

Dokumentation

zur betrieblichen Projektarbeit

DAdmin

Erstellung einer Webanwendung zur Verwaltung von registrierten Domainnamen

Andreas Schipplock

Fachinformatiker

Fachrichtung Anwendungsentwicklung

Ausbildungsbetrieb:

TeKoNet Medien GmbH
Alte Bottroper Straße 92
45356 Essen

Auszubildender:

Andreas Schipplock
Reulstraße 62
46240 Bottrop

Inhaltsverzeichnis

| | |
|---|----|
| Vorwort | 1 |
| 1. Projektdefinition | 1 |
| 1.1 Projektumfeld | 1 |
| 1.2 Ist-Zustand | 1 |
| 1.3 Soll-Zustand | 3 |
| 1.4 Abgrenzung | 3 |
| 1.5 Projektschnittstellen | 3 |
| 2. Projektplanung | 3 |
| 2.1 Einleitung | 3 |
| 2.2 Planung der Zeit | 4 |
| 2.3 Ressourcenplanung | 4 |
| 2.4 Nutzenanalyse | 5 |
| 2.4.1 Aufwandsvergleich | 5 |
| 2.4.2 Nutzwertanalyse | 6 |
| 3. Projektdurchführung | 7 |
| 3.1 Analysephase | 7 |
| 3.1.1 Ist-Zustand Analyse | 7 |
| 3.1.2 Soll-Zustand Definition | 7 |
| 3.1.3 Erstellung des Pflichtenhefts | 8 |
| 3.2 Entwurfsphase | 8 |
| 3.2.1 Erstellung des DV-Konzepts | 8 |
| 3.3 Implementationsphase | 9 |
| 3.3.1 Erstellung der Datenbankstruktur | 10 |
| 3.3.2 Erstellung des Pagecontrollers | 10 |
| 3.3.3 Erstellung des GUI | 11 |
| 3.3.4 Erstellung der einzelnen Page-Komponenten | 12 |
| 3.3.5 Erstellung der Expirecheck Komponente | 13 |
| 3.4 Testphase | 13 |
| 3.4.1 Durchführung der Komponententests | 13 |
| 3.4.2 Durchführung der Integrationstests | 14 |
| 3.4.3 Durchführung der Systemtests | 14 |
| 3.5 Erstellung der Dokumentation | 14 |
| 4. Projektabschluss | 15 |
| 4.1 Soll – Ist Vergleich | 15 |
| 4.2 Projektabschluss | 15 |

Hinweise

Unterstreichungen Begriffe, die eine einfach verständliche Definition benötigen, werden unterstrichen dargestellt. Die entsprechende Definition findet man dann im Glossar.

Vorwort

Das Projekt „Erstellung einer Webanwendung zur Verwaltung von registrierten Domainnamen“ ist ein betrieblich, internes Firmenprojekt der TeKoNet Medien GmbH zur Verbesserung der Arbeitsschritte bezogen auf die Verwaltung der registrierten Domainnamen. Das Projekt trägt den Namen „Dadmin“ und bedeutet nicht mehr als „Domain Administrator“, also sinngemäß „Domainnamenverwaltung“. Die Firma TeKoNet Medien GmbH agiert im Umfeld eines Fullservice Providers und bietet somit auch Webhosting, Serverhosting und andere Internetbezogenen Dienste an. In diesem Umfeld werden viele Domainnamen gebraucht und registriert. Die Registrierung und Verwaltung von Domainnamen stellt für TeKoNet aber nicht das Kerngeschäft dar, ist jedoch in Verbindung mit dem Angebotsspektrum ein immer wichtiger werdender Bereich.

1. Projektdefinition

1.1 Projektumfeld

Die Mitarbeiter der TeKoNet Medien GmbH müssen für Kunden Domainnamen registrieren und verwalten. Diese Domainnamenregistrierung geschieht über einen Registrar, der entsprechende Schnittstellen bereitstellt. Mit jeder Domainnamenregistrierung muss dieser Domainname dem jeweiligen Kunden in Rechnung gestellt werden.

1.2 Ist-Zustand

Derzeit wird bei einer Anfrage zur Registrierung eines neuen Domainnamens Seitens des Kunden, die meistens telefonisch erfolgt, der entsprechende Domainnamenhandle, sofern noch keiner existiert, angelegt und mit dessen dann der gewünschte Domainname registriert. Sofern die Registrierung dann erfolgreich durchgeführt wurde, muss der entsprechende Domainname mit Relation zum Kunden in einer einfachen Exceltabelle gespeichert werden, damit später nachvollzogen werden kann, welcher Kunde welchen Domainnamen registriert hat. Diese Exceltabelle ist Grundlage der zu berechnenden Kosten.

Durch diese Art der Verwaltung ergeben sich folgende Nachteile:

- **Schreibfehler**

Bei der telefonischen Bestellung eines Domainname können Verständigungsprobleme oder einfache Schreibfehler zu Unkosten führen, weil man ganz einfach den falschen Namen registriert. Ein falsch registrierter Domainname wird dem Kunden natürlich nicht in Rechnung gestellt und bedeutet deswegen direkter Verlust für die TeKoNet Medien GmbH.

- **Zeitaufwand**

In der Exceltabelle werden zu jedem Domainnamen jeweils die Kundendaten und das Registrierdatum gespeichert und im Falle einer Änderung dieser Daten ist eine Änderung bei allen Einträgen nötig und die Vergangenheit hat gezeigt, dass diese Änderungen oft zu inkonsistenten Daten führten, was wiederum darin resultierte, dass man nicht mehr ordentlich nach Kundendaten sortieren konnte und somit die Berechnung der Kosten der registrierten Domainnamen für einen Kunden nicht mehr korrekt durchgeführt werden kann. In vielen Fällen wurden die Kosten dann als deutlich zu niedrig ausgerechnet, weil eben diese Kundendaten nicht konsistent waren und somit bei einer Berechnung nicht berücksichtigt wurden, obwohl sie existierten.

- **Erneute Berechnung der Kosten**

In der Regel werden Domainnamen angelegt, die sich jedes Jahr automatisch um ein weiteres Jahr verlängern. Das bedeutet, dass ein Domainname, so lange dafür bezahlt wird, nicht ausläuft. Das bedeutet aber auch, dass die Mitarbeiter der TeKoNet Medien GmbH peinlichst darauf achten müssen, wann ein Domainname erneut berechnet werden soll. Die Erfahrungen diesbezüglich zeigten auf, dass oftmals vergessen wird, wann ein Domainname ausläuft und wieder berechnet werden muss, was ganz einfach daran liegt, dass es kein System gibt, welches einen über demnächst auslaufende oder verlängernde Domainnamen informiert, da die zu Grunde liegende Verwaltung eine Exceltabelle ist. Man muss also in der Exceltabelle nach dem Registrierdatum schauen, was sehr kontraproduktiv, zeitaufwendig und fehleranfällig ist.

Es ist auch wichtig, dass gelöschte oder umgezogene Domainnamen aus der Exceltabelle entfernt werden, da sonst ein Domainname erneut berechnet wird, den es gar nicht mehr gibt oder der schon längst bei einem anderen Provider bezahlt wird.

- **Übersicht für den Kunden**

Der Kunde hat aktuell keine Möglichkeit einzusehen, welche Domainnamen er bei der TeKoNet Medien GmbH registriert hat und oftmals stellt das ein Problem für die Kunden dar, da aus Kostengründen gerne eine solche Übersicht gewünscht wird.

1.3 Soll-Zustand

Durch die Entwicklung des Produkts „Erstellung einer Webanwendung zur Verwaltung von registrierten Domainnamen“ sollen die Arbeitsschritte bezüglich der Verwaltung von Domainnamen und Kunden durch die TeKoNet Medien GmbH Mitarbeiter deutlich verbessert werden, aber auch der Kunde soll Zugang zu dieser Webanwendung haben, um mit eingeschränkten Rechten eine klar strukturierte Übersicht aller seiner registrierten Domainnamen und dessen Kosten zu haben. Unterstützend helfen sollen hier intelligente Such und Filterfunktionen.

Mit Hilfe eines Formulars für den Kunden zur Anforderung eines neuen Domainnamen soll außerdem die Wahrscheinlichkeit einer Falschregistrierung minimiert werden. Ein Benachrichtigungssystem soll außerdem darüber berichten, wann neue Kosten für demnächst auslaufende oder verlängernde Domainnamen entstehen. Die Webanwendung soll außerdem eine Übersicht der, demnächst, auslaufenden Domainnamen bereitstellen, mit der auch der Kunde die Kosten für seine registrierten Domainnamen auf einen Blick in dieser Anwendung, zusätzlich zu der Emailbenachrichtigung, erkennen kann.

1.4 Abgrenzung

Die Kosten spielen eine wichtige Rolle in dieser Umgebung, aber aus zeitlichen Gründen kann eine Integration in die vorhandene Warenwirtschaftsumgebung nicht implementiert und entwickelt werden, da das System eine geschlossene Einheit ist. Es reicht dem Mitarbeiter von TeKoNet aber völlig aus zu wissen, dass etwas berechnet werden muss. Eine Integration würde also nur viel Zeit kosten und für den Mitarbeiter der TeKoNet Medien GmbH keinen außergewöhnlichen Mehrwert bringen.

1.5 Projektschnittstellen

Damit das Projekt die gewünschten Ziele erreicht, ist eine Zusammenarbeit sämtlicher Benutzergruppen, die für diese Software in Frage kommen, nötig. Während des Projekts stehen mir deswegen hauptsächlich Herr Herzog, Chef des Unternehmens, und Herr Hannemann, ein Kunde mit vielen, bei uns registrierten, Domainnamen, zur Verfügung. Herr Herzog wird die Zwischenergebnisse bewerten und maßgeblich an den Anforderungen mitwirken. Herr Hannemann ist mit der Menge an registrierten Domainnamen an einem Test beteiligt und bewertet diesbezüglich aus Kundensicht das Produkt, die Webanwendung.

2. Projektplanung

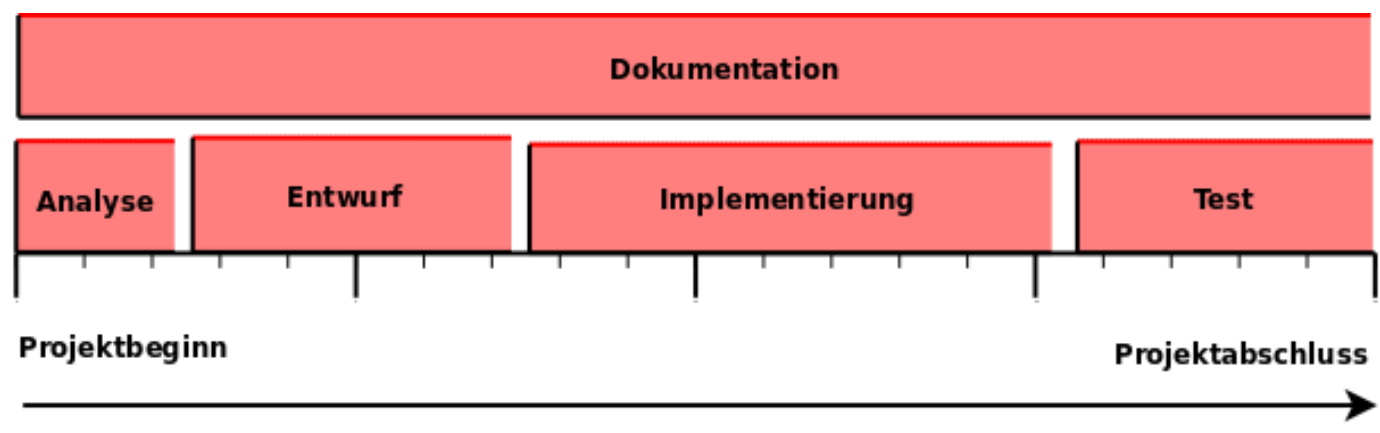
2.1 Einleitung

Das Projekt durchläuft zuerst die Analysephase, die in erster Hinsicht definiert, welche Leistungen das Produkt zu umfassen hat. In der nächsten Phase, der Entwurfsphase, wird festgelegt und dokumentiert, wie die entsprechenden Leistungen genau definiert sind. Es werden Datenstrukturen und Schnittstellen

geschaffen, die dann in der Implementierungsphase umgesetzt werden können. Anschließend wird dann das ganze Produkt getestet. Während der gesamten Projektzeit wird die Dokumentation parallel mit erstellt.

Jede Phase ist von dem Produkt der vorhergehenden Phase abhängig und bei diesem Modell muss und wird deswegen schon bei Beginn an auf die Vermeidung von Fehlern geachtet, da eine späte Änderung hierbei zeit und kostenintensive Folgen haben kann.

Die folgende Grafik verdeutlicht die Phasen in Form eines Zeitstrahls:



2.2 Planung der Zeit in Stunden

| Phase | Beschreibung | Soll |
|-----------------|---|-----------|
| Analyse | Ist-Analyse formulieren | 2 |
| | Soll-Zustand definieren | 2 |
| | Erstellung des Pflichtenhefts | 5 |
| Entwurf | Erstellung des DV-Konzepts | 13 |
| Implementierung | Umsetzung der zuvor definierten Strukturen, Ziele und Spezifikationen | 25 |
| Test | Durchführung Komponententests | 2 |
| | Durchführung Integrationstests | 2 |
| | Durchführung Systemtests | 6 |
| Dokumentation | Erstellung der Projektdokumentation | 10 |
| | Erstellung des Benutzerhandbuchs | 3 |
| Gesamt | | 70 |

2.3 Ressourcenplanung

Für die Entwicklung der Webanwendung „Dadmin“ wird lokal ein Rechner mit Debian Linux 5 aufgesetzt, der dann entsprechend Apache2 mit Mod_Perl2, Capullo , Postgresql und alle benötigten Perlmodule installiert hat.

Für die Erstellung der Programmteile wird der Editor „SciTE“ benutzt und für die Versionskontrolle wird ein lokal installierter Git-Server verwendet.

Für die einzelnen Komponententests wird das, frei verfügbare, Testing Framework „Test::More“ verwendet. Die Komponententests werden aber auch zusätzlich schon während der Implementierung manuell getätigt.

2.4 Nutzenanalyse

Da es sich hier um ein betriebliches, internes Projekt handelt, welches hauptsächlich der Firma dient, und es hier keinen Kunden gibt, der dieses Projekt in Auftrag gibt, und man deswegen keine Stundensätze für die Berechnung der Kosten verwenden kann, beschränke ich mich hier auf den Aufwandsvergleich, die Nutzwertanalyse und lasse die Kosten außen vor.

2.4.1 Aufwandsvergleich

▪ 2.4.1.1 Mehraufwand durch Annahme einer Bestellung eines Domainnamens

Eine Bestellung eines Domainnamens erfolgt immer telefonisch durch den Kunden, was bedingt durch die Art der Kommunikation einige Zeit in Anspruch nimmt. Hinzu kommt die Validierung der Daten und eventuelle Kommunikationsprobleme. Bei falscher oder fehlender Daten muss man eventuell den Kunden zurückrufen. Die Bestellung von Domainnamen wird mit der Einführung der Webanwendung alleine vom Kunden durchgeführt und der Mitarbeiter von TeKoNet bekommt nur noch eine Email mit allen relevanten Daten.

| Aktion | Alter Aufwand | Neuer Aufwand |
|--|---------------|---------------|
| Führen eines Gesprächs | 5 Minuten | 0 Minuten |
| Prüfung der Inhaber und Domaindaten | 2 Minuten | 0 Minuten |
| Gesamt bei 100 Registrierungen pro Monat | 12 Stunden | 0 Stunden |
| Hochgerechnet auf das Jahr | 144 Stunden | 0 Stunden |

▪ 2.4.1.2 Mehraufwand durch die Verwaltung der Exceltabelle

Nachdem ein Kunde die telefonische Bestellung eines Domainnamens getätigt hat, muss der Mitarbeiter von TeKoNet diesen Domainnamen registrieren und danach die Exceltabelle mit Domainname, Datum und Kundendaten pflegen. Bei der Pflege der Exceltabellen entstehen jedoch Redundanzen und bereits getätigte Eingaben müssen bei jedem Domainnamen wiederholt werden. Dies führt außerdem zu Fehlern und kostet Zeit, da der Mitarbeiter bei jeder Pflege der Exceltabelle die Daten prüfen muss. Mit der Einführung der Webanwendung werden Kundendaten nur einmalig eingegeben, was bei weiteren Domainregistrierungen den Vorgang und die Verwaltung deutlich beschleunigt und Fehlerquellen minimiert.

| Aktion | Alter Aufwand | Neuer Aufwand |
|--|---------------|---------------|
| Neue Eingabe der Kundendaten | 3 Minuten | 3 Minuten |
| Erneute Eingabe der Kundendaten | 3 Minuten | 0 Minuten |
| Verwaltung der Domainnamen pro Registrierung | 5 Minuten | 1 Minute |
| Ergibt bei 100 Neuregistrierungen pro Monat | 18 Stunden | 7 Stunden |

Hochgerechnet auf das Jahr

216 Stunden

84 Stunden

2.4.1.3 Mehraufwand durch die Prüfung von auslaufenden oder verlängernden Domainnamen

Je nach Registrierdatum laufen monatlich Domainnamen aus oder werden automatisch verlängert, was bei erster Aktion bedeutet, dass ein Mitarbeiter der TeKoNet Medien GmbH den entsprechenden Domainnamen aus der Exceltabelle löschen muss, damit dieser nicht fälschlicherweise erneut berechnet wird. Im Falle der zweiten Aktion „der Domainname wird automatisch verlängert“ muss ein Mitarbeiter mit Hilfe der Exceltabelle prüfen, wann ein Domainname verlängert wird, denn das ist Grundlage der Berechnung und Domainnamen werden in den meisten Fällen jährlich in Rechnung gestellt. Der Aufwand der Prüfung auf diese Daten ist extrem zeitintensiv. Die Prüfung der Exceltabelle bezogen auf diese Daten findet ein Mal im Monat statt. Hinzu kommt aber die Tatsache, dass die Prüfung dieser Daten mit steigender Anzahl an Domainnamen steigt. Mit dem neuen Produkt ist dieses Prüfen hinfällig, da es ein automatisches Benachrichtigungssystem geben wird, welches den entsprechenden Mitarbeiter immer rechtzeitig via Email darüber informiert, wenn ein Domainname ausläuft oder verlängert wird.

| Aktion | Alter Aufwand | Neuer Aufwand |
|--|---------------|---------------|
| Prüfen der Domaindaten auf Auslaufen oder Verlängerung | 8 Stunden | 0 Stunden |
| Ergibt im Jahr | 96 Stunden | 0 Stunden |

2.4.2 Nutzwertanalyse

Bei der Bewertung des Nutzens verwende ich folgende Bewertungen und Gewichtungsfaktoren, um den Nutzen aufzuzeigen:

| Bewertung | Faktor |
|-------------|--------|
| Sehr gut | 10 |
| Gut | 6-9 |
| Mittelmäßig | 5 |
| Schlecht | 0-4 |

| Kriterium | Gewichtung | Bewertung Alt | Bewertung Neu |
|------------------------|------------|---------------|---------------|
| Zuverlässigkeit | 4 | 2 | 9 |
| Benutzerfreundlichkeit | 4 | 1 | 7 |
| Erweiterbarkeit | 2 | 2 | 5 |
| Effizienz | 3 | 0 | 8 |
| Funktionalität | 2 | 3 | 6 |
| Summe | | 22 | 110 |

Mit Hilfe dieser Nutzwertanalyse und dieser Tabelle hier wird bereits deutlich, dass durch die Einführung des neuen Systems eine deutliche Verbesserung festzustellen ist.

3. Projektdurchführung

Das Projekt wird mit Hilfe der vier bereits genannten Phasen durchgeführt.

3.1 Analysephase

3.1.1 Ist-Zustand Analyse

Dauer : 3 Stunden

Zusammen mit Herrn Herzog, dem Geschäftsführer der TeKoNet Medien GmbH, und Herrn Erdmann aus der Technik Abteilung wird der aktuelle Zustand des Bereiches der Domainnamenverwaltung genauer analysiert und diskutiert. Da Herr Herzog, Herr Erdmann und meine Wenigkeit in diesem Bereich die meisten Erfahrungen machen konnten, existieren bereits bestimmte Kritikpunkte, die den Arbeitsablauf diesbezüglich negativ beeinflussen und nicht optimal funktionieren. Besonders in den Mittelpunkt der Kritik gerät die Zeit, die benötigt wird, um einen Domainnamen ordentlich zu verwalten, besonders mit der Tatsache im Hinterkopf, dass die Registrierung und Verwaltung von Domainnamen nicht das Kerngeschäft von TeKoNet darstellt, jedoch mit den angebotenen Dienstleistungen immer mehr an Wichtigkeit zunimmt. Besonders in der Kritik steht außerdem die Tatsache, dass man die Domainnamen nur schwer verwalten kann und keine Benachrichtigung bekommt, wenn ein Domainname ausläuft oder automatisch verlängert wird, was in einem Fall immer zu Unkosten der Firma geht und Verwirrung beim Kunden schafft, da der Kunde keine Info darüber hat, wie die Status seiner Domainnamen sind. Von Herrn Herzog wird außerdem erläutert, dass die aktuelle Situation der Domainnamenbestellung, welche rein telefonisch passiert, nicht optimal funktioniert, da dieser Prozess Zeit und automatisch Geld kostet. Er betont, dass es durchaus passieren kann, dass in diesem Prozess der Bestellung Fehler gemacht werden können, die im Falle eines solchen, Kosten für TeKoNet bedeuten. Durch diesen Bestellprozess wird außerdem der allgemeine Arbeitsablauf gestört und kostet im Endeffekt viel mehr, als nur die Zeit des Telefonates.

Diese Kritikpunkte und die Ergebnisse der Analyse werden alle samt in einem Dokument festgehalten.

3.1.2 Soll-Zustand Definition

Dauer : 2 Stunden

Mit Hilfe der Kritikpunkte und Ergebnisse der Ist-Analyse wird unter Berücksichtigung der Verbesserung des allgemeinen Arbeitsablaufes und der allgemeinen Verwaltung der Domainnamen in Bezug auf die Kunden und Kosten ein Soll-Zustand definiert und schriftlich festgehalten. Während dieser Definition ergeben sich auch bereits wichtige Kernanforderungen an das Produkt.

3.1.3 Erstellung des Pflichtenhefts

Dauer : 5 Stunden

Das Pflichtenheft wird in enger Kollaboration mit Herrn Herzog und Herrn Erdmann erstellt und es beschreibt die Zielbestimmungen, Produktfunktionen, die Produktumgebung und andere wichtige Informationen zur Spezifikation, die nötig sind, um ein detailliertes Pflichtenheft zu erstellen. Bei der Definition der Zielbestimmungen werden u.a. alle Musskriterien aufgelistet, die für das finale Produkt auf jeden Fall vorhanden und implementiert sein müssen. Neben den Musskriterien werden auch die Wunsch und Abgrenzungskriterien verfasst und notiert. Den Wunschkriterien wird aber keine besondere Gewichtung zugeordnet und es werden daher dort nur Kriterien genannt, die für die Anwendung eventuell implementiert werden können, wenn noch Zeit dafür vorhanden ist. Die Abgrenzungskriterien werden notiert und beschreiben lediglich Kriterien, die auf Grund von Zeitmangel und zu hohem Aufwand nicht berücksichtigt werden können. Zusammen mit den aufgeschriebenen Produktfunktionen ergibt sich dann ein Pflichtenheft, auf dessen Basis die spätere Anwendung, das Produkt, und das Fachfeinkonzept bzw. DV-Konzept basiert.

3.2 Entwurfsphase

3.2.1 Erstellung des DV-Konzepts

Dauer : 15 Stunden

3.2.1.1 Einleitung

Das DV-Konzept oder auch Fachfeinkonzept erfordert eine genauere Betrachtung des gesamten Systems und es teilt sich hier in Drei grobe Bereiche ein. Der erste Bereich umfasst die Systembetrachtung, die wiederum die Systemumgebung, den Systementwurf und die grafische Darstellung des Gesamtsystems beinhaltet. Der zweite Bereich, der Entwurf der Systemkomponenten, deckt die Erstellung des Datenbankentwurfs inklusive ER-Modell und Datentypendefinition ab. Es wird hier außerdem die genaue Darstellung des GUI entworfen. Anschließend werden die Beziehungen zwischen den einzelnen Programmteilen und die Abbildung wichtiger Kernfunktionen dargestellt.

3.2.1.2 Umsetzung des DV-Konzepts

Das Fachfeinkonzept beschreibt eine, zu implementierende, Webanwendung und die verwendete Programmiersprache dafür ist Perl, da dafür entsprechende Bibliotheken existieren, die die Entwicklung und Implementierung dieser Webanwendung deutlich vereinfachen und beschleunigen. Mit der hoch performanten Anbindung Mod Perl2 an den Apache2 Webserver Daemon und der Kombination aus Moose, dem postmodernen Objekt-Orientierten System, dem Webframework Capullo und dem Datenbankmanagementsystem Postgresql, können die Ziele und Anforderungen, die an die Webanwendung gestellt werden, ohne weitere Probleme realisiert werden.

Die Gesamtbetrachtung der Softwarekomponenten wird mit Hilfe einer Grafik (2.3 im DV-Konzept) verdeutlicht und soll so dabei helfen, die Übersicht während der Implementierungsphase zu bewahren.

Der Entwurf der Datenbank wird hier speziell mit Hilfe des ER-Modells realisiert (3.1 im DV-Konzept). Die „primary“ keys werden hier durch einen durchgezogenen Unterstrich gekennzeichnet und die „foreign keys“ mit einem unterbrochenen Unterstrich. Durch das ER-Modell wird die spätere Implementation des Datenbankmodells durch die grafische Darstellung der Beziehungen, Kardinalitäten und Primary bzw. Foreign Keys deutlich vereinfacht. In diesem DV-Konzept werden außerdem alle Attributnamen mit entsprechenden Datentypen und ihrer Länge definiert, was die Umsetzung nach SQL für Postgresql deutlich beschleunigt.

Die Entwürfe des GUI werden mit Unterstützung von detaillierten Grafiken, die mit Inkscape, einem SVG Programm, erstellt, was die eigentliche Umsetzung dieses GUI mit HTML, CSS, Javascript und Bildgrafiken fördern soll. Anhand dieser Grafiken wird das GUI erstellt.

Eine weitere Grafik zeigt die einzelnen Beziehungen der einzelnen Programmteile auf (3.3.1 im DV-Konzept), wie sie miteinander agieren, welche Relationen und Abhängigkeiten bestehen und wie der Fluss der Daten läuft.

Abschließend werden mit Hilfe von Struktogrammen wichtige Kernfunktionen abgebildet.

Durch die detaillierte Darstellung des GUI dauerte diese Phase zwei Stunden mehr als veranschlagt.

3.3 Implementationsphase

Dauer : 21 Stunden

Die Implementationsphase folgt direkt auf das Fachfeinkonzept und begonnen wird mit der Erstellung der Datenbankstruktur, da besonders die Kernkomponenten ohne dieses Schema nicht funktionieren. Anschließend werden alle Komponenten implementiert, wobei der Pagecontroller die Rolle des „Routers“ darstellt und alle eingehenden HTTP-Requests der jeweiligen Komponente zuordnet und sie ausführt.

Der Quellcode der Webanwendung „DAdmin“ befindet sich im Ordner `./projekt/var/www/websites/dadmin.de/`. Die Ordnerstruktur des Testsystems wurde hier im „projekt“-Ordner nachgebaut.

3.3.1 Erstellung der Datenbankstruktur

Datei/en erstellt: projekt/var/www/websites/dadmin.de/sql/schema.sql

Die Postgresql Datenbankstruktur wird mit Hilfe des, im DV-Konzept definierten, ER-Modell und der bereits formulierten Attribute mit den entsprechenden Datentypen erstellt. Es wird dazu erst eine reine Textdatei erstellt, in der der reine SQL Quellcode verfasst wird. Um die Eingabe und Ausführung des SQL Quelltextes zu vereinfachen wird das Programm „PGAdmin“, ein Postgresql Verwaltungstool, verwendet.

Es werden folgende Tabellen und Sequenzen erstellt:



Während der Implementierungsphase ist aufgefallen, dass bei bestimmten Abfragen die SQL-Statements so komplex und groß wurden, dass ein bestimmter Teil der Abfragen mit PL/Pgsql in der Postgresql Datenbank ausgelagert werden musste. Das hat den positiven Nebeneffekt der besseren Wartbarkeit und Lesbarkeit des Quellcodes.

3.3.2 Erstellung des Pagecontrollers

Datei/en erstellt: projekt/var/www/websites/dadmin.de/index.pl, projekt/var/www/websites/dadmin.de/config/.config.json

Nachdem die SQL Statements formuliert und ausgeführt wurden, wird mit der Erstellung und Implementierung des Pagecontrollers begonnen. Der Pagecontroller wird mit Hilfe des, im DV-Konzept erarbeiteten, Struktogramms und der dazugehörigen Beschreibung implementiert und erstellt. Es wird hier die „index.pl“-Datei erstellt, die den eigentlichen Controller beinhaltet. Dieser Controller erwartet dann die entsprechenden Page-Komponenten, die auch im DV-Konzept genannt werden, in dem Ordner „pages“ unterhalb des eigentlichen Rootdirectories der Webanwendung. Das Rootdirectory ist in diesem Fall auf dem Testsystem „/var/www/websites/dadmin.de“, aber je nach System kann das überall sein. Der Pagecontroller erstellt dann für jede Page-Komponente eine eigene Instanz und ruft die „run“ Methode auf, die mit jeder Page-Komponente definiert wird. Über eine „init“ Methode wird jeder Page-Komponente außerdem das „Capullo“ Objekt übergeben, damit entsprechende Funktionen zur Verarbeitung von Templates, Sessions, Requestvariablen, Datenbankzugriff uvm. zur Verfügung stehen. Die entsprechenden Entscheidungen, wann welche Komponenten geladen werden, trifft der Pagecontroller anhand der HTTP-Request Variable „page_id“. Der Pagecontroller vergleicht den Wert dieser Variable mit hardcodierten Werten und führt dann die entsprechende Page-Komponente aus. In dem Pagecontroller wird außerdem das Einlesen der Konfigurationsdatei implementiert. Im

Pagecontroller wird außerdem die Datenbankverbindung konfiguriert, so dass jede Page-Komponente darauf zugreifen kann.

3.3.3 Erstellung des GUI

Datei/en erstellt:

in projekt/var/www/websites/dadmin.de/templates : header.admin.tpl.html, header.user.tpl.html, footer.tpl.html, login.tpl.html
in projekt/var/www/websites/dadmin.de/templates/dashboard : admin.view.tpl.html, user.view.tpl.html
in projekt/var/www/websites/dadmin.de/templates/domainname : edit.tpl.html, manage.tpl.html, manage.user.tpl.html, new.tpl.html
in projekt/var/www/websites/dadmin.de/templates/price : edit.tpl.html, edit-group.tpl.html, edit-price.tpl.html, manage.tpl.html, new-group.tpl.html, new-price.tpl.html
in projekt/var/www/websites/dadmin.de/templates/status/ edit.tpl.html, manage.tpl.html, new.tpl.html
in projekt/var/www/websites/dadmin.de/templates/user/ edit.tpl.html, manage.tpl.html, new.tpl.html
in projekt/var/www/websites/dadmin.de/assets/ style.css

Im nächsten Schritt wird mit Hilfe der grafischen Darstellung aus dem Fachfeinkonzept das GUI für die einzelnen Komponenten als auch für die gesamte Webanwendung erstellt. Wie bereits erwähnt wird zur Gestaltung des GUI das Master/Detail Screenpattern verwendet, um die Übersicht und auch die Bedienung zu verbessern. Das GUI wird mit HTML, Javascript und CSS erstellt. Die Darstellung des GUI mit Hilfe von HTML wird in sogenannte Templatedateien ausgelagert, die man im „templates“ Ordner wiederfindet. Dadurch wird eine Trennung von Logik und Darstellung realisiert und verbessert somit die Wartbarkeit der Anwendung. Es wird für jede Komponente außerdem ein eigener Unterordner im Templatesordner erstellt, wo dann komponentenspezifische Templates abgelegt werden. Es wird zunächst das allgemeine Layout erstellt, um dann anschließend für jede einzelne Komponente die speziellen Templates zu erstellen.

Allgemein implementiert jede Page-Komponente drei verschiedenartige Views. Eine View zum Erstellen eines neuen Datensatzes, eine View zum Editieren eines vorhandenen Datensatzes und eine View zum Verwalten/Anzeigen vorhandener Datensätze. Besonders bei der Verwaltung werden je nach Umgebung viele Datensätze angezeigt und um bei dieser Anzahl von Datensätzen die Bedienbarkeit und Datenbankserverperformance zu erhalten, wird mit Hilfe der zusätzlich entwickelten Postgresql Datenbankfunktionen in PL/Pgsql „*_get_pagecount()“ und „*_results(pagenumber integer)“ ein Pagenavigator erstellt, mit welchem man nur einen Ausschnitt der ganzen Daten anzeigen kann. Das Design orientiert sich dabei sehr stark an die erstellten Grafiken im DV-Konzept.

Generell gibt es ein Grundlayout, welches dem Master/Detail Pattern folgt. Dieses Grundlayout definiert einen Kopf, Fuß und Menübereich. Dieses Grundlayout gibt es in zwei Varianten, die sich lediglich in dem Umfang der Menüstruktur bedingt durch die zwei verschiedenen Benutzergruppen, unterscheiden. Die Variante für den Administrator zeigt alle möglichen Menüpunkte an, die auf die Page-Komponenten verweisen, auf die nur ein Benutzer aus der „admin“ Gruppe zugreifen darf und kann. Die andere Variante beinhaltet ein sehr abgespecktes Menü für den Benutzer aus der „user“ Gruppe. Von diesem

Grundlayout ausgenommen sind die Login, Logout und Expirecheck Page-Komponenten, da sie spezielle Funktionen zu erfüllen haben, die mit der Verwaltung an sich nichts zu tun haben.

Die einzelnen Page-Komponenten entscheiden selbst, welche Variante des Grundlayouts geladen werden muss.

3.3.4 Erstellung der einzelnen Page-Komponenten

Datei/en erstellt:

in projekt/var/www/websites/dadmin.de/pages/Dadmin : Login.pm, Logout.pm, Dashboard.pm, Domainname.pm, Price.pm, Status.pm, User.pm

Diese sogenannten „Page-Komponenten“ unterhalb des „pages“ Ordners sind Perl Packages und sie definieren jeweils eine eigene Klasse mit eigenen Methoden und Funktionen. Es wird für jeden Bereich, „dashboard“, „domainname“, „user“, „pricegroup“ usw..., ein eigenes Perl Package erstellt. In jeder Page-Komponente wird die dazugehörige View bzw. das entsprechende Template geladen, verarbeitet und ausgegeben. Jede Page-Komponente greift mit Hilfe des Webframeworks Capullo hauptsächlich auf Datenbank, Templates, Sessionvariablen und HTTP-Requestvariablen zu.

Jede Page-Komponente benutzt die „Authentication“ Funktionen des Capullo Webframeworks, um zu prüfen, ob der zugreifende Client auch die nötigen Rechte hat. Das hat zur Folge, dass im aktuellen Datenbankdesign die Zuordnung der Gruppe redundant für jeden einzelnen Benutzer gespeichert wird. In der aktuellen Umgebung stellt dies aber noch kein Problem dar, da es nur einen Administrator gibt und der Rest der „user“ Gruppe zugeordnet wird und sich diese Zuordnung auch in nächster Zeit nicht ändern wird.

Die Login und Logout Page-Komponenten stellen hier eine kleine Ausnahme dar. In der Login Page-Komponente wird lediglich eine einzige View implementiert, mit der man einen Benutzernamen und Passwort eingeben kann. Für diese Eingaben wird mit Hilfe des Struktogramms aus dem DV-Konzept eine Abfrage implementiert, die prüft, ob der eingegebene Benutzername überhaupt in der Datenbank existiert und ist das der Fall, so wird geprüft, ob das eingegebene Passwort für diesen Benutzernamen auch übereinstimmt. Sind diese beiden Prüfungen abgeschlossen, so wird der Benutzer im Klartext und das Passwort als md5 Hash in einer servergespeicherten Sessiondatei abgelegt, die eine maximale Gültigkeit von 20 Minuten hat. In dem Webbrowser des Benutzers wird lediglich ein Cookie gespeichert, welches nur eine ID beinhaltet. Anhand dieser ID kann das Webframework Capullo auf die servergespeicherten Sessions zugreifen, sofern sie noch vorhanden sind. Anschließend leitet die Login Page-Komponente auf die Dashboard Page-Komponente weiter.

In der Logout Page-Komponente wird lediglich eine Methode implementiert, welche die Werte für den Benutzernamen und dessen Passwort in der servergespeicherten Sessiondatei auf Null setzt und anschließend auf die Login Page-Komponente weiterleitet.

Jede andere Page-Komponente liest die Inhalte der servergespeicherten Sessions, um zu entscheiden, ob ein Benutzer eine Aktion ausführen darf oder nicht.

3.3.5 Erstellung der Expirecheck Komponente

Datei/en erstellt:

in projekt/var/www/websites/dadmin.de/pages/Dadmin : Expirecheck.pm

Wie aus der Gesamtübersicht aus dem DV-Konzept ersichtlich wird, gibt es eine Komponente, ein Perlscript, welches monatlich vom „cron“ Dienst ausgeführt wird. Dieses Script prüft, welche Domainnamen im nächsten und übernächsten Monat auslaufen bzw. automatisch verlängert werden. Es schickt dann jeweils einen Bericht darüber via Email an den Administrator, der alle Domainnamen von allen Benutzern enthält, welche nächsten oder übernächsten Monat auslaufen bzw. verlängert werden. Der entsprechende Benutzer/Kunde bekommt den gleichen Bericht mit dem Unterschied, dass er lediglich seine eigenen Domainnamen aufgelistet bekommt. In dem Bericht selbst werden die entsprechenden Domainnamen, der dazugehörige Benutzer, der Verkaufspreis und das Auslauf bzw. Verlängerungsdatum angezeigt. Diese Darstellung beinhaltet für den Mitarbeiter der TeKoNet Medien GmbH alle nötigen Informationen, um einen Domainnamen zu berechnen oder zu löschen.

In der Planungsphase wurde vereinbart nur die Domainnamen anzuzeigen, die nächsten Monat auslaufen oder verlängert werden. Das jedoch stellte sich während der Implementation als Fehler heraus, da die Kündigungsfrist bei einem Domainname mindestens 4 Wochen beträgt und es passieren kann, dass der Benutzer den Bericht erst dann bekommt, wenn diese Frist schon nicht mehr eingehalten werden kann. Es wurde sich also nachträglich darauf geeinigt diese Berichte für zwei Monate im Voraus zu erstellen.

Für diese Komponente wurde eine PL/Pgsql Funktion „domain_expire_check“ erstellt, welche alle entsprechenden Datensätze zurückgibt.

3.4 Testphase

Dauer : 10 Stunden

3.4.1 Durchführung der Komponententests

Dauer : 2 Stunden

Mit Hilfe der Perl Bibliothek „Test::More“ werden verschiedene Tests geschrieben, die bestimmte, als auch zufällige Daten verwenden, um bestimmte Funktionen innerhalb einer Klasse, die in jeder Page-Komponente definiert ist, zu testen. Die entsprechenden Testdaten werden manuell definiert und implementiert. Diese, auch Unit-Tests genannten, Tests werden parallel zur Erstellung und Implementierung des eigentlichen Quellcodes erstellt und ausgeführt. Bei jeder Änderung werden die

Tests erneut ausgeführt, um zu testen, ob das Verhalten der Methoden der Implementierung und Änderung entsprechend ist.

3.4.2 Durchführung der Integrationstests

Dauer : 2 Stunden

Nach Abschluss jeder entwickelten und implementierten Page-Komponente bzw. nach jedem Abschluss eines Bereichs wird ein Integrationstest durchgeführt, anhand dessen festgestellt und getestet wird, ob die implementierten Komponenten miteinander funktionieren. Der Integrationstest erfolgte hier meistens direkt nach der Implementation und Fertigstellung einer einzelnen Page-Komponente wie zum Beispiel die Status Page-Komponente, die Preisgruppe Page-Komponente usw...Die festgestellten Fehler werden in einer Todo-Liste festgehalten und meistens zeitnah korrigiert.

3.4.3 Durchführung der Systemtests

Dauer : 6 Stunden

Der Systemtest wird nach Abschluss aller anderen Testphasen durchgeführt und beinhaltet hier speziell den Test durch die Mitarbeiter der TeKoNet Medien GmbH und dessen Kunden, die sich dafür freiwillig zur Verfügung gestellt haben. Der Test für das fertige Produkt wird dabei auf der Testumgebung mit Testdaten durchgeführt, um festzustellen, ob die vereinbarten Anforderungen fehlerfrei erfüllt werden.

3.5 Erstellung der Dokumentation

3.5.1 Erstellung des Benutzerhandbuchs

Dauer : 2 Stunden

Der Benutzer in dieser Umgebung ist hauptsächlich ein Mitarbeiter von TeKoNet und dessen Kunden. Da die Kunden aber nur einen minimalen Anteil der Funktionen nutzen können und sollen, wird für diesen Part eine kurze Dokumentation erstellt, die einen schnellen Einstieg in die Bedienung der Webanwendung gibt. Für die Mitarbeiter der TeKoNet Medien GmbH wird eine etwas umfassendere Dokumentation verfasst, aber bedingt durch die Ziele der Webanwendung „Dadmin“ ist die Anwendung leicht zu bedienen und vieles erklärt sich von selbst.

3.5.2 Erstellung der Projektdokumentation

Dauer : 13 Stunden

Neben dem Pflichtenheft und DV-Konzept wird hier abschließend ein Projektbericht oder auch eine Projektdokumentation verfasst.

4. Projektabschluss

4.1 Soll – Ist Vergleich

| Phase | Beschreibung | Soll | Ist |
|-----------------|---|-----------|-----------|
| Analyse | Ist-Analyse formulieren | 2 | 2 |
| | Soll-Zustand definieren | 2 | 2 |
| | Erstellung des Pflichtenhefts | 5 | 5 |
| Entwurf | Erstellung des DV-Konzepts | 13 | 15 |
| Implementierung | Umsetzung der zuvor definierten Strukturen, Ziele und Spezifikationen | 25 | 21 |
| | | | |
| Test | Durchführung Komponententests | 2 | 2 |
| | Durchführung Integrationstests | 2 | 2 |
| | Durchführung Systemtests | 6 | 6 |
| Dokumentation | Erstellung der Projektdokumentation | 10 | 13 |
| | Erstellung des Benutzerhandbuchs | 3 | 2 |
| Gesamt | | 70 | 70 |

4.2 Projektabschluss

Nachdem alle Arbeiten an dem Projekt beendet wurden und die Anforderungen erfüllt sind, wird das Produkt noch von der Testumgebung heraus in das Live-System implementiert. Es werden dann alle nötigen Kunden angelegt und dessen Domainnamen in das neue System migriert. Es erfolgt dann eine kurze Einführung des Produkts und eine Ankündigung via Email an alle Kunden, die Domainnamen bei TeKoNet haben.

Anhänge

-  Anhang A Pflichtenheft
-  Anhang B DV-Konzept
-  Anhang C Testprotokoll
-  Anhang D Benutzerhandbuch
-  Anhang E Glossar