

ZHAW Seminararbeit

Webprojekt PHP/MySQL



Feuerwehr Off-Spick

BackEnd

Michael Ott, i10b

11. Juni 2013

Inhaltsverzeichnis

1. Einleitung	5
1.1. Motivation	5
1.2. Aufgabenstellung gemäss Eingabe EBS (ZHAW)	5
1.2.1. Ausgangslage	5
1.2.2. Ziel der Arbeit	5
1.2.3. Aufgabenstellung	5
1.2.4. Erwartetes Resultat	6
1.3. Wahl des PHP-Frameworks	6
2. Randbedingungen / Kontext	7
2.1. Konventionen	7
2.2. Technische Randbedingungen	7
2.3. Termine	7
3. Projektplanung	8
3.1. Aufwandschätzung	8
3.2. Projektplan (tabellarisch)	9
3.3. Projektplan (grafisch)	10
4. Einführung CakePHP	11
4.1. Anforderungen	11
4.2. Merkmale / Funktionen	11
4.2.1. Model	11
4.2.2. Controller	12
4.2.3. Views	12
4.2.4. Scaffolding mit Cake Bake	12
4.3. Installation	13
5. Systemübersicht	14
5.1. Systemabgrenzung	14
5.2. Stakeholder	15
5.3. Funktionale Anforderungen	15
5.4. Nichtfunktionale Anforderungen	15
5.4.1. Nutzer	16
5.4.2. Administrator	17

6. Anforderungen für Module	18
6.1. Basismodul	19
6.1.1. Rahmenbedingungen	19
6.1.2. Anforderungen	20
6.1.3. Testing	20
6.2. Modul Telefonbuch	21
6.2.1. Rahmenbedingungen	21
6.2.2. Anforderungen	22
6.2.3. Testing	22
6.3. Modul News	23
6.3.1. Rahmenbedingungen	23
6.3.2. Anforderungen	23
6.3.3. Testing	23
6.4. Modul Checkliste	24
6.4.1. Rahmenbedingungen	24
6.4.2. Anforderungen	24
6.4.3. Testing	24
7. Konzept Backend-Seite	25
7.0.4. Login	25
7.0.5. Übersicht	26
7.0.6. Datensatz erfassen	27
8. Umsetzung Backend	28
8.1. Datenbank	28
8.2. MVC-Scaffold	28
8.2.1. Model erstellen	28
8.2.2. Controller erstellen	28
8.2.3. View erstellen	28
8.3. Manuelle Anpassung	28
8.3.1. Anpassung Aufbau	28
8.3.2. Anpassung Optik	29
9. Umsetzung Modul Telefonbuch	30
9.1. Neuaufbau	30
10. Umsetzung Modul Info	31
11. Integration App	32
12. Projekt-Tools	33
13. Fazit / Ideen für Erweiterungen	34
A. Abbildungsverzeichnis	35

B. Literaturverzeichnis	36
C. Quellcode	37

Glossar

Begriff	Beschreibung	Link
Apache	Freier Webserver	httpd.apache.org
CakePHP	PHP Framework	www.cakephp.org
CRUD	Create / Retrieve / Update / Delete: Grundlegende Datenbank-Operationen	
EBS	Elektronisches Bewertungssystem - System der ZHAW	https://ebs.zhaw.ch
FTP-Protokoll	FileTransfer-Protokoll: Übertragungsprotokoll für Daten	
FZWB	Feuerwehr Zweckverband Wiesendangen-Bertschikon	www.feuerwehr-wiesendangen.ch
Handheld	Mobiltelefone mit erweiterten Funktionen	
Handheld-App	Programm, welches für Einsatz auf Handheld optimiert ist	
HTTP	Hypertext Transfer Protocol - Protokoll zur Datenübertragung	
JSON	JavaScript Object Notation - kompaktes Datenformat	www.json.org
MySQL	relationalen OpenSource Datenbankverwaltungssystem	www.mysql.org
MVC-Model	Model/View/Controller: Paradigma zur klaren Trennung von Daten / Ansicht und Logik	
Nutzer	Entspricht in diesem Kontext einem Feuerwehr-Angehörigen	
OffSpick	Offizier-Spick - Telefonverzeichnis für weitere Einsatzelemente, Firmen oder Fachpersonen	
REST	Representational State Transfer - Jede Abfrage (spez. Datensatz, Update eines Datensatzes,...) erfolgt direkt über einen eigenen Link	
Scaffold	Beim Scaffolding automatisiert ein Grundgerüst der Applikation erstellt	
WebApp	Siehe Handheld-App	
ZHAW	Zürcher Hochschule für angewandte Wissenschaften	www.zhaw.ch

1. Einleitung

1.1. Motivation

Mobile Geräte ab Stufe Ortsfeuerwehr sind im Kanton Zürich zur Zeit noch nicht im Einsatz. Dieses OffSpickBackend soll Teil der ersten Generation von elektronisch unterstützen Diensten im Feuerwehr-Einsatz darstellen.

Gleichzeitig stellt diese Arbeit eine Vertiefung im Requirement Engineering und eine Einführung in das Arbeiten mit einem PHP-Framework dar.

1.2. Aufgabenstellung gemäss Eingabe EBS (ZHAW)

1.2.1. Ausgangslage

Für eine bestehende Handheld-App (OffSpick) existiert ein Backend welches für die Verwaltung der Daten zuständig ist. Dieses Backend ist jedoch sehr rudimentär und nur via Admin bedienbar.

1.2.2. Ziel der Arbeit

Überarbeitung eines bestehenden Handheld-Applikation Backend mittels Unterstützung des PHP-Frameworks 'CakePHP'. Die Verwaltung der Benutzer-/Nutzdaten soll via Homepage durch eine Nicht-IT-Person erfolgen können. Das Backend soll sich später einfach modular erweitern lassen.

1.2.3. Aufgabenstellung

1. Einarbeitung in CakePHP-Framework
2. Anforderungen möglicher Module (Telefonverzeichnis, Wochenpikett-Verantwortlicher, Standard-Checklisten, Wetterwarnungen, Hydrantensuche,...) definieren
3. Konzept für Admin-Webseite erstellen
4. Erstellen des Backend mit Session-Authentifizierung und REST-konformem Zugriff auf die Module
5. Telefonbuch-Modul implementieren
6. Einbindung/Testing der bestehenden Handheld-Applikation
7. Dokumentation erstellen
8. Fazit erstellen

1.2.4. Erwartetes Resultat

Eine ausführliche Projekt-Dokumentation und eine Handheld-App mit folgenden Eigenschaften:

1. in CakePHP-Framework eingearbeitet und Einarbeitung dokumentiert
2. Anforderungen für min. 3 mögliche Module spezifiziert (inkl. Telefonbuch-Modul)
3. Konzept für Webseite erstellt
4. Prototyp des in CakePHP umgesetzten Backend mit REST-Zugriff
5. Telefonbuch-Modul lauffähig
6. lauffähiger (ggf. angepasste) OffSpickApp
7. Dokumentation erstellt
8. Fazit erstellt

1.3. Wahl des PHP-Frameworks

Die Wahl des passenden PHP-Frameworks wurde auf Grund der folgenden Kriterien gefällt:

Framework	läuft mit PHP 5.3.4	MySQL-Support 5.0.77	letzte Version	Plugins	Forum	MVC	Gratis
CakePHP	Ja	Ja	10.06.13	Ja	++	Ja	Ja
Zend	Ja	Ja	15.05.13	Ja	++	Ja	Ja
Symfony	Ja	Ja	03.06.13	Ja	++	Ja	Ja

Da alle 3 geprüften Frameworks meine Anforderungen in etwa gleich abdecken, habe ich mich in meinem persönlichen Umfeld nach Tipps umgehört. Hier wurde mir eher CakePHP resp. Symfony empfohlen. Die Wahl fiel daher auf CakePHP.

2. Randbedingungen / Kontext

Die Applikation soll möglichst ohne Zusatzkosten auf dem bestehenden Webserver des FZWB (Feuerwehr-Zweckverband Wiesendangen-Bertschikon) betrieben werden.

2.1. Konventionen

Bereich	Bedingung
Projektdokumentation	12-15 Seiten (gem. Vorgabe ZHAW)
Versionsmanagement	Quellcode muss durch ein geeignetes System verwaltet sein

2.2. Technische Randbedingungen

Funktion	System/Details	Version	Bemerkung
Webserver	Apache	2.2.3	Aktualisierung jederzeit durch Hoster möglich
PHP	PHP	5.3.4	Aktualisierung jederzeit durch Hoster möglich
Datenbank	MySQL	5.0.77	Aktualisierung jederzeit durch Hoster möglich
Framework	CakePHP	2.3	gem. Aufgabenstellung
Zugriff	via http(s)-Webservice		nur mittels gültiger Login-Daten

2.3. Termine

Datum	Schritt	Inhalt
13.03.2013	KickOff-Meeting	Projektstart
22.03.2013	Abgabe Projektantrag	Ausformulierung eines Projektantrages in EBS
11.06.2013	Abgabe Arbeit	Abgabe Dokumentation und Software
18.06.2013	Präsentation	Projekt-Präsentation

3. Projektplanung

3.1. Aufwandschätzung

Schritt	Geschätzt (h)	Effektiv (h)	Abweichung (h)
Projekteingabe	2	2	0
Grobkonzept	2	2	0
Anforderungen definieren	15	12	-3
Layout-Konzept Backend	2	2	0
Einrichten CakePHP	2	2	0
Einarbeitung CakePHP	20	30	+10
Umsetzung Basismodul	15	25	+10
Testing Basismodul	2	3	+1
Umsetzung Telefonbuch-Modul	15	in Arbeit	
Testing Telefonbuch-Modul	3	in Arbeit	
Umsetzung News-Modul	4	4	0
Testing News-Modul	1	1	0
Umsetzung Checklisten-Modul	optional		
Testing Checklisten-Modul	optional		
Dokumentation	10	15	+5
Präsentation	4	in Arbeit	
Total	97	(98)	(+23)

Der geschätzte Arbeitsaufwand ist höher als die in der Aufgabenstellung definierten 50h, da mir eine Umsetzung sehr wichtig ist. Durch die fehlenden Kenntnisse musste daher in der Aufwandschätzung auch die Einarbeitungszeit für die nötigen Sprachen und Konzepte einberechnet werden.

Zusätzlich wird in einem späteren Schritt noch ein entsprechendes Testing durchgeführt. Auf dieses wird aber im Rahmen dieses Proof-of-Concepts verzichtet.

3.2. Projektplan (tabellarisch)
























	Vorgang ▾	Vorgangsname ▾	Dauer ▾	Anfang ▾	Fertig stellen ▾	Vorgänger ▾
		KickOff	0 Tage	Mi 13.03.13	Mi 13.03.13	
		Projektantrag erstellen	1 Tag	Mi 13.03.13	Mi 13.03.13	1
		Abgabe Projektantrag	0 Tage	Fr 22.03.13	Fr 22.03.13	2
		Grobkonzept	1 Tag	Mo 25.03.13	Mo 25.03.13	3
		Anforderungen definieren	5 Tage	Di 26.03.13	Mo 01.04.13	4
		Layout-Konzept Backend	1 Tag	Di 02.04.13	Di 02.04.13	5
		Einrichten CakePHP-Entwicklungsumgebung	1 Tag	Mi 03.04.13	Mi 03.04.13	6
		Einarbeitung CakePHP	10 Tage	Do 04.04.13	Mi 17.04.13	7
		Umsetzung Basismodul	15 Tage	Do 18.04.13	Mi 08.05.13	8
		Testing Basismodul	1 Tag			
		Umsetzung News-Modul	4 Tage	Do 09.05.13	Di 14.05.13	9
		Testing News-Modul	0 Tage	Di 14.05.13	Di 14.05.13	11
		Umsetzung Telefonbuch-Modul	5 Tage	Mi 15.05.13	Di 21.05.13	12
		Testing Telefonbuch-Modul	3 Tage	Mi 22.05.13	Fr 24.05.13	13
		Dokumentation	65 Tage	Mi 13.03.13	Di 11.06.13	
		Projekt-Abgabe	1 Tag	Di 11.06.13	Di 11.06.13	
		Präsentation	5 Tage	Mi 12.06.13	Di 18.06.13	

Abbildung 3.1.: Projektplan - Tabelle

3.3. Projektplan (grafisch)

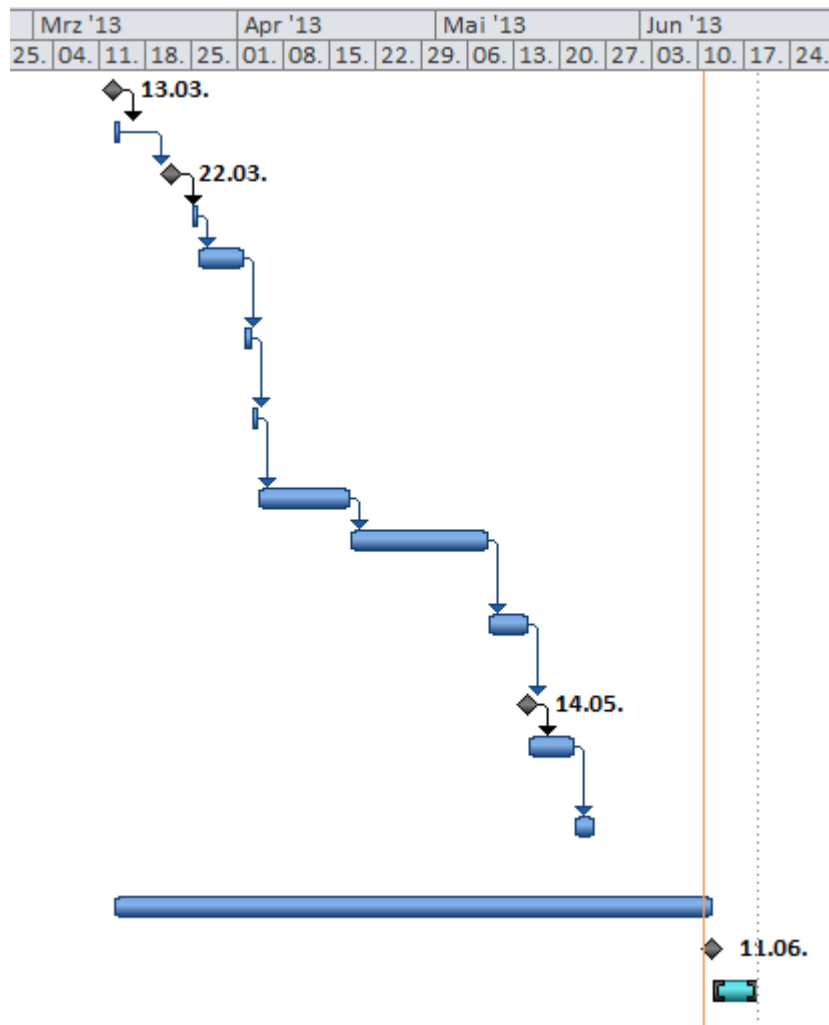


Abbildung 3.2.: Projektplan - Grafik

4. Einführung CakePHP

Das folgende Kapitel soll ein paar Punkte aus dem Framework herausheben, welche für dieses Projekt relevant waren. Für eine vollständige Dokumentation wird auf die Homepage von CakePHP verwiesen.

Zur Einarbeitung empfiehlt sich eine Installation von CakePHP [2] in einer XAMPP [4]-Installation. Anschliessend führte ich die sehr verständlich dokumentierte Tutorials [3] durch.

4.1. Anforderungen

Für die aktuelle, in diesem Projekt eingesetzte Version 2.3.5 wird ein HTTP-Webserver (z.B. Apache) mit PHP 5.2.8 oder höher benötigt. Zusätzlich wird der Einsatz einer Datenbank (z.B. MySQL 4 oder höher) empfohlen. Beide Anforderungen sind gem. Kapitel 2.1 erfüllt. Um auch die Möglichkeit vom Scaffolding zu nutzen, ist ein Shell-Zugriff nötig. Aus diesem Grund wurde die Entwicklung lokal auf einer XAMPP-Installation durchgeführt.

4.2. Merkmale / Funktionen

Das Framework zeichnet sich durch folgende Funktionen aus:

- Nutzung der MVC-Architektur um die Daten sauber von Logik und Ansicht zu trennen
- Bietet standardmässig CRUD (Create / Retrieve / Update / Delete) - Unterstützung
- integrierte Datenvalidierung bei Benutzereingaben
- Erstellen eines Grundgerüsts via Konsole (sog. Scaffolding)

Speziell für das Scaffolding empfiehlt es sich die Konventionen [1] einzuhalten:

4.2.1. Model

Das Model repräsentiert eine Tabelle in der Datenbank und ist u.a. auch für die Validierung der Daten zuständig (nur was den gegebenen Kriterien entspricht, wird auch gespeichert). Das System übernimmt die Ansteuerung mit den spezifische Datenbank-Eigenschaften. Die Model-Dateien sind im Verzeichnis `cakephp235/app/model` abgelegt.

- Model-Klassennamen sind singular und CamelCased (jeder Wortteil beginnt mit einem grossen Buchstaben)
- Feldnamen mit mehreren Wörter werden mit '_' getrennt (z.B. `news_categorie_id`)
- Die Bezeichnungen haben Einfluss auf die automatische Generierung von Models, damit werden Fremdschlüsselbeziehungen eruiert

4.2.2. Controller

Controller enthalten die Funktionen der Anwendung. Die Programmlogik im Controller greift bei Bedarf auf die Models zu oder stellt der View bestimmte Funktionen zur Verfügung. Die Controller-Dateien sind im Verzeichnis `cakephp235/app/controller` abgelegt.

4.2.3. Views

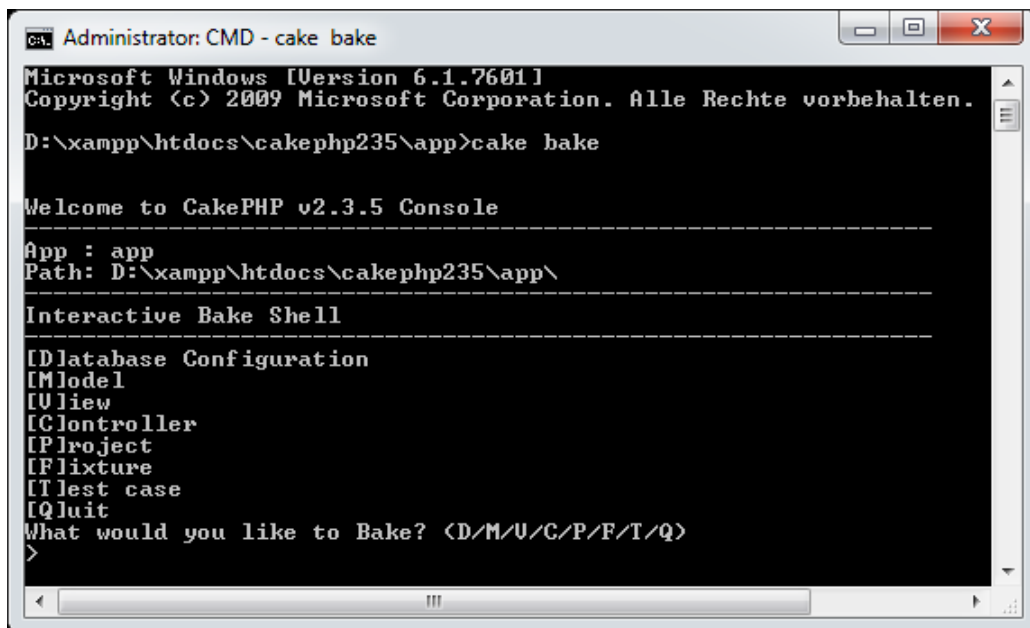
Darin sind die ganzen Layouts für die Darstellung verpackt. Zur Unterstützung stehen sog. Helpers zur Verfügung, welche gewisse Aufgaben (Formulare erstellen, Meldungen anzeigen, eMail,...) erleichtern.

Die View-Dateien sind im Verzeichnis `cakephp235/app/views` abgelegt.

4.2.4. Scaffolding mit Cake Bake

Um automatisiert ein Code-Grundgerüst zu erstellen, steht das Scaffolding Tool *cake bake* zur Verfügung. Man benötigt hierfür einen Consolen-Zugriff auf den Server. Im Fall einer lokalen XAMPP-Installation reicht der Aufruf von *cake bake* aus dem Pfad `\cakephp235\app` via Windows-Kommandozeile.

Der Aufruf funktioniert mittels Befehl *cake bake* wie folgt:



```
ca. Administrator: CMD - cake bake
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Alle Rechte vorbehalten.

D:\xampp\htdocs\cakephp235\app>cake bake

Welcome to CakePHP v2.3.5 Console
-----
App : app
Path: D:\xampp\htdocs\cakephp235\app\
-----
Interactive Bake Shell
-----
[D]atabase Configuration
[M]odel
[U]iew
[C]ontroller
[P]roject
[F]ixture
[T]est case
[Q]uit
What would you like to Bake? <D/M/U/C/P/F/T/Q>
>
```

Abbildung 4.1.: CakePHP - Scaffolding mit *cake bake*

Nach der Auswahl erscheinen nun je nach gewählter Aufgabe weitere Abfragen welche in einem einfachen und verständlichen Dialogsystem beantwortet werden müssen. Wurden beim Aufbau der Tabellen die Feldbezeichnungen (spez. bei Schlüssel) richtig aufgebaut, so schlägt *cake bake* die Schlüssel-Beziehungen entsprechend vor.

4.3. Installation

Die Installation von CakePHP erfolgt je nach Anwendungszweck: Entwicklung oder Produktion. Beide Varianten sind in der Installationsanleitung [2] auf der CakePHP-Homepage <http://book.cakephp.org/2.0/en/installation.html>.

Die folgende Kurzanleitung ist für dieses Projekt auf die Variante Entwicklung ausgerichtet.

Dieses Kapitel beschreibt die Installation auf dem Webserver oder einer XAMPP-Installation. Die Unterschiede sind jeweils mit a) oder b) bezeichnet. Für die detailliertere Informationen steht eine ausführliche Installationsanleitung bereit.

1. Download der neusten Version von CakePHP via <http://www.cakephp.org>
2. Entpacken des heruntergeladenen CakePHP Zipfile (Root-Ordner der CakePHP Installation wird mit *cakephp235* benannt)
3. Diese Ordner-Struktur muss:
 - a) mittels FTP-Client auf den Webserver geladen werden
 - b) in den lokalen Ordner `C:\xampp\htdocs\` kopiert werden
4. Anschliessend kann mittels Browser die Installation aufgerufen werden:
 - a) <http://www.feuerwehr-wiesendangen.ch/cakephp235>
 - b) `localhost/cakephp235`

Es erscheint nun eine Installationsübersicht mit den Angaben, was noch an welcher Stelle angepasst werden muss/soll:

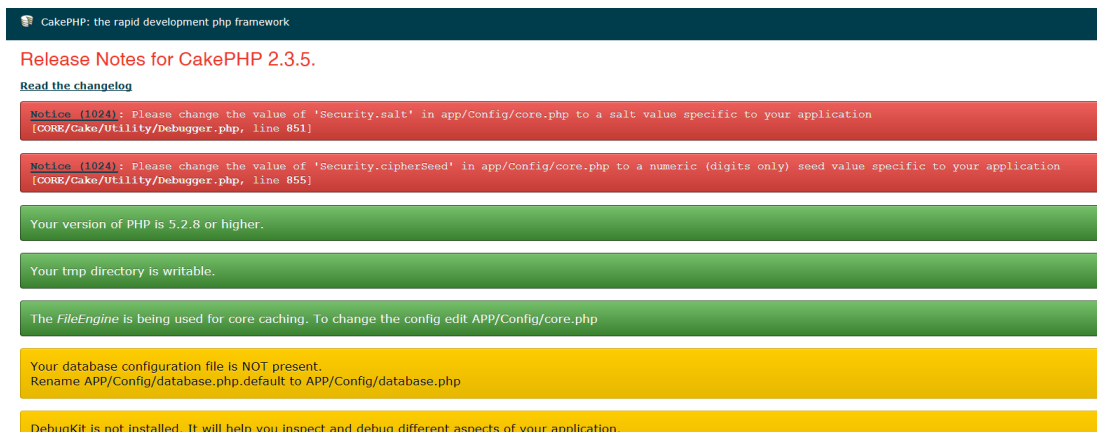


Abbildung 4.2.: CakePHP - Installationsübersicht

5. Sobald man die aufgelisteten Anpassungen ausgeführt hat, kann die Seite nochmals aufgerufen werden. Die Anzeige sollte nun nur noch grüne Einträge anzeigen. Die Installation von CakePHP ist somit abgeschlossen

5. Systemübersicht

Das folgende Kapitel gibt einen Überblick über das Gesamtsystem OffSpickBackend:

5.1. Systemabgrenzung

Das System OffSpickBackend lässt sich wie folgt abgrenzen:

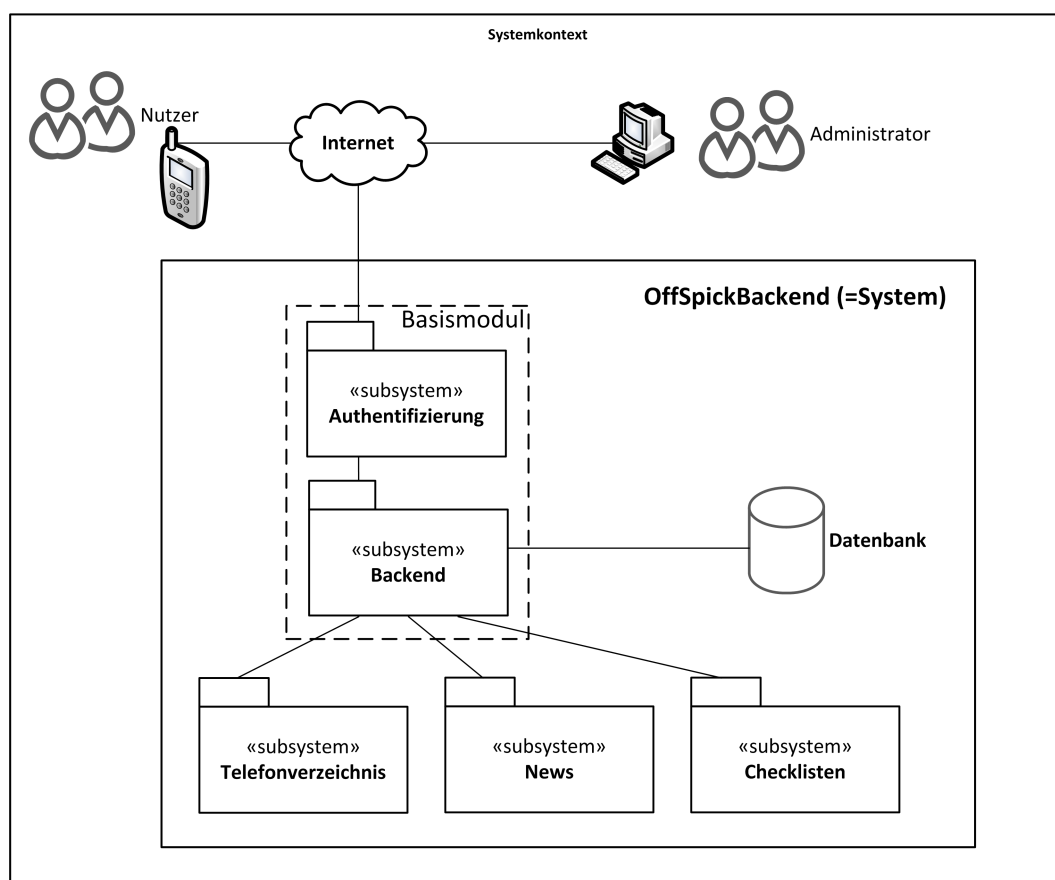


Abbildung 5.1.: Systemabgrenzung

5.2. Stakeholder

Die folgenden Stakeholder sind für dieses Projekt relevant:

Stakeholder	Relevanz	Name	Verfügbarkeit	E-Mail
Nutzer	Hoch	Michael Ott	7 x 24h (Vorzugsweise Abends)	ottmic02@students.zhaw.ch
Administrator	Mittel	Michael Ott	7 x 24h (Vorzugsweise Abends)	ottmic02@students.zhaw.ch
Experte	Hoch	Matthias Bachmann	E-Mail	m.bachmann@united.ch

Der Entwickler ist gleichzeitig Initiator, Administrator und Nutzer (Offiziersrang) dieses Projekts. Auf Grund organisatorischer Vorgaben seitens Feuerwehr wird daher kein weiterer Stakeholder beteiligt.

5.3. Funktionale Anforderungen

Detailldokumentation siehe Kapitel 6

5.4. Nichtfunktionale Anforderungen

ID	Bezeichnung
NFA-001	Der Look der Anwendung

5.4.1. Nutzer

Der Nutzer muss via Handheld-App (nicht Bestandteil dieser Arbeit) die Daten der verschiedenen Module abrufen können (z.B. Telefonbuch). Dies soll nur möglich sein, wenn er beim Aufruf authentifiziert wurde (Zugangsdaten sind in der Handheld-App hinterlegt)

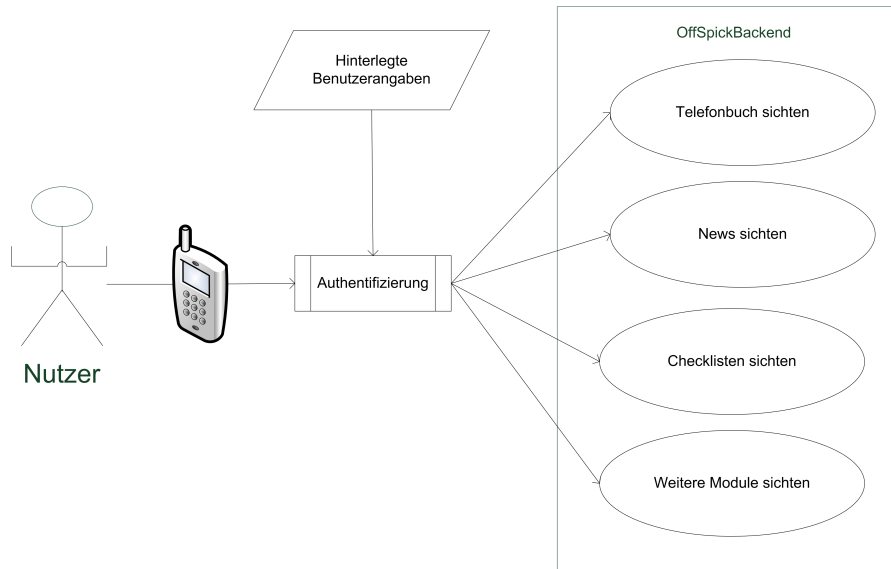


Abbildung 5.2.: UseCase Nutzer

5.4.2. Administrator

Der Administrator muss zusätzlich zu den Möglichkeiten des Nutzers einen Zugriff via Webseite auf das OffSpickBackend haben

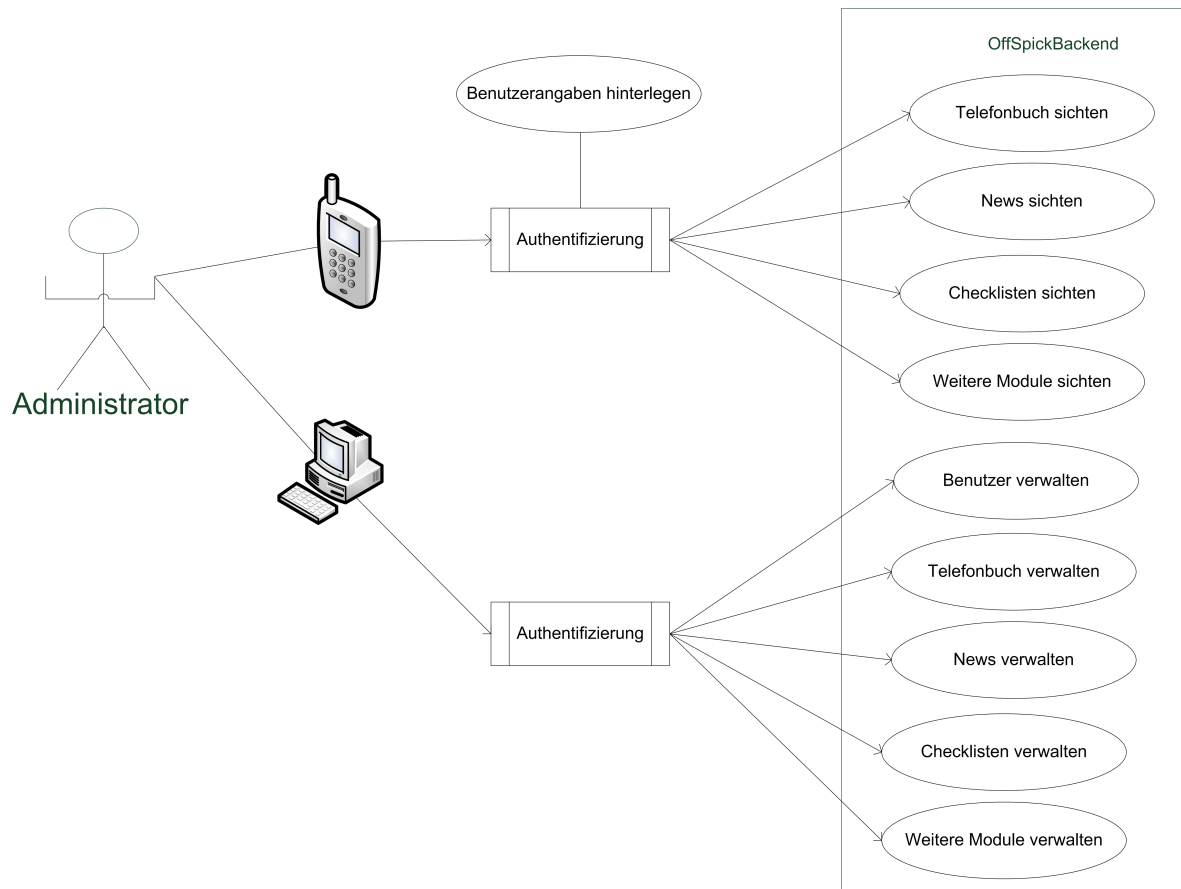


Abbildung 5.3.: UseCase Administrator

6. Anforderungen für Module

Das Backend stellt ein zentraler Bestandteil des OffSpick dar und sollte grundsätzlich in seinem Funktionsumfang modular erweiterbar sein. Die folgenden Module sollen aufzeigen, wie ein solcher Aufbau möglich sein soll.

Da gemäss der Aufgabenstellung das bestehende Backend überarbeitet wird, muss die Datenstruktur der Benutzerverwaltung und das Telefonbuch übernommen werden. Ebenso erfolgt die Authentifizierung der Clients über die Session-Authentifizierung.

Die anderen Module werden komplett neu aufgebaut.

6.1. Basismodul

Die Basis des Backend besteht aus einem Authentifizierungsteil und einem Webservice welcher die Daten gemäss der implementierten Module anbietet. Diese Basis stellt die Kommunikation sicher und ist gleichzeitig die Plattform für die weiteren Module.

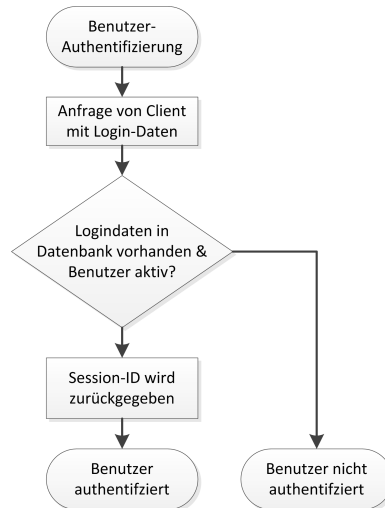


Abbildung 6.1.: Backend-Authentifizierung

6.1.1. Rahmenbedingungen

Anfänglich wurde definiert, dass die ursprüngliche User-Tabelle für die Authentifizierung 1:1 weiterverwendet wird. Nach Studium der CakePHP-Dokumentation habe ich jedoch die Entscheidung gefällt, mich an die Empfehlung von CakePHP zu halten und die Authentifizierung entsprechend neu aufzubauen. Die Tabelle wird aus diesem Grund mit einem SQL-Skript auf die neue Struktur angepasst. Das bedingt, dass der Aufwand der Anpassung des Frontend höher wird. Die Authentifizierung ist in CakePHP ebenfalls bereits über die Session möglich, daher sollte kein kompletter Umbau sondern lediglich eine Anpassung an die neue Verschlüsselung notwendig sein.

User	
ID	INT(6)
Username	VARCHAR(4)
Name	VARCHAR(40)
Password	TEXT
EMail	VARCHAR(50)
ValidFrom	DATETIME
ValidTo	DATETIME
IP	VARCHAR(15)
Indexes	

The image shows a screenshot of a database table structure for a table named 'User'. The fields listed are: ID (INT(6)), Username (VARCHAR(4)), Name (VARCHAR(40)), Password (TEXT), EMail (VARCHAR(50)), ValidFrom (DATETIME), ValidTo (DATETIME), and IP (VARCHAR(15)). There is an 'Indexes' section at the bottom which is currently collapsed.

Abbildung 6.2.: ursprüngliche USER-Tabelle

6.1.2. Anforderungen

ID	Bezeichnung	Bezug	Aufwand (h)
Req-01001	Bestehende User-Tabelle gem. CakePHP-Tutorial anpassen		
Req-01002	Es müssen Benutzer erfasst werden können	Req-01001	
Req-01003	Jeder Benutzer hat einen Gültigkeitsbereich	Req-01001	
Req-01004	Es müssen Benutzer gelöscht werden können	Req-01002	
Req-01005	Jeder Benutzer kann als Backend-User freigeschalten werden	Req-01002	
Req-01006	Der letzte Backend-User kann nicht gelöscht werden	Req-01002 Req-01005	
Req-01007	Das Backend soll mittels REST-Standard erreichbar sein		
Req-01008	Zugriff nur als registrierter User möglich	Req-01007	
Total			15

6.1.3. Testing

Das Testing des Moduls erfolgt als Funktionstest gemäss den obenstehenden Anforderungen. Die Funktionstest werden immer mehrfach (z.B. mit gültigem und ungültigem User bei Req-01008) durchgeführt

6.2. Modul Telefonbuch

Dieses Modul war eigentlich die ausschlaggebende Idee und stellt ein zentral verwaltetes Telefonbuch für die Offiziere dar. Dieses Telefonbuch soll speziell auch Pikett-Pläne mit wechselnden Telefonnummern abdecken (z.B: gilt beim Wasser-Pikett wochenweise eine andere Telefonnummer).

6.2.1. Rahmenbedingungen

Die folgende Datenstruktur ist bereits vorhanden und muss wie bereits unter 4.1 beschrieben auf die CakePHP-spezifischen Bezeichnungen angepasst werden.

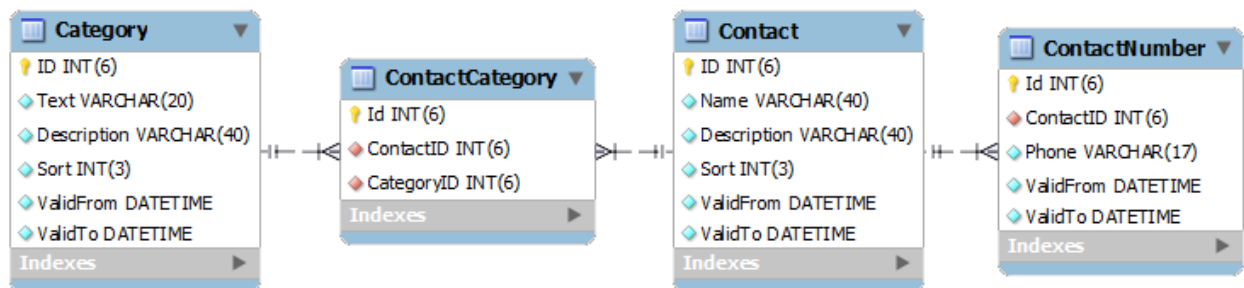


Abbildung 6.3.: Datenbank-Struktur Modul Telefonverzeichnis

Tabelle *CATEGORY* (Kategorie)

Tabelle *CONTACTCATEGORY* (Zuteilung Kontakte an Kategorien)

Tabelle *CONTACT* (Kontakt: Bezeichnung des Kontaktes)

Tabelle *CONTACTNUMBER* (Telefonnummer des Kontaktes)

6.2.2. Anforderungen

ID	Bezeichnung	Bezug	Aufwand (h)
Req-02001	Vorgegebene Datentabellen, siehe Rahmenbedingungen		
Req-02002	Es können Kategorien hinzugefügt werden	Req-02001	
Req-02003	Es können Kontakte erfasst werden (ID, Name, Beschreibung)		
Req-02004	Es können Kontaktenummern erfasst werden (ID, Telefonnummer)		
Req-02005	Einem (1) Kontakt kann eine oder mehrere Kontaktnummer hinzugefügt werden, wenn sich die Gültigkeitsdauer der Kontaktnummern nicht überschneiden		
Req-02006	Jeder Kontakt kann einer oder mehrerer Kategorien zugeteilt werden	Req-02001 Req-02003	
Req-02007	Zuteilung eines Kontaktes an eine Kategorie muss gelöscht werden können	Req-02004	
Req-02008	Kategorie-Einträge können gelöscht werden, wenn keine Kontakte zugeteilt sind	Req-02001	
Req-02009	Jeder Kontakt hat ein Gültigkeitsbereich	Req-02003	
Req-02010	Jeder Kategorie hat ein Gültigkeitsbereich (nicht gültige Kategorien werden komplett inkl. zugeteilten Kontakte ausgeblendet)	Req-02001	
Req-02011	Jede Kategorie hat eine definierbare Anzeigereihenfolge	Req-02001	
Req-02012	Jeder Kontakt hat eine definierbare Anzeigereihenfolge	Req-02003	
Req-02013	Es muss beim Aufruf eine konsolidierte, nicht normalisierte Tabelle erstellt werden		
Req-02014	Es dürfen nur gültige Kategorien / Kontakte in der Ausgabetabelle enthalten sein	Req-02011	
Req-02015	Übergabe erfolgt als JSON-Objekt (nur gültige Kategorien / Kontakte)	Req-02011	
Total			15

6.2.3. Testing

Das Testing des Moduls erfolgt als Funktionstest gemäss den obenstehenden Anforderungen. Die Funktionstest werden immer mehrfach (z.B. mit gültigen und ungültigen Kontakten bei Req-02009) durchgeführt

6.3. Modul News

Der Nutzer soll aktuelle News lesen können

Der Administrator soll einfach neue News-Einträge nach Kategorien getrennt, erfassen können

Beispiel: z.B. Hydrant XY ausser Betrieb, Strasse XY bis am .. gesperrt

Das Modul existiert bis Projektstart noch nicht

6.3.1. Rahmenbedingungen

Das Modul ist noch nicht vorhanden und hat daher keine Tabellen oder Code welcher übernommen werden muss.

6.3.2. Anforderungen

ID	Bezeichnung	Bezug	Aufwand (h)
Req-03001	Es gibt eine Kategorie-Liste		
Req-03002	Es können neue Kategorien erfasst werden	Req-03001	
Req-03003	Es können News-Einträge erfasst werden (Titel, Text)		
Req-03004	Ein News-Eintrag muss einer oder mehrerer Kategorien zugeteilt werden	Req-03001 Req-03003	
Req-03005	News-Einträge können gelöscht werden		
Req-03006	Kategorien können gelöscht werden, wenn sie keinem News-Eintrag zugeteilt sind	Req-03001	
Req-03007	Jede News hat einen Gültigkeitsbereich	Req-03003	
Req-03008	Neuster News-Eintrag ist zuoberst, der älteste zu unterst	Req-03003	
Total			4

6.3.3. Testing

Das Testing des Moduls erfolgt als Funktionstest gemäss den obenstehenden Anforderungen. Die Funktionstest werden immer mehrfach (z.B. mit gültigen und ungültigen News bei Req-03007) durchgeführt

6.4. Modul Checkliste

Der Nutzer soll im Einsatz einfach aus gegebenen Checklisten eine auswählen und diese lesen können.

Der Administrator soll einfach neue Checklisten-Items erfassen und in Checklisten zusammenstellen können.

6.4.1. Rahmenbedingungen

Das Modul ist noch nicht vorhanden und hat daher keine Tabellen oder Code welcher übernommen werden muss.

6.4.2. Anforderungen

ID	Bezeichnung	Bezug	Aufwand (h)
Req-04001	Es können Checklisten erfasst werden (ID, Bezeichnung)		
Req-04002	Es können Checklisten-Einträge erfasst werden (ID, Bezeichnung)		
Req-04003	Jeder Checklisten können Checklisten-Einträge zugeteilt werden	Req-04001 Req-04002	
Req-04004	Verbindungen von Checklisten-Einträge zu Checklisten können gelöscht werden	Req-04003	
Req-04005	Checklisten-Einträge können gelöscht werden, wenn sie keiner Checkliste zugeteilt sind	Req-04002	
Req-04006	Checklisten können gelöscht werden, wenn sie keine Checkliste-Einträge enthalten	Req-04001	
Req-04007	Jede Checkliste hat einen Gültigkeitsbereich (nicht gültige Checklisten werden komplett inkl. zugeteilten Checklisten-Einträge ausgeblendet)	Req-04001	
Req-04008	Jeder Checkliste hat eine definierbare Anzeigereihenfolge	Req-04001	
Req-04009	Die Sortierung von Checklisten-Einträgen innerhalb einer Checklisten kann definiert werden	Req-04002	
Total			optional

6.4.3. Testing

Das Testing des Moduls erfolgt als Funktionstest gemäss den obenstehenden Anforderungen. Die Funktionstest werden immer mehrfach (z.B. mit zugeteilten und nicht zugeteilten Checklisten-Einträgen bei Req-04005) durchgeführt

7. Konzept Backend-Seite

Der Aufbau der Backend-Seite soll einfach und ohne optische Spielereien gestaltet sein, da dieser Zugang lediglich zu administrativen Zwecken genutzt wird. Sie soll den Zugriff von speziell für das Backend freigeschalteten User zur Verfügung stehen

7.0.4. Login

Für den Zugriff auf das Backend ist immer ein Login erforderlich. Die Login-Maske muss lediglich den Benutzername und das Passwort abfragen:

The image is a hand-drawn sketch of a web browser window. The title bar at the top reads 'OffSpick - Backend'. The address bar shows the URL 'http://www.feuerwehr-wiesendangen.ch/offspick/Login'. Below the address bar is a header section with a small logo on the left and the text 'OffSpick-Backend Feuerwehr-Zweckverband Wiesendangen Bertschikon'. The main content area is titled 'Login' and contains two input fields: 'Benutzername' (username) and 'Passwort' (password). Below these fields is a button labeled 'Einloggen' (login). The browser window has standard navigation buttons (back, forward, stop, home) and a search icon in the top right corner.

Abbildung 7.1.: Konzept Backend-Webseite Login

7.0.5. Übersicht

In der Übersicht-Ansicht sind sämtliche erfassten Einträge als Liste in diesem und anderen Modulen.

In der Listenansicht sind die einzelnen Einträge ersichtlich. Jeder Eintrag kann direkt, sofern erlaubt, bearbeitet oder gelöscht werden (Aufruf in der letzten Spalte)

The screenshot shows a web browser window titled "OffSpick - Backend" with the URL "http://www.feuerwehr-wiesendangen.ch/offspick/Users". The page header includes a logo and the text "OffSpick-Backend Feuerwehr-Zweckverband Wiesendangen Bertschikon".

On the left is a sidebar menu with the following items:

- > Benutzerverwaltung
 - Benutzer erfassen
 - Benutzerkategorie erfassen
 - Benutzerkategorie bearbeiten
- > Telefonbuch
 - Eintrag erfassen
 - Kategorie erfassen
 - Kategorie löschen
- > News
 - News erfassen
 - News-Kategorien anzeigen
 - News-Kategorie erfassen
- > Checklisten
 - Checkliste erfassen
 - Checklisten-Item anzeigen
 - Checklisten-Item erfassen
- Ausloggen

The main content area is titled "Benutzerverwaltung" and displays a table of users:

ID	Username▲	Name▲	Gültig von↕	Gültig bis↕	Benutzergruppe▼	Admin-Benutzer▲	Aktionen
1	MiOt	Michael Ott	01.01.2000	01.01.2100	Offizier	<input checked="" type="checkbox"/>	Details Bearbeiten Löschen
2	MaMu	Max Muster	01.01.2000	01.01.2100	Offizier	<input checked="" type="checkbox"/>	Details Bearbeiten Löschen
3	HaMu	Hans Muster	01.01.2000	01.01.2100	Offizier	<input checked="" type="checkbox"/>	Details Bearbeiten Löschen
4	GuBr	Guido Brenzlig	01.01.2000	01.01.2100	Offizier	<input type="checkbox"/>	Details Bearbeiten Löschen
5	PeRa	Peter Rauch	01.01.2000	01.01.2100	Kader	<input type="checkbox"/>	Details Bearbeiten Löschen
6	AnHe	Andrea Heiss	01.01.2013	01.01.2100	Mannschaft	<input type="checkbox"/>	Details Bearbeiten Löschen

Below the table is a pagination control showing "Vorherige Seite", "6 / 6", and "Nächste Seite".

Abbildung 7.2.: Konzept Backend-Webseite Übersicht

7.0.6. Datensatz erfassen

Auf der Erfassungsmaske sind Felder gemäss dem Modul aufgeführt. Beim Speichern werden entsprechend die definierten Eingabeprüfungen gemacht und direkt angezeigt

The screenshot displays a web browser window titled "OffSpick - Backend" with the URL <http://www.feuerwehr-wiesendangen.ch/offspick/Users/Add>. The page header includes a logo and the text "OffSpick-Backend Feuerwehr-Zweckverband Wiesendangen Bertschikon".

The main content area is divided into a left sidebar and a right main panel. The sidebar contains a menu with the following items:

- > Benutzerverwaltung
 - Benutzer erfassen** (highlighted)
 - Benutzerkategorie erfassen
 - Benutzerkategorie bearbeiten
- > Telefonbuch
 - Eintrag erfassen
 - Kategorierfassen
 - Kategorie löschen
- > News
 - News erfassen
 - News-Kategorien anzeigen
 - News-Kategorie erfassen
- > Checklisten
 - Checkliste erfassen
 - Checklisten-Item anzeigen
 - Checklisten-Item erfassen
- Ausloggen

The right main panel is titled "Benutzer erfassen" and contains the following form fields:

- Benutzername:
- Name:
- E-Mail:
- Passwort:
- Gruppe: (dropdown menu)
- Backend-Benutzer: ☐
- Gültig ab: (calendar icon)
- Gültig bis: (calendar icon)
- Speichern:

Abbildung 7.3.: Konzept Backend-Webseite Erfassung

8. Umsetzung Backend

Die folgenden Schritte wurden jeweils pro Modul durchgeführt:

8.1. Datenbank

Als erster Schritt muss die Datenbank entsprechend den CakePHP-Regeln angepasst und erweitert werden. Zu diesem Zweck wurde ein entsprechendes SQL-Skript erstellt, welches mit phpMySQL-Admin auf der bestehenden Datenbank laufen gelassen werden kann

8.2. MVC-Scaffold

Um den Programmier-Aufwand gering zu halten, wurden mittels dem Scaffolding-Tool *cake bake* ein MVC-Grundgerüst für die ganze Applikation erstellt:

8.2.1. Model erstellen

Es wurde für jede vorhanden Tabelle mittels *cake bake - Model* ein passendes Model erstellt. Zusätzlich wurden bei der Erstellung der Models noch Eingabeprüfungen definiert, so das bei der Erfassung oder Anpassung von Datensätzen auf allen Eingabefelder gültige Werte vorhanden sind.

8.2.2. Controller erstellen

Die Controller wurden ohne speziellen Anpassungen via *cake bake - Controller* erstellt

8.2.3. View erstellen

Basierend auf den erstellten Models und Controller wurde ebenfalls mittels *cake bake - View* erstellt.

8.3. Manuelle Anpassung

8.3.1. Anpassung Aufbau

Auf Basis des unter 8.2 erstellten Grundgerüstets wurden dann die Views entsprechend dem Konzept aus Kapitel 7 angepasst

8.3.2. Anpassung Optik

Es wurden nur minimale Anpassungen soweit nötig an der Optik vorgenommen. Die restlichen Anpassungen gehören in die Kategorie "Nichtfunktionale Anpassungen" und werden nach Abgabe des Projekts in Angriff genommen.

9. Umsetzung Modul Telefonbuch

In einer ersten Phase wurde das MVC-Grundgerüst für jede der 4 Tabellen erstellt. Somit lassen sich sämtliche Anpassungen in diesen 4 Tabellen durchführen, jedoch ist die Ansicht so nicht bedienerfreundlich.

9.1. Neuaufbau

Basierend auf diesem Grundgerüst wird nun manuell eine konsolidierte Ansicht erstellt, so dass innerhalb einer Bildschirmseite der Kontakt mit Telefonnummer, Kategorie und Gültigkeitsbereich ersichtlich ist.

10. Umsetzung Modul Info

In einer ersten Phase wurde das MVC-Grundgerüst für beide Tabellen erstellt.

11. Integration App

Die Analyse des OffSpick-Frontend ergab, dass die Anpassungen doch weitaus grösser sind, als ursprünglich bei der Aufgabenstellung angenommen. Dies aus verschiedenen Gründen:

- Authentifizierung
Im Prinzip funktioniert die Authentifizierung noch immer gleich (via Session). Die Verschlüsselung des Passworts muss aber clientseitig auf jeden Fall angepasst werden. Dazu kommt, dass CakePHP noch weitere, sicherere Authentifizierungen anbietet, welche clientseitig zusätzlich eine Anpassung zur Folge haben.
- Neue Module
Die bisherige Applikation war primär auf das Telefonbuch ausgerichtet. Es muss hier auch in Betracht der weiteren geplanten (und noch folgenden) Module geprüft werden, mit welcher Art von Menüführung die Daten am übersichtlichsten in der Client-App dargestellt werden können.

Aus diesem Grund wird der Fokus auf das Backend gelegt und nach diesem Projekt ein Neuaufbau des Clients in Angriff genommen

12. Projekt-Tools

Für dieses Projekt wurden folgende Tools eingesetzt:

Produkt/Dienst	Beschreibung	Link
CakePHP	PHP Framework	www.cakephp.org
Dropbox	Dateiablage für sonstige Projekt-Dokumente	Auf Anfrage (relevante Daten siehe Github)
Github	Quellcode-Verwaltung	github.com/M0tt/OffSpickBackend
LaTeX	Dokumentation	www.texworks.org
mySQL V5.0.77	Datenbank	www.mysql.com
XAMPP	Apache, MySQL, PHP Distribution	www.apachefriends.org/de/xampp.html

13. Fazit / Ideen für Erweiterungen

CakePHP bietet als PHP-Framework extrem viel, fordert einem aber auch bei der Einarbeitung. So würden die zusätzlichen Funktionen (automatisiertes Testing, Templates, diversen Handler wie z.B. eMail, Tools um die Applikation auf Sicherheitslöcher zu prüfen, Migration-Tools,...) noch viel mehr bieten. Kurzum gesagt: Das Kapitel 3 könnte deutlich länger sein und würde immer noch nicht den vollen Umfang umschreiben.

Dieses Seminararbeit hat mir jedoch auch gezeigt, dass fehlende Programmier-Erfahrung den Aufwand extrem erhöhen. So mussten gewisse Codefragmente in mehreren Anläufen erstellt werden, bis der Code am Schluss brauchbar war. Dieser Aufwand dürfte bei entsprechender Routine deutlich verkleinert werden.

Das Framework macht aber auf jeden Fall Spass und so werde ich es mit Sicherheit nicht das letzte Mal verwendet haben. Auch wenn ich die gesetzten Ziele nicht komplett erreicht habe, so werde ich diese Umsetzung bis zum Schluss verfolgen um meinen Feuerwehr-Kameraden ein neues Einsatzmittel mitgeben kann.

A. Abbildungsverzeichnis

3.1. Projektplan - Tabelle	9
3.2. Projektplan - Grafik	10
4.1. CakePHP - Scaffolding mit cake bake	12
4.2. CakePHP - Installationsübersicht	13
5.1. Systemabgrenzung	14
5.2. UseCase Nutzer	16
5.3. UseCase Administrator	17
6.1. Backend Authentifizierung	19
6.2. Backend Authentifizierung - ursprüngliche USER-Tabelle	19
6.3. Datenbank-Struktur Modul Telefonverzeichnis	21
7.1. Konzept Backend-Webseite Login	25
7.2. Konzept Backend-Webseite Übersicht	26
7.3. Konzept Backend-Webseite Erfassung	27

(Quelle Grafiken: eigene Darstellung)

B. Literaturverzeichnis

- [1] <http://book.cakephp.org/2.0/en/getting-started/cakephp-conventions.html>. *CakePHP Konventionen*. cakephp.org, Stand: 03.06.2013.
- [2] <http://book.cakephp.org/2.0/en/installation.html>. *CakePHP Installationsanleitung*. cakephp.org, Stand: 03.06.2013.
- [3] <http://book.cakephp.org/2.0/en/tutorials-and-examples.html>. *CakePHP Tutorial*. cakephp.org, Stand: 03.06.2013.
- [4] <http://www.apachefriends.org/de/xampp.html>. *XAMPP*. Apachefriends.org, Stand: 03.06.2013.

C. Quellcode

Auf das Anfügen von Quellcode wurde aus Platzgründen verzichtet. Den kompletten Quellcode findet man im GIT-Repository (gemäss Angaben unter Kapitel 12)