



**HTW** Chur  
Hochschule für Technik und Wirtschaft

Fachhochschule Ostschweiz  
University of Applied Sciences

---

# ENTWURFSDOKUMENTATION

*WebCalculator*

*Version 1.3*

*Demarmels Mirko [De],  
Hassler Reinhard [Ha],  
Heinemeyer Micheael [He],  
Morais Santos Paulo [Sa],  
Pircher Ueli [Pi]*

*I Änderungsgeschichte*

<b><u>Datum</u></b>	<b><u>Version</u></b>	<b><u>Autor</u></b>	<b><u>Beschreibung</u></b>
25.08.2007	1.3	Ha, Sa	Dokument überarbeitet
26.07.2007	1.2	Ha, Sa, Pi	Dokument überarbeitet
25.07.2007	1.1	De, Ha, Sa, Pi	Dokument überarbeitet
20.07.2007	1.0	De	Dokument erstellt

*II Inhaltsverzeichnis*

<b>1</b>	<b>EINFÜHRUNG</b>	<b>6</b>
1.1	Überblick	6
1.2	Inhalt	6
1.3	Definitionen und Abkürzungen	6
1.4	Referenzen	6
<b>2</b>	<b>HINWEISE ZUM DESIGN</b>	<b>7</b>
2.1	Annahmen	7
2.2	Einschränkungen	7
2.3	Systemumgebung	7
<b>3</b>	<b>ARCHITEKTUR</b>	<b>8</b>
3.1	Überblick	8
3.2	Datenbankstruktur	11
<b>4</b>	<b>HIGH - LEVEL DESIGN</b>	<b>12</b>
4.1	Übersicht	12
4.2	Modul Home	13
4.3	Modul Employees	13
4.4	Modul Material	13
4.5	Modul Engines	14
4.6	Modul Calculation	14
4.7	Modul Products	14
<b>5</b>	<b>LOW - LEVEL DESIGN</b>	<b>15</b>
5.1	Modul Home	15
5.2	Modul Employees	15
5.3	Modul Engines	16

---

<b>5.4</b>	<b>Modul Material .....</b>	<b>16</b>
<b>5.5</b>	<b>Modul Calculation.....</b>	<b>17</b>
<b>5.6</b>	<b>Modul Products .....</b>	<b>17</b>
<b>6</b>	<b>DESIGN DER BENUTZERSCHNITTSTELLE.....</b>	<b>19</b>
<b>A</b>	<b>REFERENZEN .....</b>	<b>21</b>

### *III Abbildungsverzeichnis*

<b>Abbildung 1: Funktionsweise des WebCalculators.....</b>	<b>10</b>
<b>Abbildung 2: ER-Diagramm .....</b>	<b>11</b>
<b>Abbildung 3: Klassendiagramm .....</b>	<b>12</b>

### *IV Tabellenverzeichnis*

<b>Tabelle 1: High - Level Home - Modul.....</b>	<b>13</b>
<b>Tabelle 2: High - Level Employee - Modul .....</b>	<b>13</b>
<b>Tabelle 3: High - Level Material - Modul.....</b>	<b>13</b>
<b>Tabelle 4: High - Level Engines - Modul .....</b>	<b>14</b>
<b>Tabelle 5: High - Level Calculation - Modul .....</b>	<b>14</b>
<b>Tabelle 6: High - Level Products - Modul.....</b>	<b>14</b>
<b>Tabelle 7: Low - Level Home - Modul .....</b>	<b>15</b>
<b>Tabelle 8: Low - Level Employee - Modul .....</b>	<b>15</b>
<b>Tabelle 9: Low - Level Engines - Modul .....</b>	<b>16</b>
<b>Tabelle 10: Low - Level Material - Modul .....</b>	<b>17</b>
<b>Tabelle 11: Low - LevelCalculationI - Modul.....</b>	<b>17</b>
<b>Tabelle 12: Low - Level Products - Modul .....</b>	<b>18</b>
<b>Tabelle 13: Seitenaufbau (Theorie).....</b>	<b>19</b>
<b>Tabelle 14: Seitenaufbau (Praxis) .....</b>	<b>19</b>

# 1 EINFÜHRUNG

## 1.1 Überblick

WebCalculator ist ein Onlinerechner zur Produktkalkulation. Das System wird auf einem Webserver installiert und arbeitet datenbankbasiert. Der Benutzer bedient das System mittels Webbrowser.

## 1.2 Inhalt

In dieser Entwurfsdokumentation sind die definierten Module des WebCalculators, ihre Beziehungen und Funktionen dokumentiert.

WebCalculator soll objektorientiert programmiert sein. Für die jeweiligen Seiten Home, Material, Mitarbeiter, Maschinen und Kalkulation sollen einzelne Module erstellt werden. In den jeweiligen Modulen sind alle Funktionen zusammengefasst, die für das User-Interface und die Datenbank-Manipulationen im entsprechenden Teilbereich notwendig sind.

## 1.3 Definitionen und Abkürzungen

Apache 2	Webserver
Eclipse 3.2	Open Source Framework, Entwicklungsumgebung für Java
HTML	Hypertext Markup Language
Java	objektorientierte Programmiersprache von Sun Microsystems
JEdit	Texteditor
Linux 2.6	Open Source Betriebssystem
MacOSX	Betriebssystem von Apple
MySQL	Open Source Datenbankverwaltungssystem
PEAR	PHP Extension and Application Repository
PHP 5	Hypertext Preprocessor, Programmiersprache für dynamische Webseiten
Windows	Betriebssystem von Microsoft

## 1.4 Referenzen

siehe Anhang A auf Seite 21 dieses Dokumentes.

## **2 HINWEISE ZUM DESIGN**

### **2.1 Annahmen**

Das Ziel des WebCalculator ist eine unabhängige Software zu realisieren. Zum Gebrauch der Software benötigt man Windows, Linux oder MacOS X als Betriebssystem. Für den webbasierten Gebrauch ist Internetexplorer ab 6.x, Firefox ab 1.5.x, Opera ab 9 oder Safari 2.0 vorgesehen.

### **2.2 Einschränkungen**

Die Software wird templatebasiert programmiert. Je grösser die Datenbanken werden, desto mehr Festplattenspeicher ist von Nöten.

### **2.3 Systemumgebung**

Die Software läuft auf PC's, Notebook's oder anderen mit einem der obgenannten Betriebssysteme ausgestatteten Geräten.

## 3 ARCHITEKTUR

### 3.1 Überblick

Die Software vom WebCalculator soll so aufgebaut werden, dass eine einzige Datei (in unserem Fall index.php) die ganze Software steuert. Somit wird immer die gleiche Datei aufgerufen. Je nachdem was für Parameter dieser Datei übergeben werden (per GET, POST oder SESSION), wird das entsprechende Modul als Objekt eingebunden und gestartet.

Des Weiteren soll die ganze Anwendung templatebasiert programmiert werden. Das heisst, dass PHP - Code strikt vom HTML - Code getrennt wird. Ebenfalls soll, soweit wie möglich, das Cascading Style Sheet (CSS) und das Java - Script (JS) vom HTML - Code getrennt werden. Der Vorteil davon ist eine übersichtliche Struktur, welche das spätere Bearbeiten oder Ändern einfacher macht.

Es werden insgesamt zwei 3rd-part Objekte verwendet, welche nicht von uns selber programmiert werden. Diese werden von PEAR [1] (<http://pear.php.net>) zur Verfügung gestellt und dienen dazu das templatebasierte Programmieren zu ermöglichen (Objekt: Sigma) und vereinfachen den Umgang mit der MySQL-Datenbank (Objekt: MDB2).

Die Software soll in erster Linie für PHP 5 optimiert werden. Es soll aber mit einem kleinen Trick gewährleistet werden, dass WebCalculator auch auf PHP 4 läuft, ohne dass die ganze Software zweimal programmiert werden muss.

Dieser Trick bezieht sich auf den Klassen-Konstruktor. In PHP 4 heisst der Konstruktor gleich wie die Klasse, während in PHP 5 \_\_constructor für den Konstruktor einer Klasse steht. Das Problem wird gelöst indem man beide Konstruktoren in die Klasse einbindet. Beim PHP 4 Konstruktor verweist man direkt auf den PHP 5 Konstruktor, während dieser dann der eigentliche Konstruktor ist. Mit diesem Trick werden bei beiden PHP Versionen \_\_constructor aufgerufen.

Beispiel einer lauffähigen Klasse (Test.class.php) in PHP 4 und PHP 5:

```
<?php

class Test
{
    function Test ()
    {
```

```
        $this->__constructor();
    }

    function __constructor ()
    {
        /* hier kommt der Konstruktor für beide Versionen rein */
    }
}

?>
```

Die Verzeichnisstruktur der Software soll so logisch wie möglich aufgebaut werden.

/	Root-Ordner von WebCalculator
/config	Hier kommt die Konfigurations-Datei rein, welche einmalig bei der Installation bearbeitet werden muss.
/lib	Dieser Ordner enthält alle Library-Klassen sowie Include-Dateien.
/lib/PEAR/*	Hier befinden sich die 3rd-part-Klassen von PEAR.
/modules	Der Ordner enthält die PHP-Quelldateien aller Module.
/template/default	Hier befinden sich die HTML-Quelldateien der Software.
/template/default/images	In diesem befinden sich alle Bilder/Grafiken, welche die Software verwendet.
/template/default/includes	Dieser Ordner enthält alle CSS- und JS-Dateien, welche dann in HTML eingebunden werden.
/template/default/modules	Hier befinden sich die HTML-Quelldateien aller Module. Jedes Modul erhält einen eigenen Ordner.

Dadurch dass die Software templatebasiert ist, wäre es möglich weitere Designs zu entwickeln ohne das bestehende zu löschen. Weitere Designs werden in einem eigenen Ordner in /template/ gespeichert, und haben die gleichen Unterordner wie das Default-Design /template/default/\*.

Entwickelt wird WebCalculator mit „Zend Development Environment“ 5.2 (30-Tage Testversion).

Die nachfolgende Grafik stellt die Funktionsweise des WebCalculators dar:

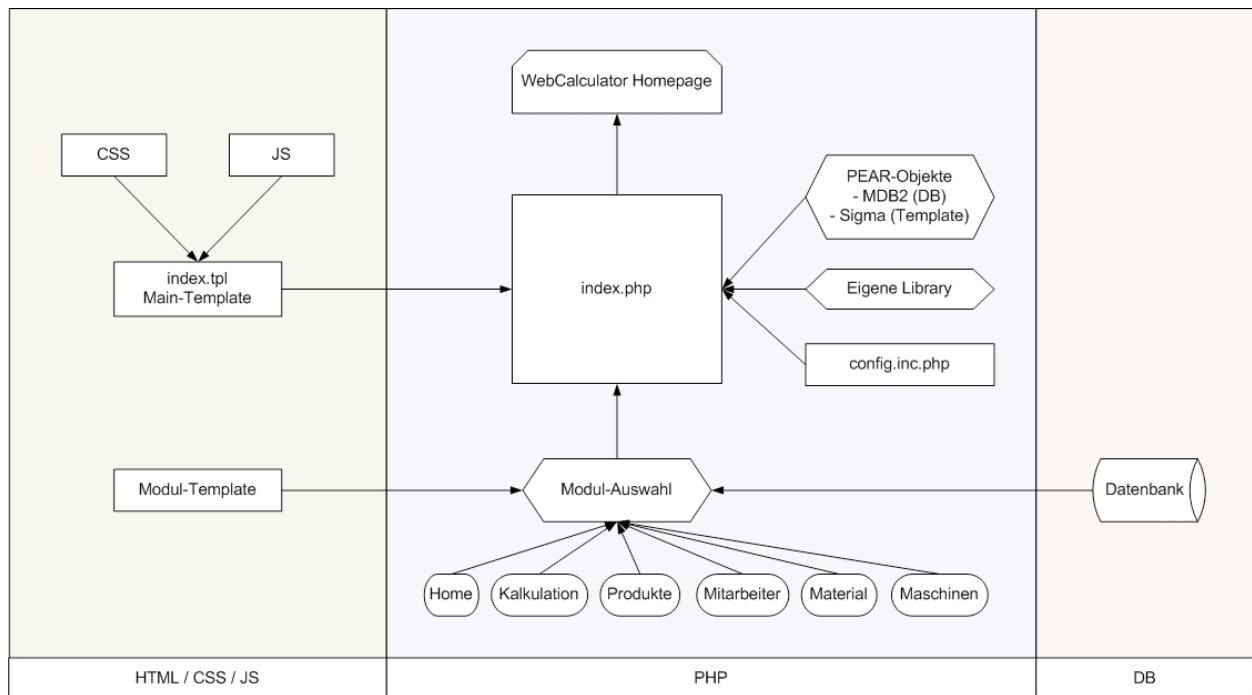


Abbildung 1: Funktionsweise des WebCalculators

### 3.2 Datenbankstruktur

Als Datenbank wird MySQL mit dem Typ MyISAM verwendet, welches keine Fremdschlüssel unterstützt. Deshalb müssen die Eigenschaften der Fremdschlüssel so gut wie möglich durch PHP ersetzt werden. Aus diesem Grund soll die Datenbankstruktur so programmierfreundlich wie möglich gelöst werden.

Die Module „products“, „employees“, „engines“ und „material“ werden in separaten Tabellen gespeichert. Die Tabelle „product\_resources“ verwaltet welches Produkt welche Ressourcen verwendet. D.h. sie verknüpft die vier anderen Tabellen miteinander. Bei der Tabelle „product\_resources“ muss unbedingt beachtet werden, dass die Spalte „res\_id“ Fremdschlüssel von drei verschiedenen Tabellen („employees“, „engines“ und „material“) verwaltet. Welcher Fremdschlüssel zu welcher gehört regelt die Spalte ‚type‘. Auf den ersten Blick erscheint diese Verknüpfung etwas kompliziert und unlogisch, sie stellt programmiertechnisch aber eine optimale Lösung dar.

Die folgende Abbildung zeigt das ER-Diagramm:

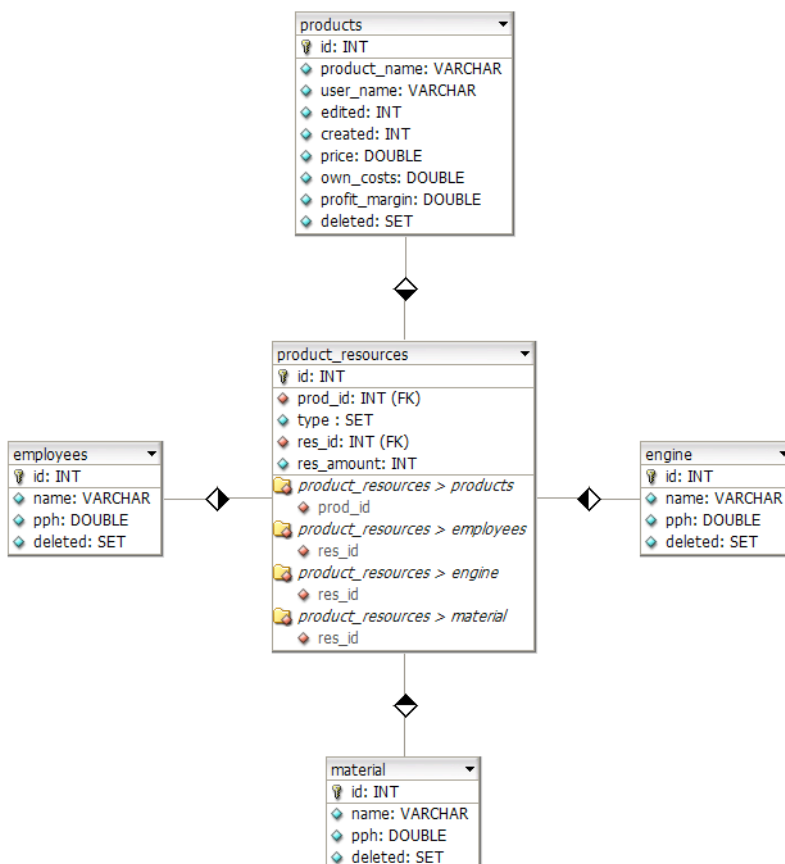


Abbildung 2: ER-Diagramm

## 4 HIGH - LEVEL DESIGN

### 4.1 Übersicht

Die folgenden Kapitel geben eine Übersicht der Funktionen aller Module. Die nachfolgende Grafik gibt einen Überblick über die Module und enthaltenen Funktionen.

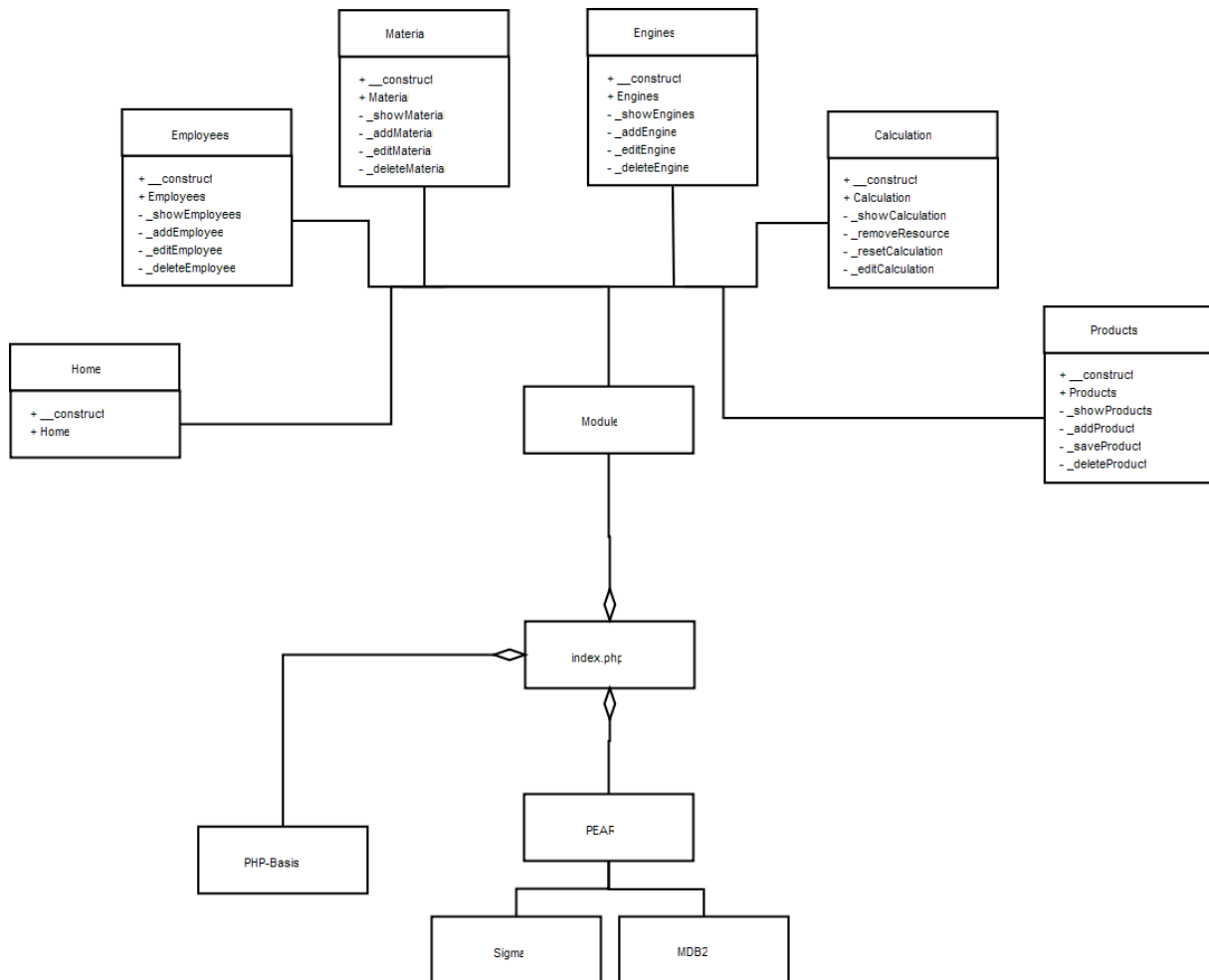


Abbildung 3: Klassendiagramm

## 4.2 Modul Home

File: Home.class.php

Beschreibung: Modul Home beinhaltet sämtliche Funktionen um die Homepage darzustellen.

Klasse:	Funktion:
Home	__construct falls php >=5
	Home

Tabelle 1: High - Level Home - Modul

## 4.3 Modul Employees

File: Employees.class.php

Beschreibung: Modul Employees beinhaltet sämtliche Funktionen um die Datensätze Mitarbeiter in der Datenbank zu bearbeiten und im Browser darzustellen.

Klasse:	Funktion:
Employees	__construct falls php >=5
	Employees
	_showEmployees
	_addEmployee
	_editEmployee
	_deleteEmployee

Tabelle 2: High - Level Employee - Modul

## 4.4 Modul Material

File: Material.class.php

Beschreibung: Modul Material beinhaltet sämtliche Funktionen um die Datensätze Material in der Datenbank zu bearbeiten und im Browser darzustellen.

Klasse:	Funktion:
Material	__construct falls php >=5
	Material
	_showMaterial
	_addMaterial
	_editMaterial
	_deleteMaterial

Tabelle 3: High - Level Material - Modul

#### 4.5 Modul Engines

File: Engines.class.php

Beschreibung: Modul Engines beinhaltet sämtliche Funktionen um die Datensätze Maschinen in der Datenbank zu bearbeiten und im Browser darzustellen.

Klasse:	Funktion:
Engines	__construct falls php >=5
	Engines
	_showEngines
	_addEngine
	_editEngine
	_deleteEngine

Tabelle 4: High - Level Engines - Modul

#### 4.6 Modul Calculation

File: Calculation.class.php

Beschreibung: Modul Calculation beinhaltet sämtliche Funktionen um eine Kalkulation zu erstellen und im Browser darzustellen.

Klasse:	Funktion:
Calculation	__construct falls php >=5
	Calculation
	_showCalculation
	_removeResource
	_resetCalculation
	_editCalculation

Tabelle 5: High - Level Calculation - Modul

#### 4.7 Modul Products

File: Products.class.php

Beschreibung: Modul Products beinhaltet sämtliche Funktionen um die Datensätze Produkte in der Datenbank zu bearbeiten und im Browser darzustellen.

Klasse:	Funktion:
Products	__construct falls php >=5
	Products
	_showProducts
	_addProduct
	_saveProduct
	_deleteProduct

Tabelle 6: High - Level Products - Modul

## 5 LOW - LEVEL DESIGN

In diesem Kapitel werden alle WebCalculator Softwaremodule beschrieben. Insbesondere werden alle im jeweiligen Modul enthaltenen Funktionen aufgelistet und erklärt. Der Gesamte Code eines Moduls befindet sich jeweils in einem einzigen File.

### 5.1 Modul Home

Funktionsname	Aufruf	Beschreibung
__construct	Beim Auswählen des Menus Home in einer der WebCalculator-Seiten	Konstruktor, ruft Funktion Home auf (PHP >= 5)
Home	durch __construct	Konstruktor für PHP < 5

Tabelle 7: Low - Level Home - Modul

### 5.2 Modul Employees

Funktionsname	Aufruf	Beschreibung
__construct	Beim Auswählen des Menus Mitarbeiter in einer der WebCalculator-Seiten	Konstruktor, ruft Funktion Employees auf (PHP >= 5)
Employees	durch __construct	Konstruktor für PHP < 5  Ruft je nach Auswahl mittels Button o. ä. die Funktionen zum Mitarbeiter ansehen (_showEmployees), neu erfassen (_addEmployee), ändern (_editEmployee) oder löschen (_deleteEmployee) auf und passt jeweils die Menus an, defaultmässig wird _showEmployees aufgerufen
_showEmployees	durch Funktion Employees	führt Datenbankabfrage nach allen erfassten Mitarbeitern aus und stellt die Datensätze dar
_addEmployee	durch Funktion Employees	Stellt Eingabefelder für die Erfassung neuer Mitarbeiter dar, tätigt Datenbankeintrag
_editEmployee	durch Funktion Employees	Stellt Eingabefelder für die Bearbeitung eines Datensatzes Mitarbeiter dar, tätigt Datenbankeintrag
_deleteEmployee	durch Funktion Employees	Löscht Datensatz Mitarbeiter aus der Datenbank

Tabelle 8: Low - Level Employee - Modul

### 5.3 Modul Engines

Funktionsname	Aufruf	Beschreibung
__construct	Beim Auswählen des Menus Maschinen in einer der WebCalculator-Seiten	Konstruktor, ruft Funktion Engines auf (PHP >= 5)
Engines	durch __construct	Konstruktor für PHP < 5  Ruft je nach Auswahl mittels Button o. ä. die Funktionen zum Maschinen ansehen (_showEngines), neu erfassen (_addEngines), ändern (_editEngines) oder löschen (_deleteEngines) auf und passt jeweils die Menus an, defaultmässig wird _showEngines aufgerufen
_showEngines	durch Funktion Engines	führt Datenbankabfrage nach allen erfassten Maschinen aus und stellt die Datensätze dar
_addEngine	durch Funktion Engines	Stellt Eingabefelder für die Erfassung neuer Maschinen dar, tätigt Datenbankeintrag
_editEngine	durch Funktion Engines	Stellt Eingabefelder für die Bearbeitung eines Datensatzes Maschine dar, tätigt Datenbankeintrag
_deleteEngine	durch Funktion Engines	Löscht Datensatz Maschine aus der Datenbank

Tabelle 9: Low - Level Engines - Modul

### 5.4 Modul Material

Funktionsname	Aufruf	Beschreibung
__construct	Beim Auswählen des Menus Material in einer der WebCalculator-Seiten	Konstruktor, ruft Funktion Material auf (PHP >= 5)
Material	durch __construct	Konstruktor für PHP < 5  Ruft je nach Auswahl mittels Button o. ä. die Funktionen zum Material ansehen (_showMaterial), neu erfassen (_addMaterial), ändern (_editMaterial) oder löschen (_deleteMaterial) auf und passt jeweils die Menus an, defaultmässig wird _showMaterial aufgerufen
_showMaterial	durch Funktion Material	führt Datenbankabfrage nach allem erfassten Material aus und stellt die Datensätze dar
_addMaterial	durch Funktion Material	Stellt Eingabefelder für die Erfassung neuen Materiales dar, tätigt Datenbankeintrag

_editMaterial	durch Funktion Material	Stellt Eingabefelder für die Bearbeitung eines Datensatzes Material dar, tätigt Datenbankeintrag
_deleteMaterial	durch Funktion Material	Löscht Datensatz Material aus der Datenbank

Tabelle 10: Low - Level Material - Modul

## 5.5 Modul Calculation

Funktionsname	Aufruf	Beschreibung
__construct	Beim Auswählen des Menus Kalkulation in einer der WebCalculator-Seiten	Konstruktor, ruft Funktion Calculation auf (PHP >= 5)
Calculation	durch __construct	Konstruktor für PHP < 5  Ruft je nach Auswahl mittels Button o. ä. die Funktionen zum Kalkulationsmenu anzeigen (_showCalculation), Ressource aus Kalkulation entfernen (_removeResource), Ressourcen-Auswahl zurücksetzen (_resetCalculation), defaultmässig wird _showCalculation aufgerufen
_showCalculation	durch Funktion Calculation	führt Datenbankabfrage nach allen erfassten Ressourcen durch sowie Produktdetails, stellt das gesamte Kalkulationsmenu dar und berechnet die Kennzahlen
_removeResource	durch Funktion Calculation	Löscht ausgewählte Ressource aus der Kalkulation
_resetCalculation	durch Funktion Calculation	Setzt ausgewählte Ressourcen und Mengen zurück
_editCalculation	durch Funktion Calculation	Ruft eine gespeicherte Kalkulation zur Bearbeitung auf

Tabelle 11: Low - LevelCalculationI - Modul

## 5.6 Modul Products

Funktionsname	Aufruf	Beschreibung
__construct	Beim Auswählen des Menus Produkte in einer der WebCalculator-Seiten sowie beim Speichern einer erstellten Kalkulation	Konstruktor, ruft Funktion Products auf (PHP >= 5)
Products	durch __construct	Konstruktor für PHP < 5

		Ruft je nach Auswahl mittels Button o. ä. die Funktionen zum Produktemenu anzeigen ( <code>_showProducts</code> ), neues Produkt sowie ein bearbeitetes Produkt in Datenbank ablegen ( <code>_addProduct</code> und <code>_saveEditedProduct</code> ), Produkt löschen ( <code>_deleteProduct</code> ), defaultmässig wird <code>_showProducts</code> aufgerufen
<code>_showProducts</code>	durch Funktion Products	führt Datenbankabfrage nach allen erfassten Produkten aus und stellt die Datensätze dar
<code>_addProduct</code>	durch Funktion Products	Erstellt aus Kalkulation Datensatz Produkt in Datenbank
<code>__saveEditedProduct</code>	durch Funktion Employees	Speichert Änderungen an einem bestehenden Produkt
<code>_deleteProduct</code>	durch Funktion Products	Löscht Datensatz Produkt aus der Datenbank

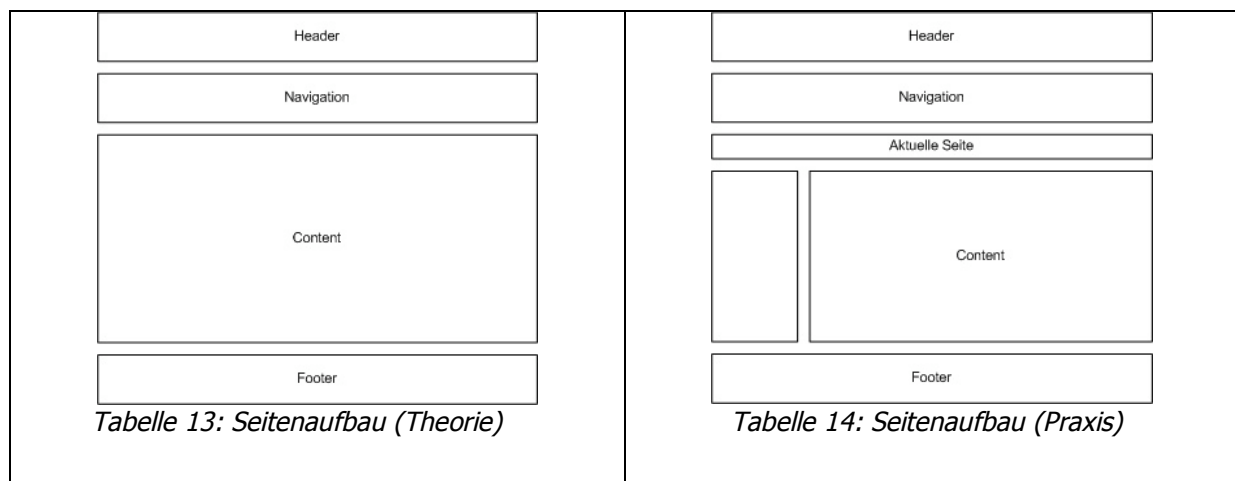
Tabelle 12: Low - Level Products - Modul

## 6 DESIGN DER BENUTZERSCHNITTSTELLE

Beim User Interface Design sollte die Einfachheit und schönes Aussehen nicht aus den Augen verloren gehen. Es soll ein Seitenaufbau entwickelt werden, welches für alle Seiten gebraucht werden kann. Ein linearer Aufbau eignet sich in allen Hinsichten perfekt.

Das Design wird anhand eines linearen Seitenaufbaus implementiert. Damit die „Schönheit“ des Designs nicht verloren geht müssen noch einige Anpassungen gemacht werden.

Die folgenden zwei Abbildungen stellen den Seitenaufbau in der Theorie und in der Praxis dar:



Die Spalten- und Zeilenstruktur werden mit Hilfe von unsichtbaren Tabellen (<table>) erzeugt. Tabellen eignen sich optimal um Layouts zu erzeugen, obwohl sie aus Sicht der Semantik nicht dazu da sind.

Bei der Strukturierung soll aber nicht allzu grosser Wert auf die Semantik gelegt werden. Sie soll so einfach wie möglich realisiert werden.

Als Doctype wird „XHTML 1.0 Transitional“ verwendet. Das <!DOCTYPE>-Tag gibt die Document Type Definition (DTD) des Dokuments an. Die DTD bestimmt die Grammatik und die Syntax der HTML-Sprache.

Da WebCalculator in Deutsch entwickelt wird, wird das Charset einfachheitshalber auf „iso-8859-1“ gesetzt.

Wichtig ist aber die HTML-Validierung. Ein HTML-Code ist erst dann validiert, wenn es keine Fehler enthält. Da HTML keine eigentliche Programmiersprache ist, ist es schwierig Fehler zu entdecken. Von

W3C gibt es aber eine Validierungs-Tool (<http://validator.w3.org/>), welches zu Hilfe genommen wird. Dadurch kann ein HTML-validierter Code erzeugt werden.

Die eingesetzten Bilder/Grafiken besitzen das Format GIF, JPG und PNG. PNG eignet sich perfekt um transparente aber dennoch qualitative gute Grafiken in Webseiten einzubinden.

Da der Internet Explorer 6 mit der Transparenz von PNG-Bildern nicht richtig umzugehen vermag, wird mit Hilfe eines HTC-Hacks (<http://www.pmsantos.ch/index.php?page=869>) Abhilfe verschafft.

Ein ähnliches Problem stellt (ebenfalls beim IE6) die Darstellung von Hover-Effekten dar. Doch auch dieses Problem kann mit Hilfe eines HTC-Hacks gelöst werden (<http://www.pmsantos.ch/index.php?page=959>).

Diese Hacks werden über das CSS eingebunden und werden nur vom IE6 verstanden. Aus diesem Grund besteht keine Möglichkeit, nebst dem HTML auch das CSS zu validieren. Dies ist jedoch nicht weiter tragisch, da die Befehle von allen anderen Browsern ignoriert werden.

## **A REFERENZEN**

[1] PEAR-<http://pear.php.net>