

Darstellung von Benutzerinformationen in einer Webapplikation  
(programmiert mit C# und ASP.NET)

**Modul T3200**

**Studienarbeit**

Studiengang Angewandte Informatik

an der Dualen Hochschule Baden-Württemberg Stuttgart Campus Horb

von

Jens-Christian Willmer

15.05.2012

Bearbeitungszeitraum	12 Wochen
Matrikelnummer, Kurs	8189582, TAI2009
Universität	DHBW Stuttgart Campus Horb
Betreuer der Universität	Schneider Ulrich

## Selbstständigkeitserklärung

Ich versichere hiermit, dass ich die vorliegende Arbeit mit dem Thema

„Darstellung von Benutzerinformationen in einer Webapplikation“

Selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

Böblingen, den \_\_\_\_\_

\_\_\_\_\_  
Jens-Christian Willmer

## Zusammenfassung

Diese Arbeit befasst sich mit der Erstellung einer Homepage, die dem angemeldeten Benutzer ermöglicht, sein E-Mail-Kontingent sowie sein verbleibendes Kontingent auf der privaten Dateiablage der Universität einzusehen.

Die Arbeit gliedert sich hierzu in zwei Phasen. Zu jeder Phase gibt es eine Ausarbeitung, wobei sich diese mit Phase zwei befasst. In diesem Teil der Arbeit wird der Schwerpunkt auf die praktische Umsetzung gelegt, wodurch der Grundlagenteil kleiner ausfällt als in der ersten Ausarbeitung.

Die Grundlagen umfassen hierbei den Microsoft Exchange Server, das Linux Rechtesystem sowie die Verbindung zweier Peers über das Secure Shell Protokoll. Das Secure Shell Protokoll wird benötigt, um eine Verbindung zwischen der Anwendung und dem Linux Server herzustellen.

Die Umsetzung des Projekts wird unter Verwendung von Visual Studio 2010 und ASP.NET MVC 3 angegangen. In dieser Ausarbeitung wird anhand von Quellcode beschrieben wie die Funktionalität der Anwendung umgesetzt wurde und welche Hürden dafür zu nehmen waren. Des Weiteren wird dem Leser im Unterkapitel Oberflächendesign das Aussehen der Oberfläche anhand von Momentaufnahmen vorgeführt und erklärt, was den Entwickler dazu bewogen hat es so und nicht anders umzusetzen.

## Inhaltsverzeichnis

Selbstständigkeitserklärung .....	I
Zusammenfassung .....	II
Inhaltsverzeichnis .....	III
1. Einleitung .....	1
1.1 Schilderung des Umfelds .....	1
1.2 Problemstellung .....	1
2. Grundlagen .....	3
2.1 Microsoft Exchange Server .....	3
2.2 Linux Rechtesystem .....	3
2.3 Secure Shell .....	4
3. Umsetzung .....	6
3.1 Benutzersuche .....	6
3.2 Auslesen der Mailquota .....	8
3.3 Auslesen des Speicherquotas .....	9
3.4 Oberflächendesign .....	10
4. Zusammenfassung .....	16
4.1 Ausgangssituation .....	16
4.2 Ziel .....	16
4.3 Ausblick .....	16
5. Administrationsanleitung .....	18
5.1 Konfigurationsdatei .....	18
5.1.1 Active Directory .....	18
5.1.2 appSettings .....	18
5.1.3 Zugangsbeschränkungen .....	19
5.1.4 Bibliotheken .....	20

---

5.1.5 Authentifizierung .....	20
5.2 CSS .....	21
Abkürzungsverzeichnis.....	V
Literaturverzeichnis .....	VI
Abbildungsverzeichnis.....	VII

## 1. Einleitung

### 1.1 Schilderung des Umfelds

Angestellte und Studierende der Dualen Hochschule Baden-Württemberg (DHBW) Stuttgart Campus Horb besitzen jeweils ein eigenes E-Mail-Postfach sowie einen privaten Speicherbereich auf dem Dateisystem der Universität.

Das E-Mail-Postfach wird von einem Exchange Server gehostet, welcher im Netzwerk der Universität steht. Der Fileserver, der sich ebenfalls im Netz der Hochschule befindet, arbeitet mit einer Unix-Plattform. Die Zugänge zu den Servern werden über ein Active Directory geregelt, welches die Anmeldedaten der Anwender enthält und bei korrekter Authentifizierung den Zugriff auf die Ressourcen erlaubt.

Sowohl das E-Mail-Postfach als auch das Benutzerverzeichnis unterliegen hierbei einer Größenbeschränkung. Eine einfache Möglichkeit den verbleibenden Speicherplatz einzusehen gibt es nicht. Es ist jedoch möglich, den verbleibenden Speicher auf Grundlage des verwendeten Speichers und der im Intranet ausgeschriebene Maximalgröße zu errechnen.

### 1.2 Problemstellung

Aufgrund des großen Aufwands, der für den Anwender entsteht, wenn er eine Übersicht über seinen verbleibenden Speicher erlangen möchte, ist dafür die Entwicklung einer Übersichtsseite (Dashboard) angedacht. Das Dashboard dient der Aufbereitung und dem Aufruf der Informationen bezüglich des jeweiligen Speichers eines Postfachs.

Des Weiteren soll diese Seite noch eine Suche enthalten, mit welcher Benutzer auf Grundlage von Vor- und Nachname einer Person deren E-Mail-Adresse suchen können. Das Dashboard soll aufgrund der existierenden Windows Server in ASP.NET und C# geschrieben werden.

Für die Anzeige der nutzerbezogenen Daten muss sich der Benutzer anmelden. Dies geschieht über einen Abgleich mit dem Active Directory. Daraufhin sendet die zu entwickelnde Webapplikation eine Anfrage an den Exchange Server, um die aktuelle und maximale Größe des Anwenderpostfachs zu erfahren.

Eine zweite Anfrage versendet das Programm an den Unix Server, um auch hiervon die Größenangaben zu erhalten. Die Kommunikation zwischen Unix Server und dem Dashboard soll hierbei über eine SSH-Verbindung stattfinden. Die von den beiden Anfragen erhaltenen Daten werden daraufhin aufbereitet und dem Anwender im Dashboard angezeigt.

Die Suche nach E-Mail-Adressen soll über das Active Directory realisiert werden. Hierbei ist angedacht, die Attribute der angelegten Personen zu durchsuchen, um deren kompletten Namen sowie die E-Mail-Adresse zu finden. Werden auf Grund von Suchanfragen, die nur einen Teil des Namens enthalten, mehrere Übereinstimmungen gefunden, so werden die ersten 15 Treffer ausgegeben.

Weiterhin wird für die Anzeige der Daten auf dem Dashboard ein Design benötigt, welches mit dem bereits vorhandenen Design des Intranets korreliert. Hierzu wird ein Designkonzept benötigt.

Die beschriebene Problemstellung ist Inhalt der Seminararbeit. Sie erstreckt sich über zwei Semester, wobei für jedes Semester eine Ausarbeitung anzufertigen ist.

Diese Arbeit stellt den zweiten Teil der Ausarbeitung dar, wodurch der Schwerpunkt auf die praktische Umsetzung gelegt wird. Sie befasst sich im Grundlagenteil mit dem Aufbau des Microsoft Exchange Servers sowie mit den Linux Rechtesystem und der Verbindung zweier Peers via Secure Shell (SSH). In Kapitel **Fehler! Textmarke nicht definiert.** wird anschließend das praktische Vorgehen anhand von ausgewähltem Quellcode und Bildern der Oberfläche beschrieben.

## 2. Grundlagen

### 2.1 Microsoft Exchange Server

Der Microsoft Exchange Server ist eine spezielle Anwendung von Microsoft für die Zusammenarbeit und den Nachrichtenversand. Die erste Version kam 1996 auf den Markt und wurde seitdem kontinuierlich verbessert. Seit 2011 befindet sich der Exchange Server in der Version 14.2.1 Seine Hauptaufgabe ist das Senden und Empfangen von Mails, des Weiteren ist es aber auch möglich Terminvereinbarungen, Kalendereinträge oder Besprechungen zu verschicken und zu planen.

Um sich mit dem Exchange Server zu verbinden stehen unterschiedliche Netzwerkprotokolle zur Verfügung. Daraus ist das für das Einsatzgebiet am besten geeignete Protokoll auszuwählen. Für die Verbindung zum Exchange Server über den Browser bietet Microsoft eine Weboberfläche namens Outlook Web Access an, wodurch lediglich ein moderner Browser benötigt wird.

Des Weiteren stellt Microsoft für die Clientseitige Verbindung Microsoft Outlook bereit, eine Software die auf allen Windows Systemen läuft und dem Anwender den gesamten Funktionsumfang des Exchange Servers über die Anwendung bereitstellt.<sup>2</sup>

Für die Skalierung des Exchange Servers können mehrere Instanzen parallel betrieben und zu einer verbunden werden. Dadurch kann ein beliebig großes System entstehen, welches die Anfragen der Benutzer verarbeitet und umsetzt.

### 2.2 Linux Rechtesystem

Das Betriebssystem Linux ist für den Mehrbenutzerbetrieb ausgelegt und besitzt hierfür ein Rechtesystem. Dieses erlaubt das Setzen von Anwender- und Gruppenberechtigungen. Das Rechtesystem besitzt hierfür drei grundlegende Rechte, welche sowohl auf Ordner als auch auf Dateien angewendet werden können. Zum einen das Leserecht (r: read) welches dem Anwender erlaubt Dateien zu lesen oder den Inhalt von Ordnern einzusehen. Zum Andern das Recht zu schreiben (w: write), wodurch Dateien verändert oder Ordnern weitere Daten hinzugefügt werden dürfen sowie das Recht zum Ausführen (x: execute). Bei Dateien erlaubt dieses dritte Recht das Ausführen der Datei mit einem Programm. Bei

---

<sup>1</sup> Siehe hierzu Carius F. (2012)

<sup>2</sup> Vgl. Robichaux, P. (1999) S.1ff



Verzeichnissen beeinflusst es lediglich das wechseln in das Verzeichnis. Dadurch ist es möglich Dateien oder Ordner nur für bestimmte Personen oder Personengruppen frei zu geben.

Mit dem Kommando `ls -l` kann der Anwender in der Kommandozeile von Linux die Berechtigungen des aktuellen Verzeichnisses einsehen. Die Ausgabe dieses Kommandos ist in Abbildung 1 zu sehen. Die darunter liegende Beschreibung der einzelnen Rechte gehört allerdings nicht dazu und soll lediglich dem Leser als zusätzliche Erläuterung dienen.<sup>3</sup>

```

-rw- r-- ---   2 wentland mitarb 2342 Apr  1 23:42 punkteliste
- - - - -
| | | | |       Owner      Group   Size  Timestamp   Name
| | | | |
| | | | |_     Rechte der Anderen    (keine)
| | | |_|     Rechte der Gruppe    (lesen)
| | |_|       Rechte des Besitzers (lesen, schreiben)
|_|          Dateityp                (- = Datei, d = Verzeichnis, l = link, ... )

```

**Abbildung 1:** *Dateiberechtigungen*<sup>4</sup>

## 2.3 Secure Shell

Die Secure Shell ist ein Netzwerkprotokoll welches eine verschlüsselte Verbindung zwischen zwei Rechner herstellt. Üblicherweise wird das Protokoll dazu verwendet, um von einem lokalen Computer aus eine Fernwartung auf einem Server durchzuführen. Hierzu öffnet sich nach der Verbindung via SSH ein Terminalfenster, über welches Befehle an die Kommandozeile des Servers gesendet werden können. Die Ausgaben des Servers werden wiederum auf der Gegenseite angezeigt, sodass eine Interaktion mit dem Server möglich wird.

Beim Verbindungsaufbau authentifiziert sich der Client via Passwort oder durch ein Zertifikat beim Server. Geschieht die Authentifizierung über ein Zertifikat, so besitzt der Server das öffentliche Zertifikat und der Client das Private, wodurch der Server dem Client den Verschlüsselungsalgorithmus für die aktuelle Sitzung sowie den Schlüssel dafür mitteilen kann. Dadurch wird gewährleistet, dass die Nachricht von

<sup>3</sup> Siehe dazu Wentland , W. & Simon , C. (2006) Chapter 13

<sup>4</sup> Aus Wentland , W. & Simon , C. (2006) Chapter 13

einem dritten ohne den dazugehörigen privaten Schlüssel nicht entschlüsselt werden kann. Dadurch ermöglicht SSH eine private Kommunikation zwischen zwei Peers in einem offenen Netz.

Für die Kommunikation zwischen Client und Server wird standardmäßig der Port 22 verwendet. Der Port kann jedoch in der Konfiguration geändert werden um es Angreifern zu erschweren eine SSH-Verbindung zu entdecken. Des Weiteren ist es mit SSH möglich eine Portweiterleitung einzurichten. Dadurch werden beliebige Ports über das SSH Protokoll getunnelt, wodurch Verbindungen von anderen Programmen ebenfalls über den verschlüsselten Kanal transportiert werden können. Als Beispiel könnte eine Software zur Sprachübertragung über SSH getunnelt werden, um zu verhindern, dass andere Benutzer im Netzwerk den Gesprächsverlauf mitverfolgen können.<sup>5</sup>

---

<sup>5</sup> Vgl. Daniel J. u.a. (2005) S1ff

### 3. Umsetzung

In diesem Kapitel wird, mit Bezug auf den Grundlagenteil und der ersten Ausarbeitung, die Umsetzung der Anwendung erklärt.

Einzige rückwirkende Änderung, welche sich bei der weiteren Implementierung des Systems ergeben hat, ist, dass die Daten, welche aus dem Active Directory gezogen werden, nichtmehr in einer Datenbank zwischengespeichert werden sondern bei Bedarf neu abgerufen werden. Damit wird umgangen, eine Zweite Datenbank zum Active Directory konsistent zu halten. Zudem können entstehende Größenänderungen am Speichervolumen des Anwenders ohne Versatz angezeigt werden.

Bei der Authentifizierung des Anwenders wird außerdem ein Cookie erstellt, wodurch der Benutzer über einen vordefinierten Zeitraum angemeldet bleibt. Diese Zeitspanne kann in der Konfigurationsdatei des Systems festgelegt werden.

#### 3.1 Benutzersuche

Die Benutzersuche bietet dem Anwender eine Möglichkeit, über den Namen eines Bekannten Anwenders die entsprechende Mailadresse in Erfahrung zu bringen. Hierzu wurde eine Suchmaske auf der Oberfläche integriert. Zu beachten ist, dass diese die Suche nur startet, wenn der eingegebene Suchterm eine mindestlänge beinhaltet. Dies geschieht als Präventionsmaßnahme um das wahllose Auslesen des Active Directories zu verhindern (Abbildung 2).

```
6 public class SearchModel
7 {
8     public SearchModel()
9     {
10         Searchvalue = string.Empty;
11         Searchresult = new List<SearchResultModel>();
12     }
13
14     [Required(AllowEmptyStrings = false, ErrorMessage = "Bitte geben sie eine Person ein, die sie suchen möchten.")]
15     [DataType(DataType.Text)]
16     [StringLength(50, ErrorMessage = "Die Mindestlänge der Suchanfrage beträgt 4 Zeichen.", MinimumLength = 4)]
17     [Display(Name = "Zu suchende Person")]
18     public string Searchvalue { get; set; }
19
20     public List<SearchResultModel> Searchresult { get; set; }
21 }
```

Abbildung 2: Modell der Suchmaske

Für die Integration ist es erforderlich, das Active Directory abzufragen. Das hierfür benötigte Vorgehen wurde in der ersten Ausarbeitung bereits erklärt. Die Anfrage an das Active Directory ist so aufgebaut, dass sie die Mailadresse und den Namen jeder Person liefert, welche den gesendeten Suchterm enthält. Die Ergebnisliste wird hierbei schon bei der Suchanfrage auf 15 reduziert, um das Active Directory nicht unnötig zu belasten (Abbildung 3).

```
#region get search result
SearchResultCollection collection;
using (var root = new DirectoryEntry(WebConfigHelpers.GetActiveDirectoryConnectionString()))
{
    using (var directorySearch = new DirectorySearcher(root))
    {
        directorySearch.Filter = string.Format("displayName={0}*", model.Searchvalue);
        directorySearch.PropertiesToLoad.Add("displayName");
        directorySearch.PropertiesToLoad.Add("mail");
        directorySearch.SizeLimit = 15;
        directorySearch.Sort = new SortOption("displayName", SortDirection.Ascending);
        collection = directorySearch.FindAll();
    }
}
#endregion
```

**Abbildung 3:** Active Directory Suche

Die gefundenen Ergebnisse werden daraufhin geparkt, in Objekten abgelegt (Abbildung 4) und in einer Liste zusammengefasst. Diese Liste wird anschließend an die Oberfläche übergeben, welche die Liste als Resultat darstellt.

```
23 | public class SearchResultModel
24 | {
25 |     public string User { get; set; }
26 |     [DataType(DataType.EmailAddress)]
27 |     public string Mail { get; set; }
28 | }
29 |
```

**Abbildung 4:** Objekt eines Suchergebnisses

### 3.2 Auslesen der Mailquota

Das Auslesen des Mailquotas geschieht nicht wie vermutet über die Abfrage des Exchange Servers, sondern ebenfalls über das Active Directory. In diesem ist angegeben, ob ein Standardwert für die Größe des Mailspeichers gesetzt ist oder eine individuelle Größe verwendet wird.

Diese Angaben werden mit der in der ersten Ausarbeitung vorgestellten Methode abgefragt und für die weitere Verarbeitung in einem User-Modell vorgehalten.

Informationen über die aktuelle Größe des Mailpostfaches und die verwendeten Ordner können im Gegensatz zu den Quotaangaben nicht aus dem Active Directory ausgelesen werden. Dazu wurde der Exchange Server über den Exchange Web Services (EWS) angesprochen. Dazu wurde, wie aus den Active Directory Anfragen bekannt, das Administratorkonto verwendet. Dies wird aus zwei Gründen wiederverwendet: Erstens wird durch die Wiederverwendung des Kontos das Anlegen eines zweiten verhindert und zweitens ist durch die Berechtigungseinschränkungen des Kontos eine hohe Sicherheit gegen unsachgemäße Nutzung vorhanden.

Mit dem Administratorkonto ist es möglich, einen beliebigen Anwender zu imitieren. Diese Eigenschaft wird benutzt, um ohne Suche auf die Kontoeigenschaften des gewünschten Anwenders zuzugreifen. Durch eine Iteration werden daraufhin die vorhandenen Ordner der obersten Ebene, die Größe der Ordner sowie die Gesamtgröße des Postfachs festgestellt (Abbildung 5). Die Funktion für die Iteration der Ordner des Postfachs wurde hierbei in eine Unterfunktion ausgelagert.

```
// Impersonate user
exchangeService.ImpersonatedUserId = new ImpersonatedUserId(ConnectingIdType.SmtpAddress, mailAddress);
...
user.MailboxFolders = ExchangeHelpers.GetMailboxFolders(exchangeService);
double space = 0;
user.MailboxFolders.ForEach(folder => space += folder.Foldersize);
user.MailSpaceInUse = Math.Round((space / 1024f), 2);
```

**Abbildung 5:** Impersonation und Postfachabfrage

### 3.3 Auslesen des Speicherquotas

Für das Auslesen des Speicherkontingents auf dem Dateisystem muss eine Anfrage an einen Unix Server gestellt werden. Dies geschieht über SSH, wobei eine SSH Bibliothek verwendet wurde<sup>6</sup>.

Hierfür wird als erstes der SSH-Client initialisiert und danach die passenden Befehle für die Abfrage der Speicherbegrenzung über eine Hilfsklasse abgerufen (Abbildung 6).

```
public static bool GetHarddriveQuota(ref UserDataModel user, string userName)
{
    try
    {
        string userFolder, userFiles, userQuota;

        using (var client = new SshClient(WebConfigHelpers.GetUnixServerAdresse(),
                                         WebConfigHelpers.GetUnixServiceAccountName(),
                                         WebConfigHelpers.GetUnixServiceAccountPassword()))
        {
            client.Connect();

            var userPath = client.RunCommand(ShellHelpers.GetUserPathCommand(userName)).Result.Replace("\n", "");
            userFolder = client.RunCommand(ShellHelpers.GetFoldersCommand(userName, userPath)).Result;
            userFiles = client.RunCommand(ShellHelpers.GetFilesCommand(userName, userPath)).Result;
            userQuota = client.RunCommand(ShellHelpers.GetUserQuotaCommand(userName)).Result;

            client.Disconnect();
        }

        user.DirectoryElements = ShellHelpers.ParseFiles(userFiles);
        user.DirectoryElements.AddRange(ShellHelpers.ParseFolders(userFolder));

        // Only the first Disk will be displayed in the view!
        user.DiskQuota = ShellHelpers.ParseUserQuota(userQuota);
    }
}
```

Abbildung 6: SSH Client Anfrage

Die Befehle bestehen hierbei aus vielen Einzelbefehlen, welche durch Pipes und Filter zusammengefügt sind (Abbildung 7). Dies bedeutet zwar einen hohen Erstaufwand beim Erstellen der Befehle, vermindert aber ein intensives Parsen der Resultate. Nach dem Parsen der Informationen die der Server zurückliefert, werden diese zu den bereits gesammelten Anwenderinformationen hinzugefügt.

<sup>6</sup> Zu finden unter SSH.NET Library (2012)

```
/// <summary>
/// Returns a command that can be used to get the folders and there size of a directory.
/// </summary>
/// <param name="owner">The owner of the folders.</param>
/// <param name="path">The path that will be queried.</param>
/// <returns>The shell command.</returns>
public static string GetFoldersCommand(string owner, string path)
{
    return
        string.Format(
            // lists all folders and there sizes passed from -> (lists all items in the dir {0} -> greps only the folders -> |
            "du -B 1024 --max-depth=0 `ls -o -U {0} | grep '^d' | awk 'BEGIN{{FS=\" \"}} {{if($3 == \"{1}\")} {{print \"{2}/\"
            path, owner, path);
}
```

**Abbildung 7:** Befehlskette die via SSH an den Server gesendet wird

### 3.4 Oberflächendesign

In diesem Kapitel wird die Gestaltung der Oberfläche vorgestellt und das Aussehen im Kontext des bestehenden Intranets gezeigt.

In Abbildung 8 ist das aktuelle Intranet der DHBW Horb zu erkennen. In dieses wurde das Dashboard integriert. Für die Nutzung des Dashboards ist leider eine zweite Anmeldung von Nöten, da es noch keine Implementierung für einen Single-Sign-On seitens des Intranets gibt. Des Weiteren fehlt in dieser Abbildung eine Verlinkung in der Navigation zum angezeigten Dashboard. Dies ist allerdings zu vernachlässigen, da es sich bei dieser Abbildung lediglich um eine statische Seite aus dem Intranet handelt, um das spätere Aussehen der Applikation in Kontext des Intranets zu visualisieren.

Die einzelnen Menüpunkte wurden hierbei in einem Menüband als Reiter angeordnet. Dadurch werden dem Anwender ein einheitliches Design und eine bekannte Navigation gewährleistet.

Das CSS der Applikation greift auf die CSS-Dateien des Intranets zurück, was bei einem Verschieben dieser, Änderungen an der Konfiguration der Applikation nach sich zieht. Durch den Einsatz der für die Applikation externen CSS-Dateien ergibt sich der Vorteil, dass Designänderungen die im Intranet vorgenommen werden auch automatisch auf die, in das Intranet eingebundene Anwendung, angewendet werden.

Mit Ausnahme von Ausrichtungsparametern, diese wurden in einer internen CSS-Datei abgelegt um vor externen Änderungen geschützt zu sein.

The screenshot displays the DHBW-Dashboard interface. At the top, there is a header with the DHBW logo and the text 'Stuttgart Campus Horb'. Below this, a navigation bar contains links for 'ALUMNITREFFEN 2010', 'ALUMNITREFFEN 2011', 'ZIMMERVERMITTLUNG', and 'INTRANET'. The breadcrumb trail indicates the current path: 'Pfad: Startseite - Home > Intranet > Dashboard'. The main heading is 'DHBW-DASHBOARD'. On the left, a vertical menu lists various services under the 'INTRANET' section, including 'Webmail', 'Zertifikate', 'Angewandte Informatik', 'Elektrotechnik', 'Informationstechnik', 'Maschinenbau', 'Mechatronik', 'Wirtschaftsingenieurwesen', 'Rechenzentrum' (with sub-items like 'Einwahl von extern', 'Benutzerordnung', etc.), 'Zimmervermittlung', 'Bildergalerie', and 'An-/Abmelden'. The central content area is titled 'Anmeldung' and includes sub-tabs for 'Übersicht', 'Personensuche', 'Einstellungen', and 'About'. A prompt asks the user to 'Geben Sie Benutzernamen und Kennwort ein.' The login form, titled 'Kontoinformationen', contains a 'Benutzername' field with the value 'a09001', a 'Kennwort' field with masked characters, a 'Speichern?' checkbox, and an 'Anmelden' button.

**Abbildung 8:** Dashboard im Kontext des Intranets

Meldet der Benutzer sich am Dashboard an, wird ein Cookie erstellt, mit dem er für 7 Tage angemeldet bleibt. Das Cookie ist verschlüsselt und enthält eine eindeutige ID sowie seinen Anmeldenamen. So weiß das System welche Daten es für die Anzeige des Dashboards abrufen muss. Nach der Anmeldung landet der Benutzer automatisch auf der Übersichtsseite (Abbildung 9). Diese fragt im Hintergrund alle in der Arbeit beschriebenen Server ab und präsentiert dem Anwender eine mit Balken- und Kuchendiagrammen aufbereitete Ansicht über die ihm verbleibenden und im Einsatz befindlichen Systemressourcen.

Für die Visualisierung der Diagramme wurde eine JavaScript Bibliothek namens Highcharts JS verwendet. Sie erhält ihre Daten über ein vom Server bereitgestelltes JSON-Objekt.<sup>7</sup> Dies bedeutet, dass der Anwender über JavaScript verfügen muss, um die Informationen angezeigt zu bekommen. Highcharts JS arbeitet bei der Anzeige der Daten mit Effekten, welche das Wachstum des Speicherverbrauchs visualisieren und dem Nutzer die Interaktion mit der Oberfläche erlauben. Damit ist

<sup>7</sup> Highchart JS (2012)



es dem Anwender beim Überfahren der Balkendiagramme möglich, genauere Informationen über seinen Ressourcenverbrauch zu erlangen und die verschiedenen Quotalevel, welche als rote Markierungen in den Balkendiagrammen angezeigt werden, aus zu lesen.

Die Anzeige für die Anzahl der Festplatten-Dateien wird dem Benutzer nur angezeigt wenn im Linux Server hierfür auch ein Grenzwert hinterlegt wurde, ansonsten fällt diese Angabe weg.

Die Kuchendiagramme bauen sich, ähnlich wie die Balkendiagramme, zur Ladezeit der Seite mit einer Animation auf. Durch einen Klick auf die einzelnen Kuchenstücke hebt sich der angeklickte Teil von den anderen Kuchenstücken ab. In der Standardanzeige wird der freie Speicherplatz beziehungsweise das Verzeichnis der obersten Ebene hervorgehoben.

Bei der Laufwerksverwendung wird nur das Verhältnis zwischen den Ordnergrößen und den Dateien auf oberster Ebene angezeigt. Der Grund dafür liegt darin, dass die Anwender auch Dateien an öffentlich zugänglichen Orten abspeichern können und es eine große Dateiabfrage für den Linux Server bedeuten würde alle Verzeichnisse des Dateisystems abzufragen. Deshalb wurde die Abfrage nur auf das Anwenderverzeichnis begrenzt, wodurch aber keine genaue Aussage über die wirkliche Größe der vom Anwender abgespeicherten Daten getroffen werden kann. Diese Information ist nur aus dem darüber liegenden Balkendiagramm ersichtlich.

Pfad: Startseite - Home &gt; Intranet &gt; Dashboard

**DHBW-DASHBOARD**Willkommen, **Fabian Burger** [ [Abmelden](#) ]**Übersicht**[Übersicht](#)[Personensuche](#)[Einstellungen](#)[About](#)

## E-Mail-Speicher



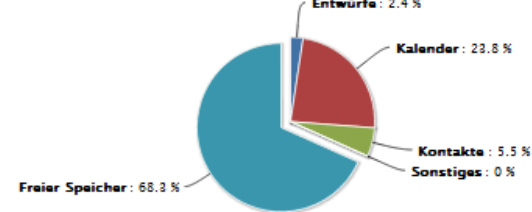
## Festplatten-Speicher



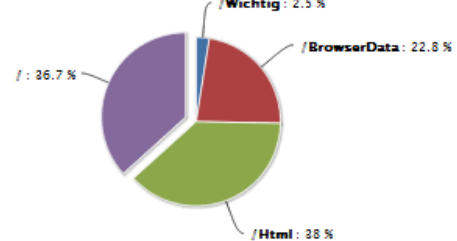
## Festplatten-Dateien



## Postfachverwendung



## Laufwerksverwendung

**Abbildung 9:** *Übersichtsseite des Dashboards*

Unter dem Reiter Personensuche befindet sich die in Kapitel 3.1 beschriebene Suche. Wie in Abbildung 10 gut zu erkennen ist, listet die Suche jegliche Namen auf, die den angegebenen Suchterm enthalten. Leider gab es in der Testumgebung nicht genügend Personen, um die Suchabfragebegrenzung zu demonstrieren. Diese wurde jedoch mit der Einstellung maximal 3 Personen aufzulisten getestet. Werden zu kurze Suchwörter eingegeben oder keine Ergebnisse gefunden, liefert die Suche eine Fehlermeldung mit hilfreichen Informationen um den Fehler bei der nächsten Suche zu vermeiden.

Pfad: Startseite - Home > Intranet > Dashboard

## DHBW-DASHBOARD

Willkommen, **Fabian Burger** [ [Abmelden](#) ]

**Personensuche**

Übersicht

Personensuche

Einstellungen

About

Suchen sie nach E-Mail-Adressen von Personen die sie kennen.

Suchanfragen benötigen eine Minimallänge von 4 Buchstaben.

### Suche

Zu suchende Person

### Ergebnisse

Name	E-Mail-Adresse
Abiata Heinrich	<a href="mailto:habiata@ba-horb.de">habiata@ba-horb.de</a>
Burger Fabian	<a href="mailto:a09001@ba-horb.de">a09001@ba-horb.de</a>
Ganter Fabian	<a href="#">Keine Adresse gefunden!</a>
Malabiano Heno	<a href="mailto:hmalabiano@ba-horb.de">hmalabiano@ba-horb.de</a>

**Abbildung 10:** Dashboard Personensuche

In Abbildung 11 ist der Reiter Einstellungen zu sehen. Über diesen Reiter lässt sich das Benutzerpasswort im Active Directory ändern. Um vorzubeugen, dass mit einem geklauten Cookie das Passwort des Anwenders geändert werden kann, muss dieser hierfür sein Passwort erneut eingeben. Des Weiteren muss er sein neues Passwort zweimal eingeben, um sicherzugehen das er sich nicht vertippt hat und sich selbst aus seinem Benutzerkonto ausschließt.

Pfad: Startseite - Home &gt; Intranet &gt; Dashboard

**DHBW-DASHBOARD**Willkommen, **Fabian Burger** [ [Abmelden](#) ]**Einstellungen**

Übersicht

Personensuche

Einstellungen

About

Ändern Sie das Kennwort mit dem unten stehenden Formular.

Neue Kennwörter müssen mindestens 7 Zeichen lang sein.

**Kontoinformationen**

Aktuelles Kennwort

Neues Kennwort

Neues Kennwort bestätigen

**Kennwort ändern**

**Abbildung 11: Dashboard Einstellungen**

In der letzten Abbildung (Abbildung 12) ist die Sicht auf die About-Seite abgebildet. In dieser gibt es einen kleinen Abstract zum Projekt sowie die Kontaktinformationen zum Ersteller und dessen, von der Universität bereitgestellten, Betreuer. So haben die Anwender die Möglichkeit, etwas über den Ursprung des Projektes zu erfahren und wissen, an wen sie sich wenden können, wenn Probleme mit dem System auftreten.

Pfad: Startseite - Home &gt; Intranet &gt; Dashboard

**DHBW-DASHBOARD**Willkommen, **Fabian Burger** [ [Abmelden](#) ]**About**

Übersicht

Personensuche

Einstellungen

About

Betreuer:

**Ulrich Schneider** **Über das Projekt**

Das Dashboard ist das Resultat einer Studienarbeit, die im Wintersemester 2011/2012 geschrieben wurde. Das Ziel der Arbeit war es, dem Anwender die Authentifizierung gegenüber dem vorhandenen Active Directory zu ermöglichen, um ihm daraufhin Informationen über das Mail-Quota des Exchange-Servers sowie das Benutzer-Quota des Unix-Dateiservers anzuzeigen.

Des Weiteren sollte es dem angemeldeten Benutzer ermöglicht werden, nach E-Mail-Adressen, von im Active Directory gepflegten Studenten und Dozenten, zu suchen.

Das Dashboard sollte in C# geschrieben und als ASP.NET-Anwendung betrieben werden. Hierzu waren außer guten Kenntnissen in C# und ASP.NET auch Kenntnisse im Umgang mit dem Active Directory, dem Exchange Server sowie der Verbindungsaufbau zu einem Server über SSH von Nöten.

Die Arbeit wurde vom Studenten durch die Anzeige und Aufbereitung der Postfach- beziehungsweise Dateisystemgröße sowie um einer Passwort-Ändern-Funktion erweitert. Für die Anpassung des Dashboards an das Design des Intranets der DHBW-Stuttgart Campus Horb wurde auf die im Intranet verwendeten CSS-Dateien verlinkt und auf eigene CSS-Klassen und -Tags weitestgehend verzichtet.

Die Daten, welche das Dashboard für den Anwender bereitstellt, werden durch die Benutzung eines Dienstkontos, welches lediglich Leserechte besitzt, bezogen. Dieses setzt die Anfragen auf das ActiveDirectory, den Exchange Server sowie den Dateiserver ab. Der Benutzername für die Abfragen wird aus dem Session-Cookie bezogen, welches bei der Anmeldung erzeugt und auf dem Client-Rechner, verschlüsselt, abgelegt wird. Es ist maximal 7 Tage gültig, danach muss sich der Anwender erneut authentifizieren. Durch die Verwendung der Cookie-Informationen wird das Zwischenspeichern des Passworts vermieden und dadurch die Sicherheit erhöht. Um das Abfangen des Cookies zu erschweren, läuft die komplette Anwendung über SSL.

**Abbildung 12: Dashboard About-Seite**

## 4. Zusammenfassung

### 4.1 Ausgangssituation

An der DHBW Stuttgart Campus Horb gibt es für jeden Studierenden und Angestellten eine private Dateifreigabe, auf welcher die Benutzer ihre Daten ablegen können. Dieses, sowie ihr von der DHBW gestelltes E-Mailpostfach, unterliegen hierbei einer Maximalgröße, die nicht überschritten werden darf.

Um den Anwendern eine einfache Möglichkeit zu bieten, den verbleibenden Platz in ihrem Postfach und ihrer Dateifreigabe einzusehen, wird eine Webseite benötigt, die diese Informationen anzeigt und aufbereitet darstellt.

Diese zu entwickeln und mit einer E-Mail-Adressen-Suche zu erweitern, ist das Ziel der aus zwei Teilen bestehenden Studienarbeit.

### 4.2 Ziel

Ziel der zweiten Ausarbeitung war es, unter Miteinbeziehung der theoretischen Grundlagen von Ausarbeitung eins, die im letzten Semester begonnene Software fertigzustellen. Hierzu wurden weitere Grundlagen in den Gebieten des Microsoft Exchange Servers benötigt sowie Informationen über die Dateisystemverwaltung von Linux und die Verbindung zwischen zwei Peers über das SSH Protokoll.

Darüber hinaus hatte die zweite Ausarbeitung das Ziel, die Vorgehensweise und die Überlegungen hinter der Konzeption des Systems zu verdeutlichen und die getroffenen Designentscheidungen zu begründen.

### 4.3 Ausblick

Als nächster Schritt könnte der Informationsgehalt des Dashboards durch die Anbindung eines weiteren Servers gesteigert werden. In Frage käme hierfür der in der Universität vorhandene Server, welcher das Druckkontingent der Studierenden verwaltet. Dadurch müssten die Studenten sich nicht extra auf dem vorhandenen Terminalserver der Universität einwählen um ihr Druckkontingent in Erfahrung zu bringen. Vor allem der Umstand, dass die Studenten sich von extern erst via VPN einwählen müssen, um eine Verbindung zum Terminal Server aufzubauen, macht eine aus dem Intranet abrufbare Anzeige des verbleibenden Kontingents attraktiv.

Des Weiteren ist darüber nach zu denken, ob es Sinn macht, dass die Inhalte des Dashboards nur angezeigt werden, wenn alle Serverabfragen erfolgreich verlaufen sind. Durch dieses Verhalten kann der Ausfall eines einzigen Servers das ganze Dashboard lahm legen, obwohl vielleicht gar nicht alle Serverabfragen benötigt werden um Teilinformationen anzeigen zu können.

Als dritte Verbesserung wäre eine Neukonzeption der Datenabfrage auf dem Linux Server eine gute Idee. Zum einen sollte anstelle des Passworts, welches beim Verbindungsaufbau via SSH zur Anmeldung benutzt wird, ein Zertifikat verwendet werden. Zum anderen muss darüber nachgedacht werden, wie umgangen werden kann, dass die Anwendung Administratorrechte benötigt, um die Quota des Servers abzufragen.

## 5. Administrationsanleitung

### 5.1 Konfigurationsdatei

Alle, für den Betrieb wichtigen Einstellungen werden in der Web.config-Datei festgelegt. Ein zusammengekürzter Auszug dieser Datei ist in Abbildung 15 und Abbildung 16 zu sehen. Im Folgenden werden anhand der Zeilennummern der beiden Abbildungen die Einstellmöglichkeiten für den Administrator besprochen.

#### 5.1.1 Active Directory

Anfangen in Zeile 4 kann der Administrator die Adresse des Active Directorys verändern, welches auf die Organisationseinheit (OU) zeigt, unter welcher die Anwender angelegt sind (Abbildung 13). Diese Adresse wird in Zeile 56 dazu benutzt, dem Active Directory Mitgliedschaftsdienst den Ort des Active Directorys bekannt zu geben. Des Weiteren wird in Zeile 51 dafür gesorgt, dass das Active Directory als Standard Mitgliedschaftsdienste verwendet wird.

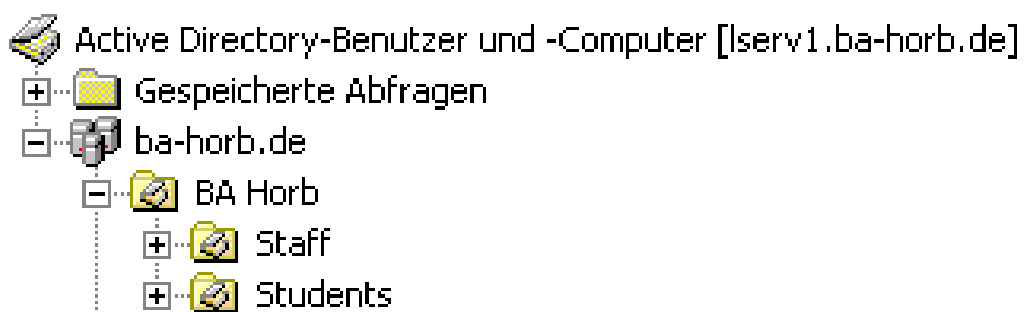


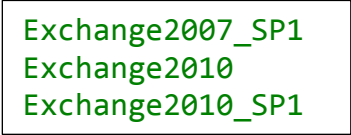
Abbildung 13: Active Directory Strukturausschnitt

#### 5.1.2 appSettings

Zwischen Zeile 6 und 19 befinden sich Schlüsselpaare, die von der Anwendung verwendet werden. Wichtig ist, dass die Schlüssel auf keinen Fall verändert werden dürfen, denn über die Namen der Schlüssel werden die hinterlegten Werte vom System gefunden. Die für den Administrator interessanten Einstellungen befinden sich in Zeile 10-18.

In Zeile 10 wird die Version des Exchange Servers angegeben. Erlaubte Werte sind Abbildung 14 zu entnehmen.

Darunter befinden sich der Anmeldename und das Passwort für den Exchange Servers. Das Passwort wird gleichzeitig für das Active Directory verwendet. Es ist hierbei darauf zu achten, dass das zu verwendende Konto lediglich Leserechte im Exchange Server sowie im Active Directory benötigt. Zusätzlich muss sichergestellt werden, dass über das Konto die Funktion „impersonation“ genutzt werden kann.<sup>8</sup>



```
Exchange2007_SP1
Exchange2010
Exchange2010_SP1
```

**Abbildung 14:** Exchange Versionsangaben

In Zeile 15 ist die Adresse des Exchange Web Services anzugeben. Sie lautet standardmäßig: „<https://servername/EWS/Exchange.asmx>“.

Darunter befinden sich die Anmeldedaten für den Linux Server. Hier wird ein Konto benötigt, das die Quotaeigenschaften der einzelnen Anwender auslesen kann. In der Regel funktioniert dies nur über das Administratorkonto.

Abschließend (Zeile 18) wird die Adresse des Linux Servers definiert. Es hat sich bewährt, hierfür ein Fully Qualified Domain Name (FQD) zu benutzen. Dadurch wird keine neue Konfiguration des Dashboards nötig, falls sich die IP-Adresse des Servers ändern sollte.

### 5.1.3 Zugangsbeschränkungen

Von Zeile 20 bis 23 erstreckt sich eine Definition, die es Anwendern, welche sich noch nicht am System angemeldet haben, erlaubt, Daten aus verschiedenen Ordnern einzusehen. Dies ist wichtig, damit die Browser von nicht angemeldeten Anwendern auf die zur Anzeige benötigten Layoutdateien und Skripte zugreifen können. Deshalb sollten diese Angaben nicht verändert werden. Das Gegenstück dazu befindet sich in Zeile 48-50. Über diese Angabe wird den angemeldeten Benutzern Zugriff auf alle Verzeichnisse gewährt, ausgenommen der Web.config-Datei.

---

<sup>8</sup> Configuring Exchange Impersonation: [http://msdn.microsoft.com/en-us/library/bb204095\(EXCHG.80\).aspx](http://msdn.microsoft.com/en-us/library/bb204095(EXCHG.80).aspx)



```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <configuration>
3   <connectionStrings>
4     <add name="ADConnectionString" connectionString="LDAP://lerv1.ba-horb.de/OU=BA_Horb,DC=ba-horb,DC=de" />
5   </connectionStrings>
6   <appSettings>
7     <add key="webpages:Version" value="1.0.0.0" />
8     <add key="ClientValidationEnabled" value="true" />
9     <add key="UnobtrusiveJavaScriptEnabled" value="true" />
10    <add key="ExchangeVersion" value="Exchange2007_SP1" />
11    <!-- ExchangeServiceAccountName is also the ActiveDirectory viewadmin username -->
12    <add key="ExchangeServiceAccountName" value="viewadmin" />
13    <!-- ExchangeServiceAccountPassword is also the ActiveDirectory viewadmin password -->
14    <add key="ExchangeServiceAccountPassword" value="asp.net" />
15    <add key="ExchangeWebServiceUrl" value="https://exmail7/EWS/Exchange.asmx" />
16    <add key="UnixServiceAccountName" value="root" />
17    <add key="UnixServiceAccountPassword" value="asp.net" />
18    <add key="UnixServerAdresse" value="herbie.ba-horb.de" />
19  </appSettings>
20  <location path="Content">
21    <system.web>
22      <authorization>
23        <allow users="*" />
24      </authorization>
25    </system.web>
26  </location>
27  <location path="Scripts">
28    <system.web>
29      <authorization>
30        <allow users="*" />
31      </authorization>
32    </system.web>
33  </location>
```

Abbildung 15: Web.config Teil 1

#### 5.1.4 Bibliotheken

Die benötigten Bibliotheken werden von Zeile 36-42 zum vorhandenen System hinzugefügt. Gleichzeitig wird darüber (Zeile 35) das Zielframework der Applikation angegeben. Dies darf nicht verändert werden da einige Methoden auf den Funktionen des aktuellen (4.0) Frameworks aufbauen und eine Änderung des Frameworks umfangreiche Änderungen an der Quellcodebasis nach sich zieht.

#### 5.1.5 Authentifizierung

In Zeile 44-47 wird die Methode der Authentifizierung festgelegt. Falls das System weiterentwickelt wird, könnte an dieser Stelle dafür gesorgt werden, dass der Anwender automatisch angemeldet wird, sobald er sich im Intranet anmeldet. Aktuell ist dies jedoch nicht vorgesehen und der Eingetragene Modus „forms“ bedeutet, dass eine formularbasierte Anmeldung notwendig ist.

In Zeile 46 wird dem System bekannt gegeben, in welches Verzeichnis nicht angemeldete Benutzer geleitet werden. In diesem Fall ist es das Verzeichnis

„Account“, in welchem die Webseite mit der Anmeldeprozedur hinterlegt ist. Das einzig interessante für den Administrator in dieser Zeile ist der Timeout. Über ihn kann angegeben werden, wie lange der angemeldete Benutzer über ein Cookie authentifiziert wird, wodurch er sich ein erneutes Anmelden erspart. Standardmäßig ist der Timeout auf 7 Tage gesetzt, die Angabe in der Web.config ist hierbei in Minuten zu hinterlegen.

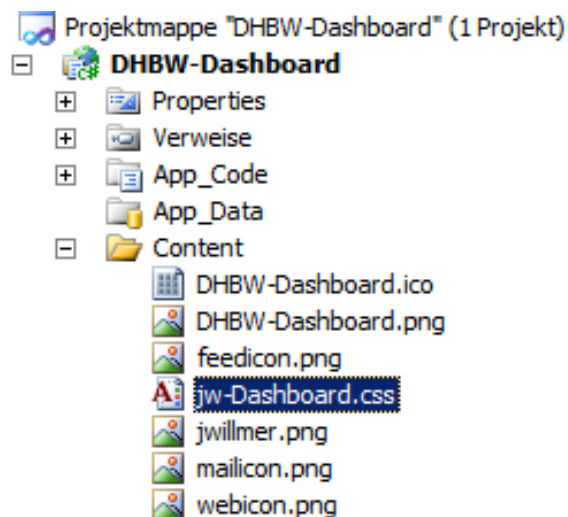
```
34 <system.web>
35 <compilation debug="true" targetFramework="4.0">
36 <assemblies>
37 <add assembly="System.Web.Abstractions, Version=4.0.0.0, Culture=neutral, PublicKeyToken=31BF3856AD364E35" />
38 <add assembly="System.Web.Helpers, Version=1.0.0.0, Culture=neutral, PublicKeyToken=31BF3856AD364E35" />
39 <add assembly="System.Web.Routing, Version=4.0.0.0, Culture=neutral, PublicKeyToken=31BF3856AD364E35" />
40 <add assembly="System.Web.Mvc, Version=3.0.0.0, Culture=neutral, PublicKeyToken=31BF3856AD364E35" />
41 <add assembly="System.Web.WebPages, Version=1.0.0.0, Culture=neutral, PublicKeyToken=31BF3856AD364E35" />
42 </assemblies>
43 </compilation>
44 <authentication mode="Forms">
45 <!-- Cookie ist 7 Tage gültig -->
46 <forms loginUrl="~/Account/LogOn" timeout="10080" name=".ADAuthCookie" requireSSL="true"/>
47 </authentication>
48 <authorization>
49 <allow users="*" />
50 </authorization>
51 <membership defaultProvider="ActiveDirectoryMembershipProvider">
52 <providers>
53 <clear />
54 <add name="ActiveDirectoryMembershipProvider" type="System.Web.Security.ActiveDirectoryMembershipProvider,
55 System.Web, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a"
56 connectionStringName="ADConnectionString" attributeMapUsername="sAMAccountName" />
57 </providers>
58 </membership>
59 <profile enabled="false">
60 <providers>
61 <clear />
62 </providers>
63 </profile>
64 <roleManager enabled="false">
65 <providers>
66 <clear />
67 </providers>
68 </roleManager>
69 <!-- ... -->
70 </system.web>
71 <!-- ... -->
72 </configuration>
```

Abbildung 16: Web.config Teil 2

## 5.2 CSS

Es ist darauf zu achten, dass Änderungen an den CSS Dateien des Intranets auch Auswirkungen auf das Dashboard haben. Für Angaben, bei denen dieses Verhalten nicht gewünscht ist - zum Beispiel Größenangaben der einzelnen Reiter - können die CSS Eigenschaften in der lokalen CSS Datei des Dashboards explizit gesetzt

werden. Die CSS Dateien zum Anpassen des Dashboards befindet sich im Ordner Content unter dem Namen jw-Dashboard.css (Abbildung 17).



**Abbildung 17:** *Dashboard CSS-Datei*

## Abkürzungsverzeichnis

VPN	Virtual Private Network
SSH	Secure Shell
AD	Active Directory
IIS	Microsoft Internet Information Services
DHBW	Dualen Hochschule Baden-Württemberg
SSL	Secure Sockets Layer
EWS	Exchange Web Services
ext4	Fourth Extended Filesystem
OU	Organizational Unit
FQD	Fully Qualified Domain Name

## Literaturverzeichnis

- SSH.NET Library (2012)      Verwendete .NET SSH Bibliothek, <http://sshnet.codeplex.com/>, 2012
- Highchart JS (2012)      JavaScript Bibliothek zur Anzeige von Diagrammen, <http://www.highcharts.com/>, 2012
- Robichaux, P. (1999)      Managing Microsoft Exchange Server, <http://books.google.de/books?id=kydUdyIJyHAC>, 1999, EBook
- Carius F. (2012)      Exchange und Outlook Build Nummern, <http://www.msxfaq.de/admin/build.htm>, 2012, Einsichtnahme: 03.05.2012
- Wentland , W. & Simon , C. (2006)      Pool-Doku des Seminars für Computerlinguistik , <http://www.cl.uni-heidelberg.de/computerpool/pooldoku/html/ch13.html>, 2006, Einsichtnahme: 03.05.2012
- Daniel J. u.a. (2005)      Ssh, the Secure Shell: The Definitive Guide, <http://books.google.de/books?id=3XzIFG3w8-YC>, 2005, EBook

## Abbildungsverzeichnis

Abbildung 1: <i>Dateiberechtigungen</i> .....	4
Abbildung 2: <i>Modell der Suchmaske</i> .....	6
Abbildung 3: <i>Active Directory Suche</i> .....	7
Abbildung 4: <i>Objekt eines Suchergebnisses</i> .....	7
Abbildung 5: <i>Impersonation und Postfachabfrage</i> .....	8
Abbildung 6: <i>SSH Client Anfrage</i> .....	9
Abbildung 7: <i>Befehlskette die via SSH an den Server gesendet wird</i> .....	10
Abbildung 8: <i>Dashboard im Kontext des Intranets</i> .....	11
Abbildung 9: <i>Übersichtsseite des Dashboards</i> .....	13
Abbildung 10: <i>Dashboard Personensuche</i> .....	14
Abbildung 11: <i>Dashboard Einstellungen</i> .....	15
Abbildung 12: <i>Dashboard About-Seite</i> .....	15
Abbildung 13: <i>Active Directory Strukturausschnitt</i> .....	18
Abbildung 14: <i>Exchange Versionsangaben</i> .....	19
Abbildung 15: <i>Web.config Teil 1</i> .....	20
Abbildung 16: <i>Web.config Teil 2</i> .....	21
Abbildung 17: <i>Dashboard CSS-Datei</i> .....	22