Filter D Filter E ProdukteGruppe C Filter A Filter E Unterkategorie 2 Produktegruppe D Filter A Filter B Filter C Produktegruppe E Filter D Filter E Hauptmenüpunkt 2 Unterkategorie 3 Produktegruppe F Filter A Filter B Filter C Produktegruppe G Filter D Filter E Hauptmenüpunkt 3 Unterkategorie 4 Produktegruppe H Filter A Filter B Filter C Unterkategorie 5 Produktegruppe I Filter D Filter E Unter-Unterkategorie 6 Produktegruppe J Filter A Filter E Unterkategorie7 Produktegruppe K Filter D	✓

5.4 Filter Verwaltung

5.4.1 Abgedeckte UseCases

Implizit abgedeckt:

- Keine

Vollständig abgedeckt:

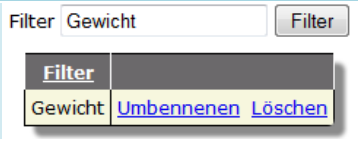
- UC06: Filter Kriterien verwalten

Teilweise abgedeckt:

- Keine

5.4.2 1. Test - Filterverwaltung Anzeige, Hinzufügen, Umbenennen, Löschen

Ausgangssituation:
Wenn der Test vom Kapitel 5.1.2 ausgeführt wurde, so sollte diese Navigation bereits unter dem Namen „Tier-Navigation Vorlage“ vorhanden sein, ansonsten muss diese erstellt werden.

Nr.	Aktion	Erwartetes Ergebnis	Status
1	Öffnen des Modules „Filter Verwaltung“		✓
2	Auswählen der Navigation „Tier-Navigation Vorlage“		✓
3	Hinzufügen eines neuen Filters mit der Beschreibung „Gewicht“	Ein neuer Filter wurde der Tabelle Filter-Tabelle hinzugefügt.	✓
4	Suchen nach dem neu hinzugefügten Filters „Gewicht“ mittels des Suchfeldes		✓
5	Über die Schaltfläche „Umbenennen“ soll „Gewicht“ nun nach „Grösse“ umbenannt werden		✓
6	Entfernen des Suchkriteriums		✓
7	Löschen des Eintrags „Grösse“	Eintrag ist nicht mehr vorhanden in der Tabelle.	✓

5.5 Benutzerverwaltung

5.5.1 Abgedeckte UseCases

Implizit abgedeckt:

- Keine

Vollständig abgedeckt:

- UC01: Benutzer verwalten CRUD

Teilweise abgedeckt:

- Keine

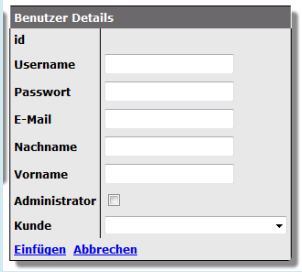
5.5.2 1. Test - Benutzerverwaltung Neue Kundengruppe

Ausgangssituation:
Es muss mit einem Administratoren-Konto eingeloggt werden, damit das Modul „Benuzterverwaltung“ sichtbar ist.

Nr.	Aktion	Erwartetes Ergebnis	Status
1	Öffnen des Modules „Benutzerverwaltung“		✓
2	Es soll eine neue Kundengruppe Namens „Mueller AG“ erstellt werden.		✓
3	Mittels der „Hinzufügen“ Schaltfläche wird die Gruppe hinzgefügt	Es wurde eine neue Kundengruppe, „Mueller AG“, in der Kundengruppen-Tabelle hinzugefügt.	✓

5.5.3 2. Test - Benutzerverwaltung Neuer Benutzer

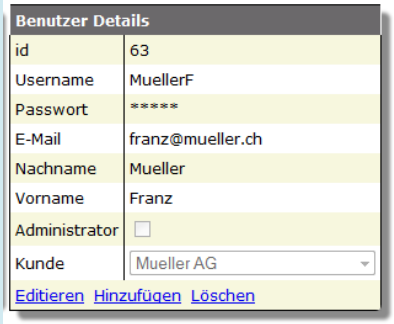
Ausgangssituation:
Es muss mit einem Administratoren-Konto eingeloggt werden, damit das Modul „Benuzterverwaltung“ sichtbar ist.

Nr.	Aktion	Erwartetes Ergebnis	Status
1	Öffnen des Modules „Benutzerverwaltung“		✓
2	Über die Schaltfläche „Neuer Benutzer“ soll ein neuer Benutzer hinzugefügt werden.	Es öffnet sich die Maske um einen neuen Benutzer zu erstellen mit leeren Feldern 	✓
3	Alle Felder sollen korrekt mit folgenden Daten gefüllt werden. Username: MuellerF Passwort: summertime E-Mail: franz@mueller.ch Nachname: Mueller Vorname: Franz Kunde: Mueller AG		✓

4	Der neue Benutzer soll über die „Einfügen“ Schaltfläche hinzugefügt werden.	Es erscheint eine Meldung, dass der Benutzer erfolgreich gespeichert wurde. Der neue Benutzer ist in der Übersichts-Tabelle ersichtlich.	✓
5	Einloggen mit dem neu erstellten Benutzer in einem neuen Fenster	Mit dem neuen Benutzer kann erfolgreich eingeloggt werden	✓

5.5.4 3. Test - Benutzerverwaltung Benutzer Verändern

Ausgangssituation:
Es muss mit einem Administratoren-Konto eingeloggt werden, damit das Modul „Benutzerverwaltung“ sichtbar ist. Wenn der Test vom Kapitel 5.1.12 ausgeführt wurde, so sollte bereits ein neuer User unter dem Username „MuellerF“ vorhanden sein.

Nr.	Aktion	Erwartetes Ergebnis	Status
1	Öffnen des Modules „Benutzerverwaltung“		✓
2	Benutzer „MuellerF“ über die „Anzeigen“ Schaltfläche auswählen	Details des Users werden angezeigt 	✓
3	Editieren des Benutzers. Die E-Mail-Adresse soll auf muellerf@gmail.com geändert werden und mittels der „Übernehmen“ Schaltfläche bestätigt werden.	Es erscheint eine Meldung, dass die Änderungen erfolgreich übernommen wurden.	✓

5.5.5 4. Test - Benutzerverwaltung Benutzer Löschen

Ausgangssituation:
Es muss mit einem Administratoren-Konto eingeloggt werden, damit das Modul „Benutzerverwaltung“ sichtbar ist. Wenn der Test vom Kapitel 5.1.12 ausgeführt wurde, so sollte bereits ein neuer User unter dem Username „MuellerF“ vorhanden sein.

Nr.	Aktion	Erwartetes Ergebnis	Status
1	Öffnen des Modules „Benutzerverwaltung“		✓
2	Benutzer „MuellerF“ über die „Anzeigen“ Schaltfläche auswählen	Details des Users werden angezeigt	✓
3	Löschen des Benutzers über die Schaltfläche „Löschen“	Es erscheint eine Meldung, dass der Benutzer erfolgreich gelöscht wurde. In der Übersichts-Tabelle ist der Benutzer nicht mehr ersichtlich.	✓

5.6 Mein Konto

5.6.1 Abgedeckte UseCases

Implizit abgedeckt:

- Keine

Vollständig abgedeckt:

- Keine

Teilweise abgedeckt:

- UC01: Benutzer verwalten CRUD
  - Eigenes Passwort ändern (Funktion die jeder Benutzer durchführen kann)

5.6.2 1. Test - Passwort ändern

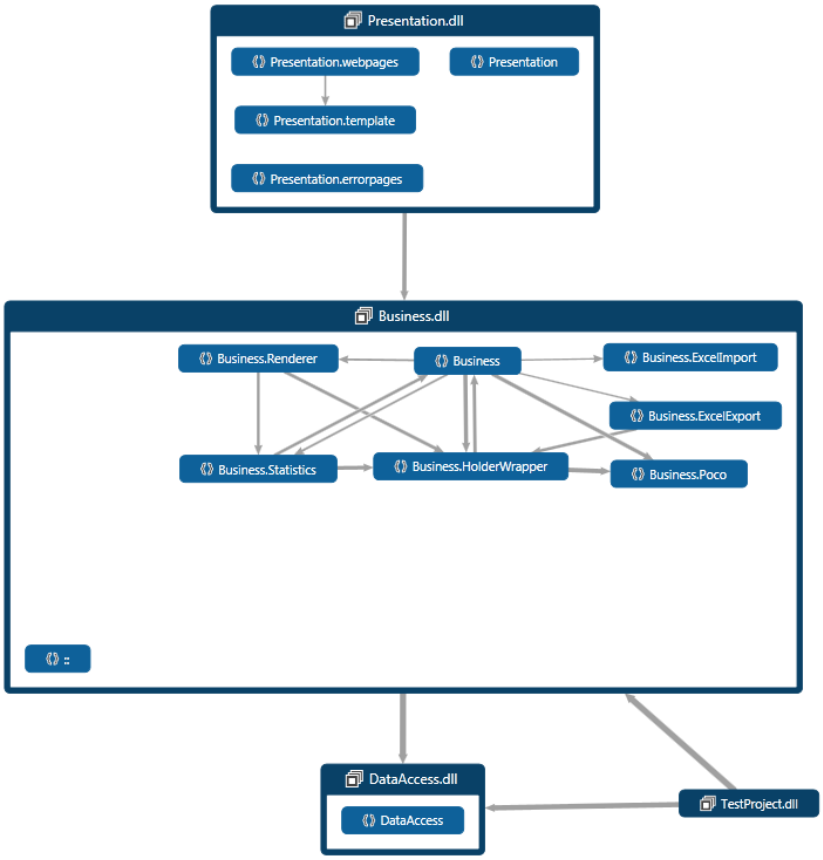
Nr.	Aktion	Erwartetes Ergebnis	Status
1	Öffnen des Modules „Mein Konto“		✓
2	Eingabe des aktuellen Passwortes sowie ein neues Passwort, welches auch korrekt bestätigt wird		✓
3	Klicken der Schaltfläche Übernehmen	Es erscheint eine Meldung, dass das Passwort erfolgreich übernommen wurde.	✓

6. Source Code Analyse

Mittels der Analyse des Source Codes, soll ein kurzer Überblick über den Code der Applikation in einer technischen Hinsicht geboten werden.

6.1 Struktur

Architektonisch wurde die Applikation in die drei Layers (Presentation, Business und Data-Access) aufgeteilt, die auch als einzelne Assemblies zur Verfügung stehen. Für die Unit-Tests wurde nochmals ein separates Projekt (TestProject.dll) erstellt.



6.2 Kennzahlen

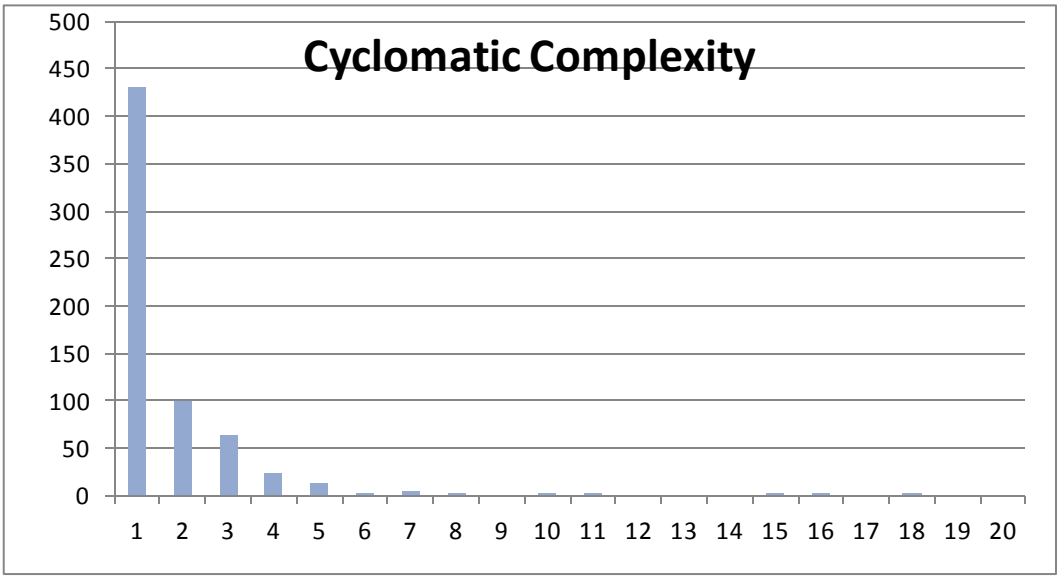
Einige Kennzahlen zur Anzahl Files bzw. Klassen, welche mit Hilfe des im Visual Studio integrierten Code Metric Plugins errechnet wurden.

Layer	File Typ	Lines of Code
Presentation	Aspx	780
	Aspx.cs (Code Behind)	325
Presentation	JavaScript	813
Presentation	CSS	394
Business	C#	1195
Data Access	C#	430

6.3 Code Metriken

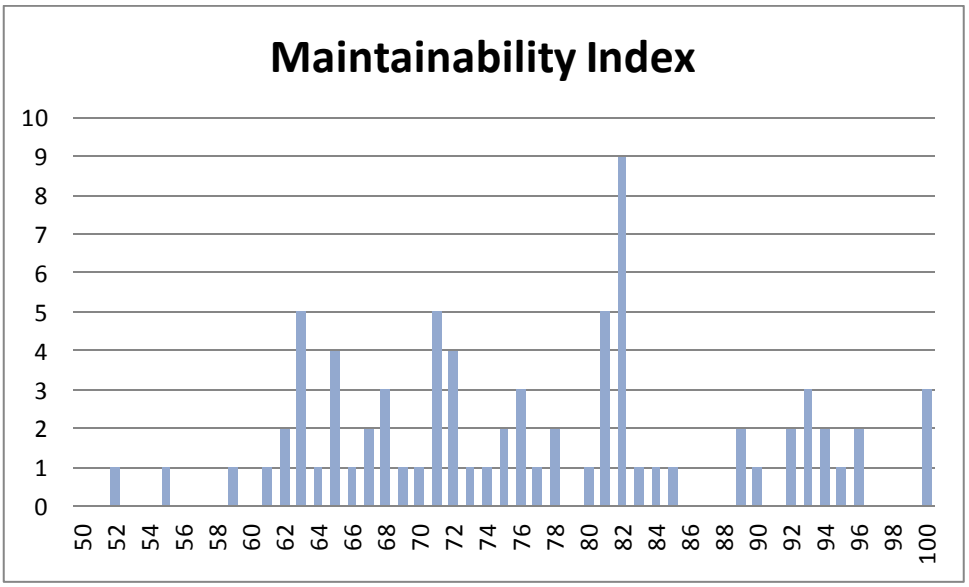
6.3.1 Cyclomatic Complexity

Der Messwert Cyclomatic Complexity beschreibt die Komplexität einer Methode anhand des Kontrollflusses. Die Komplexität einer Methode steigt mit der zunehmenden Anzahl von möglichen Kontrollflüssen, respektive mögliche Entscheidungen innerhalb der Methode. Der niedrigste mögliche Wert hierbei ist jeweils 1 bei einer leeren Methode.
Beinahe alle Methoden im gesamten Projekt halten sich hier im Bereich von einer Komplexität bis 10 auf. Erfahrungsgemäss weisen Methoden mit höheren Werten meist eine hohe oder auch zu hohe Komplexität auf und sollten bei Bedarf vereinfacht werden. Vereinzelte Ausreisser in unserem Projekt haben sich im Bereich von 10 bis 20 angesammelt und müssten bei einer Weiterführung der Entwicklung wohl. Bei der nachfolgenden Grafik handelt es sich um eine Häufigkeits-Analyse der berechneten „Cyclomatic Complexity" aller im Projekt vorhandenen Methoden.



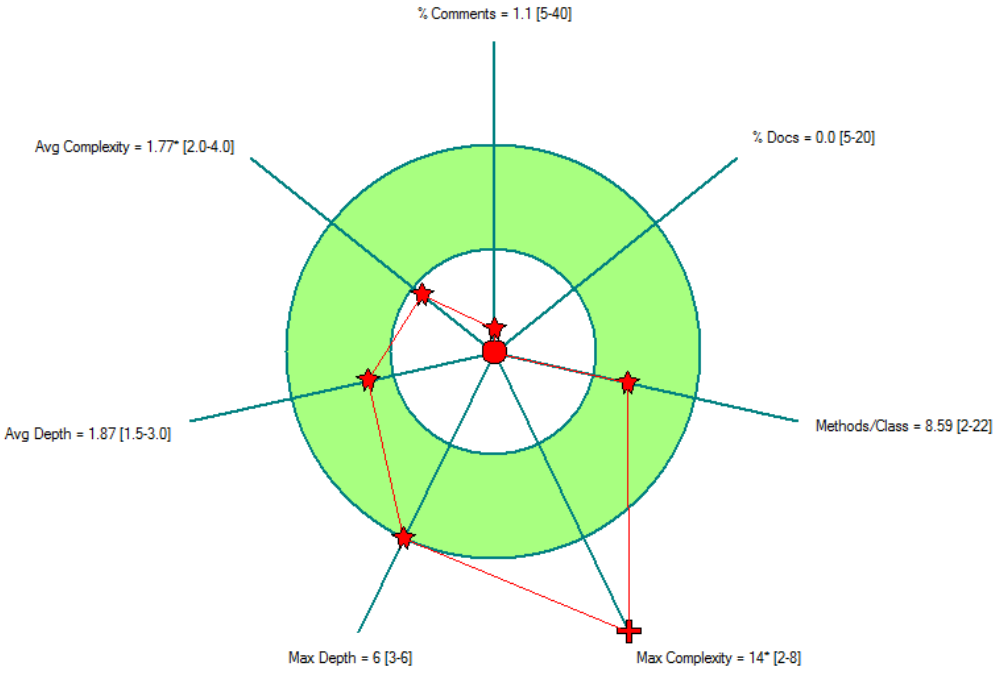
6.3.2 Maintainability Index

Der Maintainbility Index gibt Auskunft darüber, wie einfach eine Software im weiteren Verlauf unterhalten oder auch erweitert werden kann. Klassen die Werte gegen 0 haben, sind schwerer zu unterhalten als Klassen mit einem Index von 100. Der Bereich 20-100, in dem sich alle Klassen dieses Projektes befinden, wird als akzeptabel bzw. sofern der Wert gegen 100 läuft als gut befunden.
Dieser Index wird unter anderem durch die Werte „Lines of Code" und „Cyclomatic Complexity" einer Methode berechnet. Bei der nachfolgenden Grafik handelt es sich ebenfalls um eine Häufigkeits-Analyse des berechneten Maintainability-Indexes aller im Projekt enthaltenen Methoden.



6.3.3 Kiviat-Metric Graph

Mit dem Tool „SourceMonitor“ konnte noch ein Kiviat-Metric Graph erstellt werden. Es wurden dabei nur C# Files (keine JavaScript etc.) ohne Testklassen berücksichtigt.
Dieser zeigt mehrere Metriken zusammen auf und vergleicht diese dabei mit Minimum und Maximum Werten. Jede Achse stellt dabei eine eigene Metrik dar. Die Werte innerhalb des weissen Kreises sind sehr gut, im hell grünen Bereich des Kreises in Ordnung und alles noch weiter aussen ist zu hoch.
Beim Graph sieht man sofort, dass es zwei Ausreisser hat und zwar „Max Depth“ sowie „Max Complexity“. Diese Werte sind jedoch nur die absoluten Maximal- bzw. Minimalwerte, im Schnitt sind die Komplexität, sowie die Tiefe in Ordnung.
Die beiden Ausreisser sind beide auf die „NavigationStatistics“ Klasse zurück zu führen, da diese Klasse ausschliesslich dafür da ist die Statistiken zu berechnen – und diese teilweise recht komplex sein können – wurde die Klasse nicht allzu sehr unterteilt, d.h. es gibt nicht mehr Methoden als wirklich nötig.



HSR
HOCHSCHULE FÜR TECHNIK
RAPPERSWIL
FHO Fachhochschule Ostschweiz

Studienarbeit: SA eShop/PIM Prototyping

Projektrückblick

Simon Stähli (sstaehli@hsr.ch)
Patrick Zenhäusern (pzenhaeu@hsr.ch)

Inhalt

Inhalt 2
1. Einführung 3
1.1 Zweck 3
1.2 Gültigkeitsbereich 3
1.3 Referenzen 3
2. Projekt Übersicht 4
2.1 Phasen 4
2.1.1 Inception 4
2.1.2 Elaboration #1 4
2.1.3 Elaboration #2 4
2.1.4 Construction #1 4
2.1.5 Construction #2 4
2.1.6 Construction #3 5
2.1.7 Construction #4 5
2.1.8 Transition 5
3. Projektaufwände 6
3.1 Überblick 6
3.2 Grösste Abweichungen 7
3.3 Grösste Schwierigkeiten 8
3.4 Arbeitspakete 8
3.4.1 Soll- / Ist Vergleich 8
4. Persönliche Reflexion 9
4.1 Patrick Zenhäusern 9
4.2 Simon Stähli 10
4.2.1 Projektverlauf 10
4.2.2 Lessons Learned 10
4.2.3 Rückblick 10

1. Einführung

1.1 Zweck

Dieses Dokument beinhaltet einen Projektrückblick der Studienarbeit eShop/PIM Prototyping.

1.2 Gültigkeitsbereich

Gültig während des Modules „Studienarbeit Informatik" im Zeitraum des Semesters HS 12.

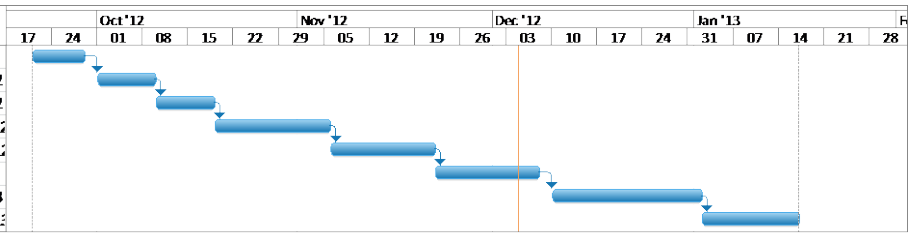
1.3 Referenzen

Redmine (http://152.96.56.49/redmine/projects/pim)
Glossar (http://152.96.56.49/redmine/projects/pim/wiki/Glossar)

2. Projekt Übersicht

In diesem Projekt wurde nach einer leicht vereinfachten Variante von RUP vorgegangen. Nachfolgend aufgelistet, finden sich die im Projekt enthaltenen Phasen, sowie deren Start- und End-Zeitpunkt.
Das Projekt wurde insgesamt über die Dauer von 14 Wochen durchgeführt.

ID		Task Mode	Task Name	Duration	Start	Finish
1			Inception	6 days	Fri 21.09.12	Fri 28.09.12
2			Elaboration #1	7 days	Mon 01.10.1	Tue 09.10.12
3			Elaboration #2	7 days	Wed 10.10.1	Thu 18.10.12
4			Construction #1	12 days	Fri 19.10.12	Mon 05.11.1
5			Construction #2	12 days	Tue 06.11.12	Wed 21.11.1
6			Construction #3	12 days	Thu 22.11.12	Fri 07.12.12
7			Construction #4	17 days	Mon 10.12.1	Tue 01.01.13
8			Transition	11 days	Wed 02.01.1	Wed 16.01.1



2.1 Phasen

2.1.1 Inception

Am Beginn des Projektes wurde ein Kickoff Meeting mit Herr Keller durchgeführt, in welchem die Problematik und die Ausgangslage des Projektes dargelegt wurden. Um einen besseren Eindruck über die Wichtigkeit einer guten Navigations-Hierarchie zu erhalten, wurden zu Beginn der Arbeit einige grosse und bekannte Online-Shops bzw. deren Navigationens-Hirarchien auf spezielle Kriterien wie Navigations-Tiefe, Kategorisierung oder auch Filter-Kriterien analysiert. Bereits am Ende dieser Phase haben wir uns auf die Technologie, ASP.NET und C#, festgelegt und somit bereits begonnen Entwicklungsumgebungen, Version-Control System sowie Projektmanagement-System einzurichten.

2.1.2 Elaboration #1

Während der ersten Elaborations-Phase wurden die genauen Systemfunktionalitäten analysiert und festgelegt, sowie deren technische Umsetzung geplant. Auch ein erster Prototyp wurde während dieser Phase entwickelt, bei welchem der Fokus auf die Kernfunktion des Bearbeitens von Listen mittels Drag& Drop, auf der Basis von JavaScript und HTML, lag. Um jedoch bereits möglichst früh auch einen Querschnitt durch die gesamte Systemarchitektur erhalten zu haben, wurde bereits zu diesem Zeitpunkt angestrebt, Daten aus einer Datenbank zu laden und für die Verwendung auf der Benutzeroberfläche vorzubereiten.

2.1.3 Elaboration #2

In der zweiten Elaborations-Phase wurde der definitive Projektplan ausgearbeitet, Arbeitspakete gebildet und entsprechende Aufwände geschätzt. Der erste Prototype wurde weiterentwickelt und als End-Ergebnis stand ein stabiles Grundgerüst der Applikation fest, welches bereits eine einfache Implementierung der Kernfunktion, das Anzeigen und Bearbeiten von Navigationshierarchien, enthalten hat.

2.1.4 Construction #1

Primärer Fokus während dieser Phase war es, die ersten wichtigen Kernfunktionalitäten zu implementieren.

- Darstellung Navigationshierarchien
- Editieren von Navigationshierarchien mittels Drag & Drop und Kontextmenü
- Login Funktionalität

2.1.5 Construction #2

Die restlichen Kernfunktionalitäten wurden während der zweiten Construction-Phase eingebaut. Der Fokus lag nach wie vor primär auf der Implementierung. Wir begannen bereits in dieser Phase Unit-

Tests für fertig gestellte Funktionalitäten zu erstellen, um eine gewisse Sicherheit zu haben falls bestehende Methoden verändert werden mussten im Laufe der Weiteren Entwicklung.

- Verwalten von Filterkriterien
- Bestehende Navigationshierarchien per Excel-File importieren

2.1.6 Construction #3

In der dritten Construction-Phase wurden die übrigen Use-Cases des Produktes abgedeckt. Unit-Tests wurden nochmals ergänzt, am GUI wurden Design-technische Anpassungen gemacht, es wurde ausgiebig getestet und Fehler wurden behoben. In dieser Phase wurde des Weiteren ein Usability Test bei einem potentiellen Kunden des Produktes, bzw. mit zwei dessen Produkte-Managern, durchgeführt. Der Usability-Test war insgesamt sehr positiv ausgefallen und es mussten nur wenige, häufig kleinere, Anpassungen an der Applikation gemacht werden. Am Ende dieser Phase stand bereits einmal ein erstes betriebsfähiges Produkt.

- Usability-Test bei der Firma OPO Oeschger
- Benutzerverwaltung für Administratoren
- GUI-Design
- Testing, Refactoring

2.1.7 Construction #4

Anfänglich wurden in dieser Phase noch Optionale Features eingeplant, diese wurden jedoch im Verlaufe des Projektes durch neu hervorgerufene Anforderungen an das Produkt ersetzt, welche alle erfolgreich implementiert werden konnten. Eine wichtige und sehr zeitraubende Aufgabe während dieser Phase waren das Ermöglichen von Referenzen in Navigationshierarchien. Dieses Feature wurde erst später im Projektverlauf als zusätzliches Feature geäussert und konnte deshalb nicht schon bereits früher umgesetzt werden. Während der Implementierung stellte sich heraus, dass dieses Feature nicht kurz und schnell eingebaut werden kann. Es stellten sich plötzlich viele neue Probleme, weil beispielsweise Referenzen auf Referenzen möglich sind und dies somit in rekursiven-Funktionen endet. Mit einem grossen Aufwand konnte es schlussendlich aber trotzdem erfolgreich implementiert werden. Ansonsten wurden viele Kleinigkeiten, welche beim Usability-Test aufgefallen waren, behoben und verbessert.

- Referenzen auf bestehende Knoten in Navigationshierarchien einfügen
- Anpassungen und Verbesserungen aufgrund des Usability-Tests
- Excel Export

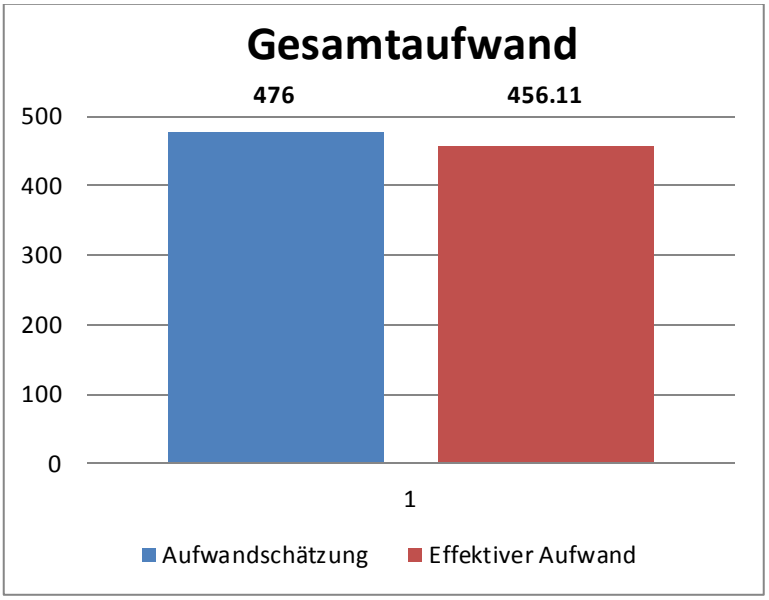
2.1.8 Transition

Während der Transition Phase ging es hauptsächlich darum, das Projekt abzuschliessen bzw. in unserem Falle die Studienarbeit. Dies bedeutete vor allem: Dokumentieren. An der Applikation mussten wir in dieser Zeit, dank einer guten Planung, nur noch kleine Änderungen vornehmen und konnten uns voll und ganz auf die Dokumentation der Arbeit konzentrieren. Ein wichtiger Anteil dieser Phase bestand auch aus dem Projektrückblick, in welchem unter anderem analysiert wurde wie die Projektdurchführung mit der Projektplanung übereinstimmte.

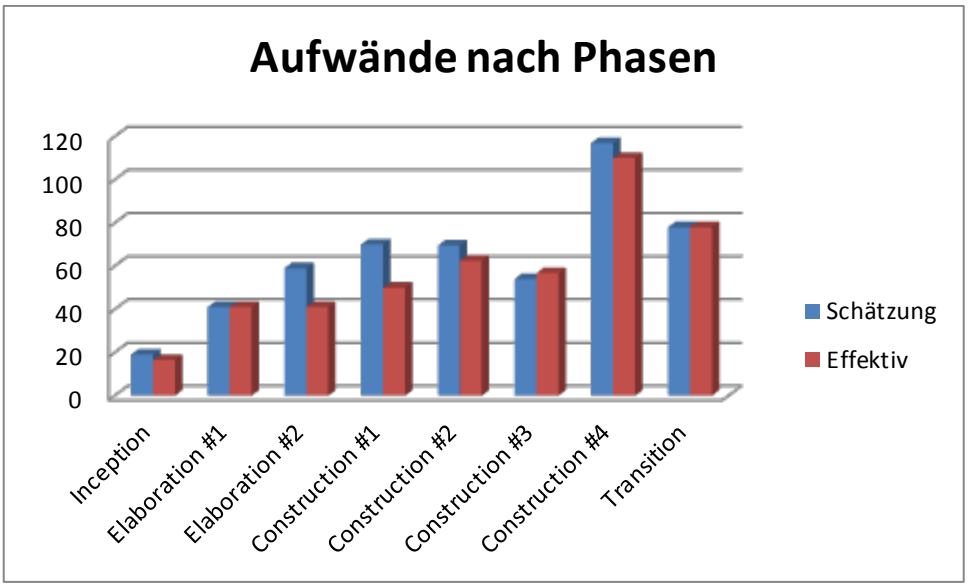
3. Projektaufwände

3.1 Überblick

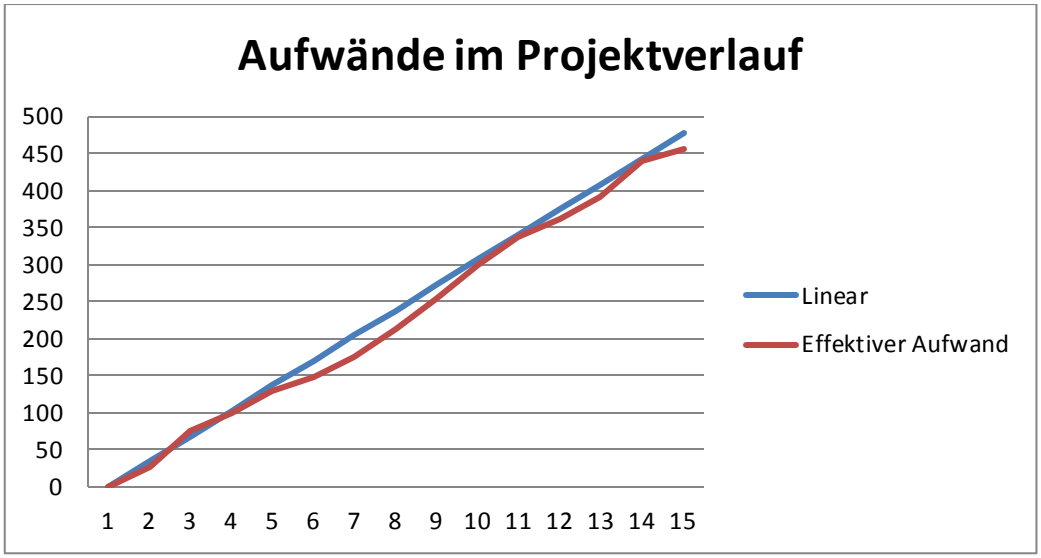
Über das ganze Projekt hinweg standen 476 Stunden zur Verfügung. Bei der Projektplanung wurde der Aufwand gleichmässig auf die verschiedenen Phasen verteilt und somit ein Schnitt von 17h pro Person und Woche angepeilt. Insgesamt wurden für das Projekt effektiv 456 benötigt. Die Differenz zu den vorgegebenen Stunden entstand dadurch, dass jeweils pro Phase eine kleine Reserve eingeplant wurde, diese aber in den meisten Fällen nicht voll ausgeschöpft wurde.



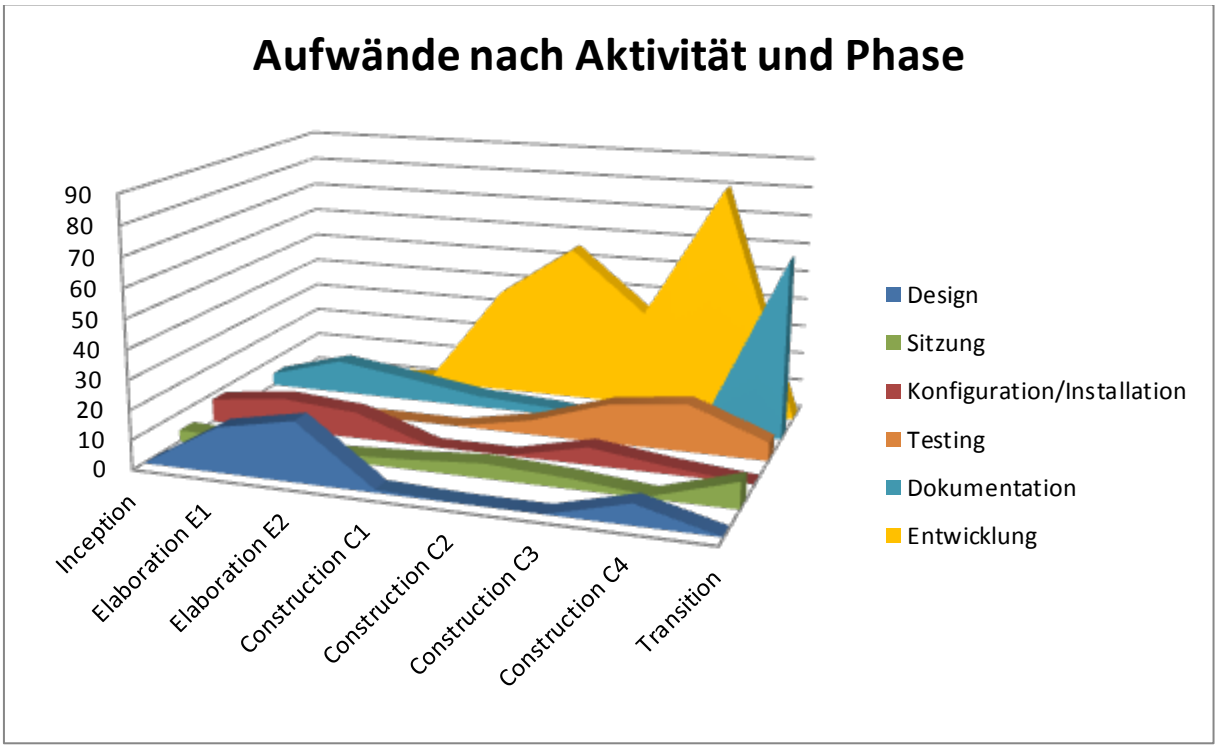
Nachfolgende Grafik stellt einen Vergleich der Aufwandschätzungen entgegen der effektiv verbrauchten Zeit pro Phase dar. In den Phasen wurde jeweils, wie bereits angemerkt, ein wenig Reserve-Zeit eingeplant, welche aber wie ersichtlich nicht immer gebraucht wurde.



Über alle Wochen hinweg wurden die Arbeiten relativ gleichmässig verteilt erledigt. Dies ist auch im nachfolgenden Diagramm ersichtlich, welches die effektiven Arbeitsaufwände über den gesamten Projektverlauf hinweg aufzeigt.



Nachfolgendes Diagrammzeigt die jeweiligen Aufwände verteilt in den einzelnen Phasen gruppiert nach der Art der Aktivität.



3.2 Grösste Abweichungen

Mehrheitlich wurden die geschätzten Aufwände pro Phase in etwa eingehalten. In der Elaboration #2 wurde der Aufwand im Voraus zu hoch geschätzt, bzw. wir kamen schneller Voran als angenommen, dies hat vor allem weil bereits viele Dokumente aus dem SE2 Projekt, zumindest als Vorlage, verwendet werden konnten und dadurch Zeit gespart werden konnte.

Auch in der ersten Construction-Phase kamen wir schneller voran als erst angenommen, dies führen wir darauf zurück, dass wir uns sehr schnell in die eingesetzte Technologie eingearbeitet haben und uns zum Teil Vorwissen im Bereich von Web-Applikationen zu Gunsten kam.
In der dritten Construction-Phase hatten wir einen leichten Mehraufwand, dies kann auf das Vorbereiten der Usability-Tests zurückgeführt werden. Die Applikation wurde in dieser Zeit erstmals auf einem Remote-Server installiert und konfiguriert. Hierbei kamen einige kleinere Probleme und Fehler auf uns zu, die wir zu diesem Zeitpunkt so nicht erwartet hatten. Dies hat uns jedoch wiederum einige Arbeit von der letzten Construction-Phase abgenommen, bei welcher diese Probleme spätestens aufgetaucht wären.
Alles in allem, wurden bei der Aufwand-Schätzung eher etwas grosszügige Annahmen getroffen.

3.3 Grösste Schwierigkeiten

Bereits im Voraus war uns, durch frühere Erfahrungen, bewusst, dass es durchaus schwer sein kann mit JavaScript saubere Lösungen zu entwickeln, da immer wieder Fehler auftreten können, die einem nicht ganz erklärbar scheinen und das Debugging nur mit zusätzlichen Tools möglich ist. Da wir uns jedoch von Anfang an auf das JavaScript Framework jQuery verlassen haben, haben sich viele dieser Probleme jedoch noch in Grenzen gehalten. Dennoch stellten sich teilweise kleinere Aufgaben als Hürden heraus, an denen man immer wieder viel Zeit, auch mit der Recherche im Internet, verlor.

3.4 Arbeitspakete

Arbeitspakete wurden detailliert im Projekt-Management-Tool Redmine verwaltet. Alle Arbeitspakete wurden soweit verfeinert, sodass deren Aufwand auf maximal 8 Stunden (natürlich gibt es immer Ausnahmen!) geschätzt werden konnte. Die Zeiterfassung wurde ebenfalls über Redmine erledigt, somit konnte am Ende des Projektes auch eine Auswertung bezüglich der Differenz von geschätzten und effektiv aufgewendeten Stunden getätigt werden.
In der Datei „Arbeitspakete.pdf“ findet sich eine Liste mit allen Arbeitspaketen. Diese ist nach Phasen, sowie nach „Ticketnummern“ (d.h. Erfassungszeitpunkt) geordnet. In der Liste sind auch die geschätzten, sowie die tatsächlichen Stunden ersichtlich.

3.4.1 Soll- / Ist Vergleich

Der Soll- / Ist Vergleich bezieht sich auf das im selben Ordner abgelegte Dokument „Arbeitspakete.pdf“ und zeigt den Vergleich der Aufwand-Schätzung gegenüber dem effektiven Zeitaufwand in Stunden und jeweils prozentual auf.
Insgesamt haben wir uns um 12.91% verschätzt, d.h. wir haben mehr eingeplant als wir gebraucht haben. Nach Herr Keller sollte man sich bei einem Projekt in dieser Grösse um nicht mehr als 20% verschätzen, mit unserer Zahl liegen wir deutlich darunter und sind deshalb zufrieden mit dem Ergebnis.
Die grössten Abweichungen sind zumeist bei kleinen Paketen, d.h. Pakete mit einem Aufwand von ca. 1 -2 Stunden. Dabei ergibt sich natürlich rasch eine Abweichung von 100%, z.B. wenn statt einer Stunde, zwei Stunden gebraucht werden. Diese Abweichungen beachten wir daher als nicht weiter tragisch.
Wirklich schlimm verschätzt haben wir uns im Prinzip bei folgenden Paketen:

- #31 – Klassenstruktur erstellen
  - Zeiten: 6h geschätzt, 1h effektiv → 83.33% Abweichung
  - Grund: Die Klassenstruktur in diesem Paket war auf die Datenbanklassen bezogen und diesen wurden durch das Entitiy-Framework automatisch generiert
- #34 – Navigationspunkte verschiebbar
  - Zeiten: 10h geschätzt, 2h effektiv → 80% Abweichung
  - Grund: Dank dem Plugin „jsTree“ wurde uns der Hauptteil dieser Arbeit abgenommen

4. Persönliche Reflexion

4.1 Patrick Zenhäusern

Ich muss zugeben, dass ich zu Beginn der Arbeit etwas verunsichert war. Zum einen hatte dies mit der Aufgabe selbst zu tun, da ich mir bis zum Kickoff-Meeting nicht wirklich vorstellen konnte, was überhaupt gemacht werden soll, zum anderen damit, dass mein Projektpartner die Arbeit mit asp.net bzw. C# durchführen wollte und ich vorher noch nie damit gearbeitet habe.
Diese Verunsicherung konnten jedoch relativ schnell beseitigt werden. Bereits nach dem ersten Meeting hatte ich ziemlich klar vor Augen, was unsere Aufgabe ist und auch wie wir diese in etwa lösen könnten. In asp.net konnte ich mich während der Inception, sowie der Elaboration 1 sehr gut einarbeiten und fand mich nach kurzer Zeit relativ gut zu Recht. Es machte sich aber natürlich deutlich bemerkbar, dass ich noch nie mit asp.net gearbeitet habe, so gab es vor allem am Anfang einige Probleme mit dem Ado.Net Framework, sowie mit LINQ. Viele Codeteile musste ich mehrmals umschreiben, da ich immer wieder eine andere, bessere Art entdeckte um etwas umsetzen. Dies war auch ein Grund dafür, dass die Architektur vermehrt geändert werden musste. Ein weiterer, stärkerer Grund dafür, dass die Architektur immer wieder angepasst werden musste, waren die ständig wachsenden Anforderungen. Mit jedem Meeting mit dem Betreuer bzw. Auftraggeber (was bei uns ein und dieselbe Person war) kamen neue Wünsche und Features hinzu, welche noch implementiert werden mussten. Einige dieser Features (wie die Undo-Funktion und vor allem die Referenzen) hatten relativ grosse Auswirkungen auf das Verhalten des Produktes und zogen deshalb auch sehr grosse, oft auch architektonische Änderungen mit sich. Ich will hier jedoch anmerken, dass diese uns nicht übermässigen Stress oder Druck erbrachten, da der Umfang Anfangs relativ knapp & klein definiert war und die Features durchaus sinnvoll sind. Da ich jedoch auch schon bei anderen Projekten (in der Wirtschaft) mit gearbeitet hatte, war ich darüber nicht sonderlich erstaunt und hatte auch keine grösseren Probleme damit. Ein grosser „Steinewerfer“ war auch der ganze JavaScript teil, ich bin überhaupt kein Fang davon, es wird einfach sehr schnell unübersichtlich und ist mühsam zum Debuggen. Gewisse Fehler habe ich am Anfang gar nie gesehen, diese kamen erst beim detaillierten Testen zum Vorschein. Das Problem war bzw. ist in meinen Augen, dass JavaScript viel zu „Wischi-Waschi“ ist, man hat zu viele Freiheiten und es wird zu wenig vorgegeben (keine Type-Safety, keine Compile-Errors) etc. Im Nachhinein denke ich auch, dass wir den JavaScript teil etwas zu wenig genau geplant hatten, vielleicht hätten wir uns dadurch einiges Kopfzerbrechen & einige Stunden Arbeit ersparen können.
Etwas nach der Hälfte des Projektes hatten wir einen Usability-Test bei einer Firma (OPO Oeschger), welcher (sehr) gut heraus kam. Ab diesem Zeitpunkt hatte ich ein sehr gutes Gefühl bei der Sache, denn ich wusste dadurch, dass unsere Designentscheide richtig waren.
In den Endphasen kam dann vor allem das Thema Dokumentation und Systemtests zum Tragen. Ich muss ehrlich sagen, dass ich kein Fan vom Dokumentieren bin und mir diese Phase deshalb am wenigsten zusagte. Auch die Systemtests waren eher mühsam, bereits nach kurzer Zeit bemerkte ich, wie gross die Testprotokolle wurden. Ich persönlich finde so etwas weder übersichtlich, noch attraktiv zum Lesen. Kurzerhand schlug ich meinem Partner vor für die wichtigsten Testszenarien Videos mit Kommentaren zu machen, was dieser als eine gute Idee empfand. Ich bin froh, dass die Videos vorhanden sind, denn zum einen leisten sie einen riesen Beitrag zum Verständnis der Tests und zum anderen wird durch sie der Funktionsumfang des Produktes relativ detailliert aufgezeigt. Schlussendlich denke ich, dass wir ein gutes Ergebnis erzielt haben und ein voll funktionsfähiges Produkt ausliefern können.
Die Arbeit insgesamt hat mir meistens Freude bereitet. Was mich persönlich etwa gestört hat, war die Zeiterfassung: Ich bin kein Fan davon, für alles immer Stunden eintragen zu müssen, viel wichtiger ist für mich das Ergebnis, dass erzielt wird! Positiv hingegen war die Zusammenarbeit mit meinem Partner, er hat seine Aufgaben immer Termin gerecht und gut erledigt, so wie ich das aus früheren Projekten mit ihm bereits kannte. Auch mit unserem Betreuer verstanden wir uns gut und konnten – so denke ich zumindest – seinen Wünschen und Anforderungen gut entgegenkommen.

4.2 Simon Stähli

4.2.1 Projektverlauf

Aufgrund der Projekt-Ausschreibung, wusste ich nicht wirklich genau was uns bei dieser Aufgabe erwartet. Fest stand jedoch auch bereits bei Beginn dieser Arbeit, dass uns sehr viel Freiraum bei der Umsetzung zur Verfügung stand. Nach dem Kickoff-Meeting mit Herrn Dr. Daniel Keller wurde uns die Aufgabe dann doch etwas klarer. Die genauen Anforderungen an die Software waren jedoch nicht bereits zu Beginn ganz klar, sondern wachsen ein wenig mit dem ganzen Projekt mit. Dies hat sich zum Teil doch auch auf die Architektur und Qualität der Applikation ausgewirkt und konnte nur durch Mehraufwand wieder bereinigt werden. Auch wenn dies in der realen-Arbeitswelt sicherlich auch immer wieder, oder zumindest häufiger als gewollt, passiert, sehe ich dies bei einer Projektarbeit als nicht passend an. Bereits zu Beginn des Projektes wurde ein detaillierter Projektplan, mit einzelnen Arbeitspaketen bis zu 8h, erstellt. Grösstenteils konnten wir uns an diesen Projektplan halten, auch wenn einige Features zwischen Phasen verschoben werden mussten. Durch ein kontinuierliches Arbeiten über alle Wochen hinweg, blieb uns auch gegen das Ende der grosse Zeitdruck erspart und die genaue Planung hat sich ausbezahlt.

4.2.2 Lessons Learned

Gerade im Bereich des Unit-Testings wurde mir klar, wie schwierig es ist, saubere und effektive Unit-Tests zu schreiben. Beispielsweise, greifen viele Funktionen dieser Applikation über das Entity Framework (OR-Mapper) auf die Datenbank zu. Um hier nun Unit-Tests im klassischen Sinne zu verwenden, müsste ein Mocking-Framework verwendet werden um das Entity Framework zu mocken. Wir haben uns jedoch der einfachheitshalber entschieden, dies Aufgrund des nicht wirklich vorhandenen Mehrwertes nicht zu machen. Somit stellen unsere Tests keine klassischen Unit-Tests im engen Sinne dar, sondern mehrheitlich Integrations-Tests.
Des Weiteren war dies mein erstes richtiges Projekt mit C#, gerade in diesem Bereich durfte ich viel neues, wie beispielsweise das Entity Framework oder auch LINQ, dazulernen.

4.2.3 Rückblick

Dass sich bei dieser Arbeit eine Web-Applikation als am besten geeignet darstellte, ist aus meiner Sicht im Nachhinein vor- und nachteilig. Aus meiner früheren Tätigkeit, hatte ich bereits viel Erfahrung im Bereich von Web-Applikationen. Diese Erfahrung kam mir doch immer wieder zu Gunsten, jedoch viel - gerade im Bereich von HTML, JavaScript und CSS - der Aspekt des Entdecken und Erforschen von neuen Technologien gänzlich weg. Welches doch ein grosser Teil der Motivation bei einer solchen Arbeit sein sollte aus meiner Sicht. Da gerade JavaScript und CSS teils relativ mühsam sein kann, hat dies zwischendurch auf die Nerven geschlagen.
Da ich beim SE2 Projekt, aufgrund eines Auslandsemesters nicht anwesend war, stellte die Studienarbeit meine erste grössere Projektarbeit an der HSR dar. Mit der Teamarbeit hat es aus meiner Sicht sehr gut geklappt und sich auch im Hinblick auf die Bachelor-Arbeit bewährt. Die Zusammenarbeit bei diesem Projekt ist sicherlich eines der positiven Erlebnisse, die ich ins nächste Semester nehme. Auch die Sitzungen mit unserem Betreuer, Herrn Dr. Keller, haben stets reibungslos geklappt.
Während des Projektes, wurden alle Aufgaben im Projekt-Management-Tool Redmine, auf der Basis einer Web-Applikation, verwaltet. Dies stellte sich als sehr effektiv heraus, da somit immer beide Team-Mitglieder auf dem aktuellsten Stand der Dinge waren und auch die getätigte Arbeit gleich vorneweg auf die einzelnen Aufgaben gebucht werden konnte.
Ein wenig Schade finde ich es, dass doch sehr viele Dokumente erstellt werden müssen, die gar nicht gross von Belangen sind und wohl kaum noch je einmal angesehen werden. Da fehlte mir doch ein wenig der Bezug zur Praxis.

Arbeitspakete

#	Ticket Beschreibung	Schätzung	Effektiv	Differenz	Abweichung
	Inception				
2	Navigationen von bestehenden WebShops analysieren	4.00	3.00	-1.00	-25.00%
3	KickOff Sitzung	2.00	2.00	0.00	0.00%
4	Clients / Entwicklungsumgebung einrichten	10.00	9.00	-1.00	-10.00%
5	1. Sitzung vom 21.09.2012, 15:00	2.00	2.00	0.00	0.00%
6	Template für Doku erstellen	1.00	0.50	-0.50	-50.00%
		19.00	**16.50**		
	Elaboration E1				
1	Server einrichten	10.00	7.50	-2.50	-25.00%
7	Anforderungsspezifikation	6.00	6.00	0.00	0.00%
8	2. Sitzung, 15:10 Uhr, Mensa	2.00	3.00	1.00	50.00%
9	Planung	5.00	4.75	-0.25	-5.00%
16	UseCases	5.00	7.50	2.50	50.00%
17	"Dummy" Prototyp - Ausprobieren	10.00	7.75	-2.25	-22.50%
23	Risikoanalyse	1.00	1.00	0.00	0.00%
26	MySQL nicht von extern erreichbar	1.00	3.00	2.00	100.00%
40	Beispieldaten in Datenbank einfüllen	1.00	0.50	-0.50	-50.00%
		41.00	**41.00**		
	Elaboration E2				
18	Funktionierender / Richtiger Prototype	10.00	6.70	-3.30	-33.00%
19	Projektplan	6.00	4.00	-2.00	-33.33%
20	Glossar	3.00	1.00	-2.00	-66.67%
21	Software Architektur	7.00	6.50	-0.50	-7.14%
22	Domain- & Klassendiagramm	10.00	5.60	-4.40	-44.00%
25	3. Sitzung, 15.10 Uhr	2.00	2.00	0.00	0.00%
27	Datenbankstruktur erstellen	2.00	2.50	0.50	25.00%
39	Einarbeitung in asp.NET / Visualstudio	10.00	9.00	-1.00	-10.00%
55	Domainanalyse	6.00	2.00	-4.00	-66.67%
56	WireFrames erstellen	3.00	2.00	-1.00	-33.33%
		59.00	**41.30**		
	Construction C1				
29	Benutzer-Authentifizierung				
31	Klassenstruktur erstellen	6.00	1.00	-5.00	-83.33%
32	Navigation darstellen	2.00	1.00	-1.00	-50.00%
33	Navigation bearbeiten können	14.00	3.50	-10.50	-75.00%
34	Navigationspunkte verschiebbar	10.00	2.00	-8.00	-80.00%
52	HTML-Page Struktur	10.00	8.50	-1.50	-15.00%
60	Session Management	4.00	1.50	-2.50	-62.50%
75	Login-Maske	1.00	1.00	0.00	0.00%
76	sichere Überprüfung der Login-Daten anhand in der ...	2.00	1.50	-0.50	-25.00%
78	Navigationen per Drag&Drop & Context Menü im Brows...	4.00	3.00	-1.00	-25.00%
79	Änderungen an Navigationsstruktur im Hintergrund t...	5.00	3.00	-2.00	-40.00%
80	Änderungen an Navigationsstruktur persistent in de...	5.00	4.50	-0.50	-10.00%
81	Code Review 1	4.00	2.00	-2.00	-50.00%
85	Bearbeiten der Anmerkungen zu EoE	4.00	4.00	0.00	0.00%
87	Persistentes Speichern funktioniert in Chrome nich...	1.00	1.00	0.00	0.00%
88	EntityFramework löscht Kinder nicht bei navigation	2.00	1.50	-0.50	-25.00%
89	Rückgabewert von allen Operationen	1.00	0.66	-0.34	-34.00%
90	Ajax Timeout / Failure	1.00	1.50	0.50	50.00%

91	Guard bei ViewOperations vereinfachen	1.00	0.40	-0.60	-60.00%
92	Config File	1.00	0.50	-0.50	-50.00%
93	4. Sitzung, 15.10 Uhr	2.00	1.20	-0.80	-40.00%
94	UnitTests für Operationen & Login Funktionen	6.00	5.00	-1.00	-16.67%
95	Toolbox vervollständigen	2.00	1.75	-0.25	-12.50%
		88.00	**50.01**		

Construction C2

24	UsabilityTest planen	2.00	2.50	0.50	25.00%
35	Filterkriterien an Navigationspunkten zuweisen kön...	10.00	10.75	0.75	7.50%
36	Filterkriterien verwalten	10.00	5.50	-4.50	-45.00%
37	Statistiken zur Navigation	6.00	5.75	-0.25	-4.17%
41	Excel Import				
59	Webnavigation/menu implementieren	3.00	2.25	-0.75	-25.00%
61	Navigation erstellen	2.00	1.50	-0.50	-25.00%
62	Vorhande Excel analysieren	1.00	1.00	0.00	0.00%
63	Format für Excel definieren	2.00	1.25	-0.75	-37.50%
64	Excel Parser	8.00	7.50	-0.50	-6.25%
65	Excel Import: Navigation & Filter anlegen	3.00	5.00	2.00	66.67%
66	Excel Import: Maske / Verwaltung dafür erstellen	4.00	3.00	-1.00	-25.00%
82	Code Review 2	5.00	4.00	-1.00	-20.00%
96	Id für eine neue Navigationitem Entry stimmt nicht...	6.50	6.00	-0.50	-7.69%
115	Refactoring / Bugfixes	5.00	5.50	0.50	10.00%
118	5. Sitzung 15.00 Uhr	2.00	1.00	-1.00	-50.00%
		69.50	**62.50**		

Construction C3

30	Benutzerverwaltung implementieren				
38	GUI-Design	8.00	9.75	1.75	21.88%
57	Usability Test durchführen	5.00	9.00	4.00	80.00%
58	Usability Test auswerten	5.00	4.50	-0.50	-10.00%
67	Passwortänderungsmöglichkeit für Benutzer	4.00	3.50	-0.50	-12.50%
68	Benutzerverwaltung: Benutzer verwalten (Administra...	4.00	5.00	1.00	25.00%
69	Benutzerverwaltung: Benutzer erstellen (Administra...	2.00	3.00	1.00	50.00%
77	Berechtigungskontrolle für Users/Administratoren	6.00	5.00	-1.00	-16.67%
83	Code Review 3	5.00	2.50	-2.50	-50.00%
119	Globaler Logger mit Log-file	5.00	4.50	-0.50	-10.00%
127	Wenn es keinen einzigen Filter gibt funktioniert z...				
128	IIS Server einrichten und Produkt dort publishen	3.00	4.00	1.00	33.33%
129	Usability Tests vorbereiten	6.00	6.00	0.00	0.00%
		53.00	**56.75**		

Construction C4

50	Mehrere Navigationen gleichzeitig anzeigen	7.00	7.00	0.00	0.00%
54	Unit Tests	15.00	14.75	-0.25	-1.67%
86	Metriken dokumentieren	4.00	3.00	-1.00	-25.00%
116	Undo Funktion von Navigations-Baum Operationen	10.00	9.00	-1.00	-10.00%
117	Kopie-Verknüpfungen / Refernezen	20.00	20.50	0.50	2.50%
120	Testdaten definieren	4.00	1.75	-2.25	-56.25%
121	HTML Renderklasse erstellen	1.00	1.50	0.50	50.00%
122	Ausschneiden, Kopieren & Einfügen anstatt nur Drag...	2.00	3.50	1.50	75.00%
123	Navigation unter neuem Namen speichern (Versionen...	4.00	6.25	2.25	56.25%
124	Filterkriterien direkt hinter Menüpunkten aufliste...	6.00	3.75	-2.25	-37.50%
125	Anzeigen falls es ungespeicherte Änderungen gibt b...	1.00	1.20	0.20	20.00%
126	(Excel) Export ermöglichen	6.00	5.00	-1.00	-16.67%
132	Umbenennen mit Doppelklick	1.00	0.25	-0.75	-75.00%
133	Direkte Weiterleitung zu Navigationverwalten nach ...	2.00	1.50	-0.50	-25.00%

134	Statistiken zur Länge von Navigationspunktenbeschr...	1.00	1.00	0.00	0.00%
135	Statistik: Anzahl Filterkriterien pro Stufe anzeig...	1.00	0.75	-0.25	-25.00%
136	Filterverwaltung: Suche für bestehende Filter	2.00	2.00	0.00	0.00%
139	Filter- und Benuzterverwaltung: Sprache überall au...	1.00	0.25	-0.75	-75.00%
142	Möglicher Fehler: "Service unavailable" nachdem ma...				
143	Hinweis / Sperren von Navigationen wenn diese bere...	4.00	3.50	-0.50	-12.50%
144	Abklärung: Baum-Div am Anfang ggf. auf "display:n...	1.00	0.25	-0.75	-75.00%
145	Toolbox erweitern um alle Kontextmenübefehle / Tas...	3.00	3.00	0.00	0.00%
146	Hinzufügen Button für Filterkriterien (nicht nur E...	0.50	0.50	0.00	0.00%
147	Suche/Filtersuche ganzes Wort markieren bei erneut...	0.75	0.60	-0.15	-20.00%
148	Input-Felder Schriftart und Feldlänge bei Hinzufüg...	1.50	1.00	-0.50	-33.33%
149	Filterverwaltung: Rückmeldung bei CRUD	1.50	1.00	-0.50	-33.33%
150	Grosses Refactoring	10.00	9.25	-0.75	-7.50%
151	6. Sitzung 15.00 Uhr	2.00	1.00	-1.00	-50.00%
153	Alle Holder müssen sich gegnsetiig kennen, Logik i...	2.00	2.00	0.00	0.00%
154	Sesionvariable pro Navigation einführen	1.00	2.50	1.50	100.00%
167	Connection-String konfigurierbar machen	1.00	2.50	1.50	100.00%
		116.25	**110.05**		
	Transition				
28	SA-Plakat erstellen	5.00	4.00	-1.00	-20.00%
43	Dokumentation überarbeiten	10.00	11.00	1.00	10.00%
44	Testen (Systemtest /Integrationstest)	10.00	9.25	-0.75	-7.50%
45	Testprotokoll	4.00	6.25	2.25	56.25%
46	Abschluss Abgabe				
47	Schlussbericht	8.00	8.50	0.50	6.25%
84	Code Review 4	5.00	5.00	0.00	0.00%
152	Dokumentation der Arbeit	8.00	8.75	0.75	9.38%
157	Kurzfassung	4.00	3.00	-1.00	-25.00%
158	Abgabe der gesamten Arbeit				
159	Abgabe CD	2.00	2.00	0.00	0.00%
160	Aufbereitung Dokumentation	8.00	10.00	2.00	25.00%
163	Installation/Konfiguration Anleitugn	10.00	6.75	-3.25	-32.50%
168	Management Summary	4.00	3.50	-0.50	-12.50%
		78.00	**78.00**		
	Total	**523.75**	**456.11**	-67.64	-12.91%

Regeln für Abweichungs-Indikator:

Grün: bis 35% Abweichung
Orange: bis 35-60% Abweichung
Rot: ab 60% Abweichung

HSR
HOCHSCHULE FÜR TECHNIK
RAPPERSWIL
FHO Fachhochschule Ostschweiz

Studienarbeit: SA eShop/PIM Prototyping

Installation / Konfiguration

Simon Stähli (sstaehli@hsr.ch)
Patrick Zenhäusern (pzenhaeu@hsr.ch)

Inhalt

Inhalt 2
1. Einführung 3
1.1 Beschreibung 3
2. Datenbank 4
2.1 Version 4
2.2 Installation 4
2.2.1 Voraussetzung 4
2.2.2 Durchführung 4
3. Web-Applikations-Server 5
3.1 Allgemein 5
3.2 Microsoft Internet Information Services (IIS) 5
3.2.1 Voraussetzung 5
3.2.2 Durchführung 6
4. Microsoft Visual Studio 8
4.1.1 Voraussetzung 8
4.1.2 Durchführung 8
5. Generelle Applikationseinstellungen 9
5.1 Installation des MySql-Connector 9
5.2 Installation des 2007 Office System Driver: Data Connectivity Components 9
5.3 Anpassen der Datenbankeinstellungen 9
5.4 Excel Import 10
5.5 Log-File 10
6. Neue Version aus Visual Studio publizieren 11
6.1 Voraussetzungen 11
6.2 Durchführung 11
7. Sicherheit 12

1. Einführung

1.1 Beschreibung

In diesem Dokument soll aufgezeigt werden, welche Schritte benötigt werden um das Produkt zu installieren und zu konfigurieren. Folgt man den Schritten dieser Anleitung, so ist das Produkt danach voll funktionsfähig und kann benutzt werden.

2. Datenbank

2.1 Version

Wie bereits aus anderen Dokumenten bekannt sein sollte, wird als Datenbank-System MySQL in der Version 5.5 verwendet. Wir empfehlen diese Version zu verwenden, allerdings ist das Produkt garantiert abwärtskompatibel bis zu Version 5.0. Das Produkt sollte auch mit tieferen Versionen funktionieren, allerdings wurde dies nicht getestet.

2.2 Installation

2.2.1 Voraussetzung

Es wird davon ausgegangen, dass bereits ein funktionsfähiger MySQL Server vorhanden ist.

2.2.2 Durchführung

Vorabhinweis: Alle SQL-Skripte befinden sich im Verzeichnis *„sql“*, welches sich im gleichen Ordner wie diese Dokumentation befindet.

1. Anmelden am MySQL Server mit einem Benutzer mit genügend Rechten.
2. Ausführen des SQL-Skriptes *„pim.sql“*
    a. Dieses Skript erstellt die Datenbank *„pim“* mit allen benötigten Tabellen und den notwendigen Basis-Daten für das Produkt.
    b. Es wird dabei **der Standardbenutzer *„pim“* mit dem Passwort *„1234“*** und der Gruppenzugehörigkeit *„PIM Gruppe“* erstellt. Dieser Benutzer hat Administratorrechte. Mit diesem können Sie sich später an der Applikation anmelden.
3. *Optional*: Ausführen des SQL-Skriptes *„grant.sql“*
    a. Dieses Skript erstellt einen MySQL **Benutzer *„pimdbuser“* mit dem Passwort *„1234“*** und gewährt diesem alle Rechte auf die Datenbank *„pim“*, welche durch das Skript *„pim.sql“* erstellt wurde. Falls Sie einen eigenen DB-Benutzer erstellen wollen oder bereits einen Benutzer haben, so können Sie das Skript auch weglassen und Ihren Benutzer in der Applikation entsprechend eintragen (siehe dazu Kapitel 5.3).

3. Web-Applikations-Server

3.1 Allgemein

Um das Produkt nutzen zu können muss natürlich die Applikation auch in irgendeiner Form installiert werden. Im Folgenden werden zwei Möglichkeiten aufgezeigt wie dies gemacht werden kann. Auf der einen Seite mit einem IIS-Server (produktiver Betrieb) und auf der anderen Seite mittels des VisualStudios (zu Testzwecken).

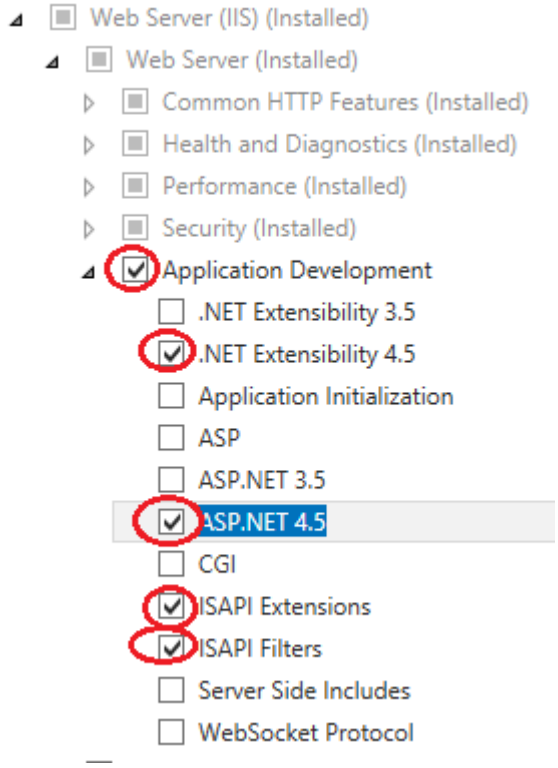
3.2 Microsoft Internet Information Services (IIS)

3.2.1 Voraussetzung

Installierter IIS-Server in der Version 7.0 oder höher. Eine bereits vorhandene Webseite (Default Website genügt).

3.2.1.1 Application Development / Applikationsentwicklungsrolle

Damit ASP.net Applikationen laufen ist es zwingend notwendig, dass beim IIS die ASP.NET 4.5 Features installiert werden. Dies kann über den Windows Server Manager oder aber auch über das Control Panel und die Funktion „Turn Features on or off“ erledigt werden. Auf folgendem Screenshot ist ersichtlich welche Features dies sind:



3.2.1.2 WebDeploy

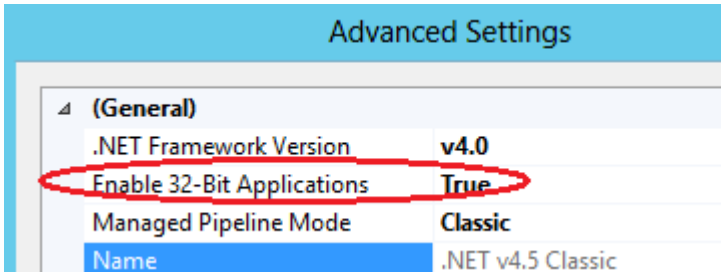
Damit man die Applikation im IIS importieren kann, ist es wichtig, dass die Erweiterung „WebDeploy 3.0“ installiert ist.
Für die Installation gibt es zwei Möglichkeiten:

1. Download von http://www.iis.net/downloads/microsoft/web-deploy
2. Installation einer bereits heruntergeladenen Datei im Verzeichnis „iis“ unter dem Namen „WDeploy.exe“

Der Installations-Prozess an sich ist selbst erklärend.

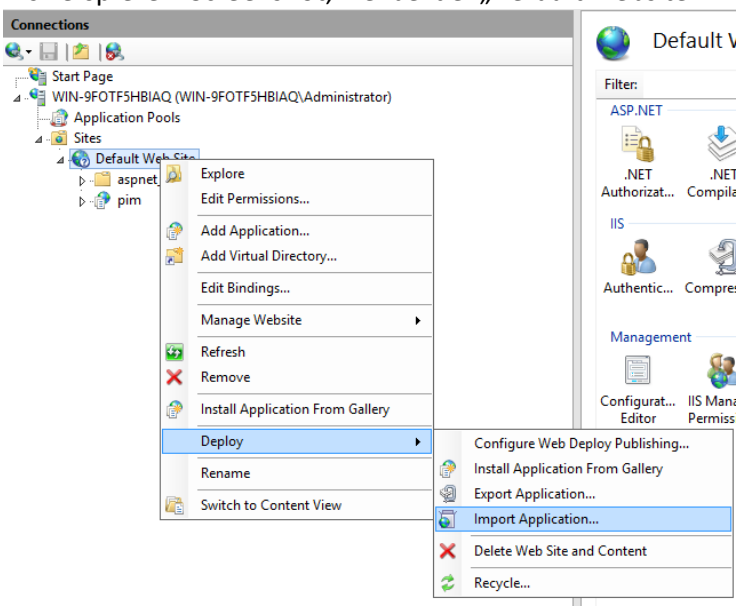
3.2.1.3 Applikationspool (Applicationpool)

Damit die Applikation korrekt läuft, muss im „Applicationpool“ die Einstellung vorgenommen werden, dass 32-Bit Anwendungen zugelassen sind.
Dafür den IIS-Manager öffnen und dann auf der linken Seite den „Applicationpool“ öffnen.
Dort muss mindestens bei allen „ASP.NET 4.5 Pools“ und dem „Default Application Pool“ unter „Erweiterte Einstellungen (Advanced Settings)“ die Einstellung „32 Bit Applikationen ermöglichen (Enable 32-Bit Applications)“ auf „Ja (Enabled)“ gesetzt werden.



3.2.2 Durchführung

1. Das ZIP-Archiv „pim.zip“ aus dem Verzeichnis „iis“ auf den Server auf dem sich der IIS-Webserver befindet kopieren.
2. Öffnen des IIS-Manager.
3. Bei der Webseite, bei der die Applikation später erscheinen soll:
   Rechtsklick → Deploy → Import Application…
   Falls dieses Menü nicht verfügbar ist, wurde entweder wie zuvor erklärt WebDeploy nicht installiert oder der IIS-Manager muss nochmals neu geöffnet werden.
   Als Beispiel ein Screenshot, hier bei der „Default Website“.



4. Im aufgehenden Fenster das „pim.zip“ Archiv auswählen.
5. Der gesamte Inhalt auswählen → alle Checkboxen markieren.
6. Dann kann ein Namen für die Webseite, welche für den späteren Zugriff auf die Applikation gebraucht wird, gewählt werden. Im Beispiel-Screenshot wurde PIM gewählt, das bedeutet im Fall der Default-Webseite, dass die Applikation dann unter http://<hostname>/pim/Login.aspx erreicht werden kann, wird der Name weggelassen, so ist die Applikation danach direkt unter dem „root-Verzeichnis“ (hier http://<hostname>/Login.aspx) erreichbar.
7. Der Import sollte dann bereits abgeschlossen sein.
8. Gegebenenfalls Upload-Verzeichnis sowie Rechte ändern, siehe dazu Kapitel 5.4

9. Es müssen jeweils noch mehr Anpassungen gemacht werden, diese sind im Kapitel Generelle Applikationseinstellungen beschrieben, da diese für IIS sowie für Visual Studio gültig sind.

4. Microsoft Visual Studio

4.1.1 Voraussetzung

Installiertes Visual Studio in der Version 2010 oder höher.

4.1.2 Durchführung

1. Visual Studio öffnen.
2. File → Open → Project/Solution…
3. Die Datei „PIM_Prototyping.sln“ im Verzeichnis „08.SourceCode\PIM_Prototyping“ (welches sich eine Stufe höher als dieses Verzeichnis befindet) öffnen.
4. Damit das Projekt gestartet werden kann (F5), muss zuerst noch ein „StartUp Project“ festgelegt werden. Dazu im SolutionExplorer Rechtsklick auf das Projekt „Presentation“ und dann „Set as StartUp Project“ wählen.
5. Es müssen gegebenenfalls noch mehr Anpassungen gemacht werden, diese sind im Kapitel *3.4 Zusätzliche Einstellungen* beschrieben, da diese für IIS sowie für Visual Studio gültig sind.

5. Generelle Applikationseinstellungen

Folgende Einstellungen müssen gegebenenfalls für beide oben angegebene Möglichkeiten durchgeführt werden.

5.1 Installation des MySql-Connector

Der MySql-Connector benötigt asp.net damit mit MySql Datenbanken gearbeitet werden kann. Dieser muss zwingend vorhanden sein.
Für die Installation gibt es zwei Möglichkeiten:

3. Download von http://www.mysql.com/products/connector/ → ADO.NET Driver for MySQL (Connector/NET)
4. Installation einer bereits heruntergeladenen Datei im Verzeichnis „utils“ unter dem Namen „mysql-connector-net-6.6.4.msi“

Die Installation selber ist selbst erklärend.

5.2 Installation des 2007 Office System Driver: Data Connectivity Components

Der Office System Driver wird benötigt, damit asp.net Excel-Dateien lesen und schreiben kann. Die Installation ist nicht immer notwendig, ist auf dem Computer auf welchem der IIS oder das VisualStudio läuft bereits Office 2007 installiert so kann dieser weggelassen werden. Wir empfehlen aber diesen so oder so zu installieren, da dieser keine destruktive Wirkung hat.
Für die Installation gibt es zwei Möglichkeiten:

1. Download von http://www.microsoft.com/en-us/download/confirmation.aspx?id=23734
2. Installation einer bereits heruntergeladenen Datei im Verzeichnis „utils“ unter dem Namen „OfficeSystemDriver.exe“

Die Installation selber ist selbst erklärend.

5.3 Anpassen der Datenbankeinstellungen

Mit hoher Wahrscheinlichkeit müssen die Datenbankeinstellungen der neuen Umgebung angepasst werden, dazu muss wie folgt vorgegangen werden:

1. Öffnen der Datei „Web.Config“
   a. **IIS**: Im Windows Explorer Ordner öffnen, in welchem die Webseite installiert ist. Falls der Pfad nicht bekannt ist: Im IIS Manager Rechtsklick auf die Webseite → Explorer. Dann die Datei „Web.Config“ öffnen.
   b. **Visual Studio:** Im Projekt „Presentation“ die Datei Web.config öffnen.
2. Die Datenbank-Einstellungen können nun in dieser Datei angepasst werden.
   Die Einstellungen sollten etwa wie folgt aussehen:

```
<!-- DB-Settings change if needed -->
<add key="dbserver" value="152.96.56.49" />
<add key="dbserverport" value="443" />
<add key="dbname" value="pim" />
<add key="dbuser" value="pimdbuser" />
<add key="dbpassword" value="1234" />
<!-- End of DB-Settings -->
```

3. Die Einstellungen sollten selbsterklärend sein. Die Einstellungen müssen natürlich so angepasst werden, damit sie mit der Konfiguration der Datenbank (aus dem Kapitel Datenbank) übereinstimmen.
4. Speichern und Schliessen der Datei „Web.config“.
   Hinweis: Ansonsten sollte in dieser Datei nichts angepasst werden!

5. **Optional in VisualStudio**: Falls das TestProject (für Unit Tests) verwendet werden soll, so muss die Datei „App.Config“, welche sich im „TestProject“ befindet, genau gleich wie die „Web.config“ angepasst werden.

5.4 Excel Import

Da die zu importierenden Excel-Dateien temporär auf dem Server zwischen gespeichert werden müssen, muss ein entsprechendes Verzeichnis auf dem Server vorhanden sein sowie die dazu nötigen Berechtigungen gesetzt werden.
Welches Verzeichnis hierzu von der Applikation verwendet wird, kann in der Datei Web.Config des Projektes festgelegt werden.

1. Öffnen der Datei „Web.Config“
   a. **IIS**: Im Windows Explorer Ordner öffnen, in welchem die Webseite installiert ist. Falls der Pfad nicht bekannt ist: Im IIS Manager Rechtsklick auf die Webseite → Explorer. Dann die Datei „Web.Config“ öffnen.
   b. **Visual Studio:** Im Projekt „Presentation“ die Datei Web.config öffnen.
2. Folgender Eintrag ist hierzu verantwortlich:
   ```
   <add key="ExcelFolderPath" value="upload"/>
   ```
   Das Verzeichnis würde in diesem Falle relative zum IIS-Applikations-Pfad (pim/webpages/upload) zeigen
3. Das Verzeichnis muss von Hand angelegt werden.
4. Dem Verzeichnis sollten die Rechte „lesen/schreiben“ für „Jeder“ (Everyone) gegeben werden

5.5 Log-File

Fehler die während der Laufzeit der Applikation auftreten werden jeweils in einem Log-File geloggt. Die Konfigurations-Möglichkeiten für den Log-Mechanismus befinden sich auch allesamt in der Web.Config Datei und sind unter dem Tag `<log4net>` zusammengefasst.
Die dabei wohl wichtigste Option bestimmt, unter welchem Pfad die Applikation das Log-File erstellen soll und ist unter dem Folgenden Eintrag zu finden:
```
<file value="C:\\temp\\PimLog.txt"/>
```

6. Neue Version aus Visual Studio publizieren

Falls an der Applikation Änderungen durchgeführt werden, so kann aus Visual Studio heraus leicht eine neue Version für den IIS-Server publiziert werden.

6.1 Voraussetzungen

Produkt ist bereits im Visual Studio installiert bzw. als Solution vorhanden.

6.2 Durchführung

1. Öffnen der Solution „PIM_Prototyping.sln“
2. Rechtsklick auf das Projekt „Presentation“ und „Publish..“ wählen.
3. *Optional*: In der Kategorie „Connection“ den „Package location“ Pfad anpassen.
   Unter diesem Pfad wird das ZIP-Archiv generiert.
4. Weitere Einstellungen sind nicht nötig. Nur noch auf „Publish“ klicken.
5. Es werden nun im angegebenen Verzeichnis einige Dateien generiert. Die einzig Relevante Datei ist „pim.zip“, der Rest kann gelöscht werden.
6. Die Datei „pim.zip“ kann nun auf dem IIS importiert werden (siehe entsprechendes Kapitel)

7. Sicherheit

Um eine sichere Übertragung der Daten zwischen Client und Server zu erreichen, müsste der IIS-Server unter dem HTTPS-Protokoll betrieben werden. Dazu müsste auf dem IIS-Webserver SSL aktiviert werden, für welches wiederum ein entsprechendes Zertifikat für die Verschlüsselung und Entschlüsselung benötigt wird. Auf eine genaue Erklärung, wie dies bewerkstelligt werden kann verzichten wir in dieser Anleitung auf verweisen auf die offizielle Anleitung von Microsoft. Welche auch online unter nachfolgendem Link verfügbar ist.
http://support.microsoft.com/kb/324069

HSR
HOCHSCHULE FÜR TECHNIK
RAPPERSWIL

FHO Fachhochschule Ostschweiz

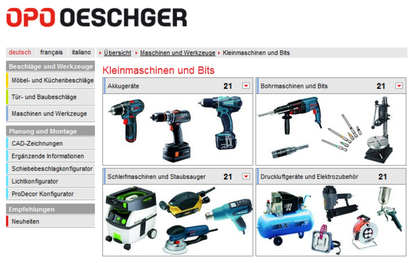
Semesterarbeit Herbstsemester 2012
Themengebiet: Software

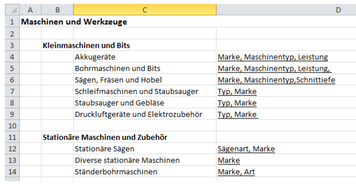
e-Shop / PIM Prototyping

Simon Stähli Patrick Zenhäusern

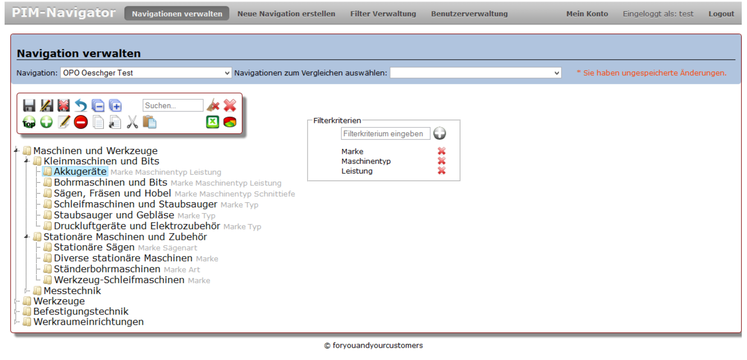
Betreuer: Dr. Daniel Keller
Experte: Dr. Daniel Keller
Projektpartner: foryouandyourcustomers AG

Ausgangslage

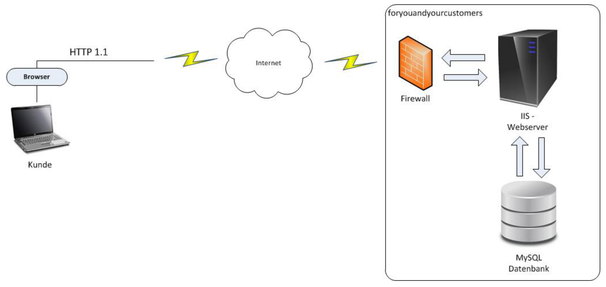




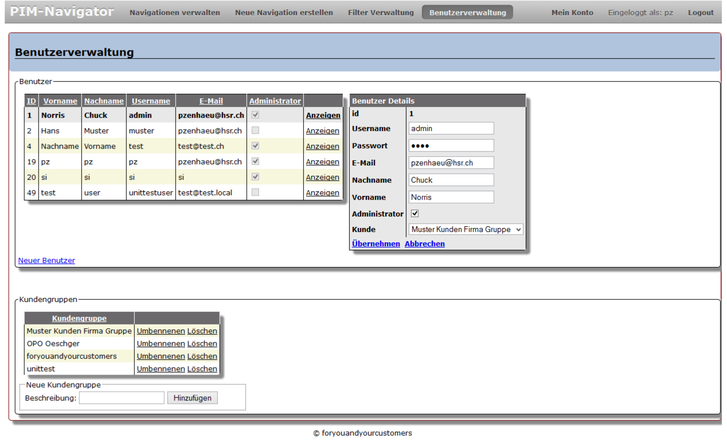
Im Auftrag der Firma foryouandyourcustomers AG, welche PIM (Product Information Management) Systeme für grössere Firmen konzipiert und realisiert, sollte eine Software für das Prototyping von Navigationsstrukuren für e-Shops und PIM-Systeme entwickelt werden. Den Kunden soll eine benutzerfreundliche Möglichkeit geboten werden, ihre zukünftigen Navigationsstrukturen zu entwickeln und zu testen. Bis zum jetzigen Zeitpunkt wurde für diese Aufgabe jeweils ein Excel-File verwendet.



Technologie

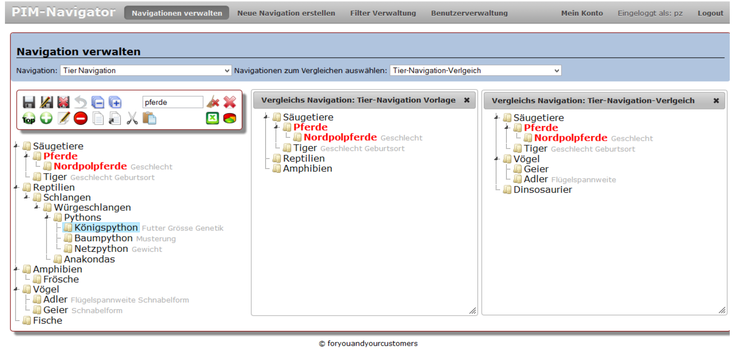


Die Applikation wurde als Web-Applikation entworfen und auf der Basis von ASP.NET realisiert. Im Hintergrund greift die Applikation auf eine MySQL Datenbank zu um die Daten zu speichern. Um den Benutzern eine möglichst dynamische und benutzerfreundliche Web-Applikation zu bieten, wurden diverse Funktionalitäten, beispielsweise Drag&Drop, mittels JavaScript implementiert.



Lösung

Als Ergebniss dieses Projektes steht eine betriebsbereit Web-Applikation, welche Produktemanagern von e-Shops oder PIM-Systemen das Erstellen, Editieren und Testen von Navigationshirarchien auf eine leichte Weise ermöglicht. Die Applikation ist Multi-User fähgig, wobei über einen Administratoren-Account neue Benutzer erstellt und verwaltet werden können. Die Kernfunktionalität stellt hierbei klar das Erstellen und Editieren von Navigationshirarchien mittels Drag&Drop dar. Um es Produktmanagern zu erlauben, diese Arbeit möglichst effektiv zu erledigen, stehen ihnen weitere Funktionalitäten wie das Rückgängig machen von Arbeitschritten, das Vergleichen von Navigationen, Statistiken oder auch das Importieren und Exportieren von Navigationshirarchien aus Excel-Dateien zur Verfügung.



HSR
HOCHSCHULE FÜR TECHNIK
RAPPERSWIL

FHO Fachhochschule Ostschweiz

Kurzfassung der Studienarbeit

Abteilung	**Informatik**
Name[n] der Studierenden	**Simon Stähli** **Patrick Zennhäusern**
Studienjahr	**HS 2012**
Titel der Studienarbeit	**e-Shop/PIM Prototyping**
Examinatorin / Examinator	**Dr. Daniel Keller**
Themengebiet	**Software**
Projektpartner	**foryouandyourcustomers AG, Pfäffikon ZH**
Institut	**Institut für Software**

Im Auftrag der Firma foryouandyourcustomers AG, welche PIM (Product Information Management) Systeme für grössere Firmen konzipiert und realisiert, sollte eine Software für das Prototyping von Navigationsstrukturen für e-Shops und PIM-Systeme entwickelt werden. Den Kunden soll eine benutzerfreundliche Möglichkeit geboten werden, ihre zukünftigen Navigationsstrukturen zu entwickeln und zu testen.

Zu diesem Zweck wurde eine Web-Applikation auf der Basis von ASP.NET konzipiert und entwickelt, welche Benutzern die Möglichkeit bietet Navigationsstrukturen zu erfassen, zu verändern und auch zu vergleichen. Da die Benutzerfreundlichkeit bei dieser Applikation ein grosser Faktor ist, wurde hierbei Client-Seitig JavaScript eingesetzt um Benutzern eine möglichst dynamische Applikation zu bieten, welche unter anderem Drag & Drop, sowie Inline-Editing unterstützt. Um eine möglichst gute Bedienbarkeit zu erreichen, wurden des weiteren Usability Tests bei potenziellen Benutzern durchgeführt. Deren Verhalten bei der ersten Benutzung der Applikation wurde analysiert und in der weiteren Entwicklung berücksichtigt.
Im Verlaufe der Arbeit kamen vom Auftraggeber noch einige zusätzliche Wünsche & Features auf, wie z.B. den Export von Navigationen in eine Excel-Datei oder die Möglichkeit Aktionen rückgängig zu machen. Da der Umfang für den zeitlichen Rahmen des Projektes am Anfang absichtlich grosszügig gewählt wurde, konnten auch alle zusätzliche Features & Wünsche implementiert werden.

Schlussendlich wurde alle Anforderungen vollständig abgedeckt und ein funktionsfähiges Produkt abgeliefert.

HSR
HOCHSCHULE FÜR TECHNIK
RAPPERSWIL

FHO Fachhochschule Ostschweiz

Vereinbarung über die Aufgabenstellung

1. Gegenstand der Vereinbarung

Mit dieser Vereinbarung wird die Aufgabenstellung für die Semesterarbeit im Herbstsemester 2012 geregelt. Diese wird durchgeführt durch die Studenten Simon Stähli und Patrick Zenhäusern unter der Betreuung von Dr. Daniel Keller und im Auftrag der Firma foryouandyourcustomers AG.

2. Aufgabenstellung

Entwerfen und implementieren Sie einen Prototyp, der die Arbeit der Daten- und Navigationsmodellierung unterstützt. Sie werden ein Tool entwickeln, das mit Baumstrukturen, Listen und evtl. UML-Klassendiagrammen die Daten- und Navigationsstrukturen gestalten hilft, sodass ein PIM-Spezialist laufend den Überblick über die Einfachheit der entworfenen Lösung behalten kann. Das Programm muss mit einem überzeugenden und leicht zu bedienenden GUI inkl. drag-and-drop ausgestattet werden. PIM-Spezialisten werden das Programm einsetzen, um VOR der Implementation eines PIM die für das jeweilige Projekt optimalen Strukturen zu definieren.

3. Unterschriften

Rapperswil, den 13.12.12 .....

Die Studenten: Simon Stähli / Patrick Zenhäusern

Rapperswil, den 13. Dez. 2012 .....

Der Betreuer der Semesterarbeit: Dr. Daniel Keller

HSR
HOCHSCHULE FÜR TECHNIK
RAPPERSWIL

FHO Fachhochschule Ostschweiz

Erklärung

Ich erkläre hiermit,

- dass ich die vorliegende Arbeit selber und ohne fremde Hilfe durchgeführt habe, ausser derjenigen, welche explizit in der Aufgabenstellung erwähnt ist oder mit dem Betreuer schriftlich vereinbart wurde,
- dass ich sämtliche verwendeten Quellen erwähnt und gemäss gängigen wissenschaftlichen Zitierregeln korrekt angegeben habe.
- das ich keine durch Copyright geschützten Materialien (z.B. Bilder) in dieser Arbeit in unerlaubter Weise genutzt habe.

Ort, Datum: Rapperswil, 13.12.2012

Name, Unterschrift: Simon Stähli,

HSR
HOCHSCHULE FÜR TECHNIK
RAPPERSWIL

FHO Fachhochschule Ostschweiz

Erklärung

Ich erkläre hiermit,

- dass ich die vorliegende Arbeit selber und ohne fremde Hilfe durchgeführt habe, ausser derjenigen, welche explizit in der Aufgabenstellung erwähnt ist oder mit dem Betreuer schriftlich vereinbart wurde,
- dass ich sämtliche verwendeten Quellen erwähnt und gemäss gängigen wissenschaftlichen Zitierregeln korrekt angegeben habe.
- das ich keine durch Copyright geschützten Materialien (z.B. Bilder) in dieser Arbeit in unerlaubter Weise genutzt habe.

Ort, Datum: Rapperswil, 13.12.2012

Name, Unterschrift: Patrick Zenhäusern