

Projektarbeit zum Rechnerpraktikum aus Programmierung

Alle zum Rechnerpraktikum gehörenden Hausübungen, die Quellcodes, der Link auf den Prototypen sowie diese Arbeit sind im Internet unter <http://rechnerpraktikum.office09.getdesigned.at> erreichbar.

Zugriff auf die unterhalb beschriebene Applikation (Prototype) erlangt man über die Adresse <http://intrastable.office09.getdesigned.at>, der Login erfolgt mit Username „demo“ und Passwort „demo“.

Problemstellung

Das von mir gewählte Problem entstammt meinem Berufsalltag. Ich habe im Jahr 2000 gemeinsam mit 2 Kollegen ein Unternehmen für neue Medien gegründet, das unter anderem klassisches Webdesign sowie dazugehörige Programmierungen in der Skriptsprache PHP anbietet.

Zu jeder Kundenbeziehung gehört die Anbotslegung, also eine bindende und daher notwendigerweise genaue Kostenschätzung für die zu erbringende Leistung. Die gelegten Angebote setzen sich üblicherweise im ersten Teil aus Standardtexten (wie z.B. die File Format Wahl für durch den Kunden zu liefernde Inhalte) sowie projektspezifischen Texten (z.B. Beschreibung einer Adressbuch Applikation für das Intranet des Kunden) zusammen. Im zweiten Teil werden alle Leistungen gegliedert aufgelistet und mit Einzelpreisen versehen, die dann am Ende des Anbots summiert ausgegeben werden. Die Praxis verlangt zusätzlich die Möglichkeit, von der Nettosumme Rabatte abzuziehen, sowie Resellerprovisionen zu berechnen.

Ein wiederholt auftretendes Problem bei der Anbotslegung ist die Unkenntnis des Kunden über die eigenen Anforderungen. Nicht selten erhalten wir Anrufe von Klein- aber auch Mittelbetrieben mit Fragen wie z.B. „Was kostet eine Homepage?“. Diese Frage wird ohne weitere Angabe von Anforderungen gestellt.

In der Theorie folgt an dieser Stelle ein ausführliches Requirements Engineering, in dem die Bedürfnisse des Kunden in Zusammenarbeit aufgedeckt werden und durch die umsetzenden Firma Verbesserungsvorschläge, Vereinfachungen bzw. sinnvolle Erweiterungen in das Projekt eingebracht werden. Der Kunden erhält dadurch mehr Einblick in sein eigenes Arbeitsgebiet und lernt seine Bedürfnisse besser kennen. Weiters versteht er den anfallenden Arbeitsaufwand, wodurch sich die Kosten für ein solches Projekt besser rechtfertigen lassen. Abschließend wird ein Anbot erstellt, das die gemeinsam beschlossenen Komponenten enthält, somit einen genauen Arbeitsumfang definiert und vom Anbot abweichenden Kosten vermeidet.

Die Praxis sieht, wie schon oberhalb beschrieben, anders aus. Der Kunde ist sich seiner Anforderungen nicht annähernd bewusst, in vielen Fällen beschäftigt er sich vor der ersten Kontaktaufnahme wenig bis überhaupt nicht mit der Thematik, oder aber er sucht sich viele Schlagwörter zusammen, die im Falle der Erstellung einer Website mit dieser in Zusammenhang bringt und verlangt basierend darauf ein Anbot.

Als seriöses Unternehmen stehen wir in dieser Situation zwischen zwei Stühlen: Auf der einen Seite, wollen wir keine Kunden durch tiefgehende Fragen verkraulen und ihn zur Konkurrenz vertreiben. Viele Unternehmen verfolgen an dieser Stelle den Ansatz, erst

einmal ein x-beliebiges Angebot durch den Kunden unterschreiben zu lassen und die tatsächliche zu erbringende Leistung erst im Nachhinein zu definieren. Diese Angebote enthalten üblicherweise lediglich Standardleistungen, ohne auf die Bedürfnisse des Kunden einzugehen. Weiters sind diese Leistungen unvollständig, d.h. dass zwar eine Leistung erbracht wird, diese aber ohne zusätzliche (nicht im Erstanbot enthaltene) Leistungen, nicht sinnvoll einsetzbar ist.

Auf der anderen Seite ist es aber zwingend notwendig, für genaue Angebote die Situation des Kunden besser zu kennen. Wir versuchen daher, im Erstgespräch die Basisanforderung herauszufinden, diese durch eigene Recherchen (im Falle der Überarbeitung einer Website z.B. durch Analyse der bisherigen Homepage) zu erweitern und darauf basierend ein Angebot zu erstellen. Dieses Angebot wird dann als Grundlage für weitere Gespräche benutzt, über- und ausgearbeitet.

Ziel muss es also sein, ohne großen Aufwand Angebote erstellen zu können, die auf Standard Vorlagen basieren, aber leicht an die Kundenbedürfnisse angepasst werden können und dennoch den Bezug zu den Vorlagen nicht verlieren, um automatische Textaktualisierungen etc. vornehmen zu können. Preise sollen sich ebenso automatisch aus der Projektplanung übernehmen lassen und je nach Wunsch bis zu einer gewissen Detailtiefe im Angebot aufscheinen.

Weiters sollte die Software zumindest innerhalb des Unternehmens von allen PCs aus nutzbar sein, bevorzugt aber über ein Extranet weltweit zur Verfügung stehen, was bei der Wahl der Umsetzungsplattform eine Webapplikation am sinnvollsten macht.

Basisfunktionen

Im Detail ergeben sich daraus folgenden Basisfunktionen für diese Applikation.

- Angebot (mit eindeutiger ID)
 - anlegen, bearbeiten, Version ablegen, PDF erstellen (drucken)
 - Elemente eines Angebots
 - Textelement bestehend aus Überschrift & Text
 - Preiselement bestehend aus Titel, Beschreibung & Preis
 - Elemente können innerhalb der Angebotsverwaltung neu hinzugefügt oder aus der Projektplanung übernommen werden (siehe unterhalb)
 - Übernahme
 - Automatische Erstellung des Angebots aus Projektplanung
 - Übernahme der Projektelemente in das Angebot
 - Erhaltung des Bezugs eines Angebotslements zum Projektelement
 - Möglichkeit zur Modifikation eines Elements ohne Änderung des zugrunde liegenden Projektelements
 - Suche nach
 - Angebotsnummer
 - Kunde
 - Status: offen / unterzeichnet

Der Standardweg zu Erstellung eines neuen Angebots geht über die Projektplanung (nicht Teil dieser Arbeit). Innerhalb der Projektplanung lässt sich ein Projekt beliebig tief verfeinern und der Zeitaufwand bzw. die Kosten schätzen. Diese Informationen bilden die Basis eines

Anbots, welches dann angepasst wird. Dabei kann definiert werden, welche Detailebenen im Anbot aufscheinen und zu welchen Preisen bestimmte Leistungen angeboten werden. Nicht in der Projektplanung enthaltene Posten lassen sich manuell ergänzen (z.B. Lizenzkosten für das Framework).

Problemlösung

Die benötigte Webapplikation wird in der Skriptsprache PHP > 4.3 unter Apache 2.x entwickelt. Als Datenbank kommt MySQL 3.2 bzw 4.x zum Einsatz. Zusätzlich zu herkömmlichen Request Techniken wird für bestimmte Funktionalitäten innerhalb der Applikation AJAX (Asynchronous JavaScript and XML) eingesetzt (mittels der Open Source PHP Klasse XAJAX).

Die in Folge beschriebenen Tabellen, Beziehungen sowie Klassen und Methoden sind teilweise theoretischer Natur, da der Prototyp selbst prozedural entwickelt ist (unter Einsatz kleinerer Funktionen sowie teilweise externer Open Source Klassen).

Beschreibung der benötigten Datenbank Tabellen.

Basis für die Anbotslegung sind die Kunden, die in der Tabelle „customer“ erfasst werden. Sie enthält, nebst einer eindeutigen ID, Kontaktinformationen wie Name, Telefon etc. sowie Flags in Form von boolean Feldern, die z.B. Aufschluss darüber geben, ob Umsatzsteuer verrechnet wird, etc..

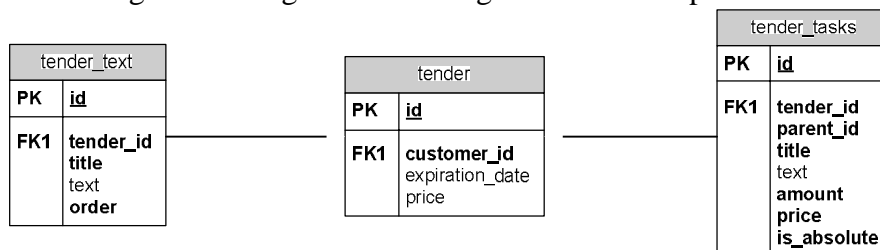
customer	
PK	<u>id</u>
	customer_nr name address1 address2 zip city country

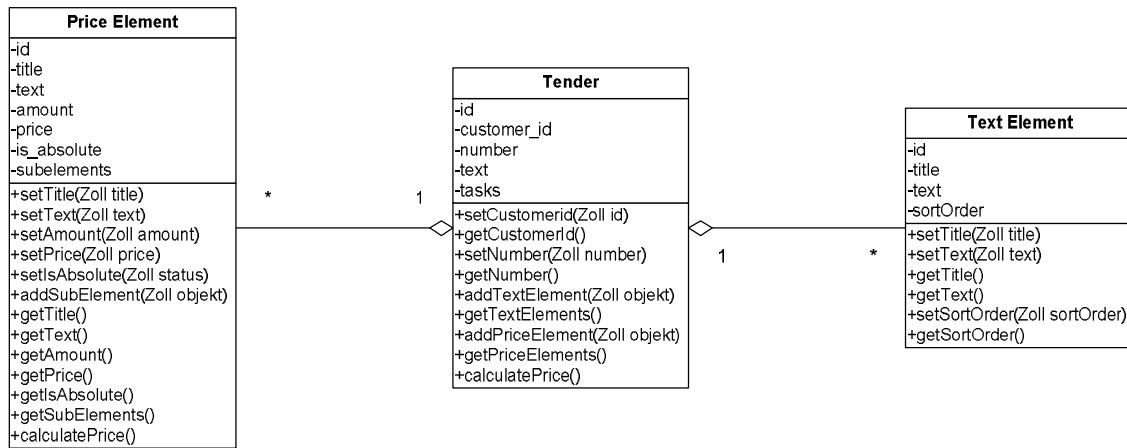
tender	
PK	<u>id</u>
FK1	customer_id expiration_date price

Die Textelemente bestehen aus Überschrift, Text und Reihenfolge innerhalb des Anbots und stehen in einem 1:n Verhältnis (mehrere Textelemente pro Anbot) zu diesem.

Die Preiselemente enthalten wie auch die Textelemente Überschrift (=Titel) und Text (=Beschreibung) sowie zusätzlich Einzelnettopreis und Menge. Da Preise hierarchisch strukturiert sind, benötigen sie zusätzlich die Information, ob und wenn ja, welchem anderen Preiselement sie untergeordnet sind. Dieses rekursive Verhältnis wird über das Feld „parent_id“ abgebildet. Befindet sich das Preiselement auf oberster Ebene, enthält diese Spalte den Wert „0“, ab der 2ten Hierarchieebene die ID des übergeordneten Tasks. Da jedes Preiselement aber selbst eine Preis („price“) enthalten kann, muss noch über ein spezielles Flag („is_absolute“) definiert sein, ob der Preis absolut ist und somit hierarchisch untergeordneten Preiselemente overruled (Spaltenwert ist true), oder der Preis ein Zwischenergebnis als Summe der darunter liegenden Preiselemente ist (Spaltenwert ist false).

Daraus ergibt sich folgendes ER-Diagramm für das Speichern der Angebote:





Data Access
+loadTender(Zoll tenderId : int = 0)
+saveTender(Zoll anbotObject : object)

Diese Voraussetzungen erfüllen viele Linux Standardinstallation. Weiters können unter Windows fertige AMP Pakete (Apache, MySQL, PHP) benutzt werden, die üblicherweise nach automatischer Installation mit einem einzigen Klick startbar sind. Zusätzliche bieten die meisten Interprovider entsprechende, kostengünstige Pakete an.

Um die Applikation zu installieren, ist ein eigener (virtueller) Host notwendig. Parallel mit anderen Applikationen im selben Document Root ist diese Anwendung nur mit Mehraufwand zu betreiben, da zahlreiche Verzeichnisse mit der fremden Anwendung kollidieren könnten und das Apache Modul `mod_rewrite` viele Datei Zugriffe noch vor der tatsächlichen Ausführung bzw. dem Ladevorgang umschreibt.

Sobald die oberhalb beschriebenen Anforderungen gestellt sind, beginnt die Installation mit dem Ablegen der zur Anwendung gehörenden Dateien im Document Root des Apache Hosts, entweder durch direkten Zugriff auf das Filesystem oder durch Upload der Dateien zum Provider z.B. per FTP oder SCP. Anschließend sollten im Document Root folgende Dateien bzw. Verzeichnisse vorhanden sein:

Document Root /	Beschreibung
<code>content / [DIR]</code>	enthält Content Dateien, die die Inhalte der Appl. enthalten
<code>functions / [DIR]</code>	enthält ausgelagerte PHP Funktionen, die von mehreren Content Seiten genutzt werden
<code>graphics / [DIR]</code>	enthält alle Grafiken im GIF und JPEG Format
<code>javascript / [DIR]</code>	enthält ausgelagerten Javascript Code, der von mehreren Content Seiten genutzt wird
<code>raxis3 / [DIR]</code>	enthält die Backend Engine, die Content und Templates mergt, DB Verbindungen aufbaut und viele zusätzliche kleine Tasks übernimmt
<code>templates / [DIR]</code>	enthält alle Template Dateien (=HTML Rahmen), in die die Content Dateien vor Versendung an den User eingefügt werden
<code>xajax / [DIR]</code>	enthält XAJAX Funktionen zum schnelleren Bearbeiten größerer Angebote
<code>.htaccess</code> <code>index.php</code>	Enthält die <code>mod_rewrite</code> Anweisungen für den Apache Webserver Zieldatei, auf die alle Content Seiten per <code>mod_rewrite</code> umgelenkt werden. Sie bindet die Backend Engine ein (raxis3) und enthält deren Konfiguration

Nachdem die Dateien erfolgreich an die vorgesehenen Positionen kopiert worden sind, muss die Datenbank eingerichtet werden.

Datenbank

Folgende Tabellen sind für die Anbots Applikation notwendig:

- customer
- tender
- tender_tasks
- tender_text

Um die Tabelle anlegen zu können, wir eine eigene Datenbank benötigt. Zu beachten ist, dass der benutzte MySQL user über die Rechte verfügen muss, eine Datenbank anzulegen sowie Tabelle anzulegen. Die Befehle können entweder über die Command Line Client Applikation

der MySQL Datenbank oder ein Webinterface wie z.B. phpMyAdmin eingegeben werden. Die Datenbank wird mit folgendem Befehl angelegt:

```
create database 'tenderapp';
```

Nachdem die Datenbank erfolgreich angelegt wurde, müssen die oberhalb beschriebenen Tabellen erzeugt werden:

```
CREATE TABLE `customer` (  
  `id` smallint(6) NOT NULL auto_increment,  
  `customer_nr` varchar(100) default NOT NULL,  
  `name` varchar(160) default NULL,  
  `address1` varchar(100) default NULL,  
  `address2` varchar(100) default NULL,  
  `zip` varchar(8) default NULL,  
  `city` varchar(50) default NULL,  
  `country` varchar(35) default NULL,  
  PRIMARY KEY (`unique_id`)  
) TYPE=MyISAM;
```

```
CREATE TABLE `tender` (  
  `id` int(11) NOT NULL auto_increment,  
  `customer_id` int(11) NOT NULL default '0',  
  `expiration_date` timestamp(14) NOT NULL,  
  `price` float(10,3) NOT NULL default '0.000',  
  PRIMARY KEY (`id`)  
) TYPE=MyISAM;
```

```
CREATE TABLE `tender_tasks` (  
  `id` int(11) NOT NULL default '0',  
  `title` varchar(30) NOT NULL default "",  
  `text` text() NOT NULL default "",  
  `amount` float(5,2) NOT NULL default '0.00',  
  `price` float(10,5) NOT NULL default '0.00000',  
  `absolute_price` enum('0','1') NOT NULL default '0',  
  `parentId` int(11) NOT NULL default '0',  
) TYPE=MyISAM;
```

```
CREATE TABLE `tender_text` (  
  `id` int(11) NOT NULL auto_increment,  
  `title` varchar(50) NOT NULL default "",  
  `text` text NOT NULL,  
  `sortOrder` int(11) NOT NULL default '0',  
  PRIMARY KEY (`id`)  
) TYPE=MyISAM;
```

Die erstellten Tabellen sollten nun in der erzeugten Tabelle sichtbar sein, zu prüfen ist dies mit dem Befehl:

```
show tables;
```

Zuletzt muss nun ein MySQL Benutzer („tenderapp“) angelegt werden, mit dem die Applikation auf die Datenbank zugreifen kann. Dieser User sollte idealerweise nur über eingeschränkte Rechte verfügen und nur von eingeschränkten Hosts aus Zugriff erhalten. Im unterhalb beschriebenen SQL Befehl darf der anzulegende User nur am localhost zugreifen. Außerdem verfügt er nur über insert, update, delete und select Rechte auf die anzulegende DB. Dazu sind 2 Befehle notwendig:

```
GRANT USAGE ON * . * TO "tenderapp"@"localhost" IDENTIFIED BY "pw";
```

```
GRANT SELECT , INSERT , UPDATE , DELETE ON `tenderapp` . * TO  
"tenderapp"@"localhost";
```

Der erste Befehl legt den User an und gestattet Zugriffe von / auf localhost, der zweite Befehl gewährt die notwendigen Zugriffsrechte auf die angelegte Datenbank „tenderapp“.

Alle SQL Befehle können auch (einen berechtigten User vorausgesetzt), über die beige stellte Daten tenderapp_sql_installation.sql Datei angelegt werden.

Abschließend sind nun der Name der angelegten Datenbank sowie der zu benutzende User mit dem dazugehörigen Passwort noch in der Applikation zu konfigurieren. Dies geschieht in der Datei index.php, auf die mittels mod_rewrite alle Zugriffe umgeschrieben werden. In dieser Datei ist folgende Zeilen entsprechend anzupassen:

```
$rx3->setDbConfig("localhost", "tenderapp", "tenderapp", "pw");
```

Die Methode setDbConfig() des rx3 Objekts (Instanz der Backend Klasse raxis3) bekommt dabei 4 Attribute übergeben: An erster Stelle den Host der MySQL Datenbank (in diesem Fall localhost), an zweiter Stelle den Namen der Datenbank, an dritter Stelle den Username und an vierter Stelle das Passwort.

Weitere Konfigurationen sind nicht notwendig.

Wartung

Die Wartung der Applikation beschränkt sich auf gelegentliche Optimierungen der Datenbank sowie regelmäßigem Backup.

Die Datenbank wird mit folgendem Befehl optimiert:

```
OPTIMIZE TABLE `tender` , `tender_text` , `tender_tasks` , `customer`
```

Das Backup beschränkt sich, sofern von den Applikationsdateien einmal eine Sicherung (z.B. auf CD-Rom) durchgeführt worden ist, auf die Sicherung der Datenbank in Form eines SQL Dumps der Tabellen. Dies hat den Vorteil, dass die Daten in einem Textformat unverschlüsselt abgelegt sind und so im Notfall mit jedem Text Editor bearbeitet werden können und außerdem im Falle eines Backups auf externe (über das Internet verbundene) Systeme bei Nutzung eines differentiellen Transfersystems wie z.B. rsync nur sehr geringe Datenmenge übertragen werden.

Weitere Informationen im Internet

Apache Webserver: <http://www.apache.org>

Modul mod_rewrite: http://httpd.apache.org/docs/1.3/mod/mod_rewrite.html

PHP: <http://www.php.net>

XAJAX: <http://www.xajaxproject.org/>

MySQL: <http://www.mysql.com>

Rsync: <http://samba.anu.edu.au/rsync/>