

# **Datenverwaltung mit WordPress**

ManagePress Suite

Bachelor-Arbeit vorgelegt von:

Ralph Büchi

Betreuender Dozent:

Dr. Alexandre de Spindler

Schriftliche Arbeit verfasst an der School of Management  
Zürcher Hochschule für Angewandte Wissenschaften (ZHAW)

Ralph Büchi  
Gehrenholz 8a  
8055 Zürich  
ralph.bu@bluewin.ch  
Matrikelnummer: S08-269-417

Winterthur, 25. Mai 2012

## **Wahrheitserklärung**

Ich erkläre hiermit, dass ich die vorliegende Arbeit selbständig, ohne Mithilfe Dritter und nur unter Benützung der angegebenen Quellen verfasst habe und dass ich ohne schriftliche Zustimmung der Studiengangleitung keine Kopien dieser Arbeit an Dritte aushändigen werde.

Ich nehme zur Kenntnis, dass die Arbeit von der Zürcher Hochschule für Angewandte Wissenschaften (ZHAW) mittels einer Plagiatserkennungssoftware überprüft werden kann. Ich bin damit einverstanden, dass zu diesem Zweck die Arbeit mit meinem Namen in eine geschützte und nur für die Hochschule zugängliche Datenbank ins Ausland übertragen wird und dort verbleibt. Ich kann bei der ZHAW jederzeit die Löschung meines Namens und allfälliger weiterer persönlicher Angaben verlangen.

Weiter nehme ich zur Kenntnis, dass gemäss § 16 Abs. 1 i.V.m. § 22 Abs. 2 FaHG die ausschliesslichen Verwendungsbefugnisse dieser Arbeit bei der ZHAW liegen. Das Recht auf Nennung der Urheberschaft bleibt davon unberührt.

Name des Studierenden (Druckbuchstaben)

.....

Name des Studierenden (Unterschrift)

.....

## Herausgabeerklärung des Dozenten

Herausgabe<sup>1</sup> der Bachelor-Arbeit „Datenverwaltung mit WordPress“

Die vorliegende Bachelor-Arbeit wird:

- ☐ nicht herausgegeben
- ☐ nicht herausgegeben werden bis ins Jahr .....
- ☐ für eine uneingeschränkte Herausgabe freigegeben.

Ort, Datum

Unterschrift der Dozentin / des Dozenten

.....

.....

<sup>1)</sup> Unter "Herausgabe" wird sowohl die Einsichtnahme im Hause wie auch die Ausleihe bzw. die Abgabe zu Selbstkostenpreisen verstanden.

Mein Dank gilt Dr. Alexandre de Spindler für die Idee zu dieser Arbeit und für seine  
Betreuung bei der Umsetzung.

## Management Summary

---

Im Rahmen dieser Arbeit wurde das WordPress-Plugin ManagePress Suite entwickelt und implementiert, um Privatpersonen sowie Kleinstunternehmen auf einfachste Weise die Möglichkeit zu eröffnen, eigene zugeschnittene Datenverwaltungssysteme zu erstellen und optimal zu nutzen. Als Grundsystem für die Entwicklung von ManagePress Suite wurde die Blogging-Plattform WordPress benutzt.

WordPress bietet eine elegante Schnittstelle zur Entwicklung von Erweiterungen und ist benutzerfreundlich aufgebaut. Im Verlauf der letzten zehn Jahre hat sich WordPress zu einem vollständigen Content Management System entwickelt und bietet durch das Post-type-Konzept die Möglichkeit verschiedene Typen von Inhalten zu generieren und zu verwalten. Das in dieser Arbeit entwickelte Plugin ManagePress Suite baut auf diesem Konzept auf und ermöglicht dem Benutzer eigene Posttypes zu generieren. Die generierten Posttypes werden in ManagePress Suite als Datentypen betrachtet und es wird die Möglichkeit geboten individuelle Eigenschaften den jeweiligen Datentypen hinzuzufügen. ManagePress Suite erweitert WordPress somit zu einem individuell anpassungsfähigen Datenverwaltungssystem. Eine weitere Funktion in ManagePress Suite ist die Meta-Plugin-Funktion. Die Funktionalität ermöglicht aus der ManagePress Suite heraus ein neues individuelles WordPress-Plugin auf der Basis von ManagePress Suite zu erstellen und zur Benutzung an Dritte weiterzugeben.

# Inhaltsverzeichnis

---

<b>Management Summary .....</b>	<b>V</b>
<b>Inhaltsverzeichnis .....</b>	<b>VI</b>
<b>Weitere Verzeichnisse .....</b>	<b>VIII</b>
Abkürzungsverzeichnis.....	VIII
Abbildungsverzeichnis.....	IX
Tabellenverzeichnis .....	X
<b>1 Einleitung.....</b>	<b>1</b>
<b>2 WordPress.....</b>	<b>2</b>
2.1 Aufstieg von Blogs bis zu WordPress .....	2
2.2 Aufbau von WordPress .....	4
2.2.1 Posttypes.....	4
2.2.2 Posts.....	5
2.2.3 Pages.....	5
2.2.4 Themes .....	5
2.2.5 Plugins .....	7
<b>3 ManagePress Suite .....</b>	<b>9</b>
3.1 Voraussetzungen.....	9
3.2 Aufbau von ManagePress Suite .....	10
3.2.1 Datenverwaltung.....	10
3.2.2 Meta-Plugin-Funktion .....	11
3.2.3 Konfigurationsdatei <i>mps_config.php</i> .....	12
3.3 Datenbankmodell .....	13
3.3.1 Tabelle: mps_core_posttype .....	14
3.3.2 Tabelle: mps_core_fields.....	15
3.3.3 Tabelle: mps_core_postconnection .....	15
3.3.4 Tabelle: mps_core_userconnection .....	16
3.3.5 Tabelle: mps_field_[field_table_name].....	16
3.3.6 Tabelle: mps_pt_[posttype] .....	16
<b>4 Anwendung.....</b>	<b>17</b>
4.1 Business Case .....	17
4.2 Konfiguration.....	18
4.2.1 Übersicht.....	18
4.2.2 Posttypes.....	19
4.2.2.1 Erstellen eines Posttypes.....	19
4.2.2.2 Löschen eines Posttypes.....	21

4.2.2.3	<i>Bearbeiten eines Posttypes</i> .....	22
4.2.3	Felder .....	23
4.2.3.1	<i>Erstellen eines Feldes</i> .....	24
4.2.3.2	<i>Löschen eines Feldes</i> .....	29
4.2.3.3	<i>Bearbeiten eines Feldes</i> .....	30
4.2.3.4	<i>Eigenschaften der Felder</i> .....	31
4.3	Export.....	33
4.4	Operative Tätigkeit .....	34
4.4.1	Erstellen eines Datenobjekts.....	35
4.4.1.1	<i>Erstellen eines Kunden</i> .....	35
4.4.1.2	<i>Erstellen eines Termins</i> .....	36
4.4.2	Dashboard.....	38
4.4.3	Frontend Ansicht .....	39
4.4.3.1	<i>Shortcode</i> .....	39
4.4.3.2	<i>Einzel-Ansicht</i> .....	41
<b>5</b>	<b>Schlussfolgerungen</b> .....	<b>42</b>
5.1	Testeinsatz .....	42
5.2	Weiterentwicklung.....	42
5.2.1	Feldtypen .....	42
5.2.2	Pflichtfelder .....	43
5.2.3	Import-Funktion .....	43
<b>6</b>	<b>Literaturverzeichnis</b> .....	<b>44</b>
<b>7</b>	<b>Anhang</b> .....	<b>45</b>
7.1	Template Hierarchie von WordPress .....	45

## Weitere Verzeichnisse

---

### Abkürzungsverzeichnis

API	Application Programming Interface
CBS	Columbia Broadcasting System
CMS	Content Management System
CSS	Cascading Style Sheets
HTML	Hypertext Markup Language
JS	JavaScript
MySQL	My Structured Query Language
PHP	Hypertext Preprocessor



## Abbildungsverzeichnis

Abbildung 1: Downloadzahlen von WordPress seit 2004, Adler (2011: XX) .....	3
Abbildung 2: Anordnung der Templates in WordPress .....	6
Abbildung 3: Beispiel Hook the_title() .....	8
Abbildung 4: Beziehung zwischen Posttypes und Felder .....	10
Abbildung 5: Beziehung zwischen Datenobjekten und WordPress Benutzern.....	11
Abbildung 6: Ablauf Export von ManagePress Suite .....	12
Abbildung 7: Datenmodel ManagePress Suite.....	14
Abbildung 8: Übersicht von ManagePress Suite .....	18
Abbildung 9: Erstellung eines Posttypes .....	19
Abbildung 10: Formular zur Erstellung des Posttypes für die Kunden.....	20
Abbildung 11: Formular zur Erstellung des Posttypes für die Termine.....	20
Abbildung 12: Löschung eines Posttypes.....	21
Abbildung 13: Bestätigung der Löschung eine Posttypes .....	21
Abbildung 14: Bearbeitung eines Posttypes .....	22
Abbildung 15: Bearbeitungsformular Posttype .....	22
Abbildung 16: Übersichtsseite, Erstellung von Feldern.....	24
Abbildung 17: Hinzufügen eines neuen Feldes .....	24
Abbildung 18: Erstellung Feld "Vorname" .....	25
Abbildung 19: Erstellung Feld "Nachname" .....	25
Abbildung 20: Erstellung Feld "Telefonnummer" .....	25
Abbildung 21: Erstellung Feld "E-Mail-Adressen" .....	25
Abbildung 22: Erstellung Feld "Wichtige Notizen" .....	25
Abbildung 23: Erstellung Feld "Adressen" .....	25
Abbildung 24: Übersichtsseite, alle Felder von den Kunden .....	26
Abbildung 25: Erstellung Feld "Datum" .....	27
Abbildung 26: Erstellung Feld "Uhrzeit" .....	27
Abbildung 27: Erstellung Feld "Terminstatus" .....	27
Abbildung 28: Erstellung Feld "Sitzplatz" .....	27
Abbildung 29: Erstellung Feld "Programm" .....	27
Abbildung 30: Erstellung Feld "Preis" .....	27
Abbildung 31: Übersichtsseite, alle Felder von den Terminen .....	28
Abbildung 32: Löschen eines Feldes.....	29

Abbildung 33: Bestätigung für die Löschung eines Feldes .....	29
Abbildung 34: Bearbeitung eines Feldes .....	30
Abbildung 35: Formular bei Bearbeitung eines Feldes .....	30
Abbildung 36: Übersicht Export.....	33
Abbildung 37: Export abschliessen .....	33
Abbildung 38: Übersicht registrierte Benutzer.....	34
Abbildung 39: Erstellung eines Kunden, Schritt 1 .....	35
Abbildung 40: Erstellung eines Kunden, Schritt 2 .....	35
Abbildung 41: Übersicht aller Kunden.....	36
Abbildung 42: Erstellung eines Termins, Schritt 1 .....	36
Abbildung 43: Erstellung eines Termins, Schritt 2 .....	37
Abbildung 44: Auswahlfenster für ein Elternelement .....	37
Abbildung 45: Dashboard mit ManagePress Suite.....	38
Abbildung 46: Beispiel Shortcode mit Posttype "kunde" .....	39
Abbildung 47: Shortcode Ansicht Tabelle .....	40
Abbildung 48: Shortcode Ansicht GoogleMaps.....	40
Abbildung 49: Frontend-Einzel-Ansicht eines Kunden .....	41
Abbildung 50: Template Hierarchie von WordPress, code.wordpress.org (2012c).....	45

## Tabellenverzeichnis

Tabelle 1: Wichtige Templates nach Adler (2011: 175) .....	6
Tabelle 2: Voraussetzungen für ManagePress Suite .....	9
Tabelle 3: Tabelle, mps_core_posttype .....	14
Tabelle 4: Datenbanktabelle mps_core_fields.....	15
Tabelle 5: Datenbanktabelle mps_core_postconnection .....	15
Tabelle 6: Datenbanktabelle mps_core_userconnection .....	16
Tabelle 7: Datenbanktabelle mps_field_[field_name_table].....	16
Tabelle 8: Datenbanktabelle mps_pt_[posttype] .....	16
Tabelle 9: Übersicht Datenobjekt Kunde .....	23
Tabelle 10: Übersicht Datenobjekt Termin .....	23
Tabelle 11: Verschiedene Typen der Felder.....	31
Tabelle 12: Verschiedene Definitionen der Felder.....	32
Tabelle 13: Feldoption - Darstellungsoption .....	32

# 1 Einleitung

---

Wie kann die Datenverwaltung einfach gemacht werden? In der heutigen Zeit werden sehr viele Daten generiert und es gibt unzählige Lösungen um diese Daten zu speichern, zu verwalten und zu organisieren. Die heutigen Lösungen sind mit viel Installations- und Konfigurationsaufwand verbunden und kosten Zeit sowie auch Geld. Für Privatpersonen und Kleinunternehmen stellen solche Aufwände ein Hindernis dar, ein Datenverwaltungssystem zu erstellen oder von einem Informatikdienstleister erstellen zu lassen.

Im Umfang dieser Arbeit wird nun eine Lösung entwickelt und implementiert, um Privatpersonen und Kleinunternehmen auf einfachste Weise die Möglichkeit zu eröffnen, individuell zugeschnittene Datenverwaltungssysteme zu erstellen und optimal zu nutzen. Die Hauptanforderungen an eine solche Lösung sind, dass ein Internetnutzer ohne spezielle Informatikkenntnisse mit geringen Kosten und wenig Zeitaufwand sein eigenes Datenverwaltungssystem kreieren kann. Eine weitere Anforderung ist, dass die Lösung mit einer Meta-Plugin-Funktion ausgestattet ist und somit die Lösung selber ein eigenes angepasstes Plugin erstellen kann. Als Grundsystem für die Implementierung der Lösung wurde im Vornherein dieser Arbeit die derzeit beliebteste Open-Source Blogging-Plattform WordPress definiert. WordPress bietet elegante Schnittstellen zum Entwickeln von Erweiterungen und ist durch ihre Benutzerfreundlichkeit eine optimale Basis.

## 2 WordPress

---

In diesem Kapitel wird die Blogging-Plattform WordPress beschrieben und es werden ihre Funktionalitäten und Eigenheiten erklärt.

### 2.1 Aufstieg von Blogs bis zu WordPress

WordPress ist eine vielseitige Websoftware, mit der ein Benutzer mit wenig Aufwand einen Blog im Web erstellen kann. Mit einem Blog kann jedermann Inhalte im Web publizieren, ohne bestimmte Vorkenntnisse davon zu haben, was Hypertext Markup Language (HTML) ist und wie die Technik hinter dem Blog funktioniert (Adler, 2011: XXI).

Der heutige Begriff Blog entstand um die Jahrtausendwende und wurde aus der Bezeichnung Weblog abgeleitet. Der Begriff Weblog bedeutet im Englischen so viel wie „Internet-Logbuch“ und wurde durch eine neue Art von Webauftritten gebildet (Adler, 2011). Die Webprogrammierer versuchten dabei Webseiten dynamischer zu erstellen und mit einer chronologischen Sortierung zu versehen, damit die neusten Inhalte direkt auf der Webseite ersichtlich sind und der Ersteller sich keine Gedanken über Informationsarchitektur machen muss. Das heisst, der Gedanke, wo der Inhalt auf der Webseite platziert werden soll oder die Ergänzung des Menüs, fällt weg (Alby, 2008: 5).

Das Bloggen hatte 2004 den Durchbruch, als die breite Masse in den USA die Blogs für sich entdeckten und die Blogger eine wichtige Rolle im sogenannten Rathergate-Skandal spielten. In diesem Skandal sammelten einige Blogger Fakten über eine nicht korrekte Darstellung des Senders CBS über den Militärdienst von George W. Bush und konnten eine öffentliche Entschuldigung von CBS erreichen. Mit den immer zahlreicheren Blogging-Plattformen, mit denen technisch Unkundige die Möglichkeiten hatten, schnell und einfach einen Weblog aufzusetzen, kamen parallel auch immer mehr Bloggingsoftware-Skripte auf den Markt, die ohne grosse Mühe auf dem eigenen Server installiert und verwendet werden konnten. In dieser Zeit erschien am 3. Januar 2004 die Version 1.0 von WordPress (Alby, 2008: 7).

WordPress entwickelte sich in den letzten Jahren stetig weiter, wodurch bis heute viele nützliche Features hinzukamen. Eines der wichtigsten Features ist, dass WordPress als vollständiges Content Management System (CMS) eingesetzt werden kann. Gemessen an der Lebensdauer anderer Webprodukten hat WordPress ein erstaunliches Alter erreicht. Die neuste Version 3.3 wurde am 12. Dezember 2011 veröffentlicht. Die Gründe für diesen Erfolg sind folgende:

- eine einfache Installation und Bedienung
- viele nützliche Features, die auch eine Verwendung als CMS ermöglichen
- eine grosse Zahl von Plugins, mit denen individuelle Anwendungen abgedeckt werden können
- eine umfangreiche Schnittstelle für die Entwicklung von Erweiterungen
- eine breite Userbasis und grosse Anwenderzahl: Somit ist ein guter Support über die Community gewährleistet und es gibt eine grosse Bandbreite an aktuellen Plugins, Hacks und Themes.
- Open-Source-Status: WordPress ist kostenlos und kann ohne Lizenzgebühren auch kommerziell genutzt werden.

(Adler, 2011: XX)

In Anlehnung an Adler (2011: XX) wird in der nachfolgenden Abbildung 1 ein Überblick über die Anzahl der Downloads von *WordPress.org* von 2004 bis 2008 gegeben. Die stetig ansteigende Anzahl Downloads zeigt auf, wie weit WordPress verbreitet ist. Gemäss *WordPress.org* (2012a) beträgt der aktuelle Downloadstand der Version 3.3 am 13. Mai 2012 17'983'054 Downloads und die Zahl steigt im Sekundentakt an.

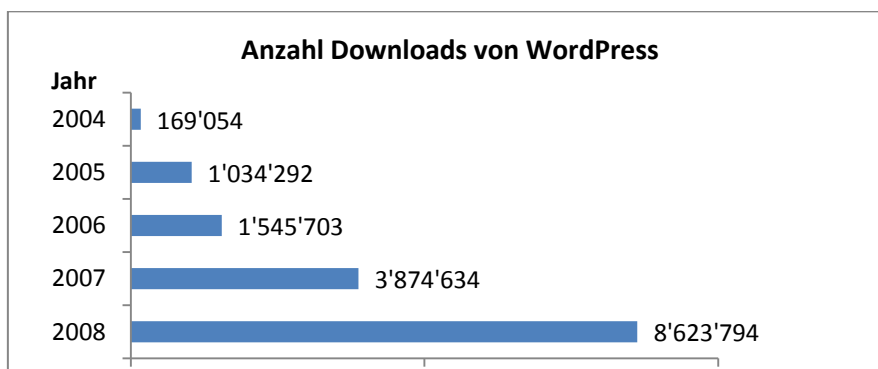


Abbildung 1: Downloadzahlen von WordPress seit 2004, Adler (2011: XX)

## 2.2 Aufbau von WordPress

Wie bereits im vorherigen Kapitel über den Aufstieg von WordPress erwähnt wurde, hat WordPress den Vorteil, dass es als vollständiges CMS eingesetzt werden kann und somit für unzählige Anwendungen zur Verfügung steht. Das war der Hauptgrund, warum für diese Arbeit WordPress als Grundgerüst gewählt wurde. In den nächsten Abschnitten werden die wichtigsten Elemente und der Aufbau von WordPress erklärt.

### 2.2.1 Posttypes

Eines der wichtigsten Basiselemente von WordPress sind die Posttypes. Durch die Posttypes kann WordPress viele verschiedene Typen von Inhalten verwalten und der Benutzer hat die Möglichkeit neue Inhaltstypen, sogenannte Posttypes, zu generieren. Dabei spielt die Idee, dass alle erstellten Inhalte von den jeweiligen Inhaltstypen intern in die gleiche Datenbanktabelle gespeichert werden, eine grosse Rolle. Die Inhalte sehen dadurch datenbanktechnisch alle gleich aus. Die einzelnen Einträge unterscheiden sich lediglich durch den „post\_type“ und können durch ihre generierte „post\_id“ weiterverwendet werden. Gemäss *WordPress.org* (2012b) sind folgende fünf Posttypes standardmässig in WordPress integriert:

- post
- page
- attachment
- revision
- nav\_menu\_item

(WordPress.org, 2012b)

### 2.2.2 Posts

Ein Post kann zwei Definitionen haben. Zum einen wird die Bezeichnung Post für einen Beitrag in einem Blog mit dem standardmässigen Posttype „post“ verwendet. Und zum anderen werden alle Inhalte in der Datenbanktabelle, egal welchem Posttype sie angehören, als Posts bezeichnet (Adler, 2011: 19).

### 2.2.3 Pages

Pages werden auch Seiten genannt und werden durch den Posttype „page“ bezeichnet. Mit dieser Art von Posttype kann WordPress als CMS eingesetzt werden. Die Pages werden im Vergleich zu den normalen Blogposts nicht chronologisch dargestellt und bekommen die Eigenschaft einer normalen Webseite. In WordPress kann so eine vollständige Webseitenstruktur und -navigation aufgebaut werden (Alby, 2008: 98).

### 2.2.4 Themes

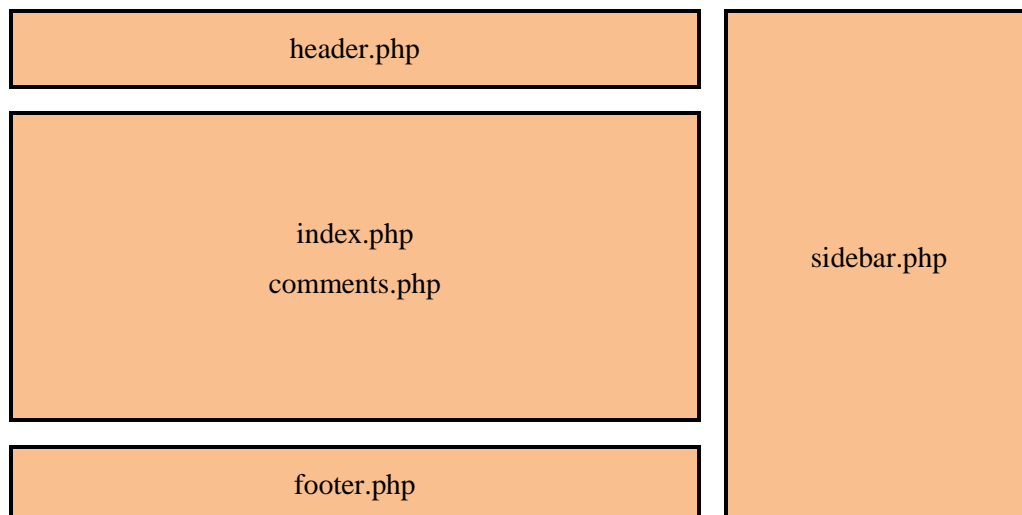
Mit der Funktion von Themes hat WordPress eine gute Möglichkeit geschaffen, das Erscheinungsbild individuell zu ändern. Themes sind Template-Pakete, welche ohne grosse Mühe in WordPress integriert werden können. Dabei können komplexere und aufwendige Themes direkt im WordPress-Backend verändert werden (Adler, 2011: 164).

Als Grundlage von diesen Themes benutzt WordPress einen bestimmten Aufbau. Jedes Theme besteht aus mehreren Unterdateien, die Templates genannt werden und auf welche WordPress zurückgreift. Alle möglichen Bezeichnungen dieser Templates sind auf der Abbildung 50 im Anhang ersichtlich. Die Hauptdatei ist die index.php. Wenn beispielsweise das Template für eine bestimmte Seite (page.php) nicht vorhanden ist, wird auf die index.php zurückgegriffen und diese angezeigt. Die wichtigsten Templates sind in der Tabelle 1 beschrieben:

Datei	Name	Beschreibung
index.php	Standardseite des Templates	Zentrale Template-Datei, die den Inhaltsbereich des Blogs definiert. Wird auch aufgerufen, wenn andere Dateien nicht vorhanden sind. – muss vorhanden sein
style.css	Stylesheet	Standard-Stylesheet, in dem die Formatierungen für das Theme festgelegt werden. – muss vorhanden sein
header.php	Header-Bereich des Templates	Entspricht dem, was man in HTML-Seiten mit <head> </head> einschliesst, plus Titelbereich des Blogs
footer.php	Footer-Bereich des Templates	Fusszeilen der Seite
sidebar.php	Seitenleiste des Templates	Die Seitenleiste, in der sich üblicherweise Archivlinks, Kommentare, Navigation usw. befinden
functions.php	Funktionsdatei	Kann verwendet werden, um Theme benutzerdefinierte PHP-Funktionen hinzuzufügen
screenshot.png	Vorschaugrafik	Wird in der Theme-Übersicht angezeigt, um eine erste optische Vorstellung vom Theme zu geben

**Tabelle 1: Wichtige Templates nach Adler (2011: 175)**

Abbildung 2 zeigt eine mögliche Darstellung, wie die einzelnen Templates angeordnet werden können.



**Abbildung 2: Anordnung der Templates in WordPress**



### 2.2.5 Plugins

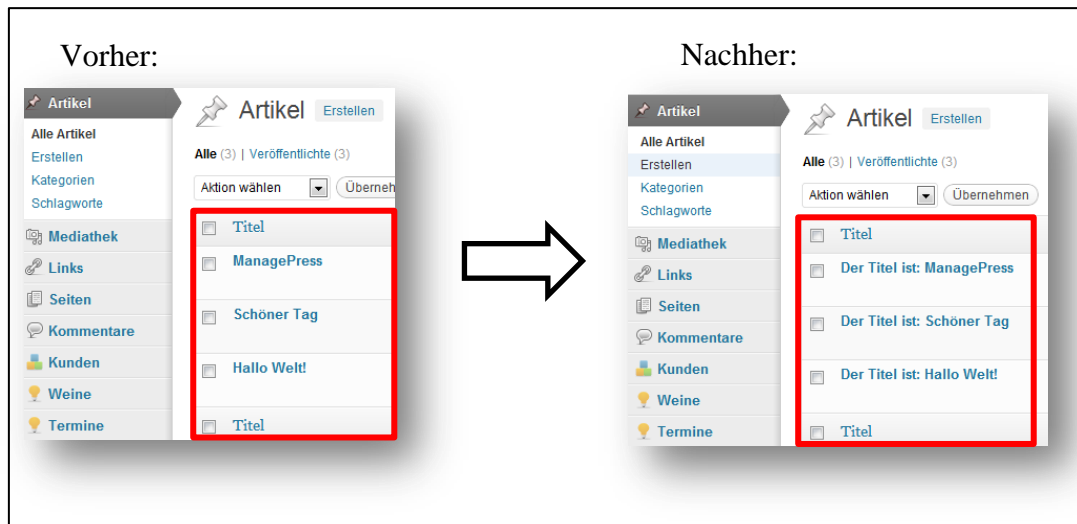
Die Plugins in WordPress sind kleine oder grössere Erweiterungen für WordPress, welche von unzähligen Entwicklern erstellt worden sind. Fast jede Anwendung, welche WordPress selbst nicht abdeckt, kann mit Plugins abgedeckt werden (Adler, 2011: 123).

Die äusserst elegante und umfangreiche WordPress-API, welche von WordPress bereitgestellt wird, ist der wichtigste Grund dafür, dass so viele Erweiterungen erstellt werden können. Ein Entwickler hat somit die Möglichkeit auf fast alle Programmteile von WordPress zuzugreifen und individuelle Anpassungen vorzunehmen. Die Anpassungen können mit sogenannten Hooks vorgenommen werden und werden direkt ins Hypertext Preprocessor (PHP) geschrieben (WordPress.org, 2012d).

Gemäss Adam Brown (2011) gibt es 1'519 Hooks für die Version 3.3 von WordPress, welche direkt in die Plugins eingebaut werden können. Nachfolgend wird ein Beispielcode gezeigt, der den Hook `the_title()` verwendet. `The_title()` wird auf den Post-Titel angewendet, der aus der Datenbank geholt wird, jedoch bevor er auf der Seite angezeigt wird und somit noch verändert werden kann. Durch den Zusatz `add_filter()` wird dann der Hook im Plugin aktiv (Alby, 2008: 261).

```
<?php
add_filter( 'the_title' , 'fn_make_title' );
function fn_make_title($title){
    return 'Der Titel ist: '.$title;
}
?>
```

Mit dem gezeigten Beispielcode wird in der Abbildung 3 folgender Effekt sichtbar. Der Titel von jedem Artikel wurde verändert dargestellt und zwar so, dass der Textbaustein „Der Titel ist:“ vor dem eigentlichen Titel eingefügt wurde.

Abbildung 3: Beispiel Hook `the_title()`

### 3 ManagePress Suite

---

In der Einleitung wurden bereits die Anforderungen an die Lösung, welche entwickelt wird, dargelegt. Diese Lösung soll WordPress zu einem benutzerfreundlichen und einfachen Datenverwaltungssystem machen, welches Privatpersonen und auch Kleinstunternehmen die Möglichkeit bietet ihre Daten auf einfachste Weise zu verwalten. Aufbauend auf den Funktionalitäten von WordPress, welche im letzten Kapitel beschrieben wurden, wurde nachfolgend das WordPress-Plugin mit dem Namen ManagePress Suite entwickelt und implementiert. Bei der Entwicklung wurde besonders auf Flexibilität und Erweiterbarkeit geachtet, damit das Plugin von einer dritten Person weiter entwickelt werden kann. Für die Implementierung wurden die Websprachen PHP, HTML, JavaScript, MySQL und CSS benutzt. Im Unterkapitel 3.1 Voraussetzungen sind die Websprachen mit den genauen Versionen aufgelistet.

#### 3.1 Voraussetzungen

Das Plugin ManagePress Suite wurde mit den folgenden Systemvoraussetzungen entwickelt und erfordert diese für einen reibungslosen Betrieb.

Websprache	Version
HTML	5
JavaScript	1.8
CSS	2.1
PHP	5.2.4
WordPress	3.3.1

**Tabelle 2: Voraussetzungen für ManagePress Suite**

## 3.2 Aufbau von ManagePress Suite

In diesem Kapitel wird der Aufbau von ManagePress Suite aufgezeigt. Aufgrund der zuvor erwähnten Anforderungen an die Lösung wurden zwei Hauptteile für den Aufbau bestimmt. Zum einen die Funktionalität der Datenverwaltung und zum anderen die Funktion des Meta-Plugins.

### 3.2.1 Datenverwaltung

Damit eine Datenverwaltung, wie sie in der Einleitung dargelegt wurde, durch WordPress realisierbar werden kann, wurde ManagePress Suite so aufgebaut, dass der zukünftige Benutzer die Möglichkeit hat eigene Posttypes zu generieren, welche als Objektklassen betrachtet werden. Die einzelnen Objekte dieser Klassen bilden dabei die einzelnen Posts, die durch die Basisfunktionalität von WordPress erstellt werden können. Bei diesem ersten Teil ist somit die volle Integration in das Grundgerüst von WordPress gegeben. Der Benutzer hat weiter die Möglichkeit zu jedem Posttype eigene Felder zu definieren. ManagePress Suite betrachtet diese Felder als Objektattribute. Mit dieser Erweiterung hat jedes Objekt bzw. Post eigene spezifische Attribute zur Verfügung und es entsteht die Möglichkeit ohne grossen Aufwand weitere Informationen zu einem Objekt hinzuzufügen.

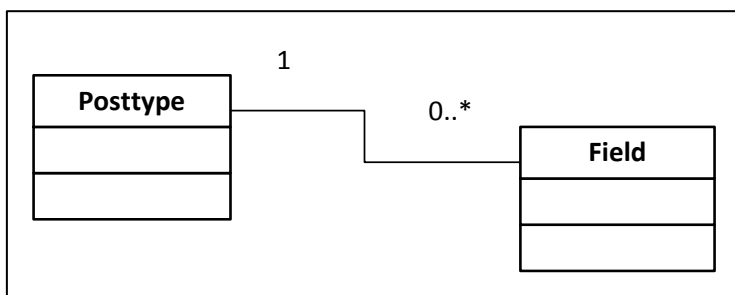


Abbildung 4: Beziehung zwischen Posttypes und Felder

In der Abbildung 4 wird der beschriebene Aufbau grafisch dargestellt, wobei man erkennt, dass zu einem Posttype mehrere Felder gehören können. Dieser erste Teil ist somit die Grundlage für die Datenverwaltung. Mit der individuellen Gestaltung der Posttypes und den Feldern kann jedes mögliche Datenmodell modelliert und genutzt werden.

Als Zusatz zur beschriebenen Feld-Posttype-Beziehung wurden zwei weitere Beziehungsarten implementiert. Die erste Beziehung ist eine Beziehung, bei welcher der Benutzer seinem erstellten Datenobjekt bzw. Post eine Benutzerverknüpfung hinzufügen kann. Dabei kann er jeden Benutzer, der in WordPress bereits registriert ist, auswählen und das Datenobjekt mit dieser zusätzlichen Information ergänzen. Die zweite neue Beziehungsart ermöglicht dem Benutzer hingegen eine Beziehung zu kreieren, bei der das erstellte Datenobjekt auf ein bereits erstelltes Datenobjekt referenziert werden kann. Mit der Eigenschaft, dass bei dieser Verknüpfung die Unterscheidung von einem Eltern-element zu einem Kindelement oder in die entgegengesetzte Richtung gemacht werden kann, wird eine Hierarchie geschaffen, welche für eine gut strukturierte Datenverwaltung äusserst wichtig ist. In der Abbildung 5 sind die beiden Beziehungsarten dargestellt.

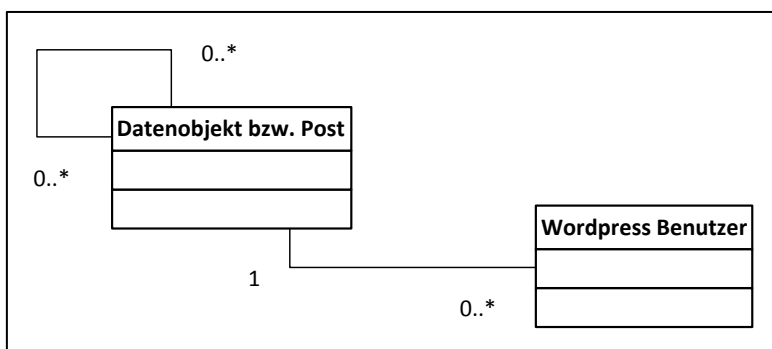


Abbildung 5: Beziehung zwischen Datenobjekten und WordPress Benutzern

### 3.2.2 Meta-Plugin-Funktion

Mit der Meta-Plugin-Funktion kommt ein weiterer Teil zur ManagePress Suite dazu. Beim Aufbau dieser Funktion wird davon ausgegangen, dass das Plugin einmal installiert und nach den individuellen Bedürfnissen konfiguriert wird. Nach dem Konfigurationsvorgang kann nun eine Kopie des aktuellen Plugins gemacht werden, die in einen geschützten Modus versetzt wird. Diese kopierte Version von ManagePress Suite kann nach dem Kopiervorgang ausgeliefert, in eine andere WordPress-Umgebung importiert und dort verwendet werden. Der beschriebene Aufbau wird in der Abbildung 6 dargestellt.

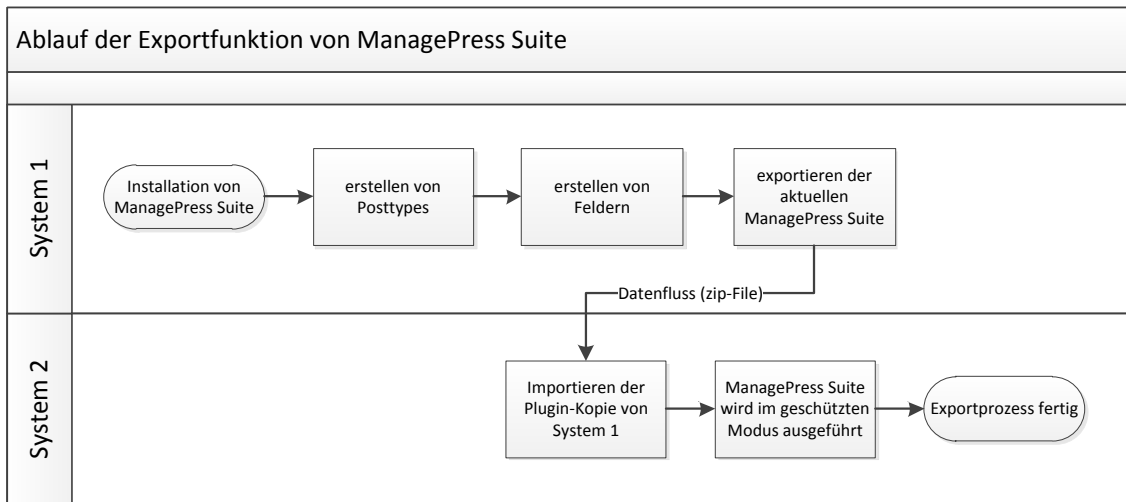


Abbildung 6: Ablauf Export von ManagePress Suite

### 3.2.3 Konfigurationsdatei *mps\_config.php*

Im Root-Verzeichnis von ManagePress Suite existiert die Datei *mps\_config.php*. Diese beinhaltet die zwei Variablen `$mps_safe_mode` und `$import_sql`. Im nachfolgenden PHP-Code wird der Standardinhalt der *mps\_config.php* gezeigt:

```
//Datei: mps_config.php
<?php
$mps_safe_mode = false;
$import_sql = '';
?>
```

Die Variable `$mps_safe_mode` ist standardmässig auf FALSE gesetzt und hat Auswirkungen auf das ganze Plugin. Sobald der Wert auf TRUE gesetzt wird, können keine Einstellungen von ManagePress Suite vorgenommen werden.

Die zweite Variable ist `$import_sql`. In dieser Variablen sind allfällige Datenbankskripte, die bei der Installation von ManagePress Suite ausgeführt werden sollen, gespeichert. Standardmässig ist `$import_sql` auf NULL gesetzt und wird bei einem Export definiert.

### 3.3 Datenbankmodell

In diesem Abschnitt wird erläutert, wie das Datenbankmodell von ManagePress Suite implementiert wird. Bei der Entwicklung des Datenbankmodells ist grossen Wert auf die Erweiterbarkeit gelegt worden, um die Anwendung individuell anzupassen.

Für die Nutzung von ManagePress Suite werden folgende vier Haupttabellen benötigt, die bei der Installation automatisch erstellt werden:

- mps\_core\_posttype
- mps\_core\_fields
- mps\_core\_postconnection
- mps\_core\_userconnection

Bei der Benutzung von ManagePress Suite können weitere zwei Tabellenvariationen hinzukommen. Zum einen wird für jeden kreierten Posttype und zum anderen für jedes erstellte Feld eine eigene Tabelle erstellt. Für die einzelnen Tabellen gelten folgende Namensgebungen:

- mps\_pt\_[posttype]
  - Bei [posttype] handelt es sich um ein Platzhalter für den Namen des Posttypes.
- mps\_field\_[field\_table\_name]
  - Bei [field\_table\_name] handelt es sich um einen Platzhalter für den Tabellennamen, der aus dem „field\_name“ und der „field\_id“ besteht.

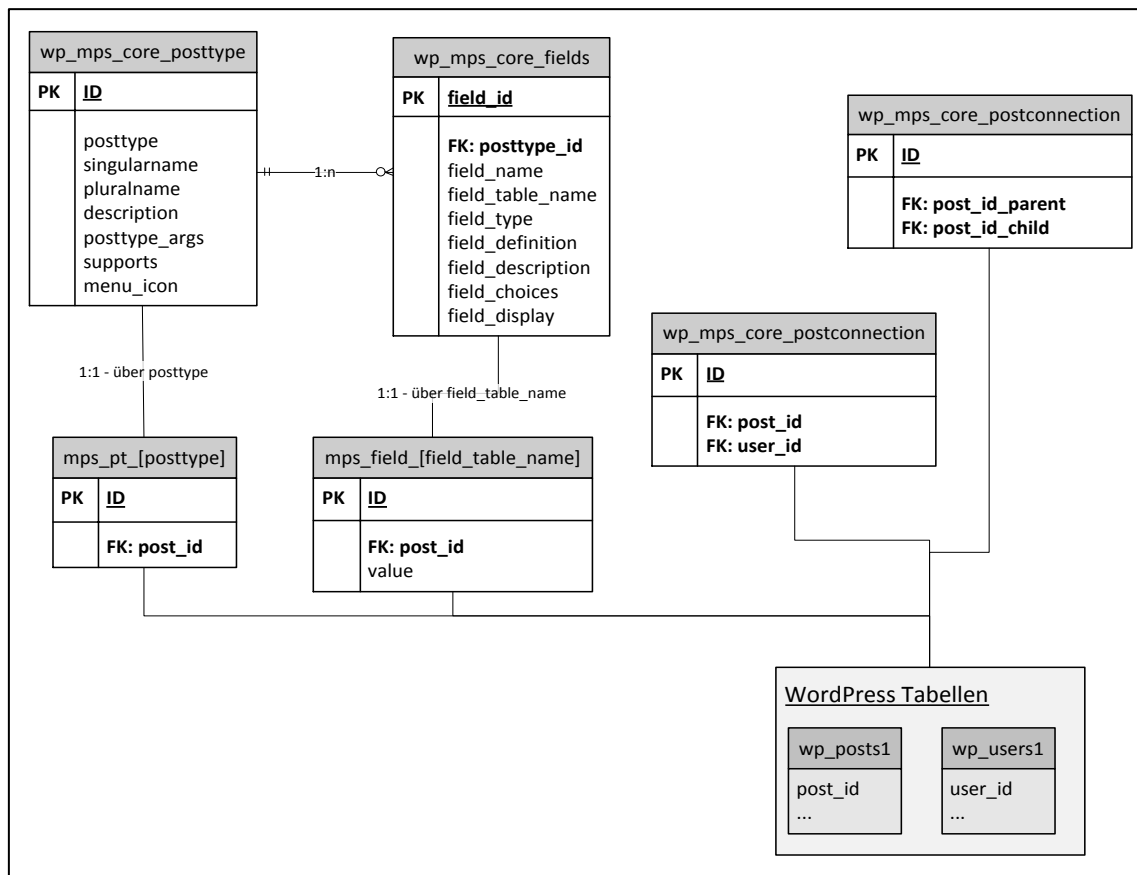


Abbildung 7: Datenmodel ManagePress Suite

### 3.3.1 Tabelle: mps\_core\_posttype

In der Tabelle „mps\_cores\_posttype“ werden alle Posttypes gespeichert, welche mit ManagePress Suite erstellt wurden.

Feld	Typ	Kollation	Beschreibung	Beispiel
ID	bigint(20)	-	Primary Key, AUTO_INCREMENT	1
posttype	text	utf8_general_ci	Name vom Posttype	haus
singularname	text	utf8_general_ci	Name von einem Objekt eines Posttypes	Haus
pluralname	text	utf8_general_ci	Der generelle Name vom Posttype im Plural	Häuser
description	longtext	utf8_general_ci	Der Beschreib vom Posttype	Alle meine Häuser.
posttype_args	longtext	utf8_general_ci	Hier werden bestimmte Einstellungen für den Posttype gespeichert. Wird in der aktuellen Version von ManagePress nicht verwendet.	empty
supports	longtext	utf8_general_ci	Spezielle Eigenschaften, welche der Posttype unterstützen soll werden in diesem Feld gespeichert	Array(title,editor)
menu_icon	text	utf8_general_ci	Ist das Icon in dem Menu	organization16.png

Tabelle 3: Tabelle, mps\_core\_posttype



### 3.3.2 Tabelle: mps\_core\_fields

Auf der Tabelle ‚mps\_core\_fields‘ sind alle erstellten Felder gespeichert. Sie gilt als Grundlage, damit die einzelnen Feldtabellen auf der Datenbank generiert werden können.

Feld	Typ	Kollation	Beschreibung	Beispiel
field_id	bigint(20)		PK, AUTO_INCREMENT	9
posttype_id	bigint(20)		FK zu mps_core_posttype.ID	1
field_name	text	utf8_general_ci	Ist der generelle Feldname	Stadt
field_table_name	text	utf8_general_ci	Ist der Tabellennamen für die automatisch generierte Feldtabelle. Besteht aus field_name und field_id	stadt_9
field_type	text	utf8_general_ci	Ist der Feldtyp	longtext
field_definition	text	utf8_general_ci	Ist eine bestimmte Definition vom Feld	singlechoice
field_description	longtext	utf8_general_ci	Der Beschreib vom Feld	Wähle eine der 2 Städte aus?
field_choices	longtext	utf8_general_ci	Hier sind allfällige Vorgaben für den Feldwert gespeichert	Array(Zürich,Basel)
field_display	int(11)		Falls 0, wird des Feld in jeder Auflistung dargestellt und falls 1 nur beim Erstellen vom Objekt	0

**Tabelle 4: Datenbanktabelle mps\_core\_fields**

### 3.3.3 Tabelle: mps\_core\_postconnection

In dieser Tabelle werden die Verknüpfungen von einem Datenobjekt zu einem andern Datenobjekt gespeichert.

Feld	Typ	Extra	Beispiel
ID	bigint(20)	PK, AUTO_INCREMENT	5
post_id_parent	bigint(20)	FK zu posts.ID	10
post_id_child	bigint(20)	FK zu posts.ID	10

**Tabelle 5: Datenbanktabelle mps\_core\_postconnection**

### 3.3.4 Tabelle: mps\_core\_userconnection

Damit die Beziehung Datenobjekt zu WordPress-User erstellt werden kann, ist die aktuelle Tabelle ‚mps\_core\_userconnection‘ notwendig.

Feld	Typ	Extra	Beispiel
ID	bigint(20)	PK, AUTO_INCREMENT	5
post_id	bigint(20)	FK zu posts.ID	10
user_id	bigint(20)	FK zu user.ID	1

**Tabelle 6: Datenbanktabelle mps\_core\_userconnection**

### 3.3.5 Tabelle: mps\_field\_[field\_table\_name]

Für jedes erstellte Feld in ManagePress Suite wird in der Datenbank eine Tabelle mit den nachfolgenden Eigenschaften erstellt. Wichtig ist dabei, dass der Feldtyp je nach Feld variieren kann. Der Name der Tabelle wird aus der Feld-ID und dem Feldnamen gebildet.

Feld	Typ	Extra	Beispiel
ID	bigint(20)	PK, AUTO_INCREMENT	5
post_id	bigint(20)	FK zu posts.ID	10
value	[field_type]	Das ist der Wert der vom Benutzer eingetragen wurde.	grün

**Tabelle 7: Datenbanktabelle mps\_field\_[field\_name\_table]**

### 3.3.6 Tabelle: mps\_pt\_[posttype]

Die Tabelle „mps\_pt\_[posttype]“ wird für jeden Posttype erstellt. Der Tabellename besteht aus dem Suffix „mps\_pt\_“ und aus der Posttype-Bezeichnung. Mit dem Erstellen eines Objektes von einem Posttype wird ein neuer Eintrag in diese Tabelle gemacht.

Feld	Typ	Extra	Beispiel
ID	bigint(20)	PK, AUTO_INCREMENT	5
post_id	bigint(20)	FK zu posts.ID	10

**Tabelle 8: Datenbanktabelle mps\_pt\_[posttype]**

## 4 Anwendung

---

Im folgenden Kapitel wird die Anwendung des Plugins beschrieben. Als Grundlage dazu dient der nachfolgende Geschäftsfall eines Coiffeur-Salons. Durch die vollständige Bearbeitung des Business Cases soll die Anwendung in Bezug auf folgende drei Teilabschnitte anschaulich erklärt werden:

1. Konfiguration
2. Exportfunktion
3. Operative Tätigkeit

### 4.1 Business Case

Um die Anwendung von ManagePress Suite möglichst praxisnah zu beschreiben, dient dieser Geschäftsfall als Grundlage.

„Der neue Coiffeur-Salon, *Hair&Care*, wurde vor drei Monaten gegründet. Das Tagesgeschäft läuft ausgezeichnet und *Hair&Care* konnte bereits einige Stammkunden gewinnen. Damit die Termine und die Kundendatei in der Zukunft effizienter und besser verwaltet werden können, haben der Geschäftsführer und seine drei Mitarbeiter an der letzten monatlichen Teamsitzung beschlossen, ein Datenverwaltungssystem einzuführen. Nach einer Woche Evaluation verschiedener Offerten entschied *Hair&Care* sein Datenverwaltungssystem bei einem kleinen innovativen Informatikunternehmen in Auftrag zu geben. Der Informatikdienstleister überzeugte mit der Lösung des Datenverwaltungssystems ManagePress Suite. Die sehr geringen Kosten und die Tatsache, dass ManagePress Suite individuell gestaltet werden kann und erweiterbar ist, gaben den Ausschlag für den Entscheid. Ein weiterer Entscheidungsgrund war, dass *Hair&Care* bereits WordPress benutzte und sich die Datenverwaltung direkt in die eigene Homepage integrieren lässt.“

## 4.2 Konfiguration

### 4.2.1 Übersicht

Abbildung 8 zeigt die Startseite des Plugins, worauf das Plugin Menu gekennzeichnet ist.

Hauptmenü von ManagePress Suite:

- ManagePress
  - Hier kann die ManagePress Suite konfiguriert werden sowie die Struktur des Datenverwaltungssystems verwaltet und bearbeitet werden.
- About
  - Wichtige Informationen über das Plugin
- Export
  - Das Plugin kann hier exportiert werden.

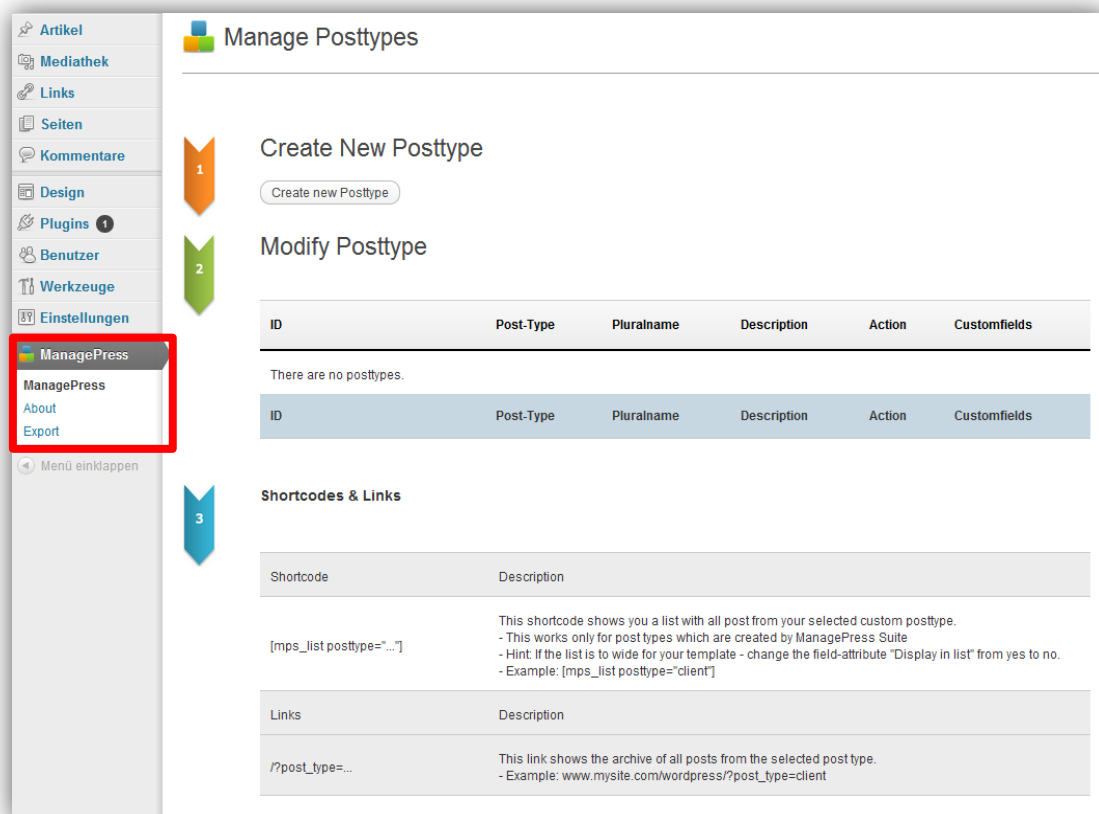


Abbildung 8: Übersicht von ManagePress Suite

## 4.2.2 Posttypes

### 4.2.2.1 Erstellen eines Posttypes

Gemäss dem Business Case werden für *Hair&Care* folgende zwei Posttypes erstellt:

- Kunde
- Termin

Als erstes muss das Menü [ManagePress] ausgewählt werden. Um nun einen neuen Posttype zu erstellen, muss auf den rot markierten Knopf [Create Posttype] geklickt werden (Abbildung 9).

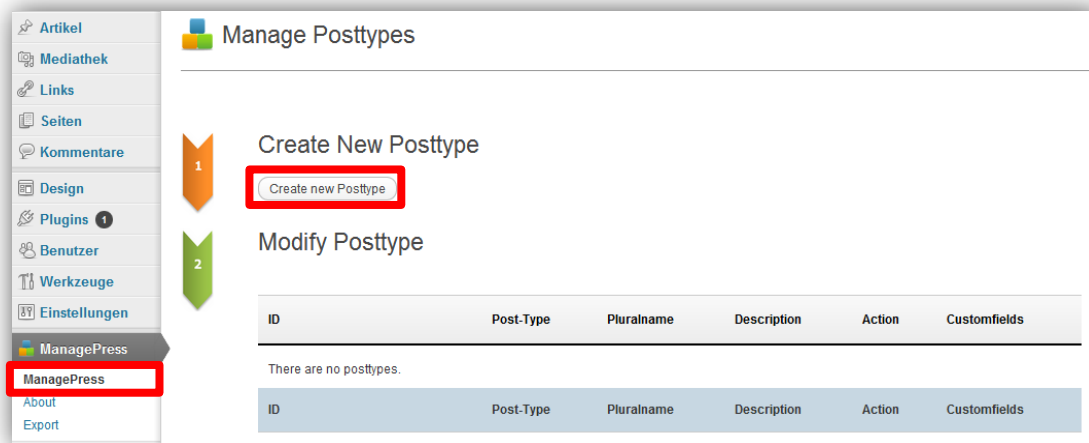


Abbildung 9: Erstellung eines Posttypes

Durch diesen Schritt wird eine neue Seite geladen. Das Formular, welches für einen neuen Posttype ausgefüllt werden muss, wird sichtbar (Abbildung 10). In den gekennzeichneten Feldern sind bereits die Angaben für den Posttype „kunde“ eingegeben und können mit dem Knopf [Save Posttype] bestätigt werden.

The screenshot shows the 'Create New Posttype' interface. On the left is a sidebar with navigation links: Artikel, Mediathek, Links, Seiten, Kommentare, Design, Plugins (1), Benutzer, Werkzeuge, Einstellungen, and ManagePress. The ManagePress section includes links for About and Export, and a 'Menü einklappen' button. The main content area is titled 'Create New Posttype' and features a 'Go Back' button. A red box highlights the 'Posttype Information' section, which includes fields for Posttype Name (kunde), Plural Name (Kunden), Singular Name (Kunde), and Description (Das sind die Kunden.). Below these are checkboxes for Supports (Title, Editor, Thumbnail, Comments, Author) and a Menu Icon selector. A green arrow labeled '1' points to the 'Posttype Information' section, and another green arrow labeled '2' points to the 'Save Posttype' button at the bottom.

Abbildung 10: Formular zur Erstellung des Posttypes für die Kunden

Der gleiche Vorgang wird für den zweiten Posttype „termin“ wiederholt und ist in Abbildung 11 dargestellt.

This screenshot shows the 'Create New Posttype' interface for creating a posttype named 'termin'. The layout is identical to the previous one, but the form fields are populated with 'termin' for the Posttype Name, 'Termine' for the Plural Name, 'Termin' for the Singular Name, and 'Das sind alle Termine' for the Description. The 'Save Posttype' button at the bottom is also highlighted with a red box. Green arrows labeled '1' and '2' indicate the same steps as in the previous image.

Abbildung 11: Formular zur Erstellung des Posttypes für die Termine

#### 4.2.2.2 Löschen eines Posttypes

In diesem Abschnitt wird der Vorgang für eine Löschung eines Posttypes beschrieben. Dieser Vorgang wird gebraucht, falls bei der Erstellung eines Posttypes ein Fehler begangen wurde oder ein Posttype nicht mehr verwendet wird. Mit dem Button [Delete] kann der Löschvorgang gestartet werden (Abbildung 12).

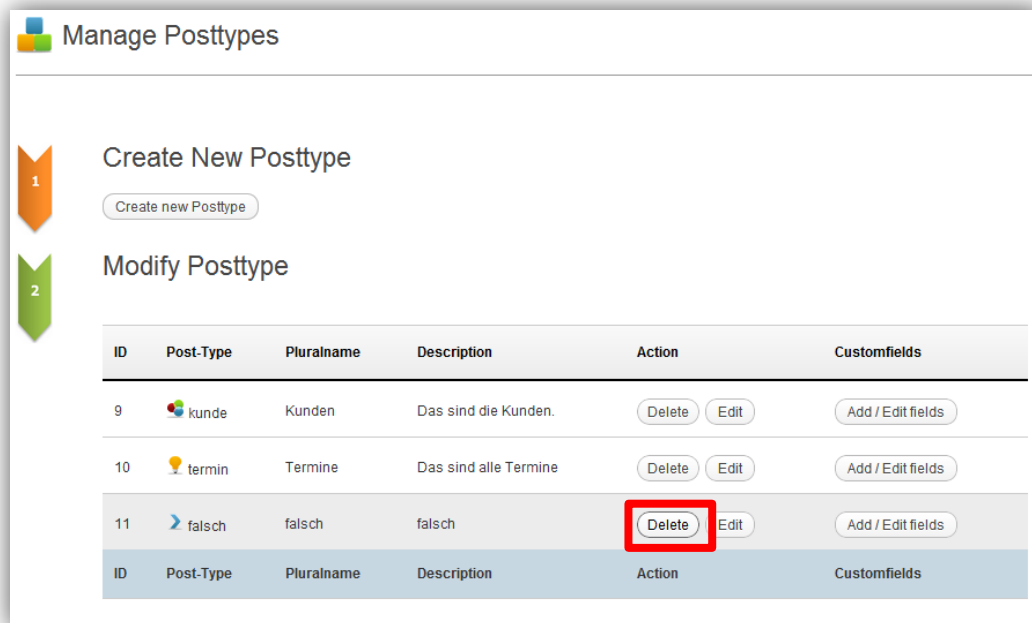


Abbildung 12: Löschung eines Posttypes

Nun erscheint eine Popup-Box und durch die Bestätigung mit [Yes] wird der Posttype gelöscht (Abbildung 13).

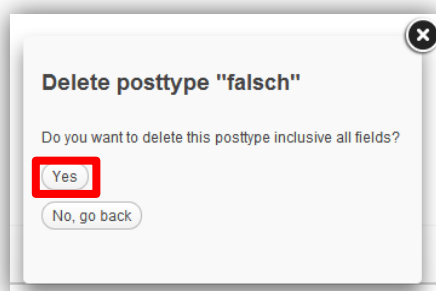


Abbildung 13: Bestätigung der Löschung eine Posttypes

Bei der Löschung werden alle Posts, Felder und Tabellen gelöscht, die mit dem ausgewählten Posttype verknüpft sind.

#### 4.2.2.3 Bearbeiten eines Posttypes

Damit ein Posttype bearbeitet werden kann, wird der Button [Edit] ausgewählt. Nach dem Klick erscheint das Posttype-Formular und der Posttype kann bearbeitet werden. Dieser Vorgang wird in Abbildung 14 und Abbildung 15 dargestellt.

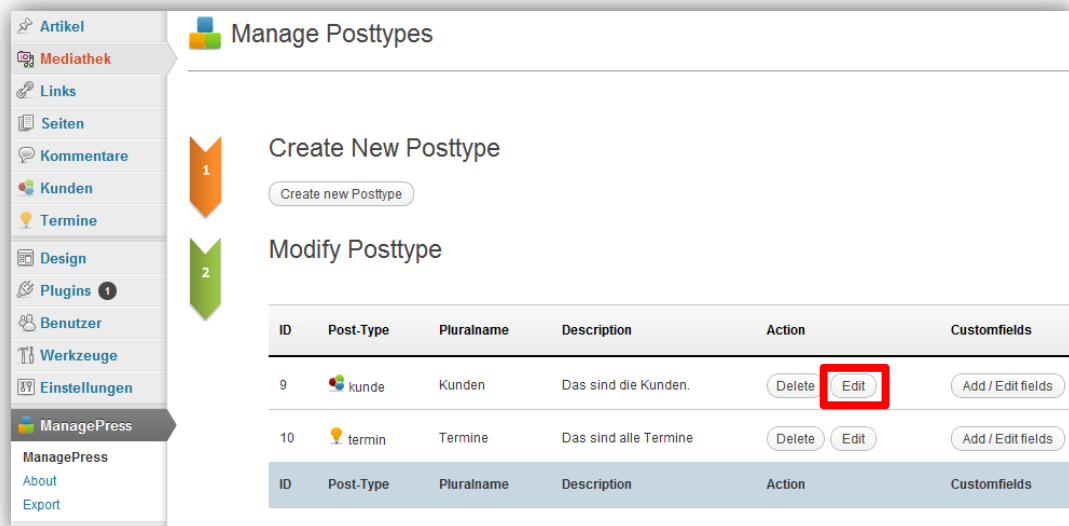


Abbildung 14: Bearbeitung eines Posttypes

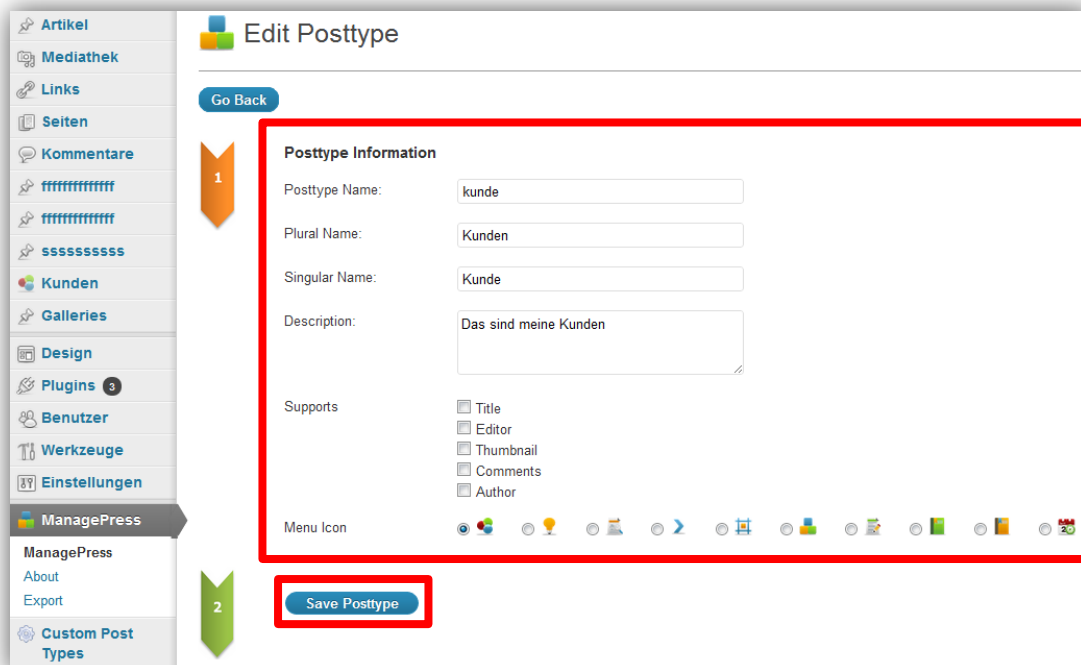


Abbildung 15: Bearbeitungsformular Posttype



### 4.2.3 Felder

Die Felder können zu einem Posttype hinzugefügt werden. Im Business Case von *Hair&Care* sind folgende Felder erforderlich:

Datenobjekt: Kunde		
Feld	Typ	Auswahlmöglichkeiten
Vorname	String	-
Nachname	String	-
Telefonnummer	String, (mehrere Auswahlmöglichkeiten)	-
E-Mail Adresse	String, (mehrere Auswahlmöglichkeiten)	-
Wichtige Notizen	Longtext	-
Adresse	Place on Googlemaps	-

**Tabelle 9: Übersicht Datenobjekt Kunde**

Datenobjekt: Termin		
Feld	Typ	Auswahlmöglichkeiten
Datum	String	-
Uhrzeit	String	-
Terminstatus	String, (nur eine Auswahl möglich)	- definitiv - provisorisch
Sitzplatz	String, (nur eine Auswahl möglich)	- Stuhl 1 - Stuhl 2 - Stuhl 3
Programm	String, (mehrere Auswahlmöglichkeiten)	- mit waschen - ohne waschen - mit Styling
Preis	Integer	-

**Tabelle 10: Übersicht Datenobjekt Termin**

#### 4.2.3.1 Erstellen eines Feldes

Für die Erstellung eines Feldes wird zwingend ein Posttype benötigt. Eine Reihe von Print-Screens illustriert diesen schrittweisen Vorgang. Die Grundlage für die Print-Screens ist dabei der Business Case von *Hair&Care*.

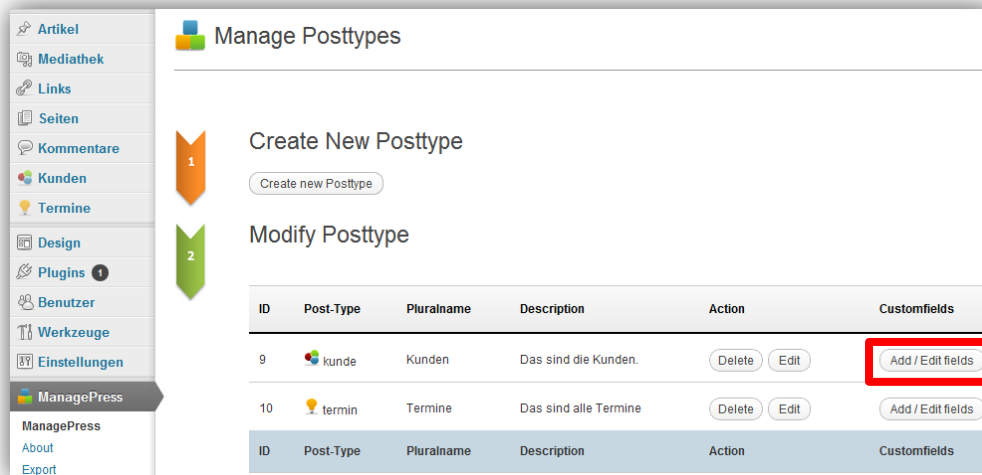


Abbildung 16: Übersichtsseite, Erstellung von Feldern

Mit der Auswahl von [Add / Edit Field] (Abbildung 16) gelangt man zur Übersichtsseite der Felder (Abbildung 17). Auf dieser Seite können Felder erstellt, bearbeitet und gelöscht werden. Mit einem Klick auf [Add Field] können neue Felder erstellt werden.

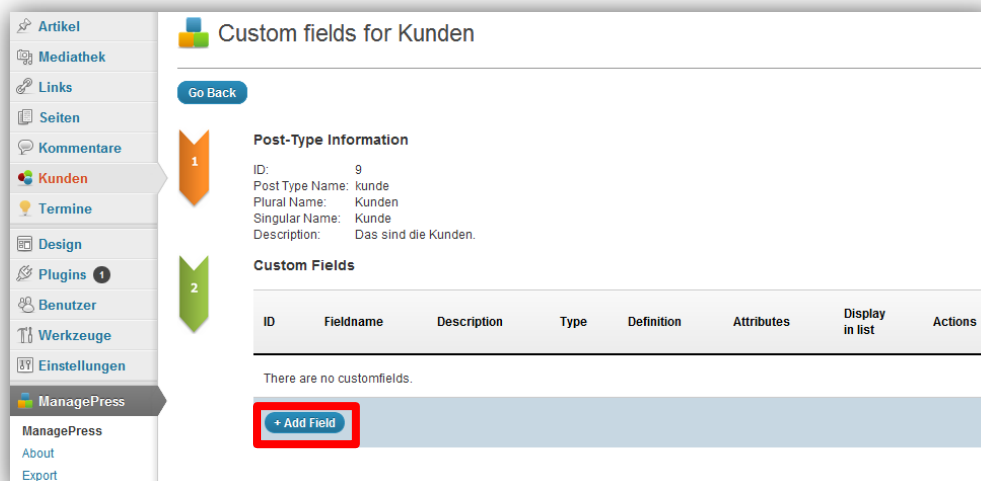


Abbildung 17: Hinzufügen eines neuen Feldes

In den nachfolgenden Abbildungen, Abbildung 18 bis Abbildung 23, sind alle Felder abgebildet, die für den Posttype „kunde“ erstellt werden. Als Basis dient dabei die Tabelle 9.

The screenshot shows the 'Add new Field' dialog box. The 'Field Name' is 'Vorname', the 'Field Description' is 'Das ist der Vorname vom Kunden', and 'Show this field in the list of all Posts?' is set to 'Yes'. The 'Field Type' is 'Text' and the 'Field Definition' is 'Single Input Field'. The 'Save Field' button is highlighted with a red box.

Abbildung 18: Erstellung Feld "Vorname"

The screenshot shows the 'Add new Field' dialog box. The 'Field Name' is 'Nachname', the 'Field Description' is 'Das ist der Nachname vom Kunden', and 'Show this field in the list of all Posts?' is set to 'Yes'. The 'Field Type' is 'Text' and the 'Field Definition' is 'Single Input Field'. The 'Save Field' button is highlighted with a red box.

Abbildung 19: Erstellung Feld "Nachname"

The screenshot shows the 'Add new Field' dialog box. The 'Field Name' is 'Telefonnummer', the 'Field Description' is 'Die Telefonnummer vom Kunden. Es können mehrere', and 'Show this field in the list of all Posts?' is set to 'Yes'. The 'Field Type' is 'Text' and the 'Field Definition' is 'Multiple Input Field'. The 'Save Field' button is highlighted with a red box.

Abbildung 20: Erstellung Feld "Telefonnummer"

The screenshot shows the 'Add new Field' dialog box. The 'Field Name' is 'E-Mail-Adressen', the 'Field Description' is 'Alle Email-Adressen, die der Kunde hat. mehrere Auswahl möglich.', and 'Show this field in the list of all Posts?' is set to 'Yes'. The 'Field Type' is 'Text' and the 'Field Definition' is 'Multiple Input Field'. The 'Save Field' button is highlighted with a red box.

Abbildung 21: Erstellung Feld "E-Mail-Adressen"

The screenshot shows the 'Add new Field' dialog box. The 'Field Name' is 'Wichtige Notizen', the 'Field Description' is 'Das ist ein Textfeld mit wichtigen Notizen.', and 'Show this field in the list of all Posts?' is set to 'No'. The 'Field Type' is 'Longtext' and the 'Field Definition' is 'Single Input Field'. The 'Save Field' button is highlighted with a red box.

Abbildung 22: Erstellung Feld "Wichtige Notizen"

The screenshot shows the 'Add new Field' dialog box. The 'Field Name' is 'Adresse', the 'Field Description' is 'Adresse vom Kunden auf GoogleMaps', and 'Show this field in the list of all Posts?' is set to 'Yes'. The 'Field Type' is 'Place on GoogleMaps' and the 'Field Definition' is 'Single Input Field'. The 'Save Field' button is highlighted with a red box.

Abbildung 23: Erstellung Feld "Adressen"

Das Ergebnis der Erstellung der einzelnen Felder wird in Abbildung 24 gezeigt. Hier sind alle Felder aufgelistet, die dem Posttype „kunde“ angehören.

**Custom fields for Kunden**

[Go Back](#)

**Post-Type Information**

ID: 9  
Post Type Name: kunde  
Plural Name: Kunden  
Singular Name: Kunde  
Description: Das sind die Kunden.

**Custom Fields**

ID	Fieldname	Description	Type	Definition	Attributes	Display in list	Actions
1	Vorname	Das ist der Vorname vom Kunden	text	input		yes	<a href="#">Edit</a> <a href="#">Delete</a>
2	Nachname	Das ist der Nachname vom Kunden	text	input		yes	<a href="#">Edit</a> <a href="#">Delete</a>
3	Telefonnummer	Die Telefonnummer vom Kunden. Es können mehrere eingegeben werden	integer	multipleinput		yes	<a href="#">Edit</a> <a href="#">Delete</a>
4	E-Mail-Adressen	Alle Email-Adressen, die der Kunde hat. mehrere Auswahl möglich.	text	multipleinput		yes	<a href="#">Edit</a> <a href="#">Delete</a>
5	Wichtige Notizen	Das ist ein Textfeld mit wichtigen Notizen.	longtext	input		no	<a href="#">Edit</a> <a href="#">Delete</a>
6	Adresse	Adresse vom Kunden auf GoogleMaps	coordinates	input		yes	<a href="#">Edit</a> <a href="#">Delete</a>

[+ Add Field](#)

Abbildung 24: Übersichtsseite, alle Felder von den Kunden

Gemäss dem Business Case von *Hair&Care* werden in Abbildung 25 bis Abbildung 30 alle Felder für den Posttype „termin“ erstellt.

**Add new Field**

Field Name: Datum

Field Description: Das Datum vom Termin.

Show this field in the list of all Posts? ☒ Yes ☐ No

Field Type: Date

Field Definition: Single Input Field

Save Field

Abbildung 25: Erstellung Feld "Datum"

**Add new Field**

Field Name: Uhrzeit

Field Description: Die Uhrzeit vom Termin

Show this field in the list of all Posts? ☒ Yes ☐ No

Field Type: Time

Field Definition: Single Input Field

Save Field

Abbildung 26: Erstellung Feld "Uhrzeit"

**Add new Field**

Field Name: Terminstatus

Field Description: Terminstatus für die Planung.

Show this field in the list of all Posts? ☒ Yes ☐ No

Field Type: Text

Field Definition: Single Choice

Attribute: definitiv

Attribute: provisorisch

add attribute

Save Field

Abbildung 27: Erstellung Feld "Terminstatus"

**Add new Field**

Field Name: Sitzplatz

Field Description: Welcher Sitzplatz wird für den Termin gebraucht?

Show this field in the list of all Posts? ☒ Yes ☐ No

Field Type: Text

Field Definition: Single Choice

Attribute: Stuhl 1

Attribute: Stuhl 2

Attribute: Stuhl 3

add attribute

Save Field

Abbildung 28: Erstellung Feld "Sitzplatz"

**Add new Field**

Field Name: Programm

Field Description: Was wünscht der Kunde genau?

Show this field in the list of all Posts? ☐ Yes ☒ No

Field Type: Text

Field Definition: Multiple Choice

Attribute: mit waschen

Attribute: ohne waschen

Attribute: mit Styling

Attribute: Beratung

add attribute

Save Field

Abbildung 29: Erstellung Feld "Programm"

**Add new Field**

Field Name: Preis

Field Description: Preis für die Auswertung und interne Planung.

Show this field in the list of all Posts? ☐ Yes ☒ No

Field Type: Integer

Field Definition: Single Input Field

Save Field

Abbildung 30: Erstellung Feld "Preis"

Abbildung 31 zeigt alle erstellten Felder vom Posttype „termin“.

**Custom fields for Termine**

[Go Back](#)

**Post-Type Information**

1  
ID: 10  
Post Type Name: termin  
Plural Name: Termine  
Singular Name: Termin  
Description: Das sind alle Termine

**Custom Fields**

2

ID	Fieldname	Description	Type	Definition	Attributes	Display in list	Actions
7	Datum	Das Datum vom Termin.	date	input		yes	<a href="#">Edit</a> <a href="#">Delete</a>
8	Uhrzeit	Die Uhrzeit vom Termin	time	input		yes	<a href="#">Edit</a> <a href="#">Delete</a>
9	Terminstatus	Terminstatus für die Planung.	text	singlechoice	- definitiv - provisorisch	yes	<a href="#">Edit</a> <a href="#">Delete</a>
10	Sitzplatz	Welcher Sitzplatz wird für den Termin gebraucht?	text	singlechoice	- Stuhl 1 - Stuhl 2 - Stuhl 3	yes	<a href="#">Edit</a> <a href="#">Delete</a>
11	Programm	Was wünscht der Kunde genau?	text	multiplechoice	- mit waschen - ohne waschen - mit Styling - Beratung	no	<a href="#">Edit</a> <a href="#">Delete</a>
12	Preis	Preis für die Auswertung und interne Planung.	integer	input		no	<a href="#">Edit</a> <a href="#">Delete</a>

[+ Add Field](#)

Abbildung 31: Übersichtsseite, alle Felder von den Terminen

#### 4.2.3.2 Löschen eines Feldes

Bei der Löschung eines Feldes wird die Tabelle „mps\_field\_[field\_table\_name] und der Eintrag in der Tabelle „mps\_core\_fields“ aus der Datenbank gelöscht. Durch die Auswahl [Delete] (Abbildung 32) wird der Löschvorgang gestartet und muss gemäss Abbildung 33 zusätzlich mit [Yes] bestätigt werden.

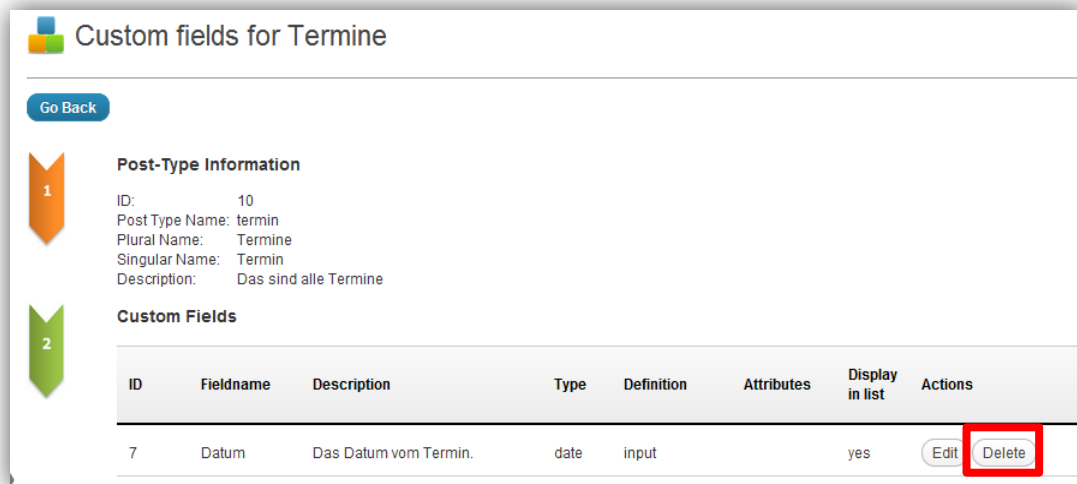


Abbildung 32: Löschen eines Feldes



Abbildung 33: Bestätigung für die Löschung eines Feldes

#### 4.2.3.3 Bearbeiten eines Feldes

Damit ein Feld bearbeitet werden kann, muss zuerst auf den Button [Edit] geklickt werden. Dieser erste Schritt ist in Abbildung 34 dargestellt.

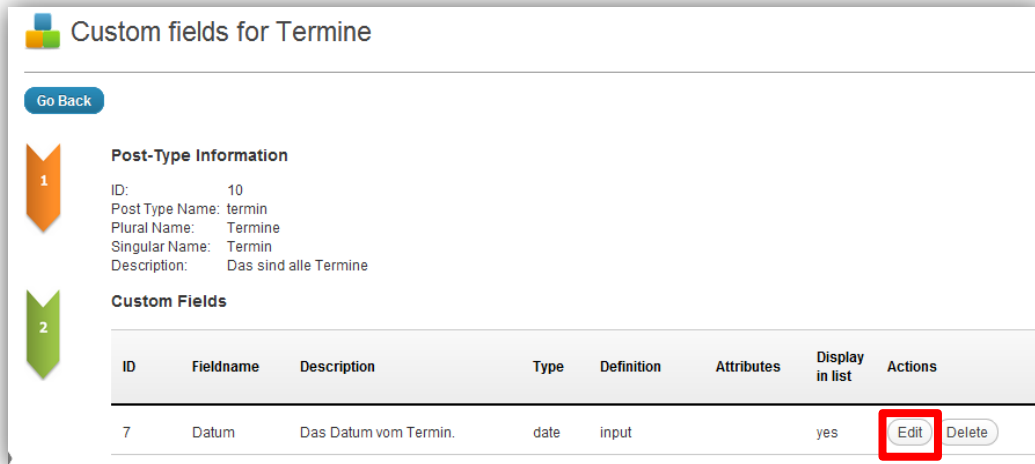


Abbildung 34: Bearbeitung eines Feldes

Im nächsten Schritt erscheint das Bearbeitungsformular gemäss Abbildung 35. Die einzelnen Felder können nun angepasst werden und mit [Save Field] bestätigt werden. Durch das Bestätigen werden alle neuen Informationen in die Datenbank geschrieben. Falls der Typ des Feldes geändert wurde, wird die entsprechende Feldtabelle in der Datenbank angepasst. Bei diesem Vorgang kann es vorkommen, dass die bereits vorhandenen Einträge gelöscht werden. Als Beispiel nehmen wir an, dass der Feldtyp von String zu Integer gewechselt wurde. Die Datenbank versucht nun jeden String in einen Integer umzuwandeln, was in den meisten Fällen nicht funktionieren wird, wodurch der Wert auf 0 gesetzt wird.

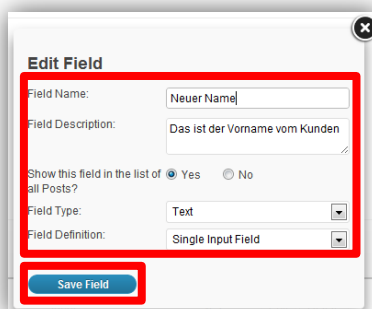


Abbildung 35: Formular bei Bearbeitung eines Feldes



#### 4.2.3.4 Eigenschaften der Felder

In Tabelle 11, Tabelle 12 und Tabelle 13 sind alle möglichen Feldoptionen dargestellt.

Feld-Typ	Beschreibung
Text	 max. Länge 2 <sup>16</sup>
Longtext	 max. Länge 2 <sup>32</sup>
Integer	 max. 4 Bytes
Big-Integer	 max. 8 Bytes
Date	
Time	
Place on GoogleMaps	

**Tabelle 11: Verschiedene Typen der Felder**



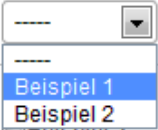
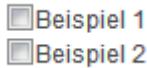
Feld-Definition	Beschreibung	Darstellung
Single Input Field	Dabei handelt es sich um ein Feld, dass nur ein Eintrag erlaubt.	
Multiple Input Field	Bei diesem Feld können beliebig viele Einträge hinzugefügt werden.	
Single Choice	Single Choice bedeutet, dass bereits Attribute vordefiniert werden und ein Attribut ausgewählt werden kann.	
Multiple Choice	Ähnlich wie Single choice, jedoch können mehrere Attribute ausgewählt werden.	

Tabelle 12: Verschiedene Definitionen der Felder

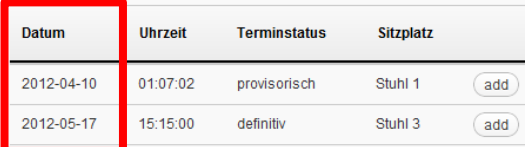
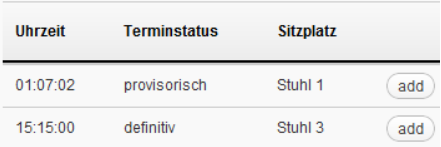
Darstellungsoption	Beschreibung
<b>“Show this field in the list of all Posts?”</b>	
Yes	 <p>→Darstellungsoption beim Feld “Datum” ist auf Yes</p>
No	 <p>→ Darstellungsoption beim Feld “Datum” ist auf No</p>

Tabelle 13: Feldoption - Darstellungsoption

## 4.3 Export

In diesem Kapitel wird die Exportfunktion von ManagePress Suite erläutert. Mit dem gekennzeichneten Menüpunkt in Abbildung 36 erscheint die Übersichtsseite vom Export. Es erscheint eine Auflistung aller erstellten Posttypes. Gemäss dem Business Case von *Hair&Care* sind die Posttypes „kunde“ und „termin“ sichtbar und können jetzt exportiert werden. Im Fall von *Hair&Care* kann somit eine aktuelle Kopie von ManagePress Suite erstellt werden und vom Informatikunternehmen an den Coiffeur-Salon ausgeliefert werden.

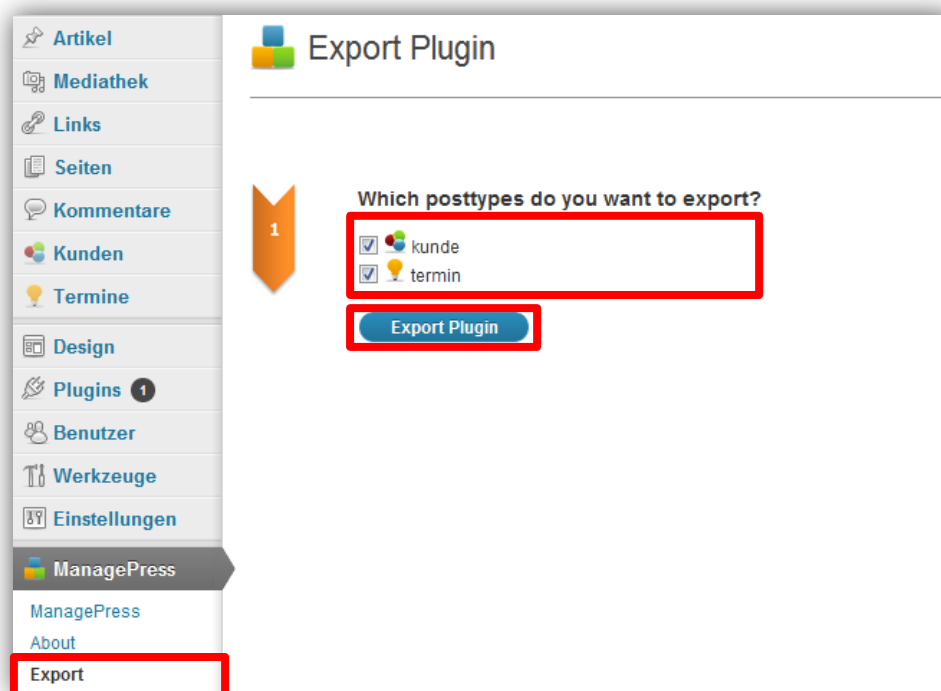


Abbildung 36: Übersicht Export

Mit einem Klick auf [Export Plugin] wird das Plugin erstellt und kann gemäss Abbildung 37 heruntergeladen werden.

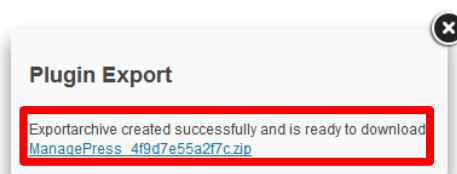
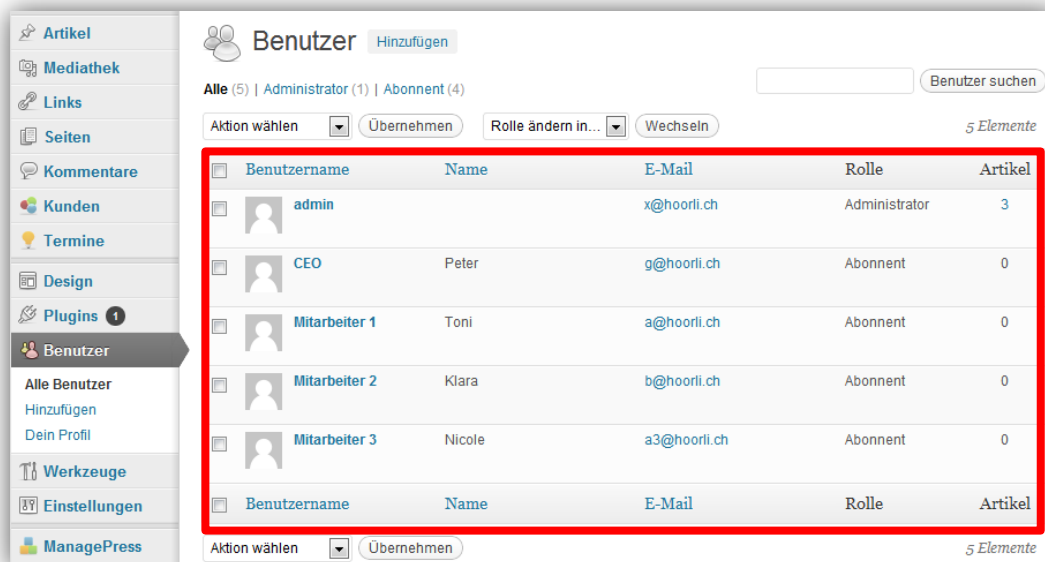


Abbildung 37: Export abschliessen

## 4.4 Operative Tätigkeit

In den Kapiteln 4.2 Konfiguration und 4.3 Export wurde die Grundlage für die operative Tätigkeit von ManagePress Suite gelegt. In diesem Kapitel wird beschrieben, wie ManagePress Suite operativ genutzt werden kann. Dabei wird immer noch vom Business Case ausgegangen, das heisst die Posttypes und Felder wurden erstellt und der Export vom Plugin war erfolgreich. Nun ist die exportierte ManagePress Suite bei *Hair&Care* installiert und für die operative Tätigkeit bereit.

Um die operative Tätigkeit für *Hair&Care* möglichst realitätsnah zu veranschaulichen, wird davon ausgegangen, dass der Geschäftsführer und seine drei Mitarbeiter bereits in WordPress als Benutzer erfasst sind (Abbildung 38).



The screenshot shows the 'Benutzer' (Users) management page in the ManagePress Suite. The left sidebar contains navigation links for Artikel, Mediathek, Links, Seiten, Kommentare, Kunden, Termine, Design, Plugins, Benutzer, Werkzeuge, Einstellungen, and ManagePress. The main content area displays a list of users. At the top, there are filters for 'Alle (5)', 'Administrator (1)', and 'Abonnent (4)', along with a search bar and a 'Benutzer suchen' button. Below the filters, there are buttons for 'Aktion wählen', 'Übernehmen', 'Rolle ändern in...', and 'Wechseln'. The user list table has the following data:

Benutzername	Name	E-Mail	Rolle	Artikel
admin		x@hoorli.ch	Administrator	3
CEO	Peter	g@hoorli.ch	Abonnent	0
Mitarbeiter 1	Toni	a@hoorli.ch	Abonnent	0
Mitarbeiter 2	Klara	b@hoorli.ch	Abonnent	0
Mitarbeiter 3	Nicole	a3@hoorli.ch	Abonnent	0

At the bottom of the table, there are buttons for 'Aktion wählen' and 'Übernehmen', and a note '5 Elemente'.

Abbildung 38: Übersicht registrierte Benutzer

Die operative Tätigkeit wird in folgende Abschnitte unterteilt:

- Erstellen eines Datenobjekts
- Dashboard
- Shortcode
- Frontend Ansicht

#### 4.4.1 Erstellen eines Datenobjekts

Dieser Abschnitt wird in zwei Unterkapitel aufgeteilt. Zum einen wird ein Datenobjekt des Posttypes „kunde“ und zum anderen ein Datenobjekt des Posttypes „termin“ erstellt.

##### 4.4.1.1 Erstellen eines Kunden

In den nachfolgenden Abbildungen wird der Ablauf beschrieben, wie ein neuer Kunde erfasst wird.

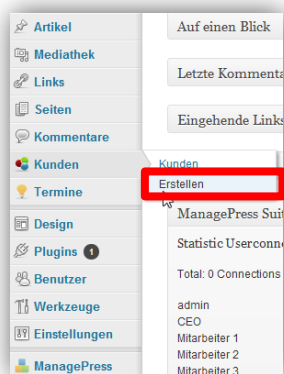


Abbildung 39: Erstellung eines Kunden, Schritt 1

A screenshot of the 'Add new Kunde' form in the WordPress dashboard. The form is titled 'Add new Kunde' and contains a 'Meta Box for Kunden' with several input fields: 'Vorname' (Manuel), 'Nachname' (Meier), 'Telefonnummer' (076 312 45 68, 044 462 59 89), 'E-Mail-Adressen' (pet@juhu.com), 'Wichtige Notizen' (- spielt Fussball), 'Adresse' (Höfliweg), and 'User Connection' (admin, CEO, Mitarbeiter 1, Mitarbeiter 2, Mitarbeiter 3). On the right side, there is a 'Veröffentlichen' section with buttons for 'Speichern', 'Vorschau', 'Status: Entwurf Bearbeiten', 'Sichtbarkeit: Öffentlich Bearbeiten', 'Sofort veröffentlichen Bearbeiten', 'In den Papierkorb legen', and a red-bordered 'Veröffentlichen' button.

Abbildung 40: Erstellung eines Kunden, Schritt 2

Nachdem ein Kunde erstellt wurde, ist der Kunde in der Übersicht sichtbar. Dabei werden in der gezeigten Auflistung nur diejenigen Felder als Spalten aufgelistet, die in der Eigenschaft „Show this field in the list of all Posts?“ auf Yes gesetzt worden sind. Bei den Kunden ist deshalb das Feld „Wichtige Notizen“ nicht sichtbar.

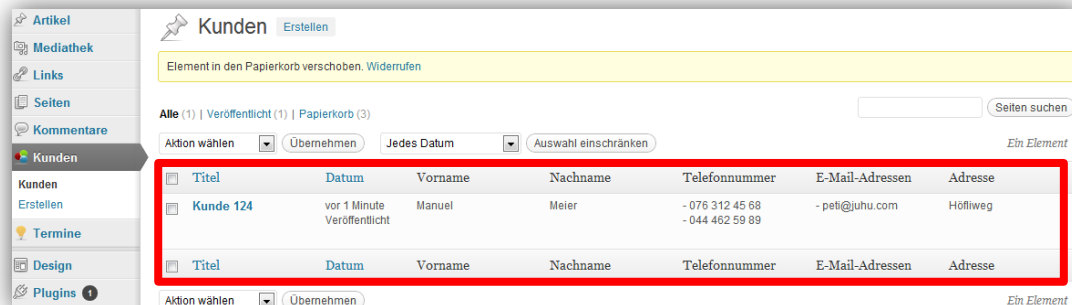


Abbildung 41: Übersicht aller Kunden

#### 4.4.1.2 Erstellen eines Termins

Das Erstellen eines Termins wird in Abbildung 42 und Abbildung 43 gezeigt. Wichtig sind dabei die Optionen „User Connection“ und „Child element of“. Mit diesen Optionen kann *Hair&Care* Beziehungen zwischen Kunden und Mitarbeitern bilden und somit ManagePress Suite optimal für die Terminverwaltung nutzen.



Abbildung 42: Erstellung eines Termins, Schritt 1

Abbildung 43: Erstellung eines Termins, Schritt 2

Mit einem Klick auf [add parent element] erscheint das Auswahlfenster für die Beziehungen. In Abbildung 44 ist das Auswahlfenster abgebildet. Durch die Auswahl des Posttypes und der Suchfunktion kann jedes beliebige Datenobjekt gesucht und mit dem Button [add] hinzugefügt werden.

Name	geburt	haare	email
Pascal	1988-01-19	-	-

Abbildung 44: Auswahlfenster für ein Elternelement

#### 4.4.2 Dashboard

Eine weitere Funktionalität für die operative Tätigkeit ist im Dashboard von WordPress sichtbar (Abbildung 45).

- ManagePress Suite / Statistic
  - In dieser Abbildung wird eine Übersicht dargestellt, wie oft ein Benutzer in der „User Connection“ mit einem Datenobjekt verknüpft wurde.
- ManagePress Suite / My Posts
  - Hier werden alle Posts aufgelistet, welche eine User Connection mit dem aktuell eingeloggten Benutzer haben.

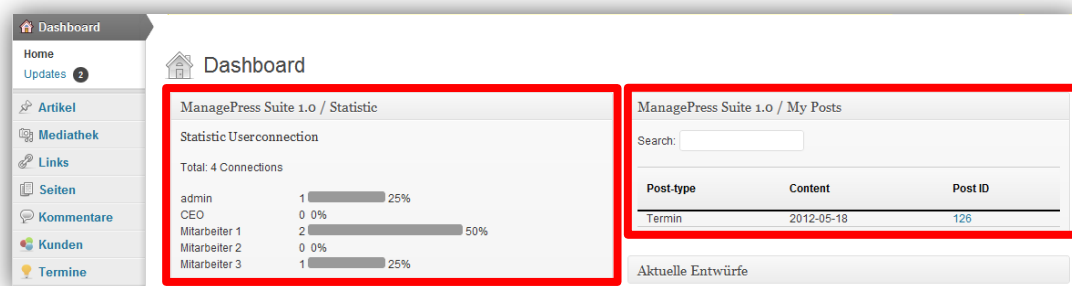


Abbildung 45: Dashboard mit ManagePress Suite



### 4.4.3 Frontend Ansicht

#### 4.4.3.1 Shortcode

ManagePress Suite unterstützt einen Shortcode. Der Shortcode ist hilfreich für das Verwalten der Daten. In der Abbildung 46 wird der Shortcode für den Posttype „kunde“ benutzt.

Shortcode: `[mps_list posttype="kunde"]`

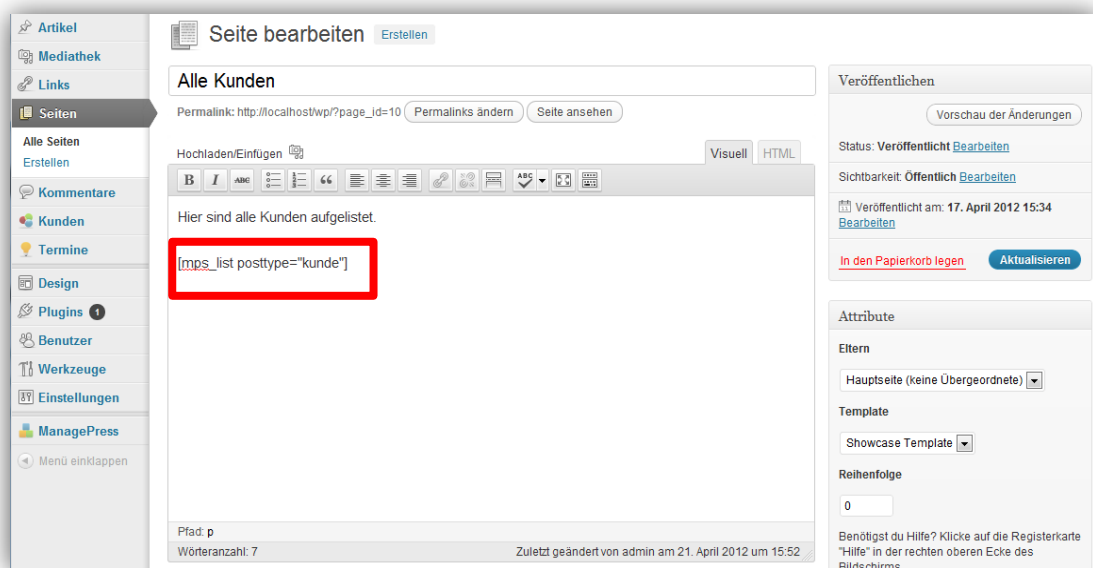


Abbildung 46: Beispiel Shortcode mit Posttype "kunde"

Sobald die Seite mit dem Shortcode veröffentlicht ist, kann die Seite im Frontend von WordPress betrachtet werden (Abbildung 47). Die Daten können in dieser Ansicht durchsucht, exportiert oder kopiert werden.



Abbildung 47: Shortcode Ansicht Tabelle

In der Abbildung 48 ist eine weitere Ansichtsmöglichkeit sichtbar. Diese Ansicht ist nur möglich, wenn ein Feld als „Place on GoogleMaps“ definiert wurde.

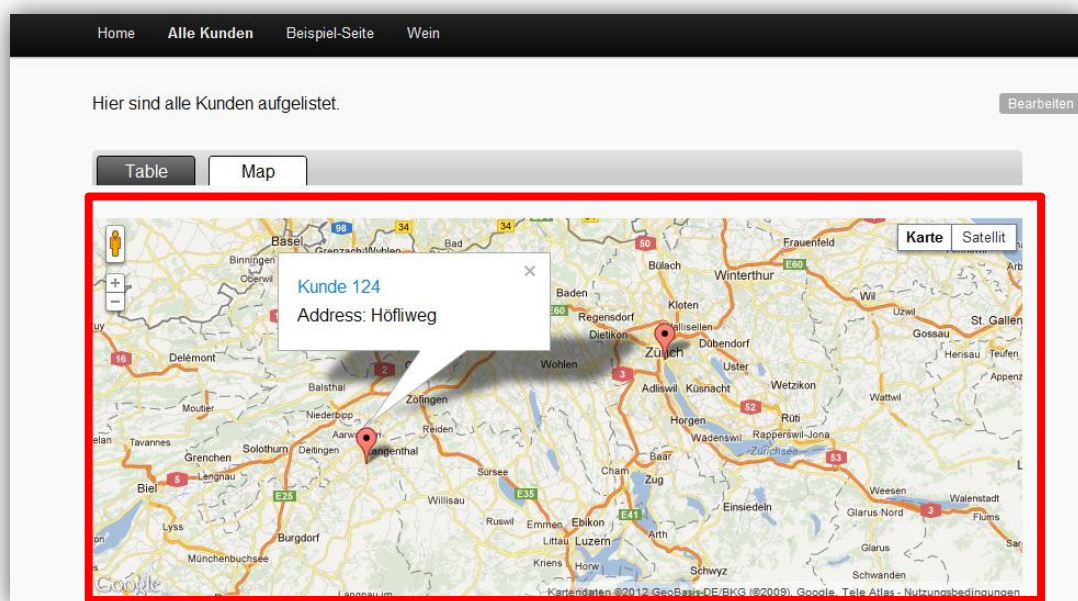


Abbildung 48: Shortcode Ansicht GoogleMaps

#### 4.4.3.2 Einzel-Ansicht

In Abbildung 49 wird die Einzel-Ansicht eines Datenobjektes von ManagePress Suite dargestellt. Dieses spezielle Template wird bei jedem Datenobjekt angewendet, welches als Grundlage die ManagePress Suite hat. Beim dargestellten Datenobjekt sind die Kundendaten vom Kunden 130 und seine Beziehungen zu den Terminen ersichtlich.

[Bearbeiten](#)

## Kunde 130

Vorname	Corinne
Nachname	Schulthess
Telefonnummer	074 846 45 12
E-Mail-Adressen	ku@mu.com
Wichtige Notizen	-
Adresse	Bahnhofstrasse 10, zürich
Userconnection	<a href="#">Mitarbeiter 3</a>

Parent element of:

### Termine

- ↻ 2012-05-17, 15:15:00, definitiv, Stuhl 3, [125](#)
- ↻ 2012-04-10, 01:07:02, provisorisch, Stuhl 1, [118](#)

Dieser Eintrag wurde von [admin](#) veröffentlicht. Lesezeichen zum [Artikel setzen](#).

Abbildung 49: Frontend-Einzel-Ansicht eines Kunden

## 5 Schlussfolgerungen

---

### 5.1 Testeinsatz

Das entwickelte Plugin ManagePress Suite wurde im Rahmen der Vorlesung „Standards, Normen & Daten“ an der Zürcher Hochschule für angewandte Wissenschaft getestet. Die Vorlesung wurde von 15 Teilzeitstudierenden des Studiengangs Wirtschaftsinformatik besucht. Die Studenten wurden zu Beginn des Unterrichts in die Funktionalitäten von ManagePress Suite eingeführt. Nach der Einführung wurde der Klasse die Aufgabe gestellt, mit ManagePress Suite eine Datenverwaltung für CDs zu erstellen. Das Ziel für die Struktur der Datenverwaltung war, mindestens zwei Datentypen mit je zwei Eigenschaften zu generieren. Die Musterlösung der Struktur der Datenverwaltung bestand aus einem Datentyp „CD“ mit den Eigenschaften „CD-Titel“ und „Erscheinungsdatum“ und einem Datentyp „Track“ mit den Eigenschaften „Track-Titel“ und „Interpret“.

Der Testeinsatz war sehr aufschlussreich. Es konnte beobachtet werden, dass die Studierenden sich zuerst an die Umgebung von WordPress gewöhnen mussten. Nach kurzer Angewöhnungszeit konnten sie ManagePress Suite ohne Schwierigkeiten benutzen. Am Schluss gelang es allen Studierenden die Aufgabe gut zu lösen. Das Feedback der Studenten war insgesamt sehr positiv.

### 5.2 Weiterentwicklung

#### 5.2.1 Feldtypen

Eine Weiterentwicklungsmöglichkeit von ManagePress Suite besteht in der Ergänzung durch weitere spezifische Feldtypen. Zum Beispiel wäre es bei einer Kundenverwaltung sinnvoll ein Feldtyp „E-Mail“ zu erstellen, damit jede eingegebene E-Mail-Adresse nach Richtigkeit verifiziert werden kann.

Mögliche Feldtypen:

- Telefonnummer
- Land
- AHV-Nummer

- Bildupload
- Website
- E-Mail

### 5.2.2 Pflichtfelder

Die Möglichkeit, dass einige Felder als Pflichtfelder konfiguriert werden können, wäre eine weitere sinnvolle Eigenschaft für die Zukunft. Somit könnten minimale Anforderungen an ein Datenobjekt vorgegeben werden, damit keine unvollständigen Daten in die Datenbank gespeichert werden und die Datenkonsistenz gewährleistet ist.

### 5.2.3 Import-Funktion

ManagePress Suite mit einer Import-Funktion zu versehen, wäre eine zusätzliche Möglichkeit einer Weiterentwicklung. Die Import-Funktion würde dem Benutzer erlauben bereits vorhandene Datenbestände in ManagePress Suite zu importieren und direkt zu nutzen. Bei der Umsetzung stellt sich jedoch die Frage, ob die Import-Funktion selbst neue Posttypes generieren kann oder nur die Daten in die bereits manuell erstellten Posttypes importiert.

## 6 Literaturverzeichnis

---

- Adler, O. (2011). *Praxiswissen WordPress*. Köln, Deutschland: O'Reilly Verlag.
- Alby, T. (2008). *Professionell bloggen mit WordPress*. München, Deutschland: Carl Hanser Verlag München.
- Brown, A. (3. 8 2011). *All plugin hooks in WordPress 3.3*. URL: [http://adambrown.info/p/wp\\_hooks/version/3.3](http://adambrown.info/p/wp_hooks/version/3.3) [13.5.2012]
- WordPress.org. (2012a). *WordPress Download Counter*. URL: <http://wordpress.org/download/counter/> [13.5.2012]
- WordPress.org. (2012b). *Post Types - WordPress Codex*. URL: [http://codex.wordpress.org/Post\\_Types](http://codex.wordpress.org/Post_Types) [13.5.2012]
- WordPress.org. (2012c). *Template\_Hierarchy*. URL: [http://codex.wordpress.org/images/1/18/Template\\_Hierarchy.png](http://codex.wordpress.org/images/1/18/Template_Hierarchy.png) [13.5.2012]
- WordPress.org. (2012d). *Plugin API - WordPress Codex*. URL: [http://codex.wordpress.org/Plugin\\_API](http://codex.wordpress.org/Plugin_API) [13.5.2012]

## 7 Anhang

### 7.1 Template Hierarchie von WordPress

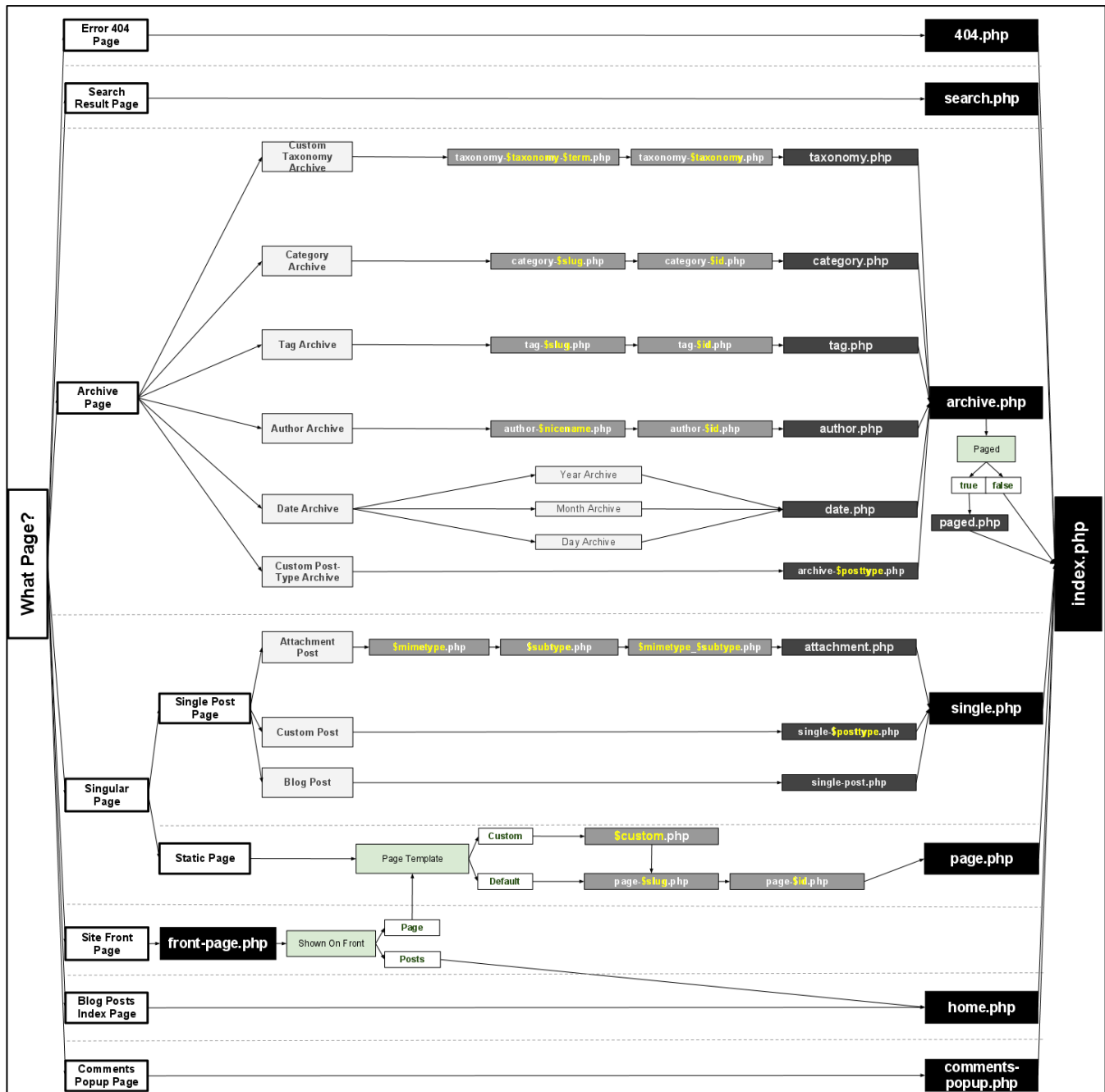


Abbildung 50: Template Hierarchie von WordPress, code.wordpress.org (2012c)