

Proj ektarbei t 2007/08

Web Frontend zu ALFSA



ausgeführt an der
Abteilung für Elektronik der
Höheren Technischen Bundeslehr- u. Versuchsanstalt St. Pölten
Waldstraße 3, A-3100 St. Pölten

im Schuljahr
2007/08

von
Andreas Brandstätter, 5AHELI-02
Christoph Klaffl, 5AHELI-08

unter Betreuung von
Dipl.-Ing. Wolfgang Alfery

St. Pölten, am 2008-02-17



Teil A

Projektmanagement

1 Inhaltsverzeichnis

1	Inhaltsverzeichnis.....	2
2	Allgemeines.....	3
2.1	Veraltete Systeme vs. Neue Lösungen.....	3
2.1.1	Datenaustausch	3
2.1.2	Statistische Auswertung	3
2.1.3	Datensicherung	3
3	Auftrag.....	4
3.1	Funktionen	4
3.2	Kontaktadressen	5
3.2.1	Auftraggeber	5
3.2.2	Auftragnehmer	5
3.2.3	Adresse für Anfragen	5
4	Projektziele	6
5	System der Feuerwehr Tulln	7
5.1	Reverse Engineering	7
6	Projektentwurf	8
7	Zeitablauf	9
7.1	Zeitplan	9
7.2	Arbeitskalender	10
7.2.1	Brandstätter Andreas	10
7.2.2	Klafl Christoph	13
8	Verzeichnisse	14
8.1	Abbildungsverzeichnis	14
8.2	Tabellenverzeichnis	14
8.3	Codeverzeichnis.....	14
9	Appendix.....	15
9.1	GPL 3.0	15

2 Allgemeines

Im Feuerwehrowesen werden Atemschutzgeräte als wesentlicher Teil der Brandbekämpfung verwendet. Auch bei sonstigen Einsätzen werden Atemschutzgeräte eingesetzt. Die Befüllung der Atemschutzflaschen mit Atemluft sowie die Wartung der Geräte sind unmittelbar für die Sicherheit der Atemschutzgeräteträger maßgeblich. Somit ist es notwendig diese Vorgänge in geeigneter Weise zu Dokumentieren.

Die Qualitätssicherung stellt somit im Atemschutzbereich der Feuerwehren eine lebenswichtige Notwendigkeit dar.

2.1 Veraltete Systeme vs. Neue Lösungen

Diese Qualitätssicherung erfolgt zurzeit in Form von verschiedenen Formularen und Tabellen in Papierform. Dieses System war lange Zeit bewährt, aber kann nicht die Vorteile Computergestützter Datenverarbeitung bieten.



Abbildung 1: Qualitätssicherung in Papierform

2.1.1 Datenaustausch

Einätze übergreifen oftmals mehrere Feuerwehren, und somit ist es notwendig die Daten der Atemschutz-Qualitätssicherung untereinander auszutauschen. Dies ist bei der Qualitätssicherung in Papierform nur mit erheblichem Aufwand durchführbar. Dies soll durch Einsatz einer Computergestützten Qualitätssicherung wesentlich erleichtert werden oder in weiten Teilen überhaupt erst ermöglicht werden.

2.1.2 Statistische Auswertung

Die Erfassung von Statistiken stellt ebenfalls bei Verwendung von Qualitätssicherung in Papierform einen nicht unwesentlichen Aufwand dar. Daten müssen bei jeder Feuerwehübergreifenden Auswertung erst gesammelt und per Hand verarbeitet werden. Im Gegensatz dazu kann bei einer Computergestützten Lösung eine statistische Auswertung viel einfacher erstellt werden. Die Daten sind immer aktuell verfügbar und Statistiken können jeder sofort generiert werden.

2.1.3 Datensicherung

Die Aufzeichnungen der Atemschutz-Qualitätssicherung müssen natürlich in geeigneter Weise gesichert und Archiviert werden. Die Aufzeichnungen werden derzeit in diversen Ordnern und Mappen abgelegt. Bei Verlust dieser Schriften sind nur teilweise Kopien vorhanden. Bei Verwendung der von Computergestützten Datenverarbeitung ist es sehr leicht periodische Sicherungen und Archive anzulegen. Außerdem kann durch einfache Maßnahmen sehr hohe Datensicherheit gegen Verlust gewährleistet werden.

3 Auftrag

Im Rahmen der Diplomarbeit „Atemluftfüllstellenapplikation (ALFSA)“ wird eine Computergestützte Lösung für die Atemschutz-Qualitätssicherung auf Basis eines verteilten Datenbank-Serversystems entworfen.

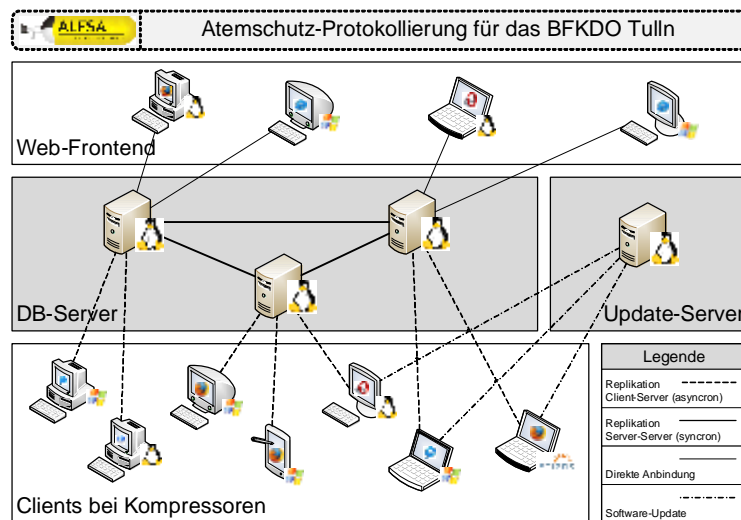


Abbildung 2: Übersicht über ALFSA

Dieses Qualitätssicherungs-System sieht die Verwendung von Clients bei jedem Atemluftkompressor vor. Diese Clients werden von den Füllberechtigten der jeweiligen Kompressoren bedient.

Aufgabe dieses Projektes ist es, eine Software für die Dateneingabe und Auswertung auf den Clients zu entwickeln.

Die Software soll auch für wenig Computererfahrene Anwender einfach zu Bedienen sein. Die Eingabe der Daten soll teilweise mit Barcodescanner und Touchscreen, oder mit Tastatur und Maus erfolgen. Außerdem soll die Software Betriebssystem-Unabhängig realisiert werden.

3.1 Funktionen

Die Client-Software soll folgende Funktionen bieten:

- § Benutzerspezifische Anmeldung der Benutzer
- § Einstellen der Grundkonfiguration
- § Software-Update initialisieren
- § Daten-Synchronisation starten
- § Anlegen von Einsätzen
- § Erfassen von Flaschenfüllungen
- § Anzeige von Flaschendaten
- § Anzeige von Füllungsdaten
- § Teilweise Bearbeitung von Flaschendaten

3.2 Kontaktadressen

3.2.1 Auftraggeber

Bezirksfeuerwehrkommando Tulln
Ansprechperson:
Johannes Ofner
+43 (664) 5334541
johannes.ofner@feuerwehr.tulln.at

3.2.2 Auftragnehmer

Andreas Brandstätter
Gerersdorferstraße 17
3443 Sieghartskirchen
+43 (664) 9246242
brandstaetter@feuerwehrjugend-noe.at

Christoph Klaffl
Birkenweg 5
3550 Langenlois
+43 (664) 4332791
christophklaffl@gmail.com

3.2.3 Adresse für Anfragen

Das Projektteam von aaron ist für sämtliche Anfragen unter der Email-Adresse

aaron@bfkdo-tulln.at

erreichbar.

4 Projektziele

Die Ziele des Projektes stellen die speziellen Anforderungen an das Projekt dar. Primärziele müssen größtenteils erreicht werden. Sekundärziele sollten erreicht werden. Tertiär- und Quartierziele stellen lediglich Möglichkeiten und Perspektiven dar.

Ebenso werden Anti-Ziele definiert. Diese stellen Dinge dar, die nicht realisiert werden sollen. Einerseits weil sie vom Auftraggeber nicht gewünscht sind, oder weil sie vom Projektteam als nicht geeignet erachtet wurden.

Typ, Grad	Beschreibung
Primärziel	einfache und intuitive Oberfläche
Primärziel	Betriebssystemunabhängigkeit
Primärziel	einfache Dateneingabe per Barcode
Primärziel	differenzierte Benutzerberechtigungen
Sekundärziel	Möglichkeit der Installation ohne VMware
Tertiärziel	innovatives Oberflächendesign
Tertiärziel	absolut reine Touchscreenbedienung
Tertiärziel	Herunterfahren aus PHP-Oberfläche
Tertiärziel	Erweiterungsmöglichkeit mit Benutzerscripten
Quartierziel	Live-Version der Client-Oberfläche für USB-Sticks
Quartierziel	lernfähige Anordnung der Bedienungselemente
Anti-Ziel	nervige Animationen
Anti-Ziel	Multilinguale Inhalte

Tabelle 1: Projektziele

5 System der Feuerwehr Tulln

Die Feuerwehr Tulln setzt seit dem Jahr 2006 ein Computergestütztes- Qualitätssicherungssystem ein. Dieses System wurde von Johannes Ofner der Feuerwehr Tulln entwickelt. Dieses System wird derzeit als Einzelstationssystem eingesetzt, obwohl es (partiell) die Möglichkeit bieten würde, mehrere Kompressoren zu verwalten.



Abbildung 3: System der Feuerwehr Tulln¹

Das System der Feuerwehr Tulln wurde in PHP programmiert. Es verwendet eine MySQL-Datenbank zu Datenhaltung. Zur Dateneingabe werden ein Barcodescanner, ein Touchscreen-Bildschirm sowie Maus und Tastatur verwendet.

Die Oberfläche ist für Internet Explorer bei einer Auflösung von 1024x768 optimiert.

5.1 Reverse Engineering

Das System der Feuerwehr Tulln wurde hinsichtlich seiner Strukturen sowie Eingabemöglichkeiten analysiert.

¹ Quelle: <http://alfsa.feuerwehr.tulln.at/>; Urheber: Johannes Ofner

6 Projektentwurf

Die Realisierung des Projektes erfolgt nach folgendem Strukturplan.

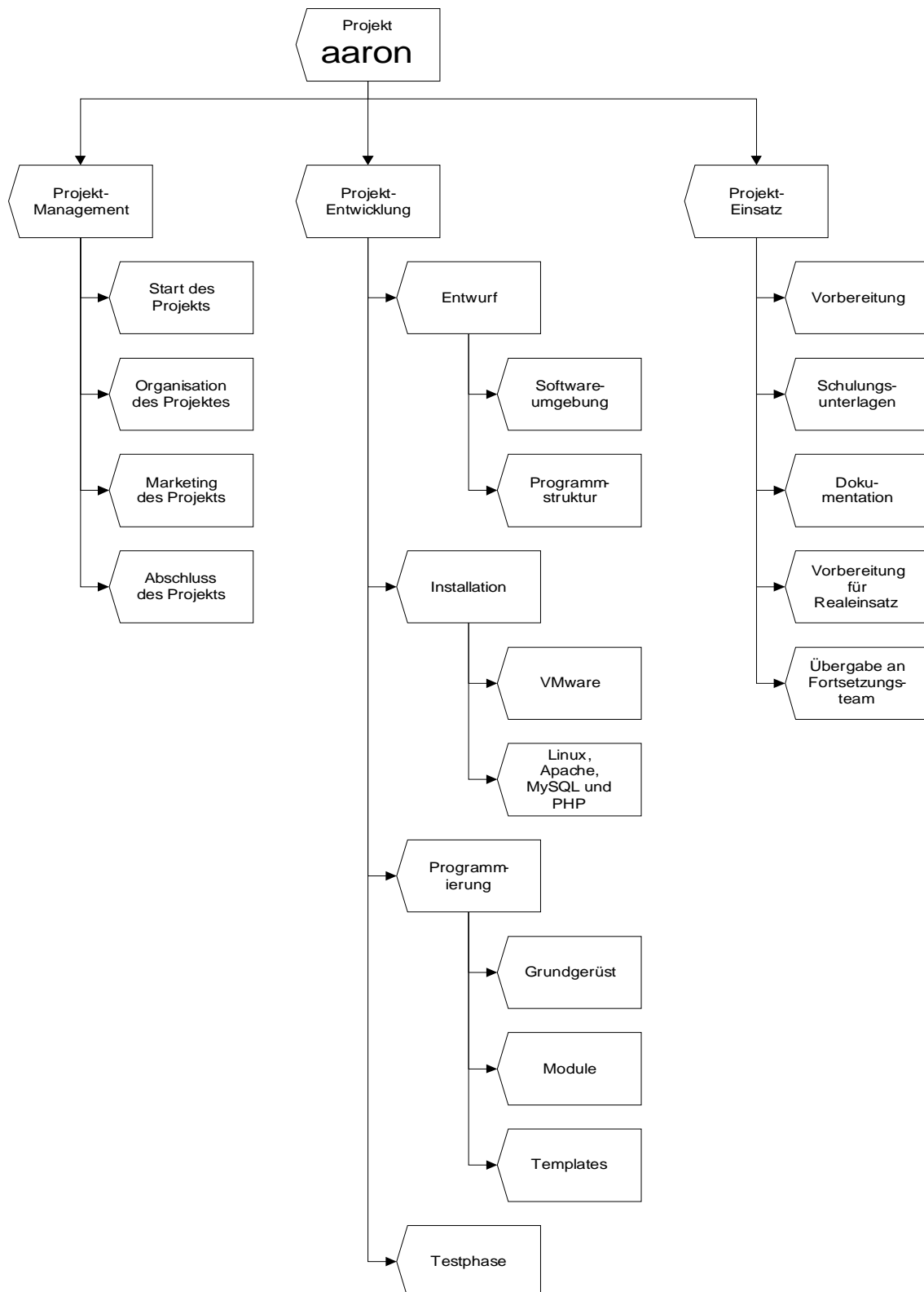


Abbildung 4: Strukturplan

7 Zeitablauf

7.1 Zeitplan

Oktober	Analyse der Projektaufgabe, Aufnahme des Ist-Zustandes, Einarbeitung in das Thema, Ausarbeitung einer Spezifikation,
November	Erörterung verschiedener Lösungs-Varianten, Detailplanung, Entwurf des Designs,
Dezember	Programmentwurf, Programmnavigation entwerfen, Erstellung der Skripts und Librarys,
Jänner	Erstellung der Skripts und Librarys, Erstellung der GUI-Templates, Testphase,
Februar	Erstellung der Skripts und Librarys, Testphase, Programmanpassungen,
März	Gesamt-Testphase, Programmkorrekturen,
April	Abschlussarbeiten, abschließende Ausführung der Dokumentation,

Tabelle 2: Zeitplan

7.2 Arbeitskalender

7.2.1 Brandstätter Andreas

von	bis	dauer	taetigkeitstyp	taetigkeit
2007-09-20 07:50:00	2007-09-20 09:30:00	1 h 40 min	Programmiert	Einarbeitung in das Thema
2007-10-10 07:50:00	2007-10-10 16:40:00	8 h 50 min	Programmiert	Neues Design Grundeinstellungsseite Informationszeile
2007-10-11 07:50:00	2007-10-11 09:30:00	1 h 40 min	Dokumentiert	Zeitplan (Projekt) Spezifikations-Entwurf (Projekt)
2007-10-13 12:00:00	2007-10-13 18:30:00	6 h 30 min	Programmiert	Bildschirmtastatur Synchronisation Fehlerbehebungen
2007-10-14 12:00:00	2007-10-14 18:30:00	6 h 30 min	Programmiert	Fehlerausgaben nach Typ differenziert und grafisch gestaltet Grundlegende Systemänderungen Synchronisation (Verbindungspasswort)
2007-10-16 08:30:00	2007-10-16 09:30:00	1 h 0 min	Programmiert	About-Seite
2007-10-18 07:50:00	2007-10-18 09:30:00	1 h 40 min	Programmiert	Stukturplan (Projekt) Spezifikation (Projekt)
2007-10-24 07:50:00	2007-10-24 16:40:00	8 h 50 min	Programmiert	Nachrichtensystem: Nachrichtenübersicht, Nachrichten lesen, Nachrichten als gelesen markieren (Templates, Pages) Fehlerausgaben für MySQL-Errors
2007-10-27 17:00:00	2007-10-27 19:30:00	2 h 30 min	Programmiert	Oberfläche für das Eintragen von Nachrichten
2007-10-27 20:30:00	2007-10-27 22:30:00	2 h 0 min	Programmiert	Anzeige von Nachrichten
2007-10-28 00:00:00	2007-10-28 01:00:00	1 h 0 min	Programmiert	Zeitanzeige in der Infozeile
2007-10-30 16:40:00	2007-10-30 17:20:00	0 h 40 min	Dokumentiert	Template-Syntax
2007-10-31 07:50:00	2007-10-31 16:40:00	8 h 50 min	Programmiert	Einsatzerfassungssystem mit Füll-Sessions (erstellen, unterbrechen, aufnehmen, abschließen von Einsätzen)
2007-10-31 19:30:00	2007-10-31 21:00:00	1 h 30 min	Programmiert	Einsatz erneut aufnehmen und wieder beenden Einsatz-Schnellstart
2007-10-31 23:00:00	2007-10-31 23:50:00	0 h 50 min	Programmiert	Einsatz-Ansicht
2007-11-03 00:00:00	2007-11-03 01:00:00	1 h 0 min	Dokumentiert	Beschreibung der Schritte für Installation der Client-Software
2007-11-03 19:00:00	2007-11-03 20:30:00	1 h 30 min	Programmiert	Proxy-Einstellung modifiziert Software-Update aus Oberfläche starten
2007-11-04 00:00:00	2007-11-04 02:00:00	2 h 0 min	Programmiert	Bildschirm-Kalender Bearbeiten von Einsätzen Haltbarkeit von Nachrichten
2007-11-04 14:00:00	2007-11-04 17:00:00	3 h 0 min	Programmiert	Barcode-Eingabefeld Crystals Barcode-Einloggen
2007-11-06 22:30:00	2007-11-06 23:00:00	0 h 30 min	Programmiert	verschiedene Buttons modifiziert
2007-11-07 07:50:00	2007-11-07 14:40:00	6 h 50 min	Programmiert	Barcode-Eingabefeld modifiziert Crystals erweitert Flaschenfuellugen (mit Barcode oder manuell) Client-Server Synchronisation (kleine Änderungen)
2007-11-09 08:00:00	2007-11-09 09:30:00	1 h 30 min	Programmiert	Charset und weitere Header-Informationen eingefügt
2007-11-11 12:30:00	2007-11-11 17:00:00	4 h 30 min	Programmiert	Mysql-Injektion verhinderung Füllvorgangs-Anzeige Scrollbuttons für Tochtscreen
2007-11-13	2007-11-13	1 h 30 min	Programmiert	Auswahl der Einsätze bzw. Nachrichten per Buttons bzw. Barcode

08:00:00	09:30:00			
2007-11-14 07:50:00	2007-11-14 10:00:00	2 h 10 min	Programmiert	Datenbanksynchronisation Client-Server modifiziert (Datenpakete) diverse Oberflächenänderungen
2007-11-28 07:50:00	2007-11-28 16:40:00	8 h 50 min	Programmiert	Umstellung der Funktionen auf Klassen Neue Login-Oberfläche mit geänderter Benutzerauswahl
2007-12-05 07:50:00	2007-12-05 16:40:00	8 h 50 min	Programmiert	Umstellung der Funktionen auf Klassen Implementierung von mehreren Kompressoren pro Füllstelle Funktionalität von Flaschenfüllungen und Einsätzen Flaschenansicht (Übersichtsliste)
2007-12-06 11:20:00	2007-12-06 12:10:00	0 h 50 min	Programmiert	kleine Änderungen der Flaschendarstellung Test der Softwareänderungen am Euklid-Server (Update, Synchronisation, usw.)
2007-12-12 07:50:00	2007-12-12 10:20:00	2 h 30 min	Programmiert	Berechtigungen für Client implementiert neue Fehlermeldung für Berechtigungsprobleme
2007-12-12 10:20:00	2007-12-12 16:40:00	6 h 20 min	Programmiert	Anzeige von Flaschen Anzeige von Mängel
2007-12-13 07:50:00	2007-12-13 09:30:00	1 h 40 min	Programmiert	Anzeige von Mängel
2007-12-13 11:20:00	2007-12-13 12:20:00	1 h 0 min	Programmiert	Anzeige von Mängel
2007-12-16 00:00:00	2007-12-16 02:00:00	2 h 0 min	Programmiert	Anzeige von Mängeln Erfassen von Mängeln Beheben von Mängeln
2007-12-18 15:00:00	2007-12-18 16:30:00	1 h 30 min	Programmiert	Buttons mit Icons
2007-12-19 07:50:00	2007-12-19 16:40:00	8 h 50 min	Programmiert	Icons auf Buttons Mängel erfassen (Mängeltypen) Mängel bei Füllungen kontrollieren Flaschen anzeigen & bearbeiten flying windows!
2007-12-19 19:30:00	2007-12-19 22:00:00	2 h 30 min	Programmiert	flying windows (Error-Meldungen)
2007-12-20 07:50:00	2007-12-20 08:40:00	0 h 50 min	Programmiert	Symbole für flying windows
2007-12-20 11:10:00	2007-12-20 12:10:00	1 h 0 min	Programmiert	Farbschema für flying windows neue Kategorie der flying windows: Hilfe
2007-12-20 12:10:00	2007-12-20 12:40:00	0 h 30 min	Programmiert	Darstellung von Hilfe-Fenstern
2007-12-20 13:30:00	2007-12-20 14:00:00	0 h 30 min	Programmiert	Positionierung von Hilfe-Fenstern
2007-12-28 15:30:00	2007-12-28 18:30:00	3 h 0 min	Programmiert	Flaschen bearbeiten Flaschen einfügen div. Änderungen
2007-12-28 18:30:00	2007-12-28 20:00:00	1 h 30 min	Programmiert	Barcode-Scanner konfiguriert für i25
2007-12-29 14:30:00	2007-12-29 15:30:00	1 h 0 min	Programmiert	Feuerwehren anzeigen
2007-12-29 15:30:00	2007-12-29 17:30:00	2 h 0 min	Programmiert	Betriebsstundenzaehler diverses
2007-12-29 17:30:00	2007-12-29 20:00:00	2 h 30 min	Programmiert	Fremdflaschen (eintragen, füllen)
2008-01-01 23:30:00	2008-01-02 01:30:00	2 h 0 min	Programmiert	extended logon form
2008-01-02 16:30:00	2008-01-02 17:30:00	1 h 0 min	Programmiert	Design der Füllansicht überarbeitet
2008-01-02 18:45:00	2008-01-02 19:30:00	0 h 45 min	Dokumentiert	Konzept des neuen Nachrichtensystems entworfen
2008-01-02 20:00:00	2008-01-02 21:00:00	1 h 0 min	Programmiert	neues Nachrichtensystem (anzeigen von Nachrichten)
2008-01-02 22:15:00	2008-01-03 01:00:00	2 h 45 min	Programmiert	neues Nachrichtensystem (eintragen von Nachrichten)

2008-01-03 01:15:00	2008-01-03 02:45:00	1 h 30 min	Programmiert	Benutzer anzeigen (Übersichtsliste und Detailansicht)
2008-01-07 22:00:00	2008-01-08 00:00:00	2 h 0 min	Dokumentiert	Projektdokumentation - Struktur und Vorbereitungen
2008-01-08 07:50:00	2008-01-08 09:30:00	1 h 40 min	Dokumentiert	Projektdokumentation - Allgemeines, Auftrag, Einsatzbedingungen
2008-01-08 11:30:00	2008-01-08 12:30:00	1 h 0 min	Dokumentiert	Projektdokumentation - Auflösung
2008-01-08 12:40:00	2008-01-08 13:10:00	0 h 30 min	Dokumentiert	Projektdokumentation - Auflösung
2008-01-08 19:00:00	2008-01-08 20:00:00	1 h 0 min	Dokumentiert	Projektdokumentation - Icons, Berechtigungen
2008-01-08 22:00:00	2008-01-09 00:00:00	2 h 0 min	Dokumentiert	Projektdokumentation - Template-Engine, Syntaxbeispiele, diverses
2008-01-09 07:50:00	2008-01-09 11:00:00	3 h 10 min	Programmiert	einfache Einstellung der ersten Grundkonfiguration

Tabelle 1: Arbeitskalender von Andreas Brandstätter, stand 2008-01-16

7.2.2 Klaffl Christoph

Datum	von	bis	Dauer	Aktion
Mittwoch, 12. September 2007	07:50	16:40	08:50	Auswahl des Projektes, Zuteilung des Betreuers, Besprechung der Aufgabenstellung und der Durchführung
Donnerstag, 13. September 2007	07:50	09:30	01:40	Besprechung der Aufgabenstellung
Mittwoch, 19. September 2007	07:50	16:40	08:50	Besprechung des genauen Projektumfangs mit dem Betreuer und des Projekt-Auftragsgebers
Donnerstag, 20. September 2007	07:50	09:30	01:40	Planung der Projektaktivitäten
Mittwoch, 26. September 2007	07:50	16:40	08:50	Abwägung der verschiedenen Programmiersprachen Java Servlets, PHP, ASP
Donnerstag, 27. September 2007	07:50	09:30	01:40	Abwägung der verschiedenen Programmiersprachen Java Servlets, PHP, ASP
Mittwoch, 03. Oktober 2007	07:50	16:40	08:50	Erste Entwürfe der Benutzeroberfläche und Erstellen der Projektspezifikation
Donnerstag, 04. Oktober 2007	07:50	09:30	01:40	Erste Entwürfe der Benutzeroberfläche und Erstellen der Projektspezifikation
Mittwoch, 10. Oktober 2007	07:50	16:40	08:50	Arbeiten an der Projektspezifikation, Benutzerverwaltung (Login Screen)
Donnerstag, 11. Oktober 2007	07:50	09:30	01:40	Benutzerverwaltung (Login Screen)
Mittwoch, 17. Oktober 2007	07:50	16:40	08:50	Umschreibung der Benutzerverwaltung (Login Screen), wegen der DB Struktur Änderung von ALFSA
Donnerstag, 18. Oktober 2007	07:50	09:30	01:40	Analysierung des Kodierungsproblem
Mittwoch, 24. Oktober 2007	07:50	16:40	08:50	Schreiben und testen der Library für die Nachrichten
Freitag, 26. Oktober 2007	22:00	00:00	02:00	Optimierung der Libraries
Mittwoch, 31. Oktober 2007	07:50	16:40	08:50	Erweitern und Anpassung der filling Library an die neue Datenbankstruktur
Mittwoch, 07. November 2007	07:50	16:40	08:50	Fehlerbereinigung
Mittwoch, 14. November 2007	07:50	16:40	08:50	Strukturänderungen
Mittwoch, 21. November 2007	07:50	16:40	08:50	Vorbereitungen für den Tag der offenen Tür
Mittwoch, 28. November 2007	07:50	16:40	08:50	Änderungen vom Auftraggeber wurden übernommen
Donnerstag, 29. November 2007	07:50	09:30	01:40	Fehlerbereinigung
Mittwoch, 05. Dezember 2007	07:50	16:40	08:50	diverse Programmierungen und Dokumentierung
Donnerstag, 06. Dezember 2007	07:50	09:30	01:40	diverse Programmierungen und Dokumentierung
Mittwoch, 12. Dezember 2007	07:50	16:40	08:50	diverse Programmierungen und Dokumentierung
Donnerstag, 13. Dezember 2007	07:50	09:30	01:40	diverse Programmierungen und Dokumentierung
Mittwoch, 19. Dezember 2007	07:50	16:40	08:50	diverse Programmierungen und Dokumentierung
Donnerstag, 20. Dezember 2007	07:50	08:40	00:50	diverse Programmierungen und Dokumentierung

8 Verzeichnisse

8.1 Abbildungsverzeichnis

Abbildung 1:	Qualitätssicherung in Papierform	3
Abbildung 2:	Übersicht über ALFSA	4
Abbildung 3:	System der Feuerwehr Tulln	7
Abbildung 4:	Strukturplan	8

8.2 Tabellenverzeichnis

Tabelle 1:	Projektziele.....	6
Tabelle 2:	Zeitplan.....	9
Tabelle 1:	Arbeitskalender von Andreas Brandstätter, stand 2008-01-16.....	12

8.3 Codeverzeichnis

Es konnten keine Einträge für ein Abbildungsverzeichnis gefunden werden.

9 Appendix

9.1 GPL 3.0

GNU GENERAL PUBLIC LICENSE ²

<<folgt>>

² Quelle: <http://www.gnu.org/licenses/gpl-3.0.html>;



Teil B

Projektentwicklung

1 Inhaltsverzeichnis

1	Inhaltsverzeichnis.....	2
2	Programmentwurf.....	4
3	Datenbankentwurf	5
3.1	Daten der Datenbank.....	5
3.2	Grundlegender Aufbau der Datenbank	6
4	Realisierungsentwurf.....	7
4.1	Virtualisierung	7
4.1.1	Vorteile.....	7
4.2	Open Source	7
4.2.1	Linux	8
4.3	Software	8
4.3.1	VMware	9
4.3.2	Debian Etch.....	9
4.3.3	Subversion	9
4.3.4	Apache 2.....	9
4.3.5	MySQL	10
4.3.6	PHP 5.....	10
4.4	Sprachen.....	10
4.4.1	Skriptsprache PHP	10
4.4.2	Markup-Sprache HTML.....	10
4.4.3	Stylesheet-Sprache CSS	11
4.4.4	Skriptsprache JavaScript.....	11
4.4.5	Skriptsprache sh.....	11
5	Umsetzung.....	12
5.1	Programmstruktur	12
5.2	Navigationsstruktur	13
5.3	Datenbankstruktur.....	14
5.4	Berechtigungsstruktur	15
5.4.1	Bereiche.....	15
5.4.2	Feststellung der Berechtigung.....	15
5.4.2.1	Beispiel	15
6	Eingabegeräte	16
6.1	Touchscreen	16
6.2	Barcode	16
6.2.1	Grundlegendes.....	17
6.2.2	Lesegeräte	17
6.2.2.1	Sicherheitshinweise.....	17
	Code39 18	
6.2.3	i25.....	18
6.3	Tastatur und Maus.....	18
7	Oberflächendesign	19
7.1	Vorgaben an das Design.....	19
7.1.1	Einsatzbedingungen	19
7.1.2	Auflösung.....	19

7.1.2.1	Häufig verwendete Browserauflösung	20
7.1.2.1.1	Web-Barometer	20
7.1.2.1.2	Feuerwehrjugend-NÖ	21
7.1.2.2	Touchscreen Bildschirme	22
7.1.2.3	Resultat	22
7.2	Farbkomposition	23
7.3	flying windows	24
7.3.1	Kategorien	24
7.4	Piktogramme	25
7.4.1	Logo-Icons	25
7.4.2	Button-Icons	25
7.4.3	Crystal-Icons	26
7.5	Design-Entwurf	26
7.6	Template-Engine	28
7.6.1	Grundlegende Verwendung	28
7.6.2	Variablen-Substitution	29
7.6.3	Feld-Substitution	29
7.6.4	Sonderzeichen-Substitution	29
7.6.5	Konstanten-Substitution	29
7.6.6	Schleifen-Substitution	30
7.6.7	Bedingte-Substitution	30
7.6.8	Erweiterte Bedingte-Substitution	31
7.6.9	Erweiterte Konstanten -Substitution	31
7.7	Konformität mit XHTML 1.0	32
7.7.1	Dokumente müssen wohlgeformt sein	32
7.7.1.1	Elemente dürfen nicht überlappen	32
7.7.1.2	Element- und Attributnamen müssen klein geschrieben werden	32
7.7.1.3	Für nicht-leere Elemente sind End-Tags erforderlich	32
7.7.1.4	Attributwerte müssen immer in Anführungszeichen eingeschlossen werden	33
7.7.1.5	Attributminimierung	33
7.7.1.6	Leere Elemente	33
7.8	Browserkompatibilität	33
7.8.1	Opera	33
7.8.2	Firefox	33
7.8.3	Internet Explorer	34
7.8.3.1	Nicht unterstützte Funktionen	34
7.8.3.2	Fazit	34
8	Verzeichnisse	35
8.1	Abbildungsverzeichnis	35
8.2	Tabellenverzeichnis	35
8.3	Codeverzeichnis	36
9	Appendix	38
9.1	Logo	38
9.2	Barcode-Tabelle	38

2 Programmmentwurf

Die Struktur der Virtualisierung, die zu verwendende Software und die Datenbankstruktur waren durch die Diplomarbeit ALFSA bereits vorgegeben. Somit entfielen nur noch geringe Entscheidungskompetenzen auf das Projekt aaron.

Der Entwurf des Programmes wurde stets mit Orientierung auf die einfache und intuitive Bedienung durchgeführt. Darauf Basierend konnte ein einfacher Entwurf der Oberfläche, angelehnt an das Bestehende System der Feuerwehr Tulln, erarbeitet werden. Diese Design-Entwürfe resultierten einerseits in genaueren Entwicklungen des Designs und des Layouts, andererseits im grundsätzlichen Aufbau der Programmstruktur. (siehe auch Kapitel 5 Umsetzung)

Eine wesentliche Kreation der Entwurfsphase stellt die Template-Engine dar. Durch eine Template-Engine ist es möglich die Daten und das Layout mit Design komplett zu trennen. Ferner können so Templates leicht ausgetauscht werden und somit der Layout-Aufbau komplett geändert werden ohne die Funktion, Sicherheit oder das Programm zu beeinflussen. Templates können außerdem von versierten Benutzern selbst geändert werden um das Programm persönlichen Bedürfnissen anzupassen.

Ebenfalls wurde die Barcode-Only-Bedienung, das heißt die Oberfläche nur mit dem Barcode-Scanner zu bedienen, in der Entwurfsphase hervorgebracht. Per Barcodeeingabe kann sich der Benutzer anmelden, durch die Oberfläche navigieren oder Daten eingeben.

Nicht zuletzt musste in der Entwurfsphase ein Codename für das Projekt gefunden werden. Nach langen Überlegungen konnte der Name „aaron“ gefunden werden. Aaron ist ein männlicher Vorname der im Alten Testament der Bibel zu finden ist. Es bedeutet etwa „Erleuchter“ oder „Großer Held“. ¹ Basierend auf dem Namen wurde das Logo der Diplomarbeit ALFSA modifiziert und so das Logo des Projektes entworfen.



Abbildung 1: aaron-Logo

¹ Quelle: <http://de.wikipedia.org/wiki/Aaron>;

3 Datenbankentwurf

Die Datenbank wurde im Zuge der Diplomarbeit ALFSA entworfen und konzipiert. Sie ist daher für das Projekt als gegeben anzusehen.

3.1 Daten der Datenbank

Die Datenbank besteht im Wesentlichen aus folgenden Daten:

- Allgemeine Daten (Füllstelle, Feuerwehr, Kompressor, ...)
- Benutzerdaten (Person, Benutzer, Berechtigungen, Gruppen, ...)
- Flaschendaten (Flaschen, Mängel, Füllungen, ...)
- Sonstige Daten (Geräte, Masken, Nachrichten, ...)

Diese Daten stehen in unterschiedlichen Beziehungen zueinander. Die Beziehungen werden in nachfolgenden Kapiteln genauer erläutert.

Ein Großteil der Allgemeinen Daten stellt sogenannte Quasistatische Daten dar. Diese Daten verändern sich in den seltensten Fällen. Sie werden auch als Stammdaten bezeichnet. Dazu gehören unter anderem Feuerwehren, Postleitzahlen, usw. Diese Daten werden vom Auftraggeber bereitgestellt und in das System importiert.

3.2 Grundlegender Aufbau der Datenbank

Die Datenbank ist grundlegend wie folgt aufgebaut.

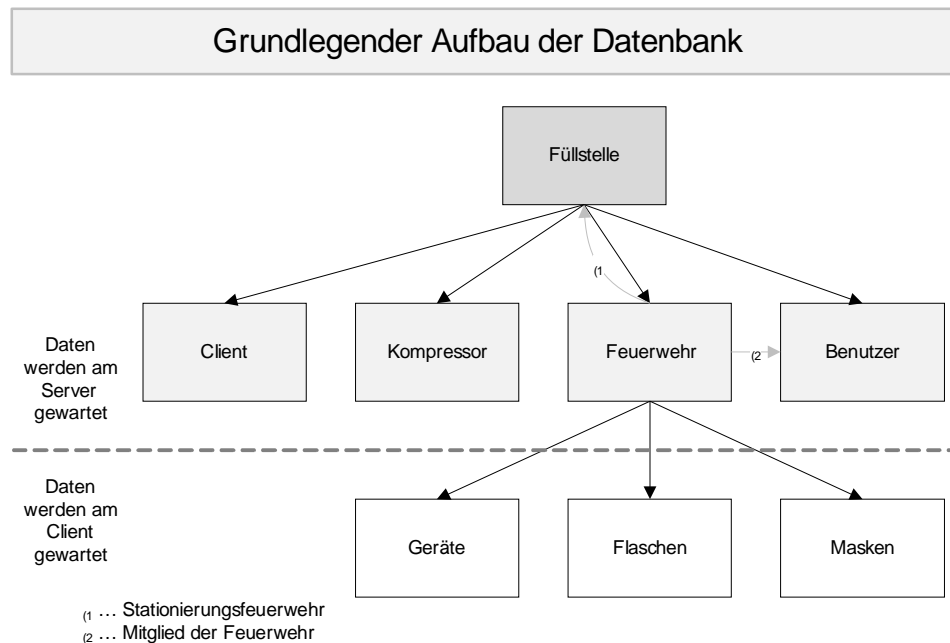


Abbildung 2: Grundlegender Aufbau der Datenbank

Zentrale Organisationseinheit ist die Füllstelle. Eine Füllstelle ist eine gemäß Kesselgesetz ermächtigte Einrichtung, die Berechtigt ist Atemluftflaschen zu füllen. Dieser Füllstelle können mehrere Kompressoren angehören. Ebenso können mehrere Clients von einer Füllstelle eingesetzt werden. Jede Feuerwehr ist einer primären Füllstelle zugeordnet. Benutzer sind ebenso einer Füllstelle zugeordnet bei der sie nach Kesselgesetz ermächtigt sind Atemluftflaschen zu füllen.

Atemluftflaschen, Geräte und Masken befinden sich im Eigentum einer Feuerwehr und werden somit dieser Feuerwehr zugeordnet. Dadurch sind Atemluftflaschen, Geräte und Masken indirekt auch zu einer dezidierten Füllstelle zugeordnet.

Folgende Daten sollten auf der jeweilig zuständigen Füllstelle bearbeitet werden können:

- Flaschendaten
- Gerätedaten
- Maskendaten

Folgende Daten werden auf allen Füllstellen eingetragen:

- Füllungen
- Mängel
- Nachrichten

Alle übrigen Daten werden ausschließlich am Server verändert und unterliegen so keinem ändernden Einfluss des Clients.

4 Realisierungsentwurf

In den folgenden Kapiteln wird der Entwurf der Client-Software dargestellt.

4.1 Virtualisierung

„Virtualisierung bezeichnet Methoden, die es erlauben, Ressourcen eines Computers aufzuteilen.“²

Ziel der Virtualisierung ist es in unserem Fall, die gesamte Software auf unterschiedlichen Betriebssystemen betreiben zu können.

Die Realisierung der Virtualisierung erfolgt wie in folgender Abbildung gezeigt:

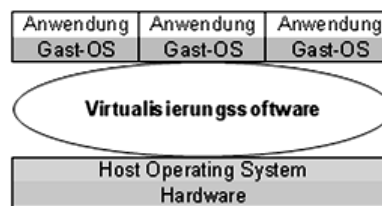


Abbildung 3: Virtualisierung³

Es wird die Software VMware eingesetzt und darin Linux als Gast-System betrieben.

4.1.1 Vorteile

Einerseits ist der Vorteil der verwendeten Virtualisierung die bereits erwähnte Plattformunabhängigkeit. Somit kann die Client-Software auf folgenden Betriebssystemen als Host verwendet werden:

- Linux
- Windows
- MacOS
- Solaris

Weiters ist die Client-Software durch die Verwendung von VMware kompakt auf ein Image reduziert. Dieses muss lediglich in die Maschine geladen und gestartet werden. Es fällt somit kein Aufwand für die Installation an. Außerdem kann bei kompletten Updates der ganzen Software das Image leicht ausgetauscht werden.

4.2 Open Source

Zur Realisierung des Projektes wird zum Größten Teil Open Source Software eingesetzt. Der Vorteil von Open Source Software liegt in der Stabilität, Sicherheit und geringen Feherrate. Durch die Offenlegung des Sourcecodes und der Möglichkeit für Jedermann die Software zu verbessern gibt es viele Entwickler, die eventuelle Fehler rasch finden und ebenso schnell beheben.

Ein nicht zu verachtender Aspekt ist auf jeden Fall der Kostenfaktor. Gerade bei einer Non-Profit-Organisation als Auftraggeber wie das Bezirksfeuerwehrkommando Tulln ist es wertvoll Kosten zu sparen.

² Quelle: [http://de.wikipedia.org/wiki/Virtualisierung_\(Informatik\)](http://de.wikipedia.org/wiki/Virtualisierung_(Informatik));

³ Quelle: http://de.wikipedia.org/wiki/Bild:Informatik_virtualisierungsvergleich.PNG; Eintrag: Virtualisierung (Informatik)

4.2.1 Linux

„Linux ist ein freies Multiplattform-Mehrbenutzer-Betriebssystem für Computer, das Unix ähnlich ist.“⁴



Abbildung 4: Linux Maskottchen/Logo Tux⁵

Das Betriebssystem Linux wird heute vorrangig auf Servern eingesetzt (auch bei der Diplomarbeit ALFSA), erfreut sich aber auch im Desktopbereich immer größerer Beliebtheit. Durch Open Source ist es sehr leicht möglich es nach eigenen Wünschen zu modifizieren. Weiters bietet die hohe Skalierbarkeit enorme Vorteile für die Verwendung als Gast-Betriebssystem. Es ist beispielsweise möglich komplett auf die Grafische Oberfläche zu verzichten. Dadurch lässt sich das Gast-Betriebssystem sehr klein und ressourcensparend halten.

4.3 Software

Am Client wird zur Gänze kostenlose Software und zum Großteil Open-Source-Software verwendet.

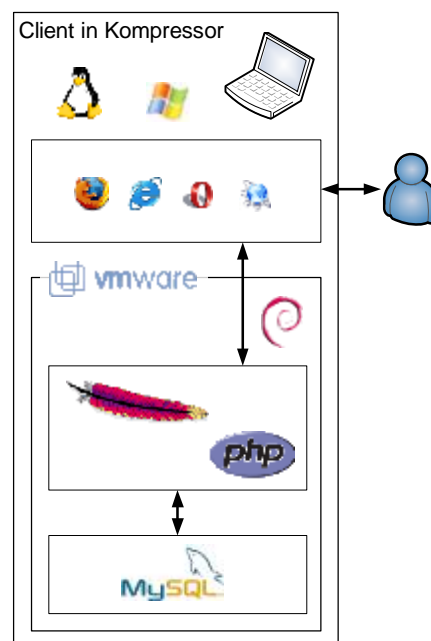


Abbildung 5: Software am Client

Diese Abbildung verdeutlicht welche Software am Client eingesetzt wird oder vom Benutzer verwendet werden kann. Ferner kann man die verschachtelte Struktur der Software erkennen.

⁴ Quelle: <http://de.wikipedia.org/wiki/Linux>;

⁵ Quelle: <http://de.wikipedia.org/wiki/Linux>;

4.3.1 VMware



Abbildung 6: VMware-Logo ⁶

VMware Server ist ein kostenloses Programm zur Virtualisierung der amerikanischen Firma VMware. Es kann auf Windows oder Linux als Wirtsbetriebssystem eingesetzt werden. Als Gastbetriebssystem ist beinahe jedes beliebige Betriebssystem möglich.

4.3.2 Debian Etch



Abbildung 7: Debian-Logo ⁷

Debian ist eine freie Linux-Distribution. Ein großer Vorteil liegt in der leichten Skalierbarkeit und Installation von zusätzlichen Komponenten durch apt-get. Debian legt Wert auf Sicherheit und Stabilität. Dadurch ist es optimal für den Einsatz im Client.

4.3.3 Subversion



Abbildung 8: Subversion-Logo ⁸

Subversion (SVN) ist eine Open-Source-Software zur Versionsverwaltung von Dateien und Verzeichnissen. Die Daten werden in einem zentralen Archiv abgelegt und in einzelnen Versionen verwaltet. Während dem Entwickeln werden die Daten auf den Entwicklungscomputer übertragen und nach der Veränderung wieder in das Archiv gespeichert. Für die Verwendung am Client werden die Daten ebenfalls aus dem Archiv kopiert.

4.3.4 Apache 2



Abbildung 9: Apache-Logo ⁹

Apache ist ein Open-Source Webserver. Der modulare aufbau von Apache ermöglicht es bei minimaler Größe genau jenen Funktionsumfang zu gewährleisten, der benötigt wird. Apache ist der meistbenutzte Webserver im Internet. ¹⁰

⁶ Quelle: <http://vmware.com/>;

⁷ Quelle: <http://www.debian.org/>;

⁸ Quelle: <http://subversion.tigris.org/>;

⁹ Quelle: <http://www.apache.org/>;

¹⁰ Quelle: http://news.netcraft.com/archives/web_server_survey.html;

4.3.5 MySQL



Abbildung 10: MySQL-Logo ¹¹

MySQL ist ein Relationales Datenbanksystem. Als Open-Source-Software kann MySQL frei verwendet werden. Der Sprachumfang von MySQL 5.0 entspricht weitgehend dem SQL3-Standard. MySQL unterstützt weiters Replikation mit Master und Slave sowie und Cluster mit einer Vielzahl an gleichartiger Nodes.

4.3.6 PHP 5



Abbildung 11: PHP-Logo ¹²

PHP ist eine serverseitige Open-Source Skriptsprache. Vorrangig wird sie zur Programmierung von Webseiten und Ähnlichem eingesetzt. Der Vorteil von PHP liegt in einfacher Datenbankbindung und einem relativ intuitivem Syntax.

4.4 Sprachen

Die verwendeten Sprachen sollen hier kurz dargestellt werden.

4.4.1 Skriptsprache PHP

Codeteil 1: PHP Syntaxbeispiel

```
<?php
    echo "Hello world";
?>
```

PHP ist eine serverseitige Open-Source Skriptsprache. Der Syntax von PHP ist C bzw. C++ sehr ähnlich und relativ intuitiv. PHP-Programmteile werden HTML-Text mit „<?php“ und „?>“ eingeschlossen.

4.4.2 Markup-Sprache HTML

Codeteil 2: HTML Syntaxbeispiel

```
<html>
    <head>
        <title>example</title>
    </head>
    <body>
        Hello world
    </body>
</html>
```

HTML ist entgegen der Meinung vieler ungebildeter Webdesigner keine Programmiersprache sondern lediglich eine Auszeichnungssprache zu Darstellung von Inhalt in Browsern. Die Formatierung erfolgt mit sogenannten Tags, die in spitzen Klammern eingeschlossen werden. Tags können ferner verschiedene Attribute beinhalten. In Kapitel 7.9 (Konformität mit XHTML 1.0) wird genauer auf den speziellen Syntax von XHTML eingegangen.

¹¹ Quelle: <http://www.mysql.de/>;

¹² Quelle: <http://www.php.net/>;

4.4.3 Stylesheet-Sprache CSS

Codeteil 3: CSS Syntaxbeispiel

```
.beispiel
{
    color: black;
    font-weight: bold;
    background-color: #AABBCC;
}
```

CSS wird als Stylesheet-Sprache verwendet um die Darstellung von HTML-Dateien grafisch zu formatieren. Mit CSS werden lediglich die Eigenschaften verschiedener Blöcke festgelegt. In HTML muss daher die Auszeichnung der einzelnen Blöcke durchgeführt werden.

4.4.4 Skriptsprache JavaScript

Codeteil 4: JavaScript Syntaxbeispiel

```
<script type="text/javascript">
    window.alert("Hello world");
</script>
```

JavaScript ist eine Clientseitige Skriptsprache. Sie wird meistens in HTML-Seiten zur dynamischen Gestaltung oder clientseitigen Datenmanipulation (zusätzliche Plausibilitätsprüfung von Formulareingaben) verwendet. Durch die Ausführung am Client sollten keine Sicherheitskritischen Vorgänge durchgeführt werden.

4.4.5 Skriptsprache sh

Codeteil 5: sh Syntaxbeispiel

```
#!/bin/sh
echo "Hello world"
```

Shell-Skripts automatisieren Befehlsfolgen des Betriebssystems, die analog auch in der Eingabeaufforderung eingegeben werden können. Die meisten Funktionen (auch einfache vergleiche) werden durch externe Programme ausgeführt. Ferner ist der Syntax sehr gewöhnungsbedürftig.

5 Umsetzung

5.1 Programmstruktur

Die Struktur der PHP-Dateien hängt maßgeblich an der index.php. Diese Datei führt alle wesentlichen Schritte durch:

- Libraries einbinden
- Template-Engine einbinden
- Prüfen, ob der User angemeldet ist
- HTML-Header und Design-Header mit Template ausgeben
- aufgerufene Seite einbinden
- Design-Footer mit Template ausgeben

Die Struktur lässt sich in folgender Abbildung darstellen:

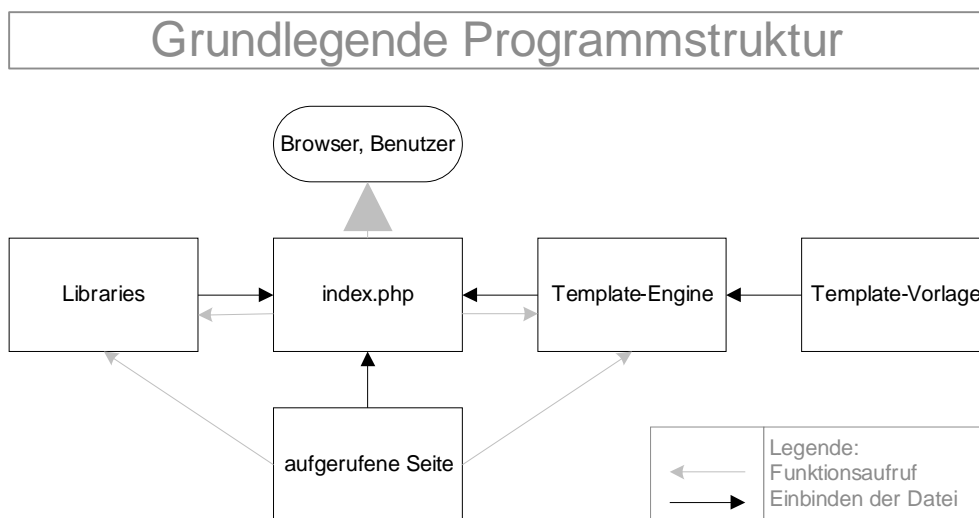


Abbildung 12: Grundlegende Programmstruktur

Diese Seiten gliedern sich in folgende Ordner:

Ordner	Inhalt	Dateityp
.	Indexdatei und verschiedene Hilfsdateien	
./conf	Konfigurationsdateien, die durch die Software erstellt werden	*.inc.php
./images	Bilder, Grafiken, Logos und Icons	*.png
./images/icons	Icons für Buttons	*.gif
./libs	Libraries mit Funktionen und Klassen	*.php
./pages	aufgerufene Seiten	*.inc.php
./sync	Dateien für Client-Server Synchronisation	*.php
./templates	Template-Dateien mit Layout für normale Bedienung	*.html
./templates/touchscreen	Template-Dateien mit Layout für Touchscreen-Bedienung	*.html

Tabelle 1: Ordnerstruktur

5.2 Navigationsstruktur

Die Navigation der Client Oberfläche wurde einfach gehalten um eine relativ intuitive Bedienung zu ermöglichen. Grundlegend ist die Anmeldung an das System, denn ohne Berechtigung dürfen keine Aktionen durchgeführt werden. Nach der Anmeldung erreicht der Benutzer die Hauptseite, von der aus alle weiteren Seiten erreicht werden können.

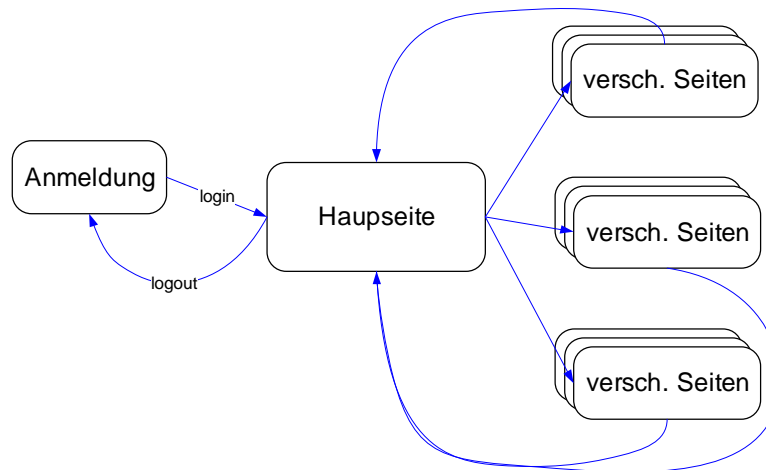


Abbildung 13: Grundlage der Navigationsstruktur

Seiten, die von der Hauptseite erreicht werden können:

(abhängig von der individuellen Berechtigung sind einige Seiten nicht vorhanden)


- Hauptfunktionen
 - Einsatz-Schnellstart (Füllvorgang)
 - Füllvorgang
 - Flaschendaten anzeigen
 - Abmelden
- Zusatzfunktionen
 - Daten synchronisieren
 - Programmupdate
 - Grundkonfiguration ändern
 - Nachrichten
 - Programminformationen
- Bearbeitungsfunktionen
 - Flaschen bearbeiten
 - Feuerwehren anzeigen
 - Benutzer anzeigen
- Adminfunktionen
 - Admin Interface
 - Error Logs anzeigen

Von jeder Seite ist es möglich einfach und rasch zur Hauptseite zurückzukehren, indem man auf das aaron-Logo klickt. Weiters ist auf allen Seiten das Barcode-Feld vorhanden, welches durch spezielle Barcodes auch Navigationsfunktionen übernehmen kann. So erreicht man Beispielsweise von der Hauptseite durch scannen einer Flasche die Seite „Flaschendaten anzeigen“.

5.3 Datenbankstruktur

Die Datenbank besteht aus folgenden Tabellen. Beziehungen werden noch nicht dargestellt, folgen jedoch in den nächsten Versionen der Dokumentation.

==> Siehe Anhang A!

In Abbildung 3 sind die Primärschlüssel der Tabellen mit den Symbolen  bezeichnet. Zusammengesetzte Primärschlüssel werden verwendet, um einen Datensatz eindeutig zu bezeichnen ohne eine eigene ID oder ähnliches vergeben zu müssen. Sequentiell oder zufällig erzeugte Schlüssel sind nicht verwendbar, da Daten auf unabhängigen Clients erzeugt werden und so die gleiche ID bekommen würden. Weitere Überlegungen und Ausführungen zum Datenbankkonzept finden sich in der Dokumentation der Diplomarbeit ALFSA.

5.4 Berechtigungsstruktur

Die Berechtigungsstruktur am Client ist relativ einfach aufgebaut. Sie umfasst einzelne Berechtigungsbereiche, die entweder Global oder Lokal erlaubt beziehungsweise verboten werden. Verbote sind stets höherrangig als Berechtigungen.

5.4.1 Bereiche

Folgende Berechtigungsbereiche sind in Verwendung:

Bereich	Inhalt	Standard
login	Der Benutzer darf sich anmelden.	global:allow
fill	Flaschen dürfen gefüllt werden. Einsätze dürfen angelegt und bearbeitet werden. Mängel dürfen erfasst werden.	local:allow
edit	Flaschendaten dürfen bearbeitet werden. Mängel dürfen behoben werden.	
show	Flaschendaten und Mängel dürfen angezeigt werden.	global:allow
config	Die Grundkonfiguration darf geändert werden.	
sync	Synchronisation und Programmupdate darf durchgeführt werden.	

Tabelle 2: Berechtigungsbereiche

Globale Rechte werden auf allen Füllstellen (inklusive der zugeordneten) angewendet, lokale Rechte nur auf der zugeordneten Füllstelle.

5.4.2 Feststellung der Berechtigung

Die Feststellung der Berechtigung für einen bestimmten Bereich erfolgt folgendermaßen in der hier dargestellten Reihenfolge:

- Berechtigungseinträge des bestimmten Bereiches (Benutzerspezifische Rechte und Standardrechte) selektieren
- Wenn nicht auf zugeordneter Füllstelle angemeldet, lokale Rechte verwerfen
- Wenn Eintrag mit „denie“ vorhanden => verweigern
- Wenn Eintrag mit „allow“ vorhanden => erlauben
- Kein Eintrag vorhanden => verweigern

5.4.2.1 Beispiel

Benutzer	Bereich	Berechtigung
Standard	login	global:allow
Standard	fill	local:allow
Hugo	fill	global:denie
Hugo	sync	local:allow
	config	<kein Eintrag>

Tabelle 3: Beispiel für Rechtekonfiguration

Hugo darf sich auf jeder Füllstelle einloggen, aber nicht füllen, nicht synchronisieren und nicht konfigurieren. Hugo darf auf eigener Füllstelle einloggen, synchronisieren, aber nicht füllen und nicht konfigurieren.

6 Eingabegeräte

Als Eingabegeräte werden komplett verschiedene Konzepte ermöglicht. Diese sind untereinander beliebig kombinierbar. So kann für jede Art der Eingabe die Möglichkeit gewählt werden, die am effizientesten und am angenehmsten für den Anwender ist.

6.1 Touchscreen

Mithilfe von Touchscreens ist es möglich direkt mit dem Ausgabegerät zu interagieren. Für gewöhnlich wird an der Angetippten Stelle ein Mausklick ausgeführt. Speziell für nicht Computer-erfahrene Anwender dürfte es von Vorteil sein, direkt darauf zu tippen, welche Aktion gesetzt werden soll. Ein weiterer Vorteil liegt darin, dass keine Maus und keine ebene Fläche für eine Maus benötigt werden. Eine Maus kann speziell bei Feuerwehreinsätzen und den verbundenen Bedingungen hinab geworfen oder verloren werden. Weiters ist bei Atemluftkompressoren oft keine ebene Fläche zur Bedienung der Maus vorhanden.



Abbildung 14: Touchscreen-Monitor ¹³

Heutige Hardware unterstützt Flächendeckend nur einen Berührungspunkt. Neue Technologien ermöglichen jedoch bereits mehrere Berührungspunkte also beispielsweise mehrere Finger zur gleichzeitigen Eingabe.

6.2 Barcode



Abbildung 15: Beispielbarcode „AARON“ ¹⁴

Barcodes finden weite Verbreitung in der Produktmarkierung und Lagerverwaltung. Barcodes sind maschinell lesbar und enthalten meist eine eindeutige Kennung in numerischer oder alphanumerischer Form.

¹³ Quelle: <http://groovytouch.j4.cc/gtm15l.html>;

¹⁴ Quelle: <http://www.bolege.de/barcode/index.html>;

6.2.1 Grundlegendes

Barcodes sind je nach Typ nach verschiedenen Gesichtspunkten optimiert:

- Hohe Datendichte
- Hohe Fehlersicherheit
- Großer Zeichensatz

Diese wird durch verschiedene Kodierungen, Checksummen oder ähnliche Verfahren realisiert.

6.2.2 Lesegeräte



Abbildung 16: Barcodescanner: Cyclone M2000 Series Scanner ¹⁵

Barcodes werden meistens mit Barcodescannern gelesen und die Daten an Computer oder ähnliche Geräte übergeben. Beim Projekt aaron wird bevorzugt ein Handscanner eingesetzt die schweren Atemluftflaschen leicht scannen zu können. Der Scanner muss Strichcodes des Typs i25 unterstützen. Für erweiterte Funktionen muss ferner der Typ Code39 unterstützt werden. Eine Erläuterung der Barcodetype folgt in den nächsten Kapiteln.

6.2.2.1 Sicherheitshinweise



Abbildung 17: Laserstrahlung ¹⁶

Abhängig vom jeweiligen Scanner kann die Laserstrahlung gefährlich sein. Dementsprechende Hinweise sind zu beachten.

¹⁵ Quelle: CycloneTM M2000 Series Product Reference Guide

¹⁶ Quelle: <http://www.zuv.uni-heidelberg.de/sw/interquellen/strahlenschutz.htm>;

Code39



Abbildung 18: Beispielbarcode „123456“¹⁷

Der Barcode des Typ Code39 bietet einen alphanumerischen Zeichensatz. Es ist möglich Großbuchstaben, Ziffern und 7 Sonderzeichen zu verwenden. Die Länge des Barcodes ist variabel. Weiters entspricht jedes Klartextzeichen genau einer definierten Strichfolge. Datensicherheit wird durch eine optionale Checksumme und ein Start- sowie ein Stopzeichen gewährleistet. Nachteilig ist die relativ geringe Informationsdichte, die sich in langen Barcodes bei bereits geringer Datenmenge auswirkt.

Code 39 ist schon relativ lange bekannt und wird daher von vielen Barcode-Scannern unterstützt.

6.2.3 i25



Abbildung 19: Beispielbarcode „123456“¹⁸

Der Typ i25 (auch genannt 2/5i, 2/5 Interleaved, ITF, Interleaved 2 of 5 oder Interleaved Two of Five) bietet nur Ziffern im Zeichensatz. Der Barcode ist nur mit gerader Ziffernanzahl verwendbar, optional kann eine Prüfziffer verwendet werden. Der Code verzahnt Stiche und Abstände so, dass die Stiche ein Zeichen bilden und die Abstände das nächste Zeichen. Daher sind nur gerade Ziffernanzahlen verwendbar.

Die Länge der dargestellten Beispiele in Abbildung 18: und Abbildung 19: verdeutlichen die hohe Informationsdichte von i25. Dieser Typ wird ebenfalls von vielen Barcode-Scannern unterstützt.

6.3 Tastatur und Maus



Abbildung 20: Tastatur und Maus¹⁹

Tastatur und Maus sind bei Computern im Consumer-Bereich äußerst verbreitet.

Die Tastatur ist zweifelsfrei die effizienteste und schnellste Methode Texte am Computer einzugeben. Für aaron ist es jedoch nicht erforderlich lange Texte zu schreiben.

Die Maus ist ebenfalls bei grafischen Oberflächen ziemlich effizient. Jedoch besitzt sie Nachteile, die in Kapitel 6.1 erläutert sind.

¹⁷ Quelle: http://studiouh.com/barcode_generator.php; Code39 mit Text „123456“

¹⁸ Quelle: http://studiouh.com/barcode_generator.php; i25 mit Text „123456“

¹⁹ Quelle: <http://de.wikipedia.org/wiki/Tastatur>; [http://de.wikipedia.org/wiki/Maus_\(Computer\)](http://de.wikipedia.org/wiki/Maus_(Computer));

7 Oberflächendesign

Das Design der Oberfläche ist von elementarer Bedeutung für die Client-Oberfläche.

7.1 Vorgaben an das Design

Das Design hat bei den nachfolgend genannten Einsatzbedingungen optimal zu funktionieren.

Außerdem ist es das Ziel neue und innovative Möglichkeiten der Darstellung im Bezug auf das Design zu entwickeln.

7.1.1 Einsatzbedingungen

Die Client-Software wird vorrangig auf TFT-Bildschirmen (Notebooks, Touchscreen-Bildschirme) dargestellt. Diese werden teilweise bei Feuerwehreinsätzen im Freien eingesetzt und betrieben. Im Speziellen werden mögliche schlechte Lichtverhältnisse angenommen.

Die Software wird außerdem teilweise von Computer-unerfahrenen Personen benutzt. Speziell diesen Benutzern soll durch ein einfaches und intuitives Design bei der Bedienung geholfen werden.

Diese Einsatzbedingungen fordern einen hohen Kontrast zur einfachen Erkenn- und Lesbarkeit der Bedienungselemente und Datendarstellung. Außerdem sind Schaltflächen großflächig auszuführen um bei einer Bedienung per Touchscreen einfache und fehlerarme Eingaben zu ermöglichen.

7.1.2 Auflösung

Um eine ideale Ausnutzung der Bildschirmfläche zu erreichen, wird die Oberfläche auf eine Minimale Auflösung optimiert. Die Nutzung mit geringerer Auflösung ist zwar möglich, aber sehr unkomfortabel. Die Verwendung mit höherer Auflösung absolut problemlos und ohne Beeinträchtigungen. Ferner wird die Client-Software immer im Vollbild-Modus verwendet. Somit ist die effektiv nutzbare Fläche gleich die Bildschirmauflösung abzüglich der Ränder des Browsers.

Die Ermittlung der minimalen Auflösung erfolgte nach nachfolgenden Aspekten.

7.1.2.1 Häufig verwendete Browserauflösung

Zur Analyse wurden zwei Quellen zur Ermittlung von häufigen Browserauflösungen herangezogen.

7.1.2.1.1 Web-Barometer

Die Firma „webhits internet design gmbh“ erfasst Statistische Daten von Zugriffen auf Internetseiten. Folgende Tabelle zeigt die Prozentuelle Verwendung der jeweiligen Auflösung.

Auflösung	Prozent
1024 x 768	44,8%
1280 x 1024	35,7%
1152 x 864	4,1%
800 x 600	3,0%
1600 x 1200	2,5%
andere	9,9%

Tabelle 4: Browserauflösungen in Prozent, Web-Barometer ²⁰

Daraus lässt sich folgendes Diagramm zeichnen. Es verdeutlicht die jeweilige Häufigkeit der einzelnen Auflösungen.

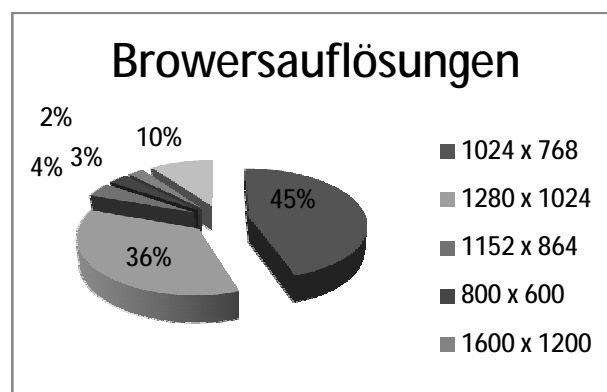


Abbildung 21: Browserauflösungen in Prozent, Diagramm basierend auf Tabelle 4:

²⁰ Quelle: <http://www.webhits.de/deutsch/index.shtml?webstats.html>; Web-Barometer: Aktuelle Daten zur allgemeinen Nutzung von Suchmaschinen, Browsern und Betriebssystemen sowie zu den Verbreitungszahlen der wichtigsten Plug-ins.; stand 2008-01-08

7.1.2.1.2 Feuerwehrjugend-NÖ

Die Homepage der Niederösterreichischen Feuerwehrjugend wird von etwa 250 Benutzern pro Tag besucht. Ein wesentlicher Teil der Besucher sind Feuerwehrmitglieder, daher kann man diese Statistik als durchaus relevant beurteilen. Die folgende Tabelle zeigt die Verwendung von Auflösungen von 15.000 Benutzern (1 Benutzer wurde ermittelt pro IP, pro Tag).

Auflösung	Prozent
1024x768	43,4%
1280x1024	26,2%
1280x800	11,5%
andere (größer als 1024x768)	7,2%
1152x864	5,0%
800x600	2,5%
1440x900	2,4%
1600x1200	2,0%

Tabelle 5: Browserauflösungen in Prozent, Feuerwehrjugend-NÖ ²¹

Ebenfalls lassen sich diese Werte als Diagramm darstellen.

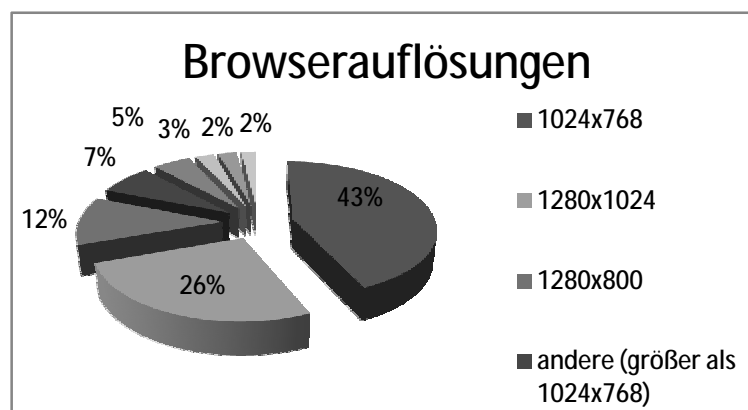


Abbildung 22: Browserauflösungen in Prozent, Diagramm basierend auf Tabelle 5:

²¹ Quelle: <http://www.feuerwehrjugend-noe.at/>; nicht öffentlich verfügbar; Urheber: Andreas Brandstätter; stand 2008-01-08

7.1.2.2 Touchscreen Bildschirme

Zur weiteren Ermittlung wird die Auflösung von aktuellen Touchscreen-Bildschirmen ermittelt. Die nachfolgende Tabelle zeigt die Auflösung einer Auswahl aktueller Bildschirme. Zur Analyse wurden Touchscreen-Bildschirme der Preisvergleichsseite

Auflösung	Prozent
1024x768	55,4%
1280x1024	35,7%
1280x800	1,8%
1600x1200	1,8%
800x600	5,4%

Tabelle 6: Touchscreen-Auflösungen in Prozent, Geizhals ²²

Das Diagramm zeigt die Verteilung der Auflösungen.

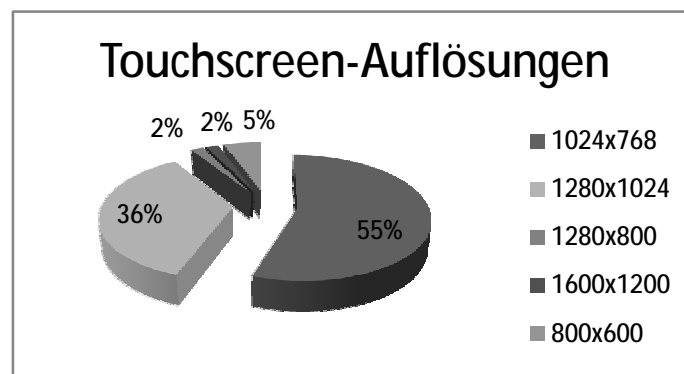


Abbildung 23: Touchscreen-Auflösungen in Prozent, Diagramm basierend auf Tabelle 6:

Somit ist eine Auflösung von mehr als 1280x800 auf jeden Fall auszuschließen. Der überwiegende Teil der aktuellen Touchscreen-Bildschirme hat eine Auflösung von 1024x768.

7.1.2.3 Resultat

Aus beiden Datenreihen der Browserauflösungen lassen sich ablesen, dass mehr als 97% eine Auflösung von 1024x786 oder größer verwenden. Ebenso ist die häufigste Auflösung aktueller Touchscreen-Bildschirme 1024x786.

Somit wurde beschlossen, die Client Oberfläche für eine Auflösung von 1024x786 zu optimieren.

²² Quelle: <http://geizhals.at/?cat=montouch>; stand: 2008-01-08

7.3 Layout-Entwurf

Das Layout wurde in zwei Bereiche unterteilt:

- Infoline
- Hauptbereich

Die folgende Abbildung zeigt die Unterteilung des Layouts:

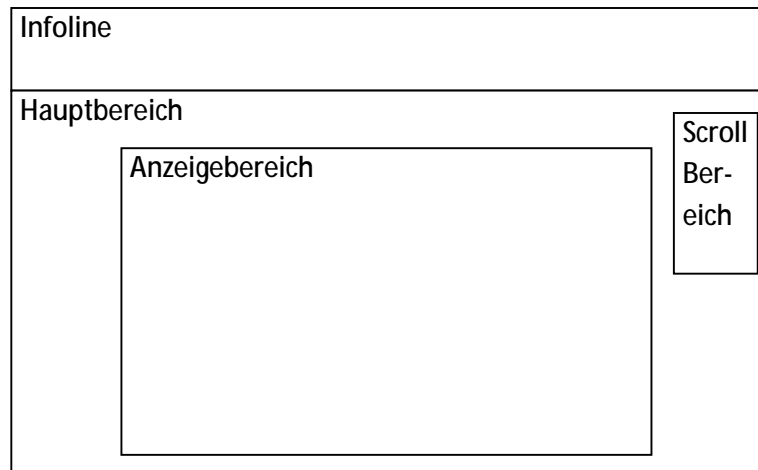


Abbildung 24: Kompletter Layout-Entwurf

Im Hauptbereich werden die tatsächlichen Daten im Anzeigebereich Form von Listen, Tabellen, usw. angezeigt. Der Scroll-Bereich stellt große Buttons für das scrollen mit Touchscreen zur Verfügung.

In der Infoline werden allgemeine statische Informationen angezeigt:

- Programmname
- Programmversion
- Kompressor
- Client
- Benutzer
- Uhrzeit
- Barcode-Feld (zur Eingabe von Barcodes mit Barcode-Scanner)
- Programm-Logo (Link zu Hauptseite)

Diese werden wie folgt verteilt:

Programmname, Programmversion	Uhr	Barcode-Feld	Programm-Logo
Kompressor, Client			
Benutzer			

Abbildung 25: Layout-Entwurf der Infoline

7.4 Farbkomposition

Zur Gestaltung der Benutzeroberfläche wurden folgende Farben gewählt:





Farbe	Muster	Hex-Code	Verwendung
Hellgrün		#E0F0E0	Hintergrund, Tabellenüberschriften
Dunkelgrün		#007000	Buttons
Hellgrau		#DDDDDD	Eingabefelder
Mittelgrau		#B0B0B0	Gruppierungsboxen

Tabelle 7: Verwendete Farben

An die Farben selbst wurden keine speziellen Anforderungen gestellt. Daher wurden beliebige aufeinander abgestimmte Farben gewählt.

7.5 flying windows

Fehlermeldungen, Hinweise und Tipps werden in sogenannten „flying windows“²³ angezeigt. Diese Textboxen können mit dem Mauszeiger per drag&drop verschoben werden. Außerdem können sie durch einen Klick kontrahiert bzw. wieder expandiert werden. Weiters ist es möglich ein „flying window“ per Mausklick zu schließen.



Abbildung 26: Beispiele für „flying windows“

Die dargestellten Beispiele für „flying windows“ zeigen die Kategorien „Information“ und „Tipp“ in expandierter Form, sowie die Kategorien „MySQL-Fehler“ und „Warnung“ in kontrahierter Form.

7.5.1 Kategorien

Diese „flying windows“ werden in folgende Kategorien unterteilt:

Typ	Langbezeichnung	Rahmenfarbe	Titelfarbe
ERROR	Fehler	#CD0101	#EB9A9A
INFO	Information	#1A5AC3	#A4BDE7
WARNING	Warnung	#FFCB01	#FFEB9A
PROG	Programmfehler	#FF9500	#FFD599
TIPP	Tipp	#357C47	#AECBB6
PERM	Berechtigungsproblem	#666666	#C2C2C2
DEBUG	Debug-Ausgabe	#1B1B1B	#A4A4A4
MYSQL	MySQL-Fehler	#186479	#A3C1CA
HELP	Hilfe	#80CB5D	#CDEBBE

Tabelle 8: Kategorien für „flying windows“

²³ Der Name „flying windows“ stellt eine Eigenkreation dar.

Die verschiedenen Farben für Fehlermeldungen wurden an die Norm DIN EN ISO 9241-110 angelehnt.



Abbildung 27: Kategorien für „flying windows“, gleiche Reihenfolge wie Tabelle 8.²⁴

Ferner werden Icons zur einfachen Erkennbarkeit der Fehlermeldungen eingesetzt.

7.6 Piktogramme

Zur leichteren Erkennbarkeit diverser Funktionen, Objekte und Datensätze werden Piktogramme (Icons) verwendet.

7.6.1 Logo-Icons



Abbildung 28: Logo als Icons

Das Logo der Projektarbeit wurde selbst gezeichnet und wird in verkleinerter Form an diversen Stellen als Piktogramm verwendet.

7.6.2 Button-Icons



Abbildung 29: Icons auf Buttons²⁵

Die Icons wurden mit folgendem Wortlaut veröffentlicht:

„my custom icon set, twotone, is now 100% free. every icon in the set, every colour, and the PSD to change the colours to whichever custom colour you want. you can use it on your blog, website, application or intranet.

what's the catch?

there isn't one. no licencing, no legal, just do whatever you want them. and if you want, post a link the comments if you do use them on a public site, so i can see where they're being used and how.“²⁶

²⁴ Quelle siehe Kapitel 7.6.3 Crystal-Icons

²⁵ Quelle: <http://www.2pt3.com/news/twotone-icons-for-free/>

²⁶ Zitat: <http://www.2pt3.com/news/twotone-icons-for-free/>

7.6.3 Crystal-Icons



Abbildung 30: Crystal Icons ²⁷

Die Crystal Icons stammen aus den Wikimedia Commons.

„Wikimedia Commons ist eine internationale freie Datenbank für Bilder, Videos, Musik und gesprochene Texte. Es ist für Benutzer kostenfrei und ohne Anmeldung nutzbar.“ ²⁸

Diese Icons stehen unter der „GNU Lesser General Public License“ und sind somit frei nutzbar.

7.7 Design-Entwurf

Nachfolgend werden einige Beispiele für das Oberflächendesign dargestellt.



Abbildung 31: Design-Entwurf: Login-Bereich

²⁷ Quelle: http://commons.wikimedia.org/wiki/Crystal_icons;

²⁸ Zitat: http://de.wikipedia.org/wiki/Wikimedia_Commons;



Abbildung 32: Design-Entwurf: Hauptübersicht

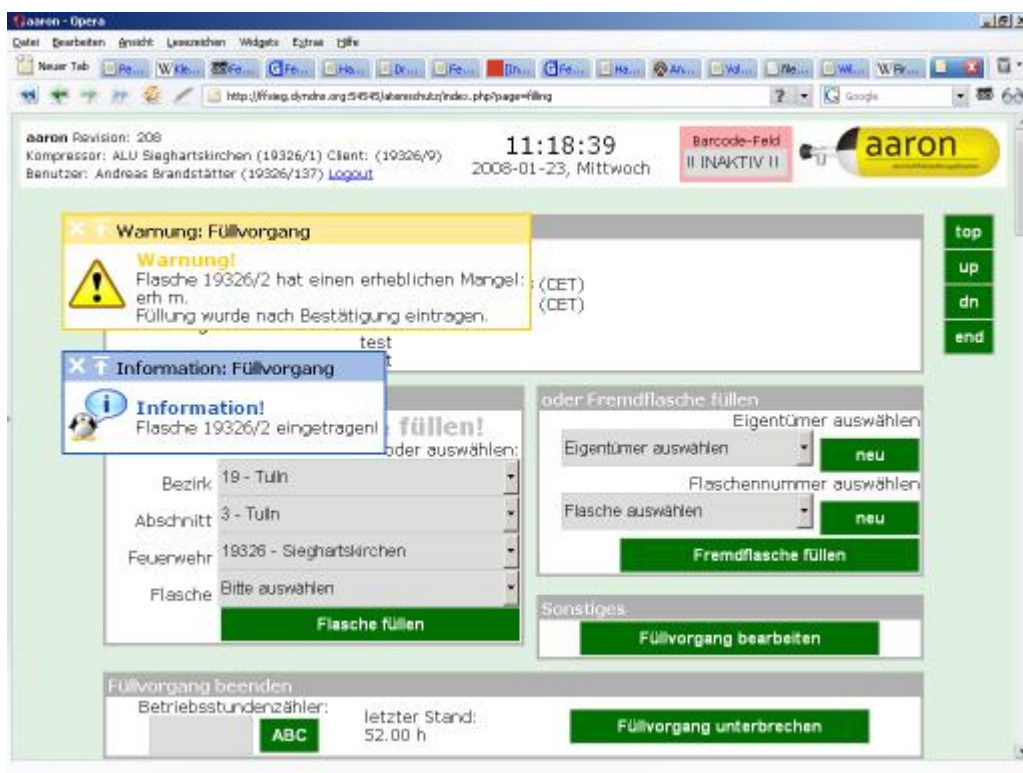


Abbildung 33: Design-Entwurf: Füllvorgang

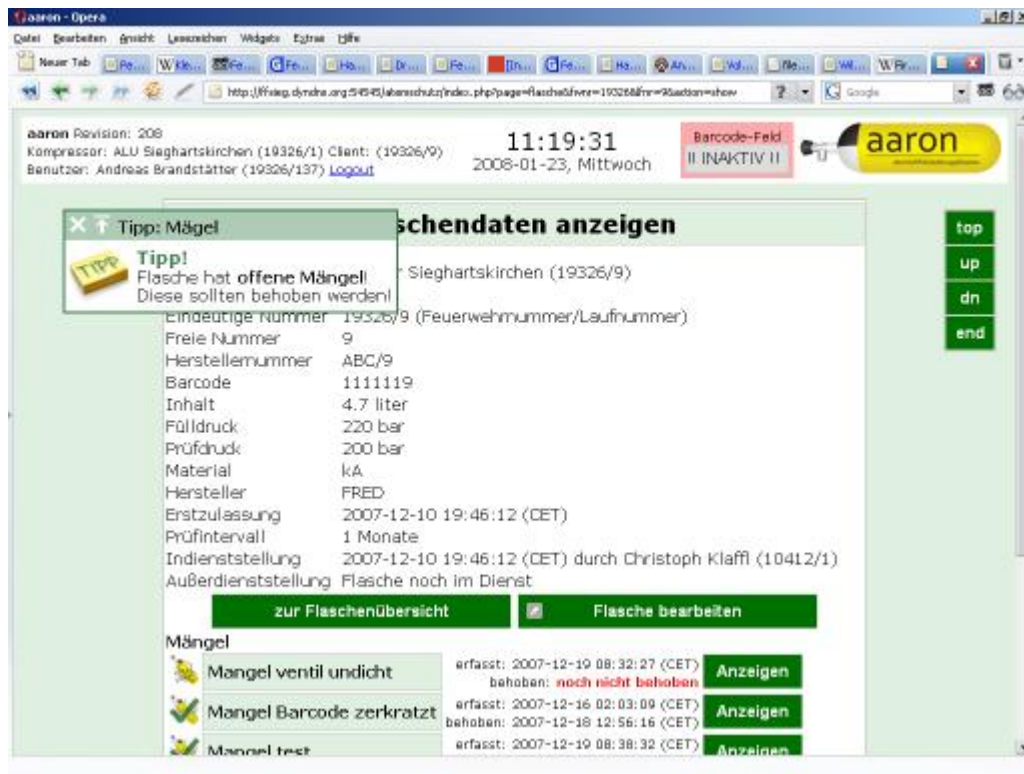


Abbildung 34: Design-Entwurf: Flaschendaten

7.8 Template-Engine

Eine Template-Engine ist eine Software, die Template (Vorlage) mit Platzhaltern verarbeitet und mit Inhalt füllt. So ist es möglich die Daten und das Layout mit Design komplett zu trennen. Außerdem so Templates leicht ausgetauscht werden und somit der Layout-Aufbau komplett geändert werden ohne die Funktion, Sicherheit oder das Programm ändern zu müssen. Templates können außerdem von versierten Benutzern selbst verändert werden um das Programm persönlichen Bedürfnissen anzupassen.

7.8.1 Grundlegende Verwendung

Die Verwendung der Template-Engine ist denkbar einfach. Mit dem Konstruktor wird das Template aus der Template-Datei geladen und erstellt. Die Funktion „set“ wird verwendet um Werte zu setzen. Mit der Funktion „display“ wird das Ergebnis ausgegeben.

Codeteil 6: Grundlegende Verwendung, PHP-Datei:

```
$test = new Template("template_name");
$test->set("variable", "gesetzter wert");
// hier diverse Ersetzungen eintragen...
$test->display();
```

Codeteil 7: Grundlegende Verwendung, Template-Datei:

```
Hello world
{variable}
```

Dieses Beispiel ergibt folgendes Resultat:

Codeteil 8: Grundlegende Verwendung, HTML-Ergebnis:

```
Hello world
gesetzter wert
```

7.8.2 Variablen-Substitution

Die Variablen-Substitution ermöglicht es einfache Inhalte aus der PHP-Datei im Template einzusetzen.

Codeteil 9: Variablen-Substitution, PHP-Datei:

```
$test->set("normal", "ersatz");
```

Codeteil 10: Variablen-Substitution, Template-Datei:

```
{normal}
```

Dieses Beispiel ergibt folgendes Resultat:

Codeteil 11: Variablen-Substitution, HTML-Ergebnis:

```
ersatz
```

7.8.3 Feld-Substitution

Die Feld-Substitution ermöglicht es einzelne Felder von Arrays im Template einzusetzen.

Codeteil 12: Feld-Substitution, PHP-Datei:

```
$test->set("namen", array("brandstaetter" => "andreas",  
                        "klaeffl" => "christoph",  
                        "alfery" => "wolfgang"));
```

Codeteil 13: Feld-Substitution, Template-Datei:

```
{namen/brandstaetter}  
{namen/klaeffl}  
{namen/alfery}
```

Dieses Beispiel ergibt folgendes Resultat:

Codeteil 14: Feld-Substitution, HTML-Ergebnis:

```
andreas  
christoph  
wolfgang
```

7.8.4 Sonderzeichen-Substitution

Da geschwungene Klammern durch die Template-Engine ersetzt werden, ist es notwendig eine Möglichkeit geschwungene Klammern im der HTML-Ergebnisdatei einzufügen.

Codeteil 15: Sonderzeichen-Substitution, PHP-Datei:

Codeteil 16: Sonderzeichen-Substitution, Template-Datei:

```
{(  
{>)}
```

Dieses Beispiel ergibt folgendes Resultat:

Codeteil 17: Sonderzeichen-Substitution, HTML-Ergebnis:

```
{  
}
```

7.8.5 Konstanten-Substitution

Konstanten können direkt durch die Template-Engine erzeugt werden. Es ist dazu kein spezieller Code in der verwendenden PHP-Datei notwendig.

Bezeichner	Ersetzung durch
------------	-----------------

use_type	Eingestellter Bedienungstyp in der Gundkonfiguration (Barcode-Scanner, Touchscreen, Maus und Tastatur)
date	Datum, Zeit und Zeitzone in der Form „Y-m-d H:i:s (T)“
self	Dateiname der aufrufenden Datei (relativ)
pagetime	Erstellungszeit der Datei in Millisekunden

Tabelle 9: Konstanten der Template-Engine

Codeteil 18: Konstanten-Substitution, PHP-Datei:

Codeteil 19: Konstanten-Substitution, Template-Datei:

```
{/use_type}
{/date}
{/self}
{/pagetime}
```

Dieses Beispiel ergibt folgendes Resultat:

Codeteil 20: Konstanten-Substitution, HTML-Ergebnis:

```
barcode
2007-10-30 21:16:51 (CET)
/tmp1/test.php
0.72 msec
```

7.8.6 Schleifen-Substitution

Mit schleifen kann ein Array in einer Dimension abgearbeitet werden. Hierbei wird ein mal pro Array-Element der Inhalt der Schleife ausgegeben.

Codeteil 21: Schleifen-Substitution, PHP-Datei:

```
$test->set("var", array(array("test" => "w1", "nochwas" => "xyz"),
                        array("test" => "w2", "nochwas" => "abc"),
                        array("test" => "w3", "nochwas" => "edf")));
```

Codeteil 22: Schleifen-Substitution, Template-Datei:

```
{%loop%var%}
schleifentest, element: {var/test}
{%/loop%}
```

Dieses Beispiel ergibt folgendes Resultat:

Codeteil 23: Schleifen-Substitution, HTML-Ergebnis:

```
schleifentest, element: w1
schleifentest, element: w2
schleifentest, element: w3
```

7.8.7 Bedingte-Substitution

Durch bedingte Substitution ist es möglich nur unter bestimmten Voraussetzungen gewisse Teile des Templates anzuzeigen.

Codeteil 24: Bedingte-Substitution, PHP-Datei:

```
$test->set("probe", "eins");
```

Codeteil 25: Bedingte-Substitution, Template-Datei:

```
{%if%probe/eins%}
bedingung erfuehlt
{%/if%}
```


Dieses Beispiel ergibt folgendes Resultat:

Codeteil 26: Bedingte-Substitution, HTML-Ergebnis:
bedingung erfuehlt

7.8.8 Erweiterte Bedingte-Substitution

Durch die erweiterte bedingte Substitution ist es möglich nur unter bestimmten komplexeren Voraussetzungen gewisse Teile des Templates anzuzeigen.

Operator	Name	Erfüllt, wenn...
eq	equal	gleich
ne	not equal	ungleich
lt	less than	kleiner
gt	greater than	größer
le	less equal	kleiner od. gleich
ge	less equal	größer od. gleich

Tabelle 10: Vergleichsoperatoren der Template-Engine

Codeteil 27: Erweiterte Bedingte-Substitution, PHP-Datei:
\$test->set("zahl", 6);

Codeteil 28: Erweiterte Bedingte-Substitution, Template-Datei:
{%if/eq%zahl/5%}eq-wahr{%/if%}
{%if/ne%zahl/5%}ne-wahr{%/if%}
{%if/lt%zahl/5%}lt-wahr{%/if%}
{%if/gt%zahl/5%}gt-wahr{%/if%}
{%if/le%zahl/5%}le-wahr{%/if%}
{%if/ge%zahl/5%}ge-wahr{%/if%}

Dieses Beispiel ergibt folgendes Resultat:

Codeteil 29: Erweiterte Bedingte-Substitution, HTML-Ergebnis:

ne-wahr

gt-wahr

ge-wahr

7.8.9 Erweiterte Konstanten-Substitution

Bei Verwendung von Schleifen-Substitutionen sind weitere Konstanten verfügbar. Die Konstante loop_counter ist jeweils nur in Schleifen gültig und gibt die aktuelle Zahl der Durchläufe an.

Codeteil 30: Schleifen-Substitution, PHP-Datei:
\$test->set("var", array(array("test" => "w1", "nochwas" => "xyz"),
array("test" => "w2", "nochwas" => "abc"),
array("test" => "w3", "nochwas" => "edf")));

Codeteil 31: Schleifen-Substitution, Template-Datei:
{%loop%var%}
schleifentest, element: {var/test} im durchlauf {loop_counter}
{%/loop%}
laenge der schleife: {%if/loopen%var%} {%/loopen%}

Dieses Beispiel ergibt folgendes Resultat:

Codeteil 32: Schleifen-Substitution, HTML-Ergebnis:

```
schleifentest, element: w1 im durchlauf 0
schleifentest, element: w2 im durchlauf 1
schleifentest, element: w3 im durchlauf 2
laenge der schleife: 3
```

7.9 Konformität mit XHTML 1.0

Bei der Entwicklung wurde vor allem darauf Wert gelegt XHTML 1.0 konforme Dateien zu produzieren.

Folgende Punkte sind hierbei wesentlich:²⁹

7.9.1 Dokumente müssen wohlgeformt sein

Wohlgeformtheit wurde durch XML eingeführt. Dies bedeutet im wesentlichen, dass alle Elemente entweder abschließende Tags haben müssen, selbst geschlossen werden und dass alle Elemente korrekt verschachtelt werden müssen.

7.9.1.1 Elemente dürfen nicht überlappen

Elemente müssen korrekt ineinander verschachtelt werden.

Codeteil 33: RICHTIG: verschachtelte Elemente

```
<p>here is an emphasized <em>paragraph</em>.</p>
```

Sie dürfen sich nicht gegenseitig überlappen.

Codeteil 34: FALSCH: Überlappende Elemente

```
<p>here is an emphasized <em>paragraph.</p></em>
```

7.9.1.2 Element- und Attributnamen müssen klein geschrieben werden

In XHTML-konformen Dokumenten werden sämtliche HTML Element- und Attributnamen klein geschrieben. Dies ist notwendig, da XML case-sensitiv ist, z.B. sind `` und `` unterschiedliche Tags.

Codeteil 35: RICHTIG: Kleinbuchstaben

```
<b>bold text</b>
```

Großbuchstaben sind für Tags nicht erlaubt.

Codeteil 36: FALSCH: Großbuchstaben

```
<B>bold text</B>
```

7.9.1.3 Für nicht-leere Elemente sind End-Tags erforderlich

Alle geöffneten Tags müssen wieder geschlossen werden.

Codeteil 37: RICHTIG: abgeschlossene Elemente

```
<p>here is a paragraph.</p><p>here is another paragraph.</p>
```

Folgenden Elemente implizieren nicht die Schließung der vorhergehenden Elemente.

Codeteil 38: FALSCH: nicht abgeschlossene Elemente

```
<p>here is a paragraph.<p>here is another paragraph.
```

²⁹ Quelle: <http://www.websitedev.de/xhtml/xhtml1/#diffs>;

7.9.1.4 Attributwerte müssen immer in Anführungszeichen eingeschlossen werden

Alle Attributwerte müssen in Anführungszeichen eingeschlossen werden, auch numerische.

Codeteil 39: RICHTIG: In Anführungszeichen gesetzte Attributwerte

```
<table rows="3">
```

Attributwerte ohne Anführungszeichen sind nicht erlaubt.

Codeteil 40: FALSCH: nicht in Anführungszeichen gesetzte Attributwerte

```
<table rows=3>
```

7.9.1.5 Attributminimierung

Attributnamen müssen immer Wertzuweisungen enthalten. Attributminimierung wird nicht unterstützt.

Codeteil 41: RICHTIG: nicht-minimierte Attribute

```
<dl compact="compact">
```

Attributnamen wie compact und checked können nicht ohne eine Angabe eines Wertes in Elementen eingetragen werden.

Codeteil 42: FALSCH: minimierte Attribute

```
<dl compact>
```

7.9.1.6 Leere Elemente

Alle Elemente müssen einen End-Tag haben oder selbst geschlossen werden mit „/“.

Codeteil 43: RICHTIG: abgeschlossene leere Elemente

```
<br /><hr />
```

Elemente ohne Ende sind nicht erlaubt.

Codeteil 44: FALSCH: nicht-abgeschlossene leere Elemente

```
<br><hr>
```

7.10 Browserkompatibilität

Durch die Verwendung einer Weboberfläche als Client-Oberfläche besteht die Notwendigkeit die Seite in einem Web-Browser darzustellen. Das World Wide Web Consortium (W3C) hat Empfehlungen der Techniken zu Darstellung von Webseiten veröffentlicht. Jedoch unterstützen nicht alle Browser diese De-facto-Standards.

7.10.1 Opera

Die Client-Oberfläche während der Entwicklung laufend in Opera getestet. Sie wird somit im Opera im Bezug auf Browserkompatibilität zu 100 Prozent richtig dargestellt.

Verwendete Version: Opera Version 9.25

7.10.2 Firefox

Die Darstellung im Firefox erfolgt ebenfalls problemlos. Nach einigen Anpassungen werden alle Elemente korrekt dargestellt.

Verwendete Version: Firefox Version 2.0.0.9

7.10.3 Internet Explorer

Aufgrund der beschränkten Kompatibilität von Internet Explorer mit Standards können einige Details nicht korrekt dargestellt werden. Die Funktion ist jedoch durch Programmiertechnische Anpassungen ohne wesentliche Abstriche gewährleistet.

Verwendete Version: Internet Explorer Version 6.0.2900.2180.xpsp_sp2_qfe.070227-2300

7.10.3.1 Nicht unterstützte Funktionen

Diese Funktionen wurden durch andere ersetzt um die Darstellung im Internet Explorer zu ermöglichen:

- § Keine transparenten PNGs
- § Keine SVG Unterstützung
- § Keine absolute Positionierung von Elementen

7.10.3.2 Fazit

Der Internet Explorer kann zwar für die Bedienung der Client-Oberfläche verwendet werden, aber es wird davon abgeraten.

Es wird daher empfohlen Opera oder Firefox zu verwenden.

8 Verzeichnisse

8.1 Abbildungsverzeichnis

Abbildung 1:	aaron-Logo.....	4
Abbildung 2:	Grundlegender Aufbau der Datenbank.....	6
Abbildung 3:	Virtualisierung	7
Abbildung 4:	Linux Maskottchen/Logo Tux	8
Abbildung 5:	Software am Client.....	8
Abbildung 6:	VMware-Logo	9
Abbildung 7:	Debian-Logo	9
Abbildung 8:	Subversion-Logo	9
Abbildung 9:	Apache-Logo	9
Abbildung 10:	MySQL-Logo	10
Abbildung 11:	PHP-Logo	10
Abbildung 12:	Grundlegende Programmstruktur.....	12
Abbildung 13:	Grundlage der Navigationsstruktur.....	13
Abbildung 14:	Tabellen der Datenbank	Fehler! Textmarke nicht definiert.
Abbildung 15:	Touchscreen-Monitor	16
Abbildung 16:	Beispielbarcode „AARON“	16
Abbildung 17:	Barcodescanner: Cyclone M2000 Series Scanner	17
Abbildung 18:	Laserstrahlung	17
Abbildung 19:	Beispielbarcode „123456“	18
Abbildung 20:	Beispielbarcode „123456“	18
Abbildung 21:	Tastatur und Maus	18
Abbildung 22:	Browserauflösungen in Prozent, Diagramm basierend auf Tabelle 4:.....	20
Abbildung 23:	Browserauflösungen in Prozent, Diagramm basierend auf Tabelle 5:.....	21
Abbildung 24:	Touchscreen-Auflösungen in Prozent, Diagramm basierend auf Tabelle 6:	22
Abbildung 25:	Beispiele für „flying windows“	24
Abbildung 26:	Kategorien für „flying windows“, gleiche Reihenfolge wie Tabelle 8:	25
Abbildung 27:	Logo als Icons	25
Abbildung 28:	Icons auf Buttons	25
Abbildung 29:	Crystal Icons	26
Abbildung 30:	Design-Entwurf: Login-Bereich	26
Abbildung 31:	Design-Entwurf: Hauptübersicht.....	27
Abbildung 32:	Design-Entwurf: Füllvorgang.....	27
Abbildung 33:	Design-Entwurf: Flaschendaten	28
Abbildung 34:	Logo aaron	38

8.2 Tabellenverzeichnis

Tabelle 1:	Berechtigungsbereiche.....	15
Tabelle 2:	Beispiel für Rechtekonfiguration	15
Tabelle 3:	Browserauflösungen in Prozent, Web-Barometer	20
Tabelle 4:	Browserauflösungen in Prozent, Feuerwehrjugend-NÖ	21
Tabelle 5:	Touchscreen-Auflösungen in Prozent, Geizhals	22

Tabelle 6:	Verwendete Farben	24
Tabelle 7:	Kategorien für „flying windows“	24
Tabelle 8:	Konstanten der Template-Engine	30
Tabelle 9:	Vergleichsoperatoren der Template-Engine	31

8.3 Codeverzeichnis

Codeteil 1:	PHP Syntaxbeispiel	10
Codeteil 2:	HTML Syntaxbeispiel	10
Codeteil 3:	CSS Syntaxbeispiel	11
Codeteil 4:	JavaScript Syntaxbeispiel	11
Codeteil 5:	sh Syntaxbeispiel	11
Codeteil 6:	Grundlegende Verwendung, PHP-Datei:	28
Codeteil 7:	Grundlegende Verwendung, Template-Datei:	28
Codeteil 8:	Grundlegende Verwendung, HTML-Ergebnis:	28
Codeteil 9:	Variablen-Substitution, PHP-Datei:	29
Codeteil 10:	Variablen-Substitution, Template-Datei:	29
Codeteil 11:	Variablen-Substitution, HTML-Ergebnis:	29
Codeteil 12:	Feld-Substitution, PHP-Datei:	29
Codeteil 13:	Feld-Substitution, Template-Datei:	29
Codeteil 14:	Feld-Substitution, HTML-Ergebnis:	29
Codeteil 15:	Sonderzeichen-Substitution, PHP-Datei:	29
Codeteil 16:	Sonderzeichen-Substitution, Template-Datei:	29
Codeteil 17:	Sonderzeichen-Substitution, HTML-Ergebnis:	29
Codeteil 18:	Konstanten-Substitution, PHP-Datei:	30
Codeteil 19:	Konstanten-Substitution, Template-Datei:	30
Codeteil 20:	Konstanten-Substitution, HTML-Ergebnis:	30
Codeteil 21:	Schleifen-Substitution, PHP-Datei:	30
Codeteil 22:	Schleifen-Substitution, Template-Datei:	30
Codeteil 23:	Schleifen-Substitution, HTML-Ergebnis:	30
Codeteil 24:	Bedingte-Substitution, PHP-Datei:	30
Codeteil 25:	Bedingte-Substitution, Template-Datei:	30
Codeteil 26:	Bedingte-Substitution, HTML-Ergebnis:	31
Codeteil 27:	Erweiterte Bedingte-Substitution, PHP-Datei:	31
Codeteil 28:	Erweiterte Bedingte-Substitution, Template-Datei:	31
Codeteil 29:	Erweiterte Bedingte-Substitution, HTML-Ergebnis:	31
Codeteil 30:	Schleifen-Substitution, PHP-Datei:	31
Codeteil 31:	Schleifen-Substitution, Template-Datei:	31
Codeteil 32:	Schleifen-Substitution, HTML-Ergebnis:	31
Codeteil 33:	RICHTIG: verschachtelte Elemente	32
Codeteil 34:	FALSCH: Überlappende Elemente	32
Codeteil 35:	RICHTIG: Kleinbuchstaben	32
Codeteil 36:	FALSCH: Großbuchstaben	32
Codeteil 37:	RICHTIG: abgeschlossene Elemente	32
Codeteil 38:	FALSCH: nicht abgeschlossene Elemente	32
Codeteil 39:	RICHTIG: In Anführungszeichen gesetzte Attributwerte	33

Codeteil 40:	FALSCH: nicht in Anführungszeichen gesetzte Attributwerte	33
Codeteil 41:	RICHTIG: nicht-minimierte Attribute.....	33
Codeteil 42:	FALSCH: minimierte Attribute	33
Codeteil 43:	RICHTIG: abgeschlossene leere Elemente	33
Codeteil 44:	FALSCH: nicht-abgeschlossene leere Elemente.....	33

9 Appendix

9.1 Logo





Abbildung 35: Logo aaron

9.2 Barcode-Tabelle

Anhang A





47 Tabellen der Datenbank

abschnitte

	abschnittnr	tinyint(3) unsigned
	bezirk	tinyint(3) unsigned
	name	varchar(255)
	edit_date	int(10) unsigned



2008-01-24 08:35:36

assp_trupp_mitglieder

	einsatz_fwnr	int(10) unsigned
	trupp_nr	int(10) unsigned
	einsatz_time	int(10) unsigned
	erfasser_fwnr	int(10) unsigned
	erfasser_stbnr	int(10) unsigned
	mitglied_fwnr	int(10) unsigned
	mitglied_stbnr	int(10) unsigned
	typ	enum('führer','mann')
	edit_date	int(10) unsigned

2008-01-24 08:35:36

assp_trupp_taetigkeit

	einsatz_fwnr	int(10) unsigned
	trupp_nr	int(10) unsigned
	einsatz_time	int(10) unsigned
	status	varchar(255)
	beginn	int(10) unsigned
	ende	int(10) unsigned
	erfasser_fwnr	int(10) unsigned
	ueberwachungsdauer	int(10) unsigned
	ort	varchar(255)
	voralarm	int(10) unsigned
	erfasser_stbnr	int(10) unsigned
	beender_fwnr	int(10) unsigned
	beender_stbnr	int(10) unsigned
	voralarm_quitiert	enum('0','1')
	edit_date	int(10) unsigned







2008-01-24 08:35:36

atempluftflaschen

	fwnr	int(10) unsigned
	laufnummer	int(10) unsigned
	herstellernummer	varchar(255)
	barcode	int(6) unsigned
	inhalt	int(3) unsigned
	fuelldruck	int(3) unsigned
	material	varchar(255)
	erstzulassung	int(10) unsigned
	pruefdruck	int(3) unsigned
	pruefintervall	int(2) unsigned
	indienst	int(10) unsigned
	indienststeller_fwnr	varchar(45)
	ausdienst	int(10) unsigned
	ausdienststeller_fwnr	int(10) unsigned
	hersteller	varchar(255)
	indienststeller_stbnr	int(10) unsigned
	ausdienststeller_stbnr	int(10) unsigned
	freie_nummer	varchar(255)
	edit_date	int(10) unsigned




2008-01-24 08:35:36

atempluftflaschen_fuellung

 atempluftflaschen_fwnr	int(10) unsigned
 atempluftflaschen_laufnummer	int(10) unsigned
 time	int(11) unsigned
 fuell_sitzung_time	int(10) unsigned
 kompressoren_fwnr	int(10) unsigned
 kompressoren_laufnummer	int(10) unsigned
edit_date	int(10) unsigned




2008-01-24 08:35:36

atempluftflaschen_maengel

 fwnr	int(10) unsigned
 laufnummer	int(10) unsigned
mangel	varchar(255)
loesung	varchar(255)
feststeller_fwnr	int(10) unsigned
 datum_feststellung	int(10) unsigned
beheber_fwnr	int(10) unsigned
datum_behebung	int(10) unsigned
feststeller_stbnr	int(10) unsigned
beheber_stbnr	int(10) unsigned
beschadigungsgrad	tinyint(3) unsigned
edit_date	int(10) unsigned



2008-01-24 08:35:36

atempluftflaschen_pruefung

 fwnr	int(10) unsigned
 laufnummer	int(10) unsigned
 datum	int(10) unsigned
hauptpruefung	char(1)
pruefer	varchar(255)
hauptpruefer	varchar(255)
edit_date	int(10) unsigned




2008-01-24 08:35:36

atemschutzgeraete

 fwnr	int(10) unsigned
 laufnummer	int(10) unsigned
aufbewahrungsort	varchar(255)
hersteller	varchar(255)
geraeteart	varchar(255)
type	varchar(255)
nr_druckminderer	varchar(255)
nr_lungenautomat	varchar(255)
nr_rahmen	varchar(255)
barcode	int(6) unsigned
baujahr	int(4) unsigned
indienst	int(10) unsigned
indienststeller_fwnr	int(10) unsigned
ausdienst	int(10) unsigned
ausdienststeller_fwnr	int(10) unsigned
wartungsintervall	int(10) unsigned
pruefintervall	int(10) unsigned
grundueberholung	int(10) unsigned
indienststeller_stbnr	int(10) unsigned
ausdienststeller_stbnr	int(10) unsigned
freie_nummer	varchar(255)
edit_date	int(10) unsigned




2008-01-24 08:35:36

atemschutzgeraete_maengel

 fwnr	int(10) unsigned
 laufnummer	int(10) unsigned
mangel	varchar(255)
loesung	varchar(255)
feststeller_fwnr	int(10) unsigned
 datum_feststellung	int(10) unsigned
beheber_fwnr	int(10) unsigned
datum_behebung	int(10) unsigned
feststeller_stbnr	int(10) unsigned
beheber_stbnr	int(10) unsigned
beschadigungsgrad	tinyint(3) unsigned
edit_date	int(10) unsigned




2008-01-24 08:35:36

atemschutzgeraete_pruefung

 fwnr	int(10) unsigned
 laufnummer	int(10) unsigned
 datum	int(10) unsigned
befund	varchar(255)
pruefung	varchar(1)
grundueberholung	varchar(1)
reparatur	varchar(1)
fremdbetreuer	varchar(255)
eigenpersonal_fwnr	int(10) unsigned
eigenpersonal_stbnr	int(10) unsigned
edit_date	int(10) unsigned



2008-01-24 08:35:36

atemschutzgeraete_wartung

 fwnr	int(10) unsigned
 laufnummer	int(10) unsigned
 datum	int(10) unsigned
personal_fwnr	int(10) unsigned
personal_stbnr	int(10) unsigned
edit_date	int(10) unsigned




2008-01-24 08:35:36

atemschutzmasken

 fwnr	int(10) unsigned
maskentyp	varchar(255)
 laufnummer	int(10) unsigned
hersteller	varchar(255)
baujahr	int(10) unsigned
pruefintervall	int(10) unsigned
indienst	int(10) unsigned
indienststeller_fwnr	int(10) unsigned
ausdienst	int(10) unsigned
ausdienststeller_fwnr	int(10) unsigned
barcode	int(6) unsigned
indienststeller_stbnr	int(10) unsigned
ausdienststeller_stbnr	int(10) unsigned
freie_nummer	varchar(255)
edit_date	int(10) unsigned




2008-01-24 08:35:36

atemschutzmasken_maengel

 fwnr	int(10) unsigned
 laufnummer	int(10) unsigned
mangel	varchar(255)
loesung	varchar(255)
feststeller_fwnr	int(10) unsigned
 datum_feststellung	int(10) unsigned
beheber_fwnr	int(10) unsigned
datum_behebung	int(10) unsigned
feststeller_stbnr	int(10) unsigned
beheber_stbnr	int(10) unsigned
beschadigungsgrad	tinyint(3) unsigned
edit_date	int(10) unsigned



2008-01-24 08:35:36

atemschutzmasken_wartung

 fwnr	int(10) unsigned
 laufnummer	int(10) unsigned
 datum	int(10) unsigned
warter_fwnr	int(10) unsigned
reinigung	varchar(1)
desinfektion	varchar(1)
pruefung	varchar(1)
reparatur	varchar(1)
warter_stbnr	int(10) unsigned
edit_date	int(10) unsigned




2008-01-24 08:35:36

benutzer

 fwnr	int(10) unsigned
 stbnr	int(10) unsigned
fuellstelle	int(10) unsigned
passwort	char(32)
angelegt	int(10) unsigned
abgeschlossen	int(10) unsigned
anleger_fwnr	int(10) unsigned
anleger_stbnr	int(10) unsigned
beender_fwnr	int(10) unsigned
beender_stbnr	int(10) unsigned
admin	enum('0','1')
edit_date	int(10) unsigned






2008-01-24 08:35:36

benutzer_anmeldung

 fwnr	int(10) unsigned
 stbnr	int(10) unsigned
 anmeldung	int(10) unsigned
abmeldung	int(10) unsigned
fuellstelle	int(10) unsigned
client_laufnummer	int(10) unsigned
kompressor_laufnummer	int(10) unsigned
edit_date	int(10) unsigned


2008-01-24 08:35:36

benutzer_schulung

 fwnr	int(10) unsigned
 stbnr	int(10) unsigned
 datum	int(10) unsigned
 bearbeiter_fwnr	int(10) unsigned
 bearbeiter_stbnr	int(10) unsigned
edit_date	int(10) unsigned



2008-01-24 08:35:36

bezirke

 bznr tinyint(3) unsigned
name varchar(255)
edit_date int(10) unsigned


2008-01-24 08:35:36

client

 fuellstelle int(10) unsigned
 laufnummer int(10) unsigned
os varchar(32)
edit_date int(10) unsigned



2008-01-24 08:35:36

dienstgrade

 nr int(10) unsigned
kurz varchar(20)
lang varchar(255)
edit_date int(10) unsigned


2008-01-24 08:35:36

einsatz

 time int(11) unsigned
 einsatzbereich_fwnr int(11) unsigned
typ enum('Einsatz','Uebung','Schulung','Diverses')
bezeichnung varchar(255)
bemerkung text
time_end int(10) unsigned
edit_date int(10) unsigned



2008-01-24 08:35:37

feuerwehren

 fwnr int(10) unsigned
fuellstelle int(10) unsigned
abschnitt tinyint(3) unsigned
name varchar(255)
bezirk tinyint(3) unsigned
plz int(10) unsigned
gemeinde varchar(255)
ortschaft varchar(255)
strasse varchar(255)
hausnummer int(10) unsigned
typ enum('Freiwillige Feuerwehr','Betriebsfeuerwehr')
edit_date int(10) unsigned







2008-01-24 08:35:37

fremdfflaschen

	fuellstelle	int(10) unsigned
	erstellung	int(10) unsigned
	eigentuemer	varchar(128)
	eig_nr	varchar(64)
	edit_date	int(10) unsigned




2008-01-24 08:35:37

fremdfflaschen_fuellung

	fremdfflaschen_fuellstelle	int(10) unsigned
	fremdfflaschen_erstellung	int(10) unsigned
	time	int(11) unsigned
	fuell_sitzung_time	int(10) unsigned
	kompressoren_fwnr	int(10) unsigned
	kompressoren_laufnummer	int(10) unsigned
	edit_date	int(10) unsigned


2008-01-24 08:35:37

fuell_sitzung

	time	int(11) unsigned
	kompressoren_fwnr	int(10) unsigned
	kompressoren_laufnummer	int(10) unsigned
	benutzer_fwnr	int(11) unsigned
	benutzer_stbnr	int(11) unsigned
	time_end	int(11) unsigned
	einsatz_time	int(10) unsigned
	einsatz_fwnr	int(10) unsigned
	betriebsstundenzaehler	float
	edit_date	int(11) unsigned


2008-01-24 08:35:37

fuellstelle

	fwnr	int(10) unsigned
	tpa_nr	varchar(32)
	edit_date	int(10) unsigned




2008-01-24 08:35:37

geraetetraeger

	fwnr	int(10) unsigned
	stbnr	int(10) unsigned
	atslehrgang	int(10) unsigned
	angelegt	int(10) unsigned
	anleger_fwnr	int(10) unsigned
	beendet	int(10) unsigned
	beender_fwnr	int(10) unsigned
	anleger_stbnr	int(10) unsigned
	beender_stbnr	int(10) unsigned
	edit_date	int(10) unsigned




2008-01-24 08:35:37

geraetetraeger_taetigkeiten

	fwnr	int(10) unsigned
	stbnr	int(10) unsigned
	datum	int(10) unsigned
	art	varchar(255)
	erfasser_fwnr	int(10) unsigned
	erfasst	int(10) unsigned
	assp_id	int(10) unsigned
	erfasser_stbnr	int(10) unsigned
	edit_date	int(10) unsigned




2008-01-24 08:35:37

geraetetraeger_untersuchung

	fwnr	int(10) unsigned
	stbnr	int(10) unsigned
	tauglich	char(1)
	dat_naechst	int(10) unsigned
	bearbeiter_fwnr	int(10) unsigned
	bearbeitet	int(10) unsigned
	dat_untersuch	int(10) unsigned
	arzt	varchar(255)
	bearbeiter_stbnr	int(10) unsigned
	edit_date	int(10) unsigned






2008-01-24 08:35:37

gruppen

	ersteller_fwnr	int(10) unsigned
	ersteller_stbnr	int(10) unsigned
	datum_erstellung	int(10) unsigned
	name	varchar(255)
	hinweise	varchar(1024)
	edit_date	int(10) unsigned



2008-01-24 08:35:37

gruppen_user

	gruppen_fwnr	int(10) unsigned
	gruppen_stbnr	int(10) unsigned
	gruppen_datum	int(10) unsigned
	user_fwnr	int(10) unsigned
	user_stbnr	int(10) unsigned
	typ	enum('verwalter','chef','mitglied')
	edit_date	int(10) unsigned

2008-01-24 08:35:37

kompressoren

	fwnr	int(10) unsigned
	laufnummer	int(10) unsigned
	seriennummer	varchar(255)
	fuellanschl_200	int(10) unsigned
	fuellanschl_300	int(10) unsigned
	fuellanschl_nieder	int(10) unsigned
	enddruck	int(10) unsigned
	hersteller	varchar(255)
	funkrufname	varchar(255)
	indienst	int(10) unsigned
	indienststeller_fwnr	int(10) unsigned
	ausdienst	int(10) unsigned
	ausdienststeller_fwnr	int(10) unsigned
	wartungsintervall	int(10) unsigned
	pruefintervall	int(10) unsigned
	tpa_pruefung	int(10) unsigned
	hochdruck_ausgang	int(10) unsigned
	hochdruck_eingang	int(10) unsigned
	quickfill	int(10) unsigned
	indienststeller_stbnr	int(10) unsigned
	ausdienststeller_stbnr	int(10) unsigned
	edit_date	int(10) unsigned




2008-01-24 08:35:37

kompressoren_maengel

	fwnr	int(10) unsigned
	laufnummer	int(10) unsigned
	mangel	varchar(255)
	loesung	varchar(255)
	feststeller_fwnr	int(10) unsigned
	datum_feststellung	int(10) unsigned
	beheber_fwnr	int(10) unsigned
	datum_behebung	int(10) unsigned
	zaehlerstand	float unsigned
	feststeller_stbnr	int(10) unsigned
	beheber_stbnr	int(10) unsigned
	beschadigungsgrad	tinyint(3) unsigned
	edit_date	int(10) unsigned






2008-01-24 08:35:37

kompressoren_pruefung

	fwnr	int(10) unsigned
	laufnummer	int(10) unsigned
	datum	int(10) unsigned
	pruefer	varchar(255)
	eigenpersonal_fwnr	int(10) unsigned
	tpa_pruefung	varchar(1)
	zaehlerstand	float unsigned
	eigenpersonal_stbnr	int(10) unsigned
	edit_date	int(11)




2008-01-24 08:35:37

nachrichten_gelesen

	person_fwnr	int(10) unsigned
	person_stbnr	int(10) unsigned
	ersteller_fwnr	int(10) unsigned
	ersteller_stbnr	int(10) unsigned
	erstell_datum	int(10) unsigned
	lese_datum	int(10) unsigned
	edit_date	int(10) unsigned




2008-01-24 08:35:37

nachrichten

kompressoren_bezirk	int(10) unsigned
kompressoren_abschnitt	int(10) unsigned
fuellstelle	int(10) unsigned
person_fwnr	int(10) unsigned
person_stbnr	int(10) unsigned
person_bezirk	int(10) unsigned
person_abschnitt	int(10) unsigned
gruppen_fwnr	int(10) unsigned
gruppen_stbnr	int(10) unsigned
gruppen_datum	int(10) unsigned
titel	varchar(255)
inhalt	text
dringlichkeit	tinyint(4)
 ersteller_fwnr	int(10) unsigned
 ersteller_stbnr	int(10) unsigned
 erstell_datum	int(10) unsigned
loescher_fwnr	int(10) unsigned
loescher_stbnr	int(10) unsigned
loesch_datum	int(10) unsigned
edit_date	int(10) unsigned





2008-01-24 08:35:37

kompressoren_wartung

 fwnr	int(10) unsigned
 laufnummer	int(10) unsigned
 datum	int(10) unsigned
eigenpersonal_fwnr	int(10) unsigned
luftqualitaet	varchar(1)
filterwechsel	varchar(1)
oelwechsel	varchar(1)
manometerkontrolle	varchar(1)
zaehlerstand	float unsigned
eigenpersonal_stbnr	int(10) unsigned
fremdpersonal	varchar(255)
edit_date	int(10) unsigned

2008-01-24 08:35:37

permission

 fwnr	int(10) unsigned
 stbnr	int(10) unsigned
 option	varchar(255)
value	enum('0','1')
 bereich	enum('global','local')
edit_date	int(10) unsigned





2008-01-24 08:35:37

person

 fwnr	int(10) unsigned
 stbnr	int(10) unsigned
vorname	varchar(255)
nachname	varchar(255)
stadt	varchar(255)
plz	int(10) unsigned
strasse	varchar(255)
hausnummer	int(10) unsigned
dienstgrad	int(10) unsigned
edit_date	int(10) unsigned






2008-01-24 08:35:37

person_kontakt

 fwnr	int(10) unsigned
 stbnr	int(10) unsigned
 typ	enum('mail','icq','msn','jabber','skype')
 value	varchar(255)
edit_date	int(10) unsigned




2008-01-24 08:35:37

person_telefon

 fwnr	int(10) unsigned
 stbnr	int(10) unsigned
typ	enum('handy','tel','fax')
 laenderkennnr	int(10) unsigned
 vorwahl	int(10) unsigned
 nummer	int(10) unsigned
edit_date	int(10) unsigned


2008-01-24 08:35:37

schulung

 datum	int(10) unsigned
vortragender	varchar(255)
 bearbeiter_fwnr	int(10) unsigned
 bearbeiter_stbnr	int(10) unsigned
ort	varchar(255)
inhalt	varchar(255)
edit_date	int(10) unsigned


2008-01-24 08:35:37

server

 name	varchar(255)
hostname	varchar(255)
ip	varchar(15)
priority	tinyint(4)
deleted	enum('0','1')
edit_date	int(11) unsigned


2008-01-24 08:35:37

server_details

 name	varchar(255)
ssh_port	int(10) unsigned
rsa	varchar(2048)
tunneld_port	int(10) unsigned
db_user	varchar(255)
db_password	varchar(255)
db_name	varchar(255)
edit_date	int(10) unsigned




2008-01-24 08:35:37

sync

 tb_name	varchar(255)
value	enum('1','0')
edit_date	int(11)



2008-01-24 08:35:37

sync_vorgang

 client_fwnr	int(10) unsigned
 client_laufnummer	int(10) unsigned
user_fwnr	int(10) unsigned
user_stbnr	int(10) unsigned
server	varchar(255)
 start_date	int(10) unsigned
end_date	int(10) unsigned
edit_date	int(10) unsigned

2008-01-24 08:35:37

verbindungskennung

 client_fwnr	int(10) unsigned
 client_laufnummer	int(10) unsigned
kennung	varchar(255)
edit_date	int(10) unsigned

2008-01-24 08:35:37