



Projektarbeit SS2000 (Freitag, 19. Mai 2000)

Verzeichnis- und E-Mail-Dienste für WAP

Dozent:
Studenten:

Dr. A. Steffen
Roman Wüest
Christoph Meier

Inhaltsverzeichnis:

1.....Aufgabenstellung

- 1.1.Aufgabenstellung der PA
- 1.2.Pflichtenheft

2.....Verwendete Software

- 2.1.ColdFusion
- 2.2.Omni HTTPd (Webserver)
- 2.3.Netscape Directory Server
- 2.4.Nokia WAP- Toolkit

3.....Realisierung der PA

- 3.1.Projektplan
- 3.2.Vorgehen
 - 3.2.1. Installation
 - 3.2.2. Erste Seiten
 - 3.2.3. ColdFusion Basics
- 3.3.Softwarespezifikationen
- 3.4.Aufbau des LDAP- Verzeichnisses
- 3.5.Zusammenwirken der Software
- 3.6.Bedienerführung
- 3.7.Implementation

4.....Schlusswort

- 4.1.Lernerfolg
- 4.2.Nicht realisierte Funktionen
- 4.3.Praktischer Nutzen der PA
- 4.4.Abschluss des Projektes

A.Anhang

- A.1.Testprotokoll
- A.2.Skript der CFM- Seiten
- A.3.Glossar
- A.4.Links

1. Aufgabenstellung

1.1.Aufgabenstellung der PA

Kommunikationssysteme (KSy)

Projektarbeiten SS 2000 - Sna10

Verzeichnis- und E- Mail Dienste für WAP

Studierende:

Christoph Meier, IT3a
Roman Wüest, IT3a

Termine:

Ausgabe: Dienstag, 14.03.2000 15:30 - 16:30 im E509
Abgabe: Freitag, 19.05.2000

Beschreibung:

Wegen der eingeschränkten Darstellungs- und Eingabemöglichkeiten der derzeitigen WAP- fähigen GSM Mobilgeräte (z. Bsp. Nokia 7110) möchte man beim Versenden von E- Mails den Mail-Empfänger mit möglichst wenigen Tastaturbefehlen eingeben können. Dies kann zum Beispiel durch die Präsentation einer kurzen Auswahlliste erreicht werden, die dynamisch nach gewissen Kriterien (Freunde, Familie, Geschäft, usw.) generiert, von einem Verzeichnisserver im Internet bezogen werden kann.

In dieser Arbeit soll serverseitig das Tool ColdFusion eingesetzt werden, das sich auf dem Web Server Gebiet einen guten Namen geschaffen hat. ColdFusion besitzt sowohl Schnittstellen zu LDAP- basierten Verzeichnisdiensten, als auch zu Mail- Diensten. Die von einem LDAP- Server durch eine Abfrage gelieferte Auswahl von E- Mail Adressen sollen als WML- Seite aufgebaut werden und die vom Benutzer am WAP- Terminal erstellte E- Mail wird anschliessend durch ColdFusion an einen Mail- Server weitergeleitet.

Aufgaben:

Studium des Aufbaus und der Abfragemöglichkeiten von LDAP- Verzeichnissen.

Einarbeiten in das ColdFusion Tool

Definition eines Pflichtenhefts auf der Basis von Kundenbedürfnissen

Erstellen der SW-Spezifikation auf der Basis des Pflichtenhefts

Definition der Schnittstellen und Module

Konzept der WML- basierten Bedienungsführung

Implementation und Austesten der Anwendung

Dokumentation der Projektarbeit

1.2. Pflichtenheft

Login

- Angabe von Benützername und Passwort
- Passwort sollte nur aus Ziffern bestehen (ein PIN), User kann aber auch ein Passwort mit Buchstaben wählen.
- Username und Passwort nach Möglichkeit speichern, sodass der Benützer diese Eingaben nicht machen muss. (z.B. via ein Bookmark bereits den Namen setzen oder das Passwort muss nicht eingegeben werden)

Menü

- **Directories:** ermöglicht das browsen eines oder mehrerer vorher konfigurierten LDAP Verzeichnisse.
 - Verschiedene LDAP- Server konfigurierbar. Einer davon ist der Standard Server, der zu Beginn benutzt wird.
 - Das Durchsuchen des Verzeichnisses startet auf dem Standard LDAP- Server, in einem Start- Eintrag. (z.B. im persönlichen Verzeichnis)
 - Bei jedem Eintrag soll die Telefonnummer oder die E- Mail Adresse in das persönliche Verzeichnis aufgenommen werden können.
- **E- Mail:** enthält ein persönliches Adressbuch und vordefinierte E- Mails, sowie auch die Möglichkeit, ein neues E- Mail zu verfassen.
 - Es sollen vordefinierte E- Mail Vorlagen vorhanden sein, welche man auch verändern oder ergänzen kann. Evtl. Möglichkeit, E- Mail offline zu schreiben.
- Möglichst einfache Auswahl der Empfänger, z.B. über eine Select- Box.
 - Die Texteingabe muss so einfach wie möglich gestaltet werden, da die Eingabe von Text auf den Mobiltelefonen nur mühsam möglich ist und da die Telefone nach einer gewissen Zeit einen Timeout erreichen und aufhängen.
- **Last Used:** die letzten (10) benützten Einträge oder Adressen
- **Help/Infos:** Hilfe und Informationen über diesen Dienst

User Interface / Generelles

- Konsistenz: immer wenn eine Telefonnummer oder eine E- Mail Adresse im Display angezeigt wird, soll man sehr einfach einen Anruf an diese Nummer tätigen können, direkt eine E- Mail senden, oder den betreffenden Eintrag ins persönliche Verzeichnis aufnehmen können.
- Nebst dem WAP Teil, soll auch ein WWW-Interface zur Verfügung stehen, das soweit möglich dieselbe Funktionalität bietet, wie die WAP Applikation. Gewisse Einstellungen (z.B. LDAP Server) sind sogar einfacher auf dem Web einzurichten.
- Einfache Konfiguration der WAP Seiten, damit man sie einfach auf spätere WAP Standards oder andere WAP- Mobiltelefone anpassen kann.
- Dem WAP angepasstes einfaches User- Interface, wenig Grafik, einfache und prägnante Links. Möglichst kurze und übersichtliche WML Seiten. (Nokia maximum compiled page size: 1397 bytes) Überall ein „Back“ Knopf.
- Nach Möglichkeit Englisch und Deutsch. Ebenfalls einfach konfigurierbar via Konfigurationsfiles.

2. Verwendete Software

2.1. ColdFusion

Einführung:

ColdFusion ist ein Produkt der Firma Allaire [1]. Seit einigen Jahren ist es die Alternative zu anderen Skript Sprachen und CGI-Methoden wie ASP und PHP. Dabei besticht ColdFusion, dass nebst Windows NT auch für Solaris und Linux erhältlich ist, durch Einfachheit der Programmierung, durch verschiedenste Interfaces und durch eine leichte Integration. ColdFusion ist momentan ab ca. 2'650.- CHF erhältlich.

Fähigkeiten:

- Rapid Development
Ein innovatives Konzept zur Programmierung mit Hilfe von HTML ähnlichen Tags ermöglicht ein produktives und schnelles Erstellen von Web-Applikationen.
- Scalable Deployment
Auch lässt das Programm in Sachen Performance keine Wünsche offen. Trotz der Arbeitsweise eines Parser/Interpreters ist ColdFusion dank multithreading und just-in-time compilation sehr schnell. Zusätzlich bietet es load balancing und failover insurance, so dass auch bei höchsten Zugriffszahlen alles rund läuft
- Open Integration
Mit einfachsten Tags ist der direkte Zugriff auf Datenbanken, E-Mail, Verzeichnisse, Dateisysteme, Java, XML möglich. Ideal für Webapplikationen, wo dynamische Seiten mit Inhalten aus Datenbanken heute genauso zum Standard gehören, wie ein einfaches Feedback Formular, dass per E-Mail versandt wird.
- Complete Security
Nicht zuletzt werden auch verschiedene Sicherheitsmechanismen unterstützt. So können ganze Umgebungen spezifisch angepasst werden, so dass Entwickler nur auf die erlaubten Ressourcen Zugriff haben.

Was ist ColdFusion?

ColdFusion wird als Middleware bezeichnet. Es kann über verschiedene Schnittstellen mit einem Webserver kommunizieren und bildet selbst die Schnittstelle zwischen dem (statischen) Webserver und verschiedenen Ressourcen wie Datenbanken und E-Mail. Dabei werden die ColdFusion Programmtags direkt in Webseiten integriert. So ist es möglich, bestehende Webseiten zu erweitern. Aufgaben wie eine SQL Abfrage oder eine Datumsberechnung oder das Suchen eines regulären Ausdrucks sind in ColdFusion trivial zu programmieren. ColdFusion parsert also Seiten (meist mit der Endung .cfm, für „ColdFusion Markup Language“) und liefert den Output als statisches HTML Content zurück an den Webserver.

Aufgaben von ColdFusion:

ColdFusion übernimmt die Erstellung von dynamischen WML-Seiten. Zusätzlich bietet es die nötigen Schnittstellen zum LDAP-Server und zum E-Mail versenden. ColdFusion übernimmt also die zentrale Rolle in der Steuerung unserer WAP Applikation. ColdFusion muss auch den Content-Type (MIME-Type) der Seite festlegen und dem Webserver melden. Nicht zuletzt übernimmt ColdFusion auch das Session-Management, also die Kontrolle und Führung eines einzelnen Benutzers.

2.2. Omni HTTPd (Webserver)

Einführung:

Als Webserver wählten wir Omni HTTPd der Firma Omnicron Technologies Co. [2]. Für ColdFusion kämen natürlich auch andere Webserver wie Microsoft IIS oder Apache durchaus in Frage. Omni HTTPd ist ein kompakter und einfacher Webserver, der aber die wichtigsten Funktionen unterstützt. Zudem ist er Freeware, d.h. die Benutzung dieses Programms kostet nichts.

Fähigkeiten:

Für uns von Bedeutung sind nur die Eigenschaften wie Standard CGI Support und Lauffähig unter Windows NT.

Aufgaben des Webservers:

Der Webserver gibt in unserer Arbeit beinahe alle Requests an den ColdFusion Service weiter und liefert danach den Output von ColdFusion dem Benutzer zurück. Bei uns ist der Benutzer normalerweise ein Inhaber eines mobilen WAP- Telefons, mit dem er sich über einen WAP- Gateway Zugang zum Internet verschaffen kann. Für die wenigen zu erwartenden Benutzer und die kleinen Datenmengen, die transferiert werden müssen, reicht dieser Webserver in jedem Fall.

2.3. Netscape Directory Server

Einführung:

Netscape [3] bietet in ihrer Serverreihe auch einen LDAP Server an, den Netscape Directory Server. Dieser LDAP Server gehört sicher zu den stabileren und schnelleren Servern auf dem Markt.

Aufgaben des LDAP Servers:

Der LDAP Server ist nicht eine zentrale Stelle unserer Applikation, sondern wir zeigen vielmehr, wie man mit ColdFusion WML Seiten erstellen, die Daten aus einem LDAP Server enthalten. Der Netscape Directory Server liefert uns diese Daten.

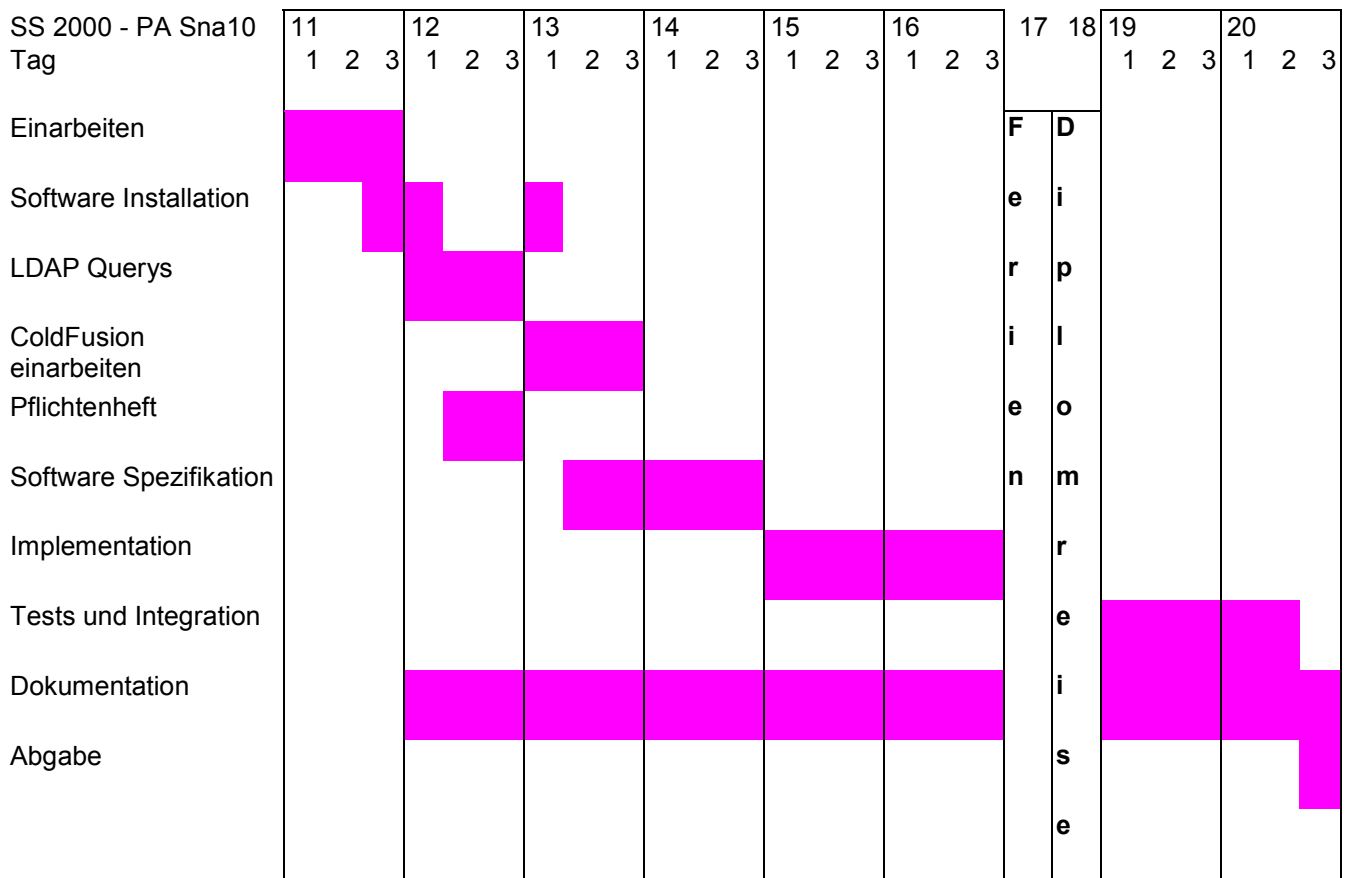
2.4. Nokia WAP- Toolkit

Einführung:

Nokia [4] bietet einen WAP Toolkit an. Dieser besteht nebst der Dokumentation zu WML und dem Nokia 7110 Mobiltelefon auch aus einem Simulator, der es ermöglicht, WAP Applikationen auf einem PC zu testen. Dieser Toolkit ist frei erhältlich.

3. Realisierung der PA

3.1. Projektplan



3.2. Vorgehen

3.2.1. Installation

Als erstes installierten wir den Webserver. Die Wahl fiel dabei zuerst auf den Microsoft Internet Information Server. Dieser installierte sich auf der Workstation Version von NT lediglich als ein Microsoft Personal Webserver, bei dem wir keine Möglichkeit fanden, neue MIME-Types zu definieren.

Die Wahl fiel sodann auf den Omni HTTPd, einen kleinen aber feinen Webserver. Dort konnten wir ohne Probleme die notwendigen MIME-Types einrichten (siehe weiter unten). Eine einfache HTML Seite und eine WML Seite überzeugten uns schnell, dass die Installation geglückt ist.

Danach installierten wir den ColdFusion Application Server. Dieser setzt voraus, dass bereits ein Webserver installiert ist. Auch hier traten keine Probleme auf.

3.2.2. Erste Seiten

Zuerst sollte man sicherlich mit ColdFusion mal Tests anstellen, wie man eine WML Seite generiert, so dass auch der MIME- Type der Seite korrekt beim Endgerät ankommt. Die WAP- fähigen Mobiltelefone zeigen heute noch keine HTML Seiten an. Die MIME- Types sind in der folgenden Tabelle aufgelistet:

Dokument Typ	MIME Typ	Dateiendung
WML Dokument	text/vnd.wap.wml	.wml
WBMP Bilder	image/vnd.wap.wbmp	.wbmp
WML Dokument kompiliert	application/vnd.wap.wmlc	.wmlc
WML Script	text/vnd.wap.wmls	.wmls
WML Script kompiliert	application/vnd.wap.wmlscriptc	.wmlsc

WAP MIME Typen

Also definieren wird zuerst in der Seite den MIME- Type. Das geschieht mit dem ColdFusion Befehl `<CFCONTENT>` . Danach lesen wir aus der WML Dokumentation, wie der Header und die Struktur einer einfachen WML Seite aussehen müssen, und programmieren folgende erste Seite:

```
<CFCONTENT TYPE="text/vnd.wap.wml ">

<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">

<wml>
  <card id="Test" title="WML Test Seite">
    <p>Hello World!</p>
  </card>
</wml>
```

Sourcecode von test1.cfm

Im oberen Teil ist ein längerer Header zu sehen, der zuerst festlegt, dass es sich um XML Inhalt handelt. Danach wird der Dokument Typ anhand einer DTD festgelegt. Im zweiten Teil ist dann ein WML Deck mit einer einfachen Card gezeigt.



Output von test1.cfm

3.3.3. ColdFusion Basics

Grundsätzlich werden die ColdFusion- Tags direkt in HTML Seiten eingebaut. Damit der Webserver solche Seiten auch erkennt, müssen die ColdFusion Source Files die Endung '.cfm' erhalten.

Ein weiterer Test könnte also so beginnen:

```
<HTML>
<HEAD><TITLE>Test Seite 2</TITLE></HEAD>
<BODY>

<CFSET myvariable = 5>
<CFSET myresult = myvariable * myvariable>

<CFOUTPUT>
Das Resultat ist: #myresult#
</CFOUTPUT>

</BODY></HTML>
```

Sourcecode von test2.cfm



Output von test2.cfm

Man kann also HTML Tags und ColdFusion Befehle mischen. Dabei werden ColdFusion Variablen nur dann ausgegeben, wenn sie zwischen zwei Raute-Zeichen (#) stehen und wenn diese wiederum in einem CFOUTPUT- Block sind. Will man ein Raute- Zeichen selbst ausgeben, so geschieht dies durch ##.

Auch das Email Interface ist sehr einfach zu verstehen. Der <CFMAIL> Tag hat unter anderem folgende Attribute:

```
<CFMAIL
  SERVER="mail.zhwin.ch"
  PORT="25"
  FROM="#sender_email#"
  TO="#recipient_email#"

  SUBJECT="#subject#"
  This is the email body.
</CFMAIL>
```

Attribute von CFMAIL

Zwischen dem öffnenden und schliessenden Tag von CFMAIL befindet sich der eigentliche Inhalt der zu versendenden Email. Wir haben erfolgreich einige solcher Emails versendet.

Von zentraler Bedeutung sind natürlich auch die LDAP Queries. In ColdFusion gibt es den CFLDAP Tag, der folgenden Attribute haben kann:

```
<CFLDAP
  SERVER="#LDAPServer#"
  USERNAME="#LDAPUSERNAME#"
  PASSWORD="#LDAPPASSWORD#"
  ACTION="query"
  NAME="abfrage1"
  START="ou=#username#, #LDAPStart#"
  SCOPE="ONELEVEL"
  FILTER="(cn=#username#)"
  ATTRIBUTES="cn,ou,unserpassword">
```

Attribute von CFLDAP

Neben den selbsterklärenden Attributen gibt es unter anderen folgende Attribute:
ACTION:

Je nach Art des LDAP Requests können die Werte ADD, DELETE, MODIFY, MODIFY DN und QUERY verwendet werden.

NAME:

Mit dem Namen bezeichnet man die Query. Über diesen Namen hat man Zugriff auf die Resultate.

DN:

Der Distinguished Name. Nötig bei den Actions ADD, MODIFY und MODIFYDN.

SORT:

Eine kommagetrennte Liste von Attributen, nach denen das Resultat sortiert werden soll, z.B. SORT="Ort ASC, Alter DESC"

Was auch bald ein mal getestet wurde, war das Client- und Session-Management. Für normale Web Applikationen kann man mit Coldfusion sehr leicht Client abhängige Variablen setzen:

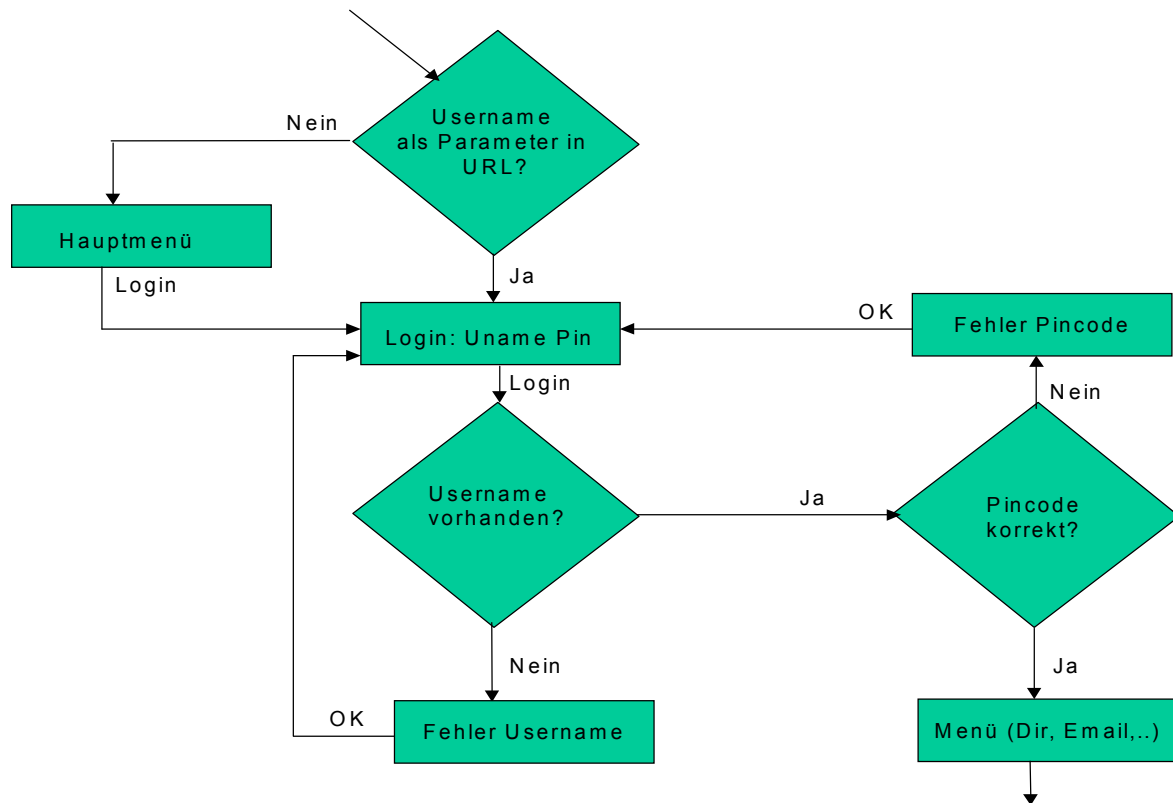
```
<CFSET Client.Username = "Roman Wueest">
```

Leider muss dazu im Browser ein Cookie gesetzt werden und weder das Nokia 7110 noch das Ericsson R320 unterstützen Cookies.

Deshalb müssen wir in der Applikation immer darum besorgt sein, dass der Client zu jeder Zeit identifiziert werden kann. Dies geschieht mit Hilfe der beiden Coldfusion Variablen CFID und CFTOKEN. Wenn diese Variablen immer bekannt sind, so kann man Client Variablen auch serverseitig setzen.

3.3. Softwarespezifikationen

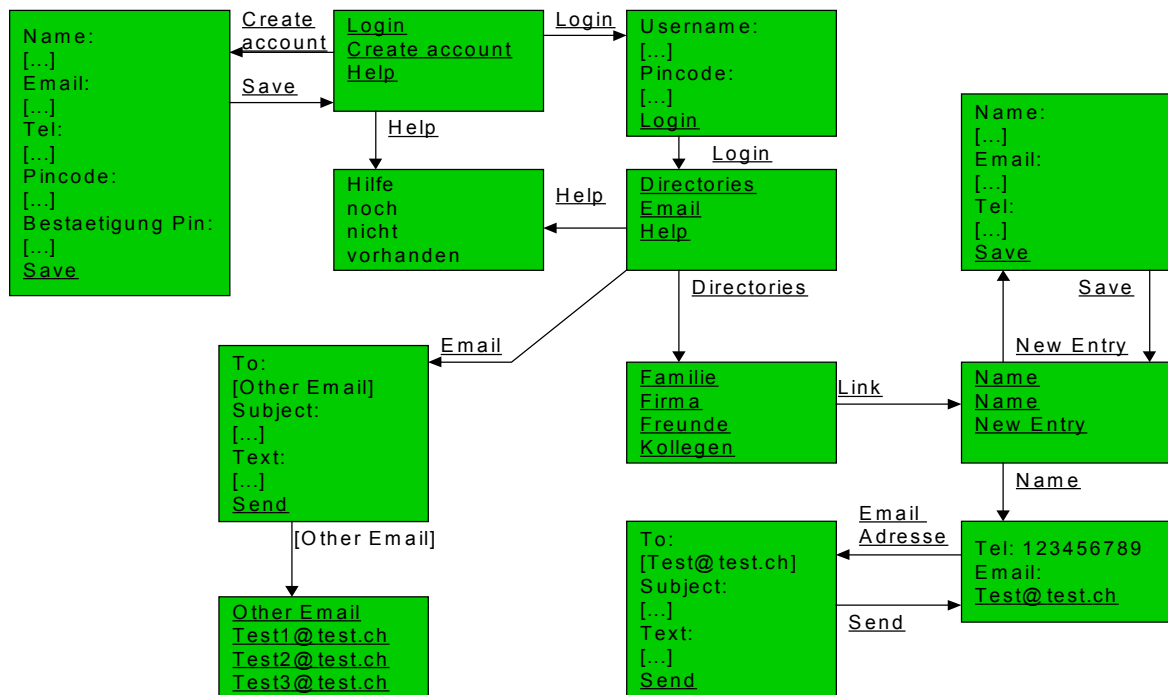
Der Login- Vorgang soll wie folgt ablaufen:



Zuerst wird geprüft, ob mit der URL auch ein Username als Parameter mitgegeben wird. Falls Ja, kommt man zur Login- Seite (Username und Pincode eingeben, Username schon durch Parameter ausgefüllt). Falls kein Parameter mitgeliefert wird, kommt man zum Hauptmenü, wo man sich einloggen kann, oder einen neuen Account erstellen. Durch die Wahl von Login kommt man dann auch zur Login- Seite.

Nachdem man auf der Login- Seite alles eingegeben und auf Login gewählt hat, wird zuerst überprüft, ob der Username in der Datenbank vorhanden ist und wenn ja, ob der Pincode mit dem gespeicherten übereinstimmt. Falls der Username nicht vorhanden ist, oder der Pincode nicht stimmt, wird eine entsprechende Fehlermeldung ausgegeben. Stimmen die Angabe kommt man zur Menüauswahl, wo man die Verschiedenen Dienste benutzen kann.

Die Menüführung soll wie folgt gestaltet sein:



Bei jeder Seite kommt noch ein Back und ein Optionen Knopf hinzu. (rechter und linker Auswahlknopf auf dem Handy). Nicht aufgeführt sind die Bestätigungen die beim Speichern auf den LDAP- Server sowie beim verschicken von E- Mails den Erfolgreichen Abschluss melden.

3.4. Aufbau des LDAP- Verzeichnisses

Das LDAP- Verzeichnis ist folgendermassen aufgebaut:

Servername, -port: ksy110.zhwin.ch:389
Organisation: o=zhwin.ch
Organisatorische Einheit: ou= wapsign

In der Unit wapsign wird dann für jeden Benutzer eine Organisatorische Einheit mit dem Usernamen eingerichtet (Bsp. ou= Roman Wueest).

In dieser Unit befinden sich nun die Informationen über den Benutzer und seine Daten (Hier Tel. Nr. und E- Mail Adressen seiner Freunde, Bekannten,...).

Die Daten sind in 4 Ordnern (Organisatorische Einheiten) eingeordnet: Familie, Firma, Freunde und Kollegen.

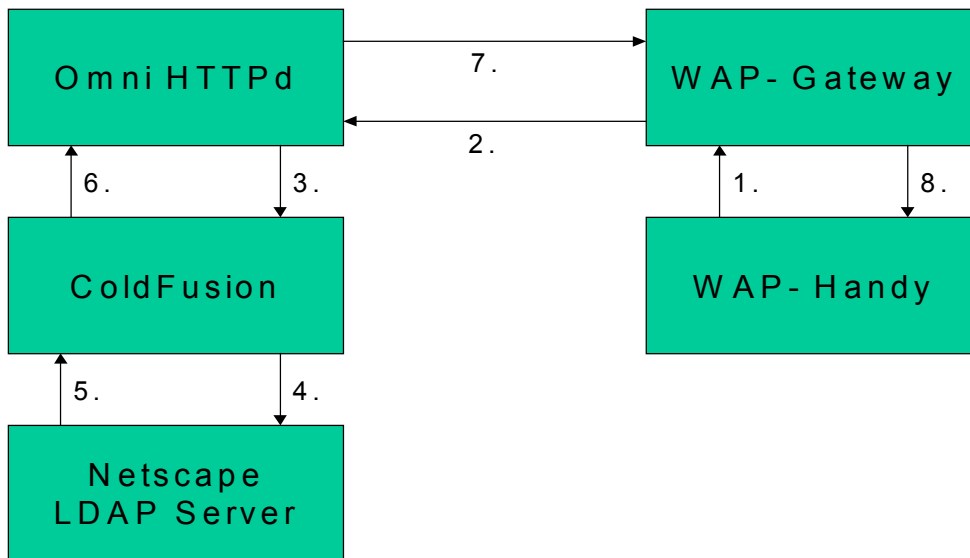
Für die Informationen des Users und die Dateneinträge haben wir ein eigenes Schema benützt, welches unseren Anforderungen entspricht.

Name des Schemas: usereintrag

Benötigte Attribute: cn (Name. Bsp. Roman Wueest)
mail (E- Mail Adresse. Bsp. i7meier@zhwin.ch)
telephonenumber (Telephonnummer. Bsp. 0522677511)
Erlaubte Attribute: unserpasswort (Pincodes des Users. Bsp. 1234)

3.5. Zusammenwirken der Software

Folgendes Schema zeigt das zusammenwirken der verschiedenen Programme



- 1.) User wählt eine Internet Seite an.
- 2.) WAP- Gateway wandelt den WAP- Request in einen HTTP- Request um und sendet ihn an den WEB- Server.
- 3.) WEB- Server erkennt CF- Tags und sendet die Seite weiter an ColdFusion.
- 4.) ColdFusion führt allfällige LDAP- Queries durch.
- 5.) LDAP- Server sendet die Resultate der Query an ColdFusion.
- 6.) ColdFusion sendet eine dynamisch erstellte HTML- Seite an den WEBV- Server zurück.
- 7.) Der WEB- Server sendet die erhaltene Seite via HTTP an den WAP- Gateway.
- 8.) WAP- Gateway wandelt die HTML- Seite in WAP- taugliche binärform um und sendet diese via WAP an das Handy.

3.6. Benutzerführung

Wenn Sie auf die Seite "http://wap.sign.ch/" kommen, werden Sie nebenstehendes Bild antreffen. Falls Sie schon registriert sind, können Sie sich mit [Login](#) an den Server anmelden. Falls Sie noch nicht registriert sind, können Sie mit [Create account](#) einen neuen Account erstellen. Mit [Help](#) erhalten Sie Hilfe zur aktuellen Seite.

[Login](#)
[Create account](#)
[Help](#)

Falls Sie sich entschieden haben, einen neuen Account zu erstellen, werden Sie auf nebenstehende Seite kommen. Wenn Sie die Eingabefelder [...] auswählen, können Sie die entsprechenden Daten, wie Name, E- Mail Adresse, Telephonnummer und Pincode, eingeben.

Name:
[...]
Email:
[...]
Tel:
[...]
Pincode:
[...]
Bestaetigung Pin:
[...]
[Save](#)

Da Sie nun einen Account besitzen, haben Sie sicher den Link [Login](#) angewählt. Sie befinden sich nun auf der Login- Seite. Wählen Sie nun die Eingabefelder [...] und geben ihren Namen und Pincode ein. Durch Wahl von [Login](#) wird Ihr Pincode und Username geprüft.

Username:
[...]
Pincode:
[...]
[Login](#)

Falls die Angaben korrekt sind, kommen Sie zur Menüauswahl. Mit [Help](#) erhalten Sie wieder Hilfe zur Seite. Falls Sie gleich ein E- Mail senden wollen, wählen Sie [Email](#) an. Mit [Directories](#) können Sie ihr Persönliches Adressverzeichnis durchforsten.

[Directories](#)
[Email](#)
[Help](#)

Sie wollen gleich eine E- Mail schicken? Sie werden sich auf dieser Seite wiederfinden. Wählen Sie die Eingabefelder [...] um das Subject ihrer E- Mail anzugeben und den Text zu schreiben. Wählen Sie das Eingabefeld [Other Email] um eine Liste von allen Personen zu erhalten, welche Sie in ihrem Verzeichnis gespeichert haben. Sie können auch selber von Hand eine E- Mail Adresse eingeben, indem Sie [Other Email] so belassen und Send wählen. Falls Sie keinen Empfänger ausgewählt habe, können Sie nun einen selber eingeben.

To:
[Other Email]
Subject:
[...]
Text:
[...]
Send

Falls Sie sich zuvor entschieden haben, ihr Verzeichnis zu durchforsten, werden Sie auf diese Auswahl-Seite gelangen. Wählen Sie einen dieser 4 Links, um alle Personen, welche in der gewählten Kategorie sind, aufgelistet zu haben oder einen neuen Eintrag zu speichern.

Familie
Firma
Freunde
Kollegen

Die Seite, welche Sie sehen, wird ähnlich der nebenstehenden aussehen. Wählen Sie einen Namen, um die gespeicherten Informationen dieser Person zu sehen. Wollen Sie einen neuen Eintrag erstellen, Wählen Sie New Entry.

Name
Name
New Entry

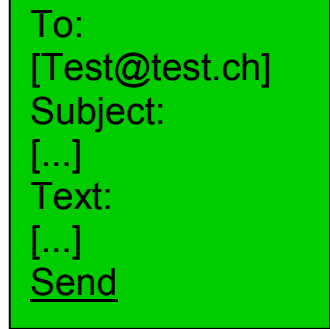
Falls Sie einen neuen Eintrag speichern wollen, müssen Sie den Namen der Person, die Tel. Nr. und die E- Mail Adresse angeben. Wählen Sie dafür die Eingabefelder [...] aus. Mit Save wird Ihr Eintrag gespeichert.

Name:
[...]
Email:
[...]
Tel:
[...]
Save

Wenn Sie die Informationen über eine Person ansehen wollen, haben Sie sicher diesen Namen gewählt. Die Seite wird folgendermassen aussehen. Mit einem Nokia 7110 können Sie den grünen Telephonhörer auf dem Handy drücken, um die eingblendete Telephonnummer auszuwählen und ein Anruf zu tätigen. Wählen Sie die E- Mail Adresse, können Sie dieser Person eine E- Mail senden.

Tel: 123456789
Email:
Test@test.ch

Wie Sie sehen, ist der Empfänger schon eingetragen. Sie müssen nun mit den Eingabefeldern nur noch das Subject und den Text eingeben. Die E- Mail wird mit Send gesendet.



To:
[Test@test.ch]
Subject:
[...]
Text:
[...]
Send

Mit dem Back Knopf (rechte Auswahl Taste auf ihrem Handy) können Sie zur jeweils vorherigen Seite zurückkehren.

Viel Spass beim E- Mail versenden mit ihrem Handy.

3.7. Implementation

Im folgenden möchten wir kurze Erläuterung zu einigen der Sourcefiles angeben. Viele Kommentare sind auch direkt in den Sourcefiles angegeben.

application.cfm

Dieses Seite ist durch den speziellen Namen so definiert, dass sie automatisch als globale Einstellungsdatei verwendet wird. Das heisst, wenn immer ein ColdFusion Sourcecode aus diesem Verzeichnis ausgeführt wird, wird zuerst das application.cfm ausgeführt.

index.cfm

Im index.cfm, welches als erstes geladen wird (per Definition im Webserver), fragen wir die CGI Variable HTTP_ACCEPT ab. Falls diese 'wml' beinhaltet, nehmen wir an, dass es sich um einen WML-fähiges Endgerät handelt und binden die WML Startseite ein.

option.cfm

Dies ist eine Subroutine, um möglichst einfach Links und Optionen in WML Seiten einzubauen. Man kann wählen ob man nur einen Link (anchor), nur eine Option (do) oder beides einbinden will.

Subroutinen werden als Tag mit vorangestelltem CF aufgerufen, hier also <CF_OPTION>. Ebenfalls kann man Attribute übergeben z.B. mit <CF_OPTION TYPE="anchor">

insertlink.cfm

Wird von option.cfm verwendet, um die Angaben für das Client-Management jeweils von Seite zu Seite weiterzugeben.

Alle anderen Files sind entweder direkt im Sourcecode dokumentiert oder im Anhang A.1. erwähnt.

4. Schlusswort

4.1. Lernerfolg

Die Projektarbeit brachte uns vorallem in Sachen Coldfusion Programmierung und LDAP Querys einiges an Wissen und interessanten Problemen. Auch bei der Installation und dem Zusammenspiel der einzelnen Komponenten konnte man einiges in Erfahrung bringen.

Die Arbeit hat uns sehr viel Spass gemacht. Coldfusion machte bei uns den besten Eindruck. Selten gab es von dieser Seite Probleme, die unlösbar waren. Im Gegensatz dazu war der LDAP Server nicht ganz leicht zu administrieren und die Mobiltelefone (Nokia 7110 und Ericsson R320s) stürzten oft ab.

Im Bereich WAP wird sich sicherlich noch einiges tun in nächster Zeit. So z.B. mit dem GPRS Update Anfangs 2001, der das ganze WAP dann etwas schneller machen dürfte.

4.2. Nicht realisierte Funktionen

Bezüglich Pflichtenheft haben wir folgende Funktionen nicht relisiert. Die meisten davon wegen Zeitmangels.

- Bisher ist nur ein LDAP-Server abrufbar (2.1.1.)
- Keine vordefinierten Mails (2.2.)
- Da noch keine Emails gelesen werden können, können diese auch nicht geforwarded oder beantwortet werden (2.2.4.)
- Last Used nicht implementiert (2.3.)
- Help noch nicht implementiert (2.4.)

4.3. Praktischer Nutzen der PA

Die Projektarbeit hat gezeigt, dass Coldfusion ein hervorragendes Tool ist um Webserver, WAP-Inhalte und Verzeichnisdienste (LDAP) zu kombinieren. Sehr geeignet ist eine solche Applikation für eine Firma, die bereits die User-Daten in einem LDAP Verzeichnis speichert. Mit Coldfusion ist es sehr leicht einen WAP-LDAP Gateway zu realisieren, wie auch ein WEB-LDAP oder auch Gateways mit Email Funktionen.

Coldfusion eignet sich darüber hinaus für jegliche Applikationen, die einige der Interfaces Datenbank, Email, Web, Files und FTP unterstützen müssen. Durch die wirklich einfache Programmierung und die direkte Einbindung in bereits bestehende HTML Seiten, ist der Einarbeitungsaufwand relativ gering.

4.4. Abschluss der Projektarbeit

Da unser Arbeits-PC (ksy110.zhwin.ch) nach der Arbeit anders verwendet wird, werden wir die Arbeit auf einen externen Server umziehen (<http://wap.sign.ch>). Unter dieser Adresse werden wir auch Informationen zur Arbeit und die Dokumentation ablegen.

A. Anhang

A.1. Testprotokoll

Hauptseite (login.cfm#main)

- √ Via <http://wap.sign.ch/> auf die Seite kommen.
- √ Mit den Links: Login, Create Account und Help zu den entsprechenden Seiten gelangen.

Login (login.cfm#login, error.cfm)

- √ Username kann eingegeben und editiert werden.
- √ Pincod kann eingegeben und editiert werden.
- √ Link Login funktioniert.
- √ Bei ungültigem Username / Pincod wird mit Fehlermeldung der Zugriff verweigert.

Menüauswahl (menu.cfm)

- √ Mit den Links: Directory, E- mail und Help zu den entsprechenden Seiten gelangen.

Directory Auswahl (list.cfm)

- √ Mit den Links: Familie, Firma, Freunde und Kollegen zu den entsprechenden Seiten gelangen.

Directory (users.cfm, user_detail.cfm)

- √ Auflistung aller Personen im Verzeichnis.
- √ Auflistung des Links: New entry
- √ Durch Auswahl eines Namen zur Tel. Nr. und E- Mail Adresse der entsprechenden Person gelangen.
- √ Durch Auswahl der E- Mail Adresse zur entsprechenden Seite gelangen.
- √ Bei Nokia 7110 durch Betätigung des grünen Telefonhörers zur Nummernauswahl gelangen.
- √ Durch Auswahl des Links "New entry" zur entsprechenden Seite gelangen.

New entry (insertuser.cfm, insertuser_save.cfm)

- √ Eingabe und editieren des Namen, E- Mail Adresse und Tel. Nr.
- √ Durch Auswahl von Save den Eintrag auf dem LDAP- Server speichern.

E- Mail (email.cfm, sendmail.cfm)

- √ Durch Directory ausgewählte E- Mail Adresse übernommen.
- √ Bei Wahl des E- Mail links (Menüauswahl) Liste mit allen E- Mail Adressen anzeigen (Namen der Person)
- √ E- Mail Adresse von Hand eingeben.
- √ Subjekt und Text eingeben und editieren.
- √ Durch Auswahl von Send E- Mail absenden.

Create Account (create.cfm, create_save.cfm)

- √ Eingabe und editieren von Name, Tel. Nr., E- Mail und Pincode
- √ Durch Auswahl von Save den User hinzufügen.

Allgemein

- √ Back Knopf funktioniert bei jeder Seite

Ort und Datum: Winterthur, 19.5.2000 Visum: Chm

A.2. Skript der CFM- Seiten

application.cfm

```
<!-- application.cfm --->
```

```
<!-- this file is automatically pre-pended to every page in  
this directory and in all subdirectories --->
```

```
<!-- define the content type to be wml for every page --->  
<CFCONTENT TYPE="text/vnd.wap.wml">
```

```
<!-- reload the page everytime --->  
<CFHEADER NAME="Cache-Control" VALUE="no-cache">
```

```
<!-- enable only cfoutput --->  
<CFSETTING ENABLECFOUTPUTONLY="Yes">
```

```
<!-- define the Session and Clientmanagement --->  
<CFAPPLICATION NAME="wapsign"  
    CLIENTMANAGEMENT="Yes"  
    SETCLIENTCOOKIES="No"  
    SESSIONMANAGEMENT="Yes"
```

```
>
```

```
<!-- LDAP parameters --->  
<CFSET LDAPSERVER="localhost">  
<CFSET LDAPSTART="ou=wapsign,o=zhwin.ch">  
<CFSET LDAPUSERNAME="cn=Directory Manager">  
<CFSET LDAPPASSWORD="admin123">
```

```
<!-- parameter passing for session management --->  
<CFSET passuser="CFID="&CFID&"&CFTOKEN="&CFTOKEN">
```

create.cfm

```
<!-- create.cfm --->
```

```
<!-- create a new account in the ldap
      this means, create the directories for the entries and the users account --->
```

```
<CFOUTPUT><?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">

<wml>
  <card id="create" title="Create account">
    <onevent type="onenterforward">
      <refresh>
        <setvar name="name" value=""/>
        <setvar name="email" value=""/>
        <setvar name="tel" value=""/>
        <setvar name="pin" value=""/>
        <setvar name="pin2" value=""/>
      </refresh>
    </onevent>
    <onevent type="onenterbackward">
      <prev/>
    </onevent>
    <do type="prev" label="Back">
      <prev/>
    </do>
    <p>Name: <input name="name" type="text" title="Name" value=""/></p>
    <p>Email: <input name="email" type="text" title="Email" value=""/></p>
    <p>Tel: <input name="tel" type="text" title="Tel" format="*N" value=""/></p>
    <p>Pincode: <input name="pin" type="password" title="Pincode" format="*N"
value=""/></p>
    <p>Bestaetigung Pin: <input name="pin2" type="password" title="Pincode"
format="*N" value=""/></p>
```

```
<p>
<anchor>Save
<go href="create_save.cfm">
  <postfield name="name" value="$(name)"/>
  <postfield name="email" value="$(email)"/>
  <postfield name="tel" value="$(tel)"/>
  <postfield name="pin" value="$(pin)"/>
  <postfield name="pin2" value="$(pin2)"/>
  <CF_INSERTLINK>
</go>
</anchor>
</p>

</card>
</wml>
</CFOUTPUT>
```


create_save.cfm

```
<!-- create_save.cfm -->

<!-- create a new account in the ldap
      this means, create the directories for the entries and the users account -->

<!-- check if a name has been entered -->
<CFIF Len(Name) LT 3>
    <CFSET reason="shortname">
    <CFINCLUDE TEMPLATE="error.cfm">
    <CFABORT>
<!-- check if the two pins correspond -->
<CFELSEIF pin NEQ pin2>
    <CFSET reason="pinsnotidentical">
    <CFINCLUDE TEMPLATE="error.cfm">
    <CFABORT>
</CFIF>

<!-- create the main user directory -->
<CFLDAP
    SERVER="#LDAPSERVER#"
    USERNAME="#LDAPUSERNAME#"
    PASSWORD="#LDAPPASSWORD#"
    ACTION="ADD"
    ATTRIBUTES="objectclass=top, organizationalunit; ou=#name#"
    DN="ou=#name#,ou=wapsign,o=zhwin.ch">

<!-- create the user configuration, type usereintrag -->
<CFLDAP
    SERVER="#LDAPSERVER#"
    USERNAME="#LDAPUSERNAME#"
    PASSWORD="#LDAPPASSWORD#"
    ACTION="ADD"
    ATTRIBUTES="objectclass=top, usereintrag;
                cn=#name#;
                mail=#email#;
                telephonenumber=#tel#;
                unserpassword=#pin#"
    DN="cn=#name#,ou=#name#,ou=wapsign,o=zhwin.ch">

<!-- create the folders for the entries -->
<CFSET foldernames="Familie,Firma,Freunde,Kollegen">
```

```
<CFLOOP LIST="#foldernames#" INDEX="currentfolder">
  <CFLDAP
    SERVER="#LDAPSERVER#"
    USERNAME="#LDAPUSERNAME#"
    PASSWORD="#LDAPPASSWORD#"
    ACTION="ADD"
    ATTRIBUTES="objectclass=top, organizationalunit; ou=#currentfolder#"
    DN="ou=#currentfolder#,ou=#name#,ou=wapsign,o=zhwin.ch">
  </CFLOOP>

<CFOUTPUT><?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">

<wml>
  <card id="create" title="Create account">
    <do type="prev" label="Back">
      <prev/>
    </do>
    <p>Your Account has been created.</p>
    <p>
      <anchor>OK<prev/></anchor>
    </p>
  </card>
</wml>
</CFOUTPUT>---->
```

email.cfm

```
<!-- email.cfm --->
```

```
<CFLDAP SERVER="#LDAPServer#"
    USERNAME="#LDAPUSERNAME#"
    PASSWORD="#LDAPPASSWORD#"
    ACTION="query"
    NAME="getemails"
    START="ou=#Client.WapSignUsername#,#LDAPStart#"
    SCOPE="SUBTREE"
    ATTRIBUTES="cn,mail"
    FILTER="!(objectclass=organizationalunit)"
    SORT="cn"
```

```
>
```

```
<!-- define some parameters --->
```

```
<CFPARAM NAME="to" DEFAULT="">
```

```
<CFPARAM NAME="subject" DEFAULT="">
```

```
<CFOUTPUT>
```

```
<?xml version="1.0"?>
```

```
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
```

```
"http://www.wapforum.org/DTD/wml_1.1.xml">
```

```
<wml>
```

```

    <card id="main" title="Send Mail">
    <onevent type="onenterbackward">
    <prev/>
    </onevent>
    <onevent type="onenterforward">
    <refresh>
    <setvar name="to" value="<CFIF to IS "Other
Email">email?<CFELSE>#to#</CFIF>"/>
    <setvar name="subject" value="#subject#"/>
    </refresh>
    </onevent>
    <p>To:
</CFOUTPUT>
```

```

    <cfif to is "">
        <CFOUTPUT>
            <select name="to">
                <option value="Other Email">Other Email</option>
            </cfoutput>
            <cfoutput query="getemails">
                <option value="#mail#">#xmlformat(cn)#</option>
            </cfoutput>
            <cfoutput></select></cfoutput>
        <cfelse>
            <CFOUTPUT>
                <input type="text" name="to" value="#to#" />
            </CFOUTPUT>
        </cfif>
    <CFOUTPUT>
        </p>
        <p>Subject:
            <input type="text" name="subject" value="#subject#" />
        </p>
        <p>Text:
            <input type="text" name="mailtext" />
        </p>

        <do type="prev" label="Back">
            <prev />
        </do>

        <p>

            <CF_OPTION label='Send' href='sendmail.cfm'
            extra='
                <postfield name="to" value="$(to)" />
                <postfield name="subject" value="$(subject)" />
                <postfield name="mailtext" value="$(mailtext)" />
            '>

        </p>
    </card>
</wml>
</CFOUTPUT>

```

error.cfm

```
<!-- error.cfm --->
```

```
<CFPARAM NAME="reason" DEFAULT="unknown">
```

```
<CFOUTPUT><?xml version="1.0"?>
```

```
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
```

```
"http://www.wapforum.org/DTD/wml_1.1.xml">
```

```
<wml>
```

```
  <card id="Error" title="Error">
```

```
    <do type="prev" label="Back"><prev/></do>
```

```
    <p>
```

```
      <CFIF reason IS "wrongpass">
```

```
        Password incorrect! Please try again.
```

```
      <CFELSEIF reason IS "wronglogin">
```

```
        Login incorrect! Please try again.
```

```
      <CFELSEIF reason IS "shortname">
```

```
        You entered a too short name, please use a longer name.
```

```
      <CFELSEIF reason IS "pinsnotidentical">
```

```
        Your pincodes do not match. Please try again.
```

```
      <CFELSE>
```

```
        An unknown error has occurred.
```

```
      </CFIF>
```

```
    </p>
```

```
    <p>
```

```
      <anchor>OK<prev/></anchor>
```

```
    </p>
```

```
  </card>
```

```
</wml>
```

```
</cfoutput>
```

help.cfm

```
<!-- help.cfm --->
```

```
<CFOUTPUT><?xml version="1.0"?>  
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"  
"http://www.wapforum.org/DTD/wml_1.1.xml">
```

```
<wml>  
  <card id="Help" title="Helpt">  
    <do type="prev" label="Back">  
      <prev/>  
    </do>  
    <p>Sorry, no help available yet.</p>  
    <p>  
      <anchor>OK<prev/></anchor>  
    </p>  
  </card>  
</wml>  
</CFOUTPUT>
```

index.cfm

```
<!-- index.cfm --->
```

```
<!-- decide what browser requests this page, and redirect automatically  
to wap or web page --->
```

```
<!-- if browser seems to be wap enabled, then show him wml pages... --->
```

```
<CFIF CGI.HTTP_ACCEPT CONTAINS "wml">  
  <CFCONTENT TYPE="text/vnd.wap.wml">  
  <CFINCLUDE TEMPLATE="login.cfm">
```

```
<!-- if not (wap enabled) show the web page --->
```

```
<CFELSE>  
  <CFCONTENT TYPE="text/html">  
  <CFINCLUDE TEMPLATE="web/index.cfm">  
</CFIF>
```

insertlink.cfm

```
<!-- insertlink.cfm --->
```

```
<!-- cfid en cftoken need to be passed on to make sure  
client.user_id is still there--->
```

```
<CFOUTPUT>
```

```
<postfield name="CFID" value="#CFID#"/>
```

```
<postfield name="CFTOKEN" value="#CFTOKEN#"/>
```

```
</CFOUTPUT>
```


insertuser.cfm

```
<!-- insertuser.cfm --->
```

```
<!-- inserts a new entry to the ldap server in the chosen folder --->
```

```
<CFOUTPUT><?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">
```

```
<wml>
```

```
<card id="Test" title="#ou#">
```

```
<!-- if user enters to this page normally, delete variables, that might be --->
```

```
<onevent type="onenterforward">
```

```
<refresh>
```

```
<setvar name="name" value=""/>
```

```
<setvar name="mail" value=""/>
```

```
<setvar name="tel" value=""/>
```

```
</refresh>
```

```
</onevent>
```

```
<!-- if user enters from the saving page, put him/her back to main menu --->
```

```
<onevent type="onenterbackward">
```

```
<prev/>
```

```
</onevent>
```

```
<!-- back button --->
```

```
<do type="prev" label="Back"><prev/></do>
```

```
<!-- form --->
```

```
<p>Name: <input name="name" type="text" title="Name" value=""/></p>
```

```
<p>Email: <input name="mail" type="text" title="Email" value=""/></p>
```

```
<p>Tel: <input name="tel" type="text" title="Tel" format="*N" value=""/></p>
```

```
<!-- Save Button --->
```

```
<p>
```

```
<CF_OPTION TYPE="anchor" LABEL="Save"
HREF="insertuser_save.cfm"
extra='
```

```
<postfield name="ou" value="#ou#"/>
```

```
<postfield name="cn" value="$(name)"/>
```

```
<postfield name="mail" value="$(mail)"/>
```

```
<postfield name="telephonenumber" value="$(tel)"/>
```

```
'>
```

```
</p>
```

```
</card>
```

```
</wml>
```

```
</cfoutput>
```

insertuser_save.cfm

```
<!-- insertuser_save.cfm --->
```

```
<CFPARAM NAME="ou" DEFAULT="">
```

```
<CFLDAP
```

```
  SERVER="#LDAPSERVER#"
```

```
  USERNAME="#LDAPUSERNAME#"
```

```
  PASSWORD="#LDAPPASSWORD#"
```

```
  ACTION="ADD"
```

```
  ATTRIBUTES="objectclass=top, usereintrag;
```

```
    cn=#cn#;
```

```
    mail=#mail#;
```

```
    telephonenumber=#telephonenumber#"
```

```
DN="cn=#cn#,ou=#ou#,ou=#Client.WapSignUsername#,ou=wapsign,o=zhwin.ch">
```

```
<CFOUTPUT><?xml version="1.0"?>
```

```
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
```

```
"http://www.wapforum.org/DTD/wml_1.1.xml">
```

```
<wml>
```

```
  <card id="Main" title="#ou#">
```

```
    <do type="prev" label="Back"><prev/></do>
```

```
    <p>Entry has been added!</p>
```

```
    <p>
```

```
      <anchor>OK<prev/></anchor>
```

```
    </p>
```

```
  </card>
```

```
</wml>
```

```
</cfoutput>
```

list.cfm

```
<!-- list.cfm --->
```

```
<CFLDAP SERVER="#LDAPSERVER#"
  USERNAME="#LDAPUSERNAME#"
  PASSWORD="#LDAPPASSWORD#"
  ACTION="query"
  NAME="name1"
  START="ou=#Client.WapSignUsername#,ou=wapsign,o=zhwin.ch"
  SCOPE="ONELEVEL"
  ATTRIBUTES="ou,objectclass"
>
```

```
<CFOUTPUT><?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">
```

```
<wml>
  <card id="Test" title="Directorys">
    <do type="prev" label="Back"><prev/></do>
    <p>
  </CFOUTPUT>
```

```
<CFOUTPUT QUERY="name1">
  <CFIF objectclass CONTAINS "organizationalunit">
    <a href="users.cfm?ou=#ou#&#passuser#">#ou#</a>
    <br/>
  </CFIF>
</cfoutput>
```

```
<CFOUTPUT>
  </p>
</card>
</wml>
</cfoutput>
```

login.cfm

```
<!-- login.cfm --->
```

```
<CFPARAM NAME="username" DEFAULT="">
<CFIF ParameterExists(Client.WapSignUsername)>
  <CFSET username=WapSignUsername>
</cfif>
```

```
<CFOUTPUT><?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">
```

```
<wml>
<CFIF username IS "">
  <card id="main" title="wap.sign.ch">
    <do type="prev" label="Back">
      <prev/>
    </do>
    <p>
      <a href="##login">Login</a>
      <br/>
      <a href="create.cfm">Create account</a>
      <br/>
      <a href="help.cfm">Help</a>
      <br/>
    </p>
  </card>
</CFIF>
```

```
<card id="login" title="Login">
  <onevent type="onenterforward">
    <refresh>
      <setvar name="username" value="#username#"/>
      <setvar name="pincode" value=""/>
    </refresh>
  </onevent>
  <do type="prev" label="Back">
    <prev/>
  </do>
  <p>Username: <input name="username" type="text" title="Username"
value="#username#"/></p>
  <p>Pincode: <input name="pincode" type="password" title="Pincode"
format="*N" value=""/></p>
```

```
<p>
  <anchor>Login
  <go href="validate.cfm">
    <postfield name="username" value="$(username)"/>
    <postfield name="pincode" value="$(pincode)"/>
    <CF_INSERTLINK>
  </go>
</anchor>
</p>

</card>
</wml>
</CFOUTPUT>
```

menu.cfm

```
<!-- menu.cfm --->
```

```
<CFOUTPUT><?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">
```

```
<wml>
  <card id="Test" title="#Client.WapSignUsername#">
    <do type="prev" label="Back"><prev/></do>
    <p><a href="list.cfm?#passuser#">Directorys</a></p>
    <p><a href="email.cfm?#passuser#">Email</a></p>
    <p><a href="help.cfm?#passuser#">Help</a></p>
  </card>
</wml>
</cfoutput>
```

option.cfm

```
<!-- option.cfm --->
```

```
<!-- usage: <CF_OPTION TYPE="do|anchor|both" EXTRA="additional postfields">
--->
```

```
<!-- extra information to be passed on as post variables --->
```

```
<cfparam name="attributes.extra" default="">
```

```
<!-- does the information appear as DO, as ANCHOR or BOTH ways --->
```

```
<cfparam name="attributes.type" default="both">
```

```
<cfoutput>
```

```
<cfif (attributes.type is "both") or (attributes.type is "do")>
```

```
<do type="accept" label="#attributes.label#">
```

```
<go href="#attributes.href#">
```

```
    <CF_INSERTLINK>
```

```
    #attributes.extra#
```

```
</go>
```

```
</do>
```

```
</cfif>
```

```
<cfif (attributes.type is "both") or (attributes.type is "anchor")>
```

```
<anchor>#attributes.label#<go href="#attributes.href#">
```

```
    <CF_INSERTLINK>
```

```
    #attributes.extra#
```

```
</go>
```

```
</anchor>
```

```
<br/>
```

```
</cfif>
```

```
</cfoutput>
```

sendmail.cfm

```
<!-- sendmail.cfm -->

<CFIF TO IS "Other Email">
  <CFINCLUDE TEMPLATE="email.cfm">
  <CFABORT>
</CFIF>

<!-- get the email address of the user -->
<CFLDAP SERVER="#LDAPServer#"
  USERNAME="#LDAPUSERNAME#"
  PASSWORD="#LDAPPASSWORD#"
  ACTION="query"
  NAME="getemail"
  START="ou=#Client.WapSignUsername#,#LDAPStart#"
  SCOPE="ONELEVEL"
  FILTER="(cn=#Client.WapSignUsername#)"
  ATTRIBUTES="cn,ou,unserpasswort,mail"
>

<cfmail server="mail.zhwin.ch"
  to="#TO#"
  from="#getemail.mail#"
  subject="#subject#">
#mailto#
</cfmail>

<CFOUTPUT>
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">
<wml>

  <card id="main" title="Send Mail">
    <p>
      Mail sent!
    </p>
    <p><anchor>OK<prev/></anchor></p>
  </card>
</wml>
</CFOUTPUT>
```


user_detail.cfm

```
<!-- user_detail.cfm -->
```

```
<CFPARAM NAME="ou" DEFAULT="">
```

```
<CFLDAP SERVER="#LDAPSERVER#"
    USERNAME="#LDAPUSERNAME#"
    PASSWORD="#LDAPPASSWORD#"
    ACTION="query"
    NAME="name1"
    START="ou=#ou#,ou=#Client.WapSignUsername#,ou=wapsign,o=zhwin.ch"
    FILTER="(cn=#cn#)"
    SCOPE="ONELEVEL"
    ATTRIBUTES="cn,mail,telephonenumber"
>
```

```
<CFOUTPUT><?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">
```

```
<wml>
  <card id="User" title="#cn#">
    <do type="prev" label="Back"><prev/></do>
</CFOUTPUT>
```

```
<CFOUTPUT QUERY="name1">
  <p>Tel: #telephonenumber#</p>
```

```
  <p>Email: <a href="email.cfm?to=#mail#&#passuser#">#mail#</a></p>
</cfoutput>
```

```
<CFOUTPUT>
  </card>
</wml>
</cfoutput>
```

user.cfm

```
<!-- users.cfm -->

<!-- define the content type to be wap -->
<CFCONTENT TYPE="text/vnd.wap.wml">
<!-- no cache -->
<CFHEADER NAME="Cache-Control" VALUE="no-cache">
<!-- enable only cfoutput -->
<CFSETTING ENABLECFOUTPUTONLY="Yes">

<CFPARAM NAME="ou" DEFAULT="">

<CFLDAP SERVER="#LDAPSERVER#"
        USERNAME="#LDAPUSERNAME#"
        PASSWORD="#LDAPPASSWORD#"
        ACTION="query"
        NAME="name1"
        START="ou=#ou#,ou=#Client.WapSignUsername#,ou=wapsign,o=zhwin.ch"
        SCOPE="ONELEVEL"
        ATTRIBUTES="cn"
        SORT="cn"
    >

<CFOUTPUT><?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">

<wml>
    <card id="Test" title="#ou#">
        <do type="prev" label="Back"><prev/></do>
    </CFOUTPUT>

<CFOUTPUT QUERY="name1">
        <p><a
href="user_detail.cfm?cn=#URLEncodedFormat(cn)#&ou=#ou#&#passuser#>#cn#</a>
        </p>
    </cfoutput>

<CFOUTPUT>
        <p>
            <CF_OPTION LABEL="New Entry"
href="insertuser.cfm?ou=#ou#&#passuser#" TYPE="both">
        </p>
    </card>
</wml>
</cfoutput>
```

validate.cfm

```
<!-- validate.cfm -->

<!-- the cfldap has to be in a CFTRY block, because a non existing
object throws an exception (in case of wrong username) -->
<CFTRY>
  <CFLDAP SERVER="#LDAPServer#"
    USERNAME="#LDAPUSERNAME#"
    PASSWORD="#LDAPPASSWORD#"
    ACTION="query"
    NAME="checkpassword"
    START="ou=#username#,#LDAPStart#"
    SCOPE="ONELEVEL"
    FILTER="(cn=#username#)"
    ATTRIBUTES="cn,ou,unserpasswort"
  >
  <CFCATCH TYPE="any">
    <!-- somethings wrong, suppose the username is not in the ldap -->
    <CFSET reason="wronglogin">
    <CFINCLUDE TEMPLATE="error.cfm">
    <CFABORT>
  </CFCATCH>
</CFTRY>

<CFIF checkpassword.RecordCount IS 1>
  <CFIF checkpassword.unserpasswort IS pincode>
    <!-- Password correct, set uservariable and redirect -->
    <CFSET Client.WapSignUsername=username>
    <CFINCLUDE TEMPLATE="menu.cfm">
    <CFABORT>
  <CFELSE>
    <!-- Password seems incorrect -->
    <CFSET reason="wrongpass">
    <CFINCLUDE TEMPLATE="error.cfm">
    <CFABORT>
  </CFIF>
</CFIF>

<CFELSE>
  <!-- somethings wrong with the login, i.e. no username found -->
  <CFSET reason="wronglogin">
  <CFINCLUDE TEMPLATE="error.cfm">
  <CFABORT>
</cfif>
```

A.3. Glossar:

WAP Server:

Eigentlich nichts anderes als ein normaler Web Server, der Dokumente via http Protokoll ins Internet liefert. Nokia stiftet zusätzlich Verwirrung, indem Ihr WAP Gateway Produkt den Namen WAP Server trägt. Ein Web Server hat die Aufgabe Content (also Inhalt) zu liefern (to serve = englisch für servieren).

WAP Gateway:

Ein WAP Gateway verbindet die WAP Endgeräte (meist Mobiltelefone) mit dem Internet oder dem WAP Server (Content Server). Die Aufgabe des Gateways ist, die WML Seiten im binären (kompilierten) Format ans WAP Endgerät zu senden. Das Gateway erhält vom Endgerät ebenfalls Informationen, sowie die Identifikation des Endgerätes, die Identifikation der SIM Karte (Telefonnummer) und bald auch die geographische Position des Mobiltelefons. Natürlich muss das Gateway auch die Anfragen, die vom Endgerät kommen entgegen nehmen und verarbeiten.

WML Card:

Ein Block im Sinne einer grafischen Navigationseinheit. Mehrere solcher Karten können in einem Spiel (engl. Deck) zusammengefasst sein.

WML Deck:

Sammlung von einer oder mehreren WML Cards. Das WML Deck bildet auch die physikalische Einheit einer WML Datei. Es ist möglich via einem URL Parameter direkt eine bestimmte WML Card in einem Deck anzuspringen.

SGML:

Standardized Generalized Markup Language

User Agent:

Das WAP Endgerät verfügt über ein Programm, das die WML Seiten interpretiert und darstellt. Das ist dann ein User Agent.

WML- Script:

Angelehnt an JavaScript, aber viel mehr Restriktionen und viel weniger Funktionalität.

XML:

Die eXtensible Markup Language wurde von W3C standardisiert und vorgeschlagen. WML ist ein Subset von XML. Und XML wiederum ist in SGML definiert.

HDML:

Handheld Markup Language

DTD:

Document Type Definition. Definiert welche Elemente mit welchen anderen vermischt werden dürfen, oder wie und in welcher Anzahl sie relativ zueinander stehen dürfen. Es wird weiter definiert, welche Tags auch geschlossen werden müssen und welche Attribute und Werte erlaubt sind.

A.4. Links

Online Dokumentation:	http://wap.sign.ch/
[1] Allaire:	http://www.allaire.com/
[1] ColdFusion:	http://www.ColdFusion.com/
[2] Omnicron Technologies	http://www.omnicron.ab.ca/
[3] Netscape Directory Server	http://home.netscape.com/directory/v4.0/
[4] Nokia	http://www.nokia.com/
W3C (World Wide Web Consortium) :	http://www.w3c.org/