Fachhochschule Nordwestschweiz

Hofackerstrasse 73

CH-4132 Muttenz

Diplomarbeit

angewandte Informatik, Internet Engineering & E-Business

Webbased Design of Heat Exchanger Systems

Autor: Manuel Schneider

Auftraggeber: HES Heat Exchanger Systems GmbH

Dozent: Bradley Richards

Abgabetermin: 21.12.2007

Diese Diplomarbeit widme ich jenen Personen, welche mich bei meinem Entschluss zum Studium und in dessen Verlauf auf vielfältige Weise unterstützt haben:

Meinen Eltern und der Familie Meller.

Inhaltsverzeichnis

1 Anforderungen

Die Firma HES – Heat Exchanger Systems baut Wärmetauscher, welche an Industriebetriebe in der ganzen Welt verkauft werden. Der Vertrieb geschieht oftmals über Vertriebsagenten, Tochterfirmen im Ausland oder Partnerfirmen.

Für die Konstruktion des Kernprodukts der Firma, den Spiralwärmetauschern, existiert im Gegensatz zu anderen Wärmetauschersystemen keine öffentlich erhältliche Auslegungssoftware. Aus diesem Grund wurden in der Firma HES zwei Auslegungsprogramme intern entwickelt, eine in Pascal geschriebene DOS-Applikation sowie eine Excel-Lösung.

Den externen Agenten und Mitarbeitern soll eine Auslegungssoftware zur Verfügung gestellt werden. Um Software-Updates zu vereinfachen und -Diebstahl zu verhindern möchte der Auftraggeber diese Software als Webapplikation realisiert haben.

1.1 Sicherheit

Es ist zu befürchten, dass Mitarbeiter externer Partner und Agenten Passwörter und Zugangsdaten mitnehmen oder an Unbefugte weitergeben könnten. Daher wird vom Auftraggeber verlangt, dass der Login in die Weboberfläche zusätzlich mit Hardware – ähnlich einem Dongle – abgesichert wird. Diese Hardware wird den Benutzern zur Verfügung gestellt und kann mittels einer Garantieübereignung abgesichert werden. Verliert der Benutzer das Gerät, so erhält die Firma HES einerseits eine finanzielle Entschädigung, ausserdem hat der betreffende Benutzer selbst keinen Zugriff mehr. Es ist daher davon auszugehen, dass die Anwender dafür Sorge tragen werden, den Dongle nicht in fremde Hände geraten zu lassen.

1.2 Funktionsumfang

Die Software soll von einem Anwender eine Zahl Parameter entgegennehmen um einen Vorschlag zu errechnen, wie ein thermisches Problem mittels eines Spiralwärmetauschers gelöst werden kann.

Hierbei gibt es drei Problemstellungen:

- Kühlen: Vorgegeben sind die Parameter der heissen Seite
- Wärmen: Vorgegeben sind die Parameter der kalten Seite
- Wärmerückgewinnung: Vorgegeben sind die Parameter beider Seiten

In einem iterativen Verfahren – welches das spezielle Know-How der Firma HES darstellt – kann ein Lösungsvorschlag errechnet werden. Dieser soll dem Benutzer ausgegeben werden.

Der Benutzer soll ausserdem die Möglichkeit haben bereits getätigte Berechnungen abzuspeichern um diese später erneut durchführen oder abändern zu können. Es wäre hierbei sinnvoll, könnten Mitarbeiter von HES diese abgespeicherten Parametersätze einsehen um gegebenenfalls einem Anwender bei der Problemlösung behilflich zu sein – zum Beispiel durch das Anpassen von Parametern auf technisch plausiblere Werte.

Eine so ausgearbeitete Lösung könnte direkt aus der Webapplikation als Anfrage an die Firma HES gesandt werden. Dem Ingenieur bei HES lägen somit alle notwendigen Daten vor.

Für die Zukunft – vom Umfang her vermutlich nicht in dieser Diplomarbeit zu realisieren – wäre eine direkte Preisberechnung denkbar.

1.3 Infrastruktur

Die Webapplikation soll auf einem internen Webserver laufen, welcher öffentlich erreichbar ist.

Als Server bietet sich der in der Firma lokal vorhandene Linux-Router an, auf dem bereits ein Apache-Webserver und MySQL-Datenbankserver installiert sind.

Der Rechner ist über eine Dial-Up-ADSL-Verbindung mit dem Internet verbunden. Durch dynamische DNS-Updates bei der Einwahl ist der Router unter einem statischen Hostnamen aus dem Internet erreichbar.

Zur Programmierung würden sich daher verschiedene Script-Sprachen oder CGI-Programme eignen. Da ein Administratorzugang besteht, wäre es auch möglich eine zusätzliche Webserver-Software zu installieren, erscheint dies anhand der Auswahl der Programmiersprache sinnvoll.

1.4 Beschreibung der Spiralwärmetauscher

Spiralwärmetauscher dienen zum Wärmeaustausch zwischen Flüssigkeiten, Gasen oder Schlämmen.

1.4.1 Mechanischer Aufbau

Zwei Bleche (Coil) werden an ein Kernrohr angeschweisst. Die beiden Bleche werden dann das Kernrohr herum aufgewickelt. Dadurch bilden sich zwischen den Wicklungen zwei Kanäle, durch welche die Medien durchfliessen können. Um eine konstante Kanalbreite sicherzustellen, werden beim Wickeln Bolzen auf das Coil aufgeschweisst. Je höher die Belastung des Kanals, desto kleiner muss dabei das Bolzenraster gewählt werden. Ein kleines Bolzenraster erhöht den Druckverlust des Wärmetauschers, zudem wird die Verwirbelung des Mediums und somit die Turbulenz im Kanal erhöht.

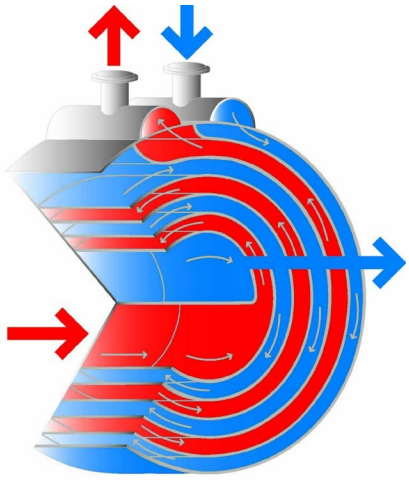


Abbildung 1: Blick auf die Spirale

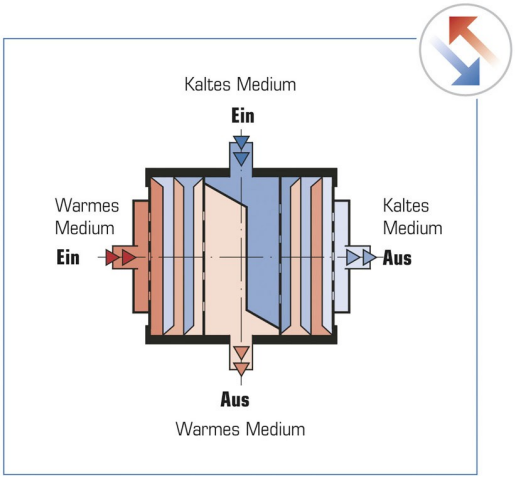


Abbildung 2: Apparat im Querschnitt

Quelle Abbildungen 1 und 2: HES-Firmenprospekt

1.4.2 Verfahrenstechnische Auslegung der Apparate

Der Temperaturübergang zwischen den Medien ist von vielen Faktoren abhängig, welche von der Verfahrenstechnik berechnet und berücksichtigt werden müssen. So setzen bestimmte Medien entsprechend widerstandsfähige Materialien voraus, hoher Druck erfordert dickes Blech, dies wiederum hat einen direkten Einfluss auf den Wärmeübergang. Die Medien haben unterschiedliche Stoffeigenschaften:

Der Wärmeübergang ist von der Fliessgeschwindigkeit abhängig, welche durch die Breite und Höhe des Kanals bestimmt ist. Ausserdem sind die Wärmekapazitäten und Wärmeleitkoeffizienten spezifisch für jedes Medium temperaturabhängig.

Aus diesen Parameter – für beide Stoffe mit deren jeweiligen Eingangs- und Ausgangstemperaturen – werden die zur Auslegung benötigten verfahrenstechnischen Faktoren berechnet.

Der Wärmeübergang ist physikalisch bestimmt durch die Wärmeleitfähigkeit der beteiligten Medien. Bei Festkörpern ist dieser Wert konstant, weshalb bei Werkstoffen auf ein vordefinierter Wert – je nach Material unterschiedlich – zurückgegriffen werden kann. Bei Fluiden ist die Wärmeleitfähigkeit nicht nur vom Stoff abhängig. Geschwindigkeit und Turbulenz beeinflussen den Wärmeübergang zusätzlich. Diese sind wiederum abhängig von der Viskosität des Fluids, welche sich mit der Temperatur ändert. Der Massenstrom definiert sich physikalisch als Volumen pro Zeiteinheit, wobei das Volumen eines Stoffes sich mit der Dichte und somit ebenfalls mit der Temperatur verändert.

Die Stoffdaten werden daher in eine mathematische Gleichung eingesetzt, woraus die Reynolds-Zahl – die Strömung des Mediums im Kanal – berechnet werden kann. Die Reynolds-Zahl wird in die Gleichung zur Berechnung der Prandtl-Zahl eingesetzt, um die

Temperaturleitfähigkeit in Abhängigkeit der Viskosität anzugeben. Die Reynolds- und Prandtl-Zahl werden weiterhin für die Bestimmung der Nusselt-Zahl als Korrekturfaktor des Wärmeübergangkoeffizienten benötigt. Anschliessend kann α berechnet werden. Es gibt den mittleren Wärmeübergangkoeffizienten des strömenden Mediums an, durch den der Gesamtwiderstand der Wärmeübertragung K berechnet werden kann.

Der Prozess, für den der Wärmetauscher eingesetzt werden soll, definiert ausserdem Rahmenbedingungen wie Druckverlust und die erforderliche Wärmeleistung. Anhand derer werden die Kanalhöhen (Abstand der Coils zueinander), -Breiten (Breite des Coils, das heisst die Höhe des Apparats) und -Längen (Länge des Coils beziehungsweise des gewundenen Kanals) bestimmt. Hierbei hat jede Änderung eines Parameters Auswirkung auf den gesamten Apparat und seine Kapazität.

Abbildung 3: Spiralwärmetauscher

Quelle Abbildung 3: HES-Firmenprospekt

2 Analyse

Die Infrastruktur – das System, auf dem die Webapplikation laufen soll – ist bereits vorhanden. Es gilt nun zu entscheiden, welche Programmiersprache eingesetzt werden soll. Diese Entscheidung hängt auch von dem eingesetzten Webframework sowie dem Vorhandensein einer API für die Sicherheitslösung ab. Dementsprechend wurde die Reihenfolge für die Entscheidungen wie folgt festgelegt:

- Evaluation Sicherheitslösung
- Evaluation Programmiersprache / Framework

2.1 Evaluation der Sicherheitslösung

Da Webapplikationen selbst nicht direkt auf eventuell vorhandene Hardware zugreifen können, muss eine Middleware dafür eingesetzt werden. Auf dem Markt gibt es diverse Anbieter von Sicherheitslösungen für den Zugriff auf Webapplikationen. Leider wird dafür fast immer zusätzliche Software auf dem Clientrechner vorausgesetzt, was die Lösungen meist plattform- und betriebssystemabhängig macht.

Kriterien für die auszuwählende Sicherheitslösung:

- Zuverlässigkeit
- Sicherheit
- einfache Handhabung
- Einfluss auf System / Netzwerk des Anwenders
- Unabhängigkeit von der Firewall-Konfiguration
- einfache Integration von Proxyservern
- Aufwand zur Implementierung

- Hardwarevoraussetzungen
- Betriebssystemunabhängigkeit
- Preis

Die meisten Lösungen auf dem Markt arbeiten mit USB-Sticks, auf denen ein kryptographischer Schlüssel hinterlegt ist. Die Webapplikation verwendet als Schnittstelle zur Hardware entweder einen eigenen, vom Hersteller mitgelieferten Browser oder ein PlugIn für einen vorhanden Browser. Alternativ zum USB-Stick bieten manche Hersteller auch Smart-Cards oder Dongles für die Druckerschnittstelle an, die Funktionsweise ist jedoch gleich.

Da Browserplugins nicht nur betriebssystem- sondern zugleich auch software- und versionsabhängig sind, wurden diese nicht weiter berücksichtigt.

2.1.1 USB-Dongles

Eine Lösung, speziell für diesen Anwendungsfall entwickelt, bietet die Firma Matrix an. Da dieser Anbieter ein kostengünstiges Test-Kit verkauft, wurde beschlossen dieses Produkt zu evaluieren. Ausserdem bewirbt die Firma Matrix ihr Produkt damit, dass es auch unter Linux und Mac OS X lauffähig ist.

Mit der mitgelieferten Software können eigene Programme in diversen Programmiersprachen geschrieben werden, welche über den Dongle abgesichert werden. Für Windows wird auch ein Werkzeug mitgeliefert um einen speziellen Webclient zu bauen, eine Software welche direkt mit der Webapplikation kommuniziert und somit die Authentifizierung übernimmt.

Zur Kommunikation wird das HTTP-Protokoll verwendet, über GET-Parameter können Datenpakete, Session-IDs und Schlüssel ausgetauscht werden.

Vorteile:

- relativ einfache Implementierung
- erprobtes Produkt, welches für genau diesen Anwendungsfall Schutz verspricht
- unkompliziert zu handhaben: Die Dongles werden einmalig registriert und sind somit über ihre Seriennummer identifizierbar
- günstig: Ein Dongle kostet zwischen 20 und 30 EUR, je nach Abnahmemenge
- es stehen APIs für diverse Programmiersprachen auf Windows, Mac OS X und Linux zur Verfügung
- kein Eingriff in das Netzwerk des Anwenders
- es existiert auch eine Netzwerklösung um ein Dongle im Netzwerk an mehreren Arbeitsstationen parallel nutzen zu können, dies ist für den Fall ideal, bei dem die Verwendung der USB-Schnittstelle an den Arbeitsstationen verboten ist, oder die Firma ihren Dongle nicht an die Mitarbeiter aushändigen möchten
- es können auch Dongles für den Parallel-Port verwendet werden
- falls im Client-Netzwerk ein Proxyserver eingesetzt wird, werden die notwendigen Einstellungen des Systems (Internetoptionen) automatisch vom Webclient verwendet

Nachteile:

- eine Client-Software muss installiert werden, somit funktioniert diese Lösung nur betriebssystemabhängig (Windows)

2.1.2 VPN-Adapter

Eine andere Überlegung war, den Zugriff nur über einen SSL-Proxy mit einem entsprechend signierten SSL-Zertifikat zuzulassen. Auf der Suche nach Hardware, welche wie eine Bridge in die Netzwerkverbindung eingeschleift werden kann und dann nur die Zugriffe auf die Webapplikation anhand ihrer Zieladresse über einen dort laufenden Proxyserver routet, wurde man auf den USB VPN & Firewall Adapter von Linksys aufmerksam. Trotz intensiver Recherche schien dies das einzige Gerät dieser Bauart auf dem Markt zu sein. Da es in Europa nicht verfügbar ist, musste es zur Evaluation aus den USA beschafft werden.

Das Gerät meldet sich am Betriebssystem als USB-Netzwerkkarte an. Unter Linux kann es über die im Kernel integrierten Treiber verwendet werden. Es kommt dort das „pegasus“-Kernelmodul zum Einsatz. Für Windows werden Treiber auf CD mitgeliefert.

Das Gerät baut eine VPN-Verbindung via IPSec zum vorkonfigurierten Endpunkt auf. Es enthält einen DHCP-Server, so dass dem angeschlossene Rechner automatisch eine IP-Adresse erteilt wird. Über die statische IP-Adresse des Adapters kann auf eine Konfigurationsoberfläche zugegriffen werden um die VPN-Parameter zu ändern.

Es wäre nun denkbar, den VPN-Adapter beim Anwender als Netzwerkkarte einzusetzen und alle Zugriffe auf die Webapplikation durch das VPN zu routen. Der betreffende virtuelle Host auf der Serverseite dürfte dann nur durch das VPN erreichbar sein. Dadurch würde garantiert, dass Benutzer nur über eine VPN-Verbindung dort zugreifen könnten. Da die Zugangsdaten vor Versand des Geräts dort eingespeichert und dem Anwender nicht bekannt gemacht werden hat dieser ohne die Hardware keine Möglichkeit den Schutz zu umgehen.

Vorteile:

- absolut plattform- und betriebssystemunabhängig
- keine Client-Software notwendig
- neben dem Zugriffschutz wäre zugleich auch der Datentransfer gesichert

Nachteile:

- verlangt beim Anwender eine bestimmte Infrastruktur, entweder ein zweiter Netzwerkanschluss oder die Möglichkeit, den gesamten Datenverkehr über den Adapter abzuwickeln, was in LANs aufgrund der Firewall-Funktionalität des Adapters Schwierigkeiten verursachen kann
- die Geräte sind relativ teuer und nur in den USA erhältlich, es scheint bislang auch nur einen Hersteller zu geben. Das zur Evaluation beschaffte Testgerät kostete 90 EUR inklusive Versandkosten.
- der Aufbau eines IPSec-Routers ist relativ aufwendig, so man nicht auf bereits fertige Hardware-Router zurückgreift
- benötigt eine funktionierende USB-Schnittstelle an der betreffenden Arbeitsstation
- benötigt unter Umständen Änderungen an der Firewall-Konfiguration im Client-Netzwerk, damit IPSec-Pakete geroutet werden

2.1.3 Internet Smart Card (Giesecke & Devrient)

Giesecke & Devrient bietet auf seiner Webseite eine Smart Card an, welche einen Webserver enthält. Gemäss Beschreibung kann dieser zum sicheren Speichern von Benutzerdaten verwendet werden, aber auch zum sicheren Zugriff auf Webseiten. Da das Gerät am USB-Port als Netzwerkkarte erkannt wird ist es plattformunabhängig zu verwenden. Die Software läuft auf der CPU der SmartCard und kann somit vom Anwender nicht manipuliert werden.

Vorteile:

- keine spezielle Treiber- oder Clientsoftware notwendig, daher plattform- und betriebssystemunabhängig
- kein Eingriff in das Netzwerk des Anwenders
- verschlüsselte Kommunikation über den USB-Stick
- keine Firewall-Konfiguration notwendig, da die Kommunikation via HTTPS erfolgt

Nachteile:

- benötigt eine funktionierende USB-Schnittstelle an der betreffenden Arbeitsstation
- Kosten und Implementierungsaufwand unbekannt, trotz mehrerer Anfragen keine Reaktion von Giesecke & Devrient
- Falls im Clientnetzwerk ein Proxyserver eingesetzt wird, muss die Software auf der SmartCard entsprechend individuell angepasst werden.

2.1.4 Auswertung

Kriterium	**Dongles**	**VPN-Adapter**	**Smart Card**	**Gewicht**
Zuverlässigkeit	+	-	?	*4*
Sicherheit	+	++	++	*4*
einfache Handhabung	-	+	++	*3*
Einfluss auf System / Netzwerk des Anwenders	++	-	++	*3*
Unabhängigkeit von der Firewall-Konfiguration	++	-	+	*3*
einfache Integration von Proxyservern	++	+	-	*2*
Aufwand zur Implementierung	++	+	?	*2*
Hardwarevoraussetzungen	+	-	-	*1*
Betriebssystemunabhängigkeit	-	++	++	*1*
Preis	++	-	?	*1*
Ergebnis	**31**	**17**	**25**	

„+“ - erfüllt „++“ - besonders herausragend

„-“ - nicht erfüllt „?“ - keine Angaben erhältlich (nicht erfüllt)

2.1.5 Entscheidung

Nach Vorlage der oben angegebenen Daten und Argumenten entschied sich die Firma HES für den Einsatz der USB-Dongles. Die Einschränkung durch die Systemvoraussetzungen erschienen als nicht so gewichtig im Vergleich zu den zu erwarteten Schwierigkeiten durch den Einsatz der VPN-Adapter in einem Firmennetzwerk. Auch ist zu erwarten, dass der Implementierungsaufwand geringer ist als bei den anderen beiden Lösungen.

2.2 Evaluation der Programmiersprachen

Durch die Wahl der Hardwarelösung hat sich keine Abhängigkeit zur verwendenden Programmiersprache ergeben. Daher stehen weiterhin alle in der Einleitung genannten Optionen zur Verfügung.

Kriterien zur Auswahl der Programmiersprache:

- Sicherheit und Stabilität
- Performanz
- wenig zusätzlich zu installierende Software auf dem Server
- einfache Wartbarkeit des Codes / der Webapplikation
- schnelle Implementierung
- Wiederverwendbarkeit von bestehendem Code
- geringer Lernaufwand

2.2.1 C++-Webapplikation mit tntnet / tntdb

Mit tntnet steht ein stabiler, hochperformanter Webserver für C++-Webapplikationen zur Verfügung. Es würde sich anbieten die Webapplikation in C++ zu entwickeln, die Anbindung an eine Datenbank liesse sich dann via tntdb realisieren. Tntdb ist eine Datenbank-Abstraktionsschicht zum Zugriff auf sqlite (Filedatenbank), MySQL, PostgreSQL und Oracle. Ein Sessionmanagement und SSL-Verschlüsselung sind ebenfalls bereits implementiert.

2.2.2 Scriptsprache PHP

Auf dem Serversystem ist die Scriptsprache PHP bereits im Einsatz. Sie hat gegenüber C++ den Vorteil, dass sie sofort eingesetzt werden kann, das heisst die Einarbeitung in tntnet, cxxtools und tntdb entfällt. Des weiteren entfällt der Compilationsprozess, die Applikation kann direkt im Kontext des bestehenden Apache Webservers laufen, Änderungen an den Programmdateien sind umgehend sichtbar.

Die Nachteile einer Scriptsprache sind dagegen, dass sie verglichen mit einem Compilat, wie das einer in C++ geschrieben Applikation, in der Ausführungsgeschwindigkeit deutlich langsamer sind. Gerade tntnet profitiert durch seine enorme Performanz. Der Nachteil kann durch die deutlich höhere Entwicklungsgeschwindigkeit, der geringeren Komplexität des Codes und somit der einfacheren Wartbarkeit des selbigen aufgewogen werden.

Zusätzlich wird PHP zur Laufzeit auf dem System durch den Einsatz des Zend-Encoders und -Caches auf Geschwindigkeit optimiert. Dabei wird der Code vom PHP-Interpreter durch eine Erweiterung des selben als vorcompilierter Zwischencode, ähnlich eines Java-Compilats in einem Cache gespeichert. Somit steigert sich die Ablaufgeschwindigkeit des normalerweise interpretierten PHP-Codes um ungefähr den Faktor 10.

Ein weiterer Vorteil des Einsatzes von PHP ist, dass für die

Grundstruktur der Webseite – wie die grafische Darstellung der Webseite, die Sessionverwaltung und der Datenbankzugriff auf eigene Vorarbeiten von der HES-Webseite zurückgegriffen werden kann. Das bei der Entwicklung der HES-Webseite entstandene Framework könnte verwendet und weiterentwickelt werden um den Ansprüchen des Auftrages zu genügen. Dabei würde die Webapplikation ohne Zusatzaufwand dem Corporate Design der Firma HES entsprechen.

2.2.3 Auswertung

Kriterium	**PHP**	**tntnet**	**Gewicht**
Sicherheit und Stabilität	+	++	*4*
Performanz	+	++	*3*
wenig zusätzlich zu installierende Software auf dem Server	++	-	*2*
einfache Wartbarkeit des Codes / der Webapplikation	++	+	*3*
schnelle Implementierung	++	-	*4*
Wiederverwendbarkeit von bestehendem Code	++	-	*3*
geringer Lernaufwand	++	-	*1*
Ergebnis	**32**	**17**	

„+“ - erfüllt „++“ - besonders herausragend

„-“ - nicht erfüllt „?“ - keine Angaben erhältlich (nicht erfüllt)

2.2.4 Entscheidung

Wie aus der Auswertung klar hervorgeht, hat tntnet seine Stärken durchaus im technologischen Bereich: Die Applikationen sind schnell, und durch die vorhandenen Bibliotheken auch sicher. Die zwingende Objektorientierung und Möglichkeiten der Sprache C++ bieten auch die Grundlage für elegantes Software-Design. Die verfügbaren Komponenten erlauben auch in kleinem Rahmen die Wiederverwertbarkeit von Code beziehungsweise entlasten den Programmierer vom Erstellen eigener Komponenten. Diesen Vorteil bietet allerdings das bereits selbst entwickelte Framework in noch grösserem Masse, da in tntnet noch kein entsprechender eigener Code zur Verfügung steht, welcher wiederverwendet werden könnte.

Wie beschrieben kann auch PHP durch den Einsatz des Encoders und Caches sehr performant sein. Es ist ausserdem schon auf dem betreffenden Server im Einsatz, tntnet und seine Abhängigkeiten müssten hier zuerst manuell installiert werden.

Grösstes Argument für die Verwendung von PHP ist allerdings die Tatsache, dass das vorhandene Framework eine Eigenentwicklung ist. Ein ausreichend grosses Vorwissen über die Programmiersprache ist bereits vorhanden, es müssen keine Handbücher studiert und neue Bibliotheken getestet werden. Im Bezug auf die begrenzte Zeit der Diplomarbeit ein wichtiger Vorteil, den diese Programmiersprache hier bietet.

3 Entwurf

Als Basis für die Entwicklung der Webapplikation benötigt man eine Infrastruktur. Diese soll die Standard-Funktionen der Webapplikation übernehmen, welche bei jedem Seitenzugriff benötigt werden. In diese Infrastruktur / Framework kann dann die Webapplikation als Modul eingehängt werden, ohne dass beim Programmieren der Module auf die Standardfunktionen Rücksicht genommen werden muss.

Folgende Module – aufgeteilt nach Funktionalität – werden benötigt:

- Anwenderdaten (superglobale Variable, Parameter) einlesen und filtern
- Benutzerdaten bereitstellen (Hostname, IP-Adresse, Spracheinstellungen)
- Datenbankverbindung
- Zugriffe auf nicht vorhandene Module abfangen
- Sprache ermitteln / Sprachdatei laden
- Sessionmanagement
- Design
- ggf. Fehlerausgabe
- Protokollierung von Transaktionen und Fehlern

Da die Zugriffe mit einer Hardwarelösung abzusichern sind, wird diese direkt in das Sessionmanagement eingebunden. Der Login-Vorgang wird durch die PIN-Eingabe der Webclient-Software ersetzt, die Benutzerkonten werden durch die Dongles, respektive ihrer Seriennummer und dem damit verbundenen Dongle-Key abgebildet.

3.1 Sicherheitslösung / Sessionmanagement

Die Funktionsweise des Sessionmanagements ist vom Hersteller der Dongles vorgegeben.

Es gibt vier Aktionen, welche über durch die Webclient-Software auf dem Dongle ausgeführt werden können, die Antwort wird als HTTP-Request des Webclients an den Webserver weitergereicht. Die Webclient-Software agiert dabei als Proxyserver: Um eine Aktion aufzulösen, muss vom Benutzer ein URL auf http://localhost:8096/ aufgerufen werden. Per Parameter wird die gewünschte Aktion übertragen, je nach Aktion werden noch Timeout, Krypto-Schlüssel oder der URL angehängt, welche nach der Aktion durch den Proxyserver aufgerufen werden sollen.

Die Aktionen:

- **clientOk** – es wird ein beliebiger 32 Bit-Schlüssel (Session-Key), eine Session-ID und der URL der Webapplikation übertragen. Diese Daten werden im Webclient gespeichert und für zukünftige Aktionen verwendet. Als Antwort wird „clientOk“ gesandt, woran die Webapplikation erkennen kann, dass der Webclient läuft. Da der Krypto-Schlüssel zurückgesendet wird, kann auch geprüft werden, ob der Client wie erwartet reagiert.

- **enterPin** – der Webclient wird gebeten innerhalb eines per Parameter definierten Zeitfensters die PIN des Benutzers abzufragen. Der zuvor initialisierte Session-Key wird zerlegt und mit der PIN und der Seriennummer des Dongles mit Hilfe des im Dongle gespeicherten Schlüssel (Dongle-Key) verschlüsselt. Zusätzlich wird der Session-Key und die Seriennummer im Klartext übertragen. Schlägt die PIN-Eingabe fehl, wird anstelle der Seriennummer eine Fehlernummer <= 0 übertragen. Um die Daten zu entschlüsseln – um Session-Key, PIN und Seriennummer zu prüfen – benötigt die

Webapplikation den Dongle-Key. Dieser muss bei der Einrichtung des Benutzers auf dessen Dongle als auch in der Benutzerdatenbank der Applikation gespeichert werden. Mit der mitgelieferten Software des Dongle-Herstellers kann dieser Schlüssel im Dongle geschützt gespeichert werden. Es ist anschliessend nicht mehr möglich diesen Schlüssel zu extrahieren. Da er bei der Authentifizierung nicht übertragen wird, sondern beiden Partner im Voraus bekannt ist, kann er während der Datenübertragung nicht abgehört werden.

- **readData** und **writeData** – mit diesen beiden Aktionen können Daten im Dongle abgespeichert und gelesen werden. Auch hier wird mit dem Dongle-Key verschlüsselt, zusätzlich wird der verschlüsselte UserCode übertragen. Der UserCode identifiziert den Eigentümer der Dongles und ist geheim zu halten, er wurde bei der Herstellung des Dongles kundenspezifisch, aber unveränderbar, gespeichert. Da beide Funktionen nicht eingesetzt werden, wird an dieser Stelle nicht weiter darauf eingegangen.

3.1.1 Zustandstabelle

Um das Sessionmanagement abzubilden müssen wir hauptsächlich sechs Parameter in einem Entscheidungsbaum abfragen. Im Detailentwurf werden später noch weitere Parameter geprüft um gegebenenfalls mit einer Fehlermeldung entsprechend zu reagieren.

Zur Entwicklung des Ablaufes und zur Vereinfachung des Entscheidungsbaumes wurde folgende Tabelle erstellt. Es ist eine Anlehnung an ein Karnaugh-Veitch-Diagramm, welches hier aufgrund der vielen Parameter nicht verwendet werden kann.

Parameter						**Zustand**
PHPSESSID gesetzt	**PHPSESSID in DB**	**action = ClientOk**	**action = pinEntered**	**status > 0**	**S/N in DB**	
-						Fall 1: Login Start, SID erzeugen und ClientOk abfragen
+	-					Fehlermeldung: Session ungültig
+	+	-	-			Fehlermeldung: Webclient läuft nicht
+	+	-	+	-		Fehlermeldung: Dongle nicht vorhanden
+	+	-	+	+		Fall 3: Login erfolgreich, S/N in DB speichern
+	+	+			-	Fall 2: PIN / S/N abfragen
+	+	+			+	Fall 4: gewünschte Seite anzeigen

Die genannten Fälle sind in den Kommentaren des Quellcodes wieder entsprechend bezeichnet.

„+“ erfüllt

„-“ nicht erfüllt

leer egal (don't care / nicht relevant)

3.1.2 Ablaufdiagramm

Der übliche Ablauf - vom ersten Aufruf der Webseite bis zur erfolgreichen Anmeldung – besteht aus vier Schritten:

1. erster Aufruf der Webapplikation. Es gibt keine Session-ID; daher werden keine weiteren Parameter geprüft. Es wird nun eine Session-ID und ein Session-Key erzeugt und in der Datenbank gespeichert. Der Benutzer wird umgeleitet, so dass der Webclient diese beiden Daten über die Aktion „clientOk“ erhält und somit initialisiert wird.

2. Die Webapplikation wird nun mit einer Session-ID aufgerufen, welche in der Datenbank gefunden wird. Die Antwort des Webclients lautet „clientOk“ und der gespeicherte Session-Key entspricht dem übertragenen. Noch besitzt die Session aber keine Seriennummer, der Benutzer ist also noch nicht identifiziert. Über eine Umleitung auf den lokalen URL des Webclients mit der Aktion „enterPIN“ wird die PIN-Abfrage initiiert.

3. Beim nächsten Aufruf erhält die Webapplikation erneut eine Session-ID, welche in der Datenbank gefunden werden kann. Die Antwort des Webclients ist aber nicht „clientOk“ sondern „pinEntered“. Der Statuswert ist grösser 0, somit war die PIN-Abfrage erfolgreich und eine Seriennummer wurde übertragen. Das verschlüsselte Datenpaket wird zerlegt, mit dem in der Datenbank gespeicherten Dongle-Key entschlüsselt und dann verglichen. Ist alles in Ordnung wird die Seriennummer in der Session abgespeichert, der Benutzer ist angemeldet und erhält die gewünschte Seite angezeigt.

4. Alle zukünftigen Zugriffe werden durch den Webclient getunnelt, um bei jedem Zugriff über „clientOk“ das Vorhandensein des Webclients zu prüfen. Somit besitzt jeder Zugriff auf die Webapplikation eine gültige Session-ID, die Antwort „clientOk“

sowie eine der Session zugeordnete Seriennummer.

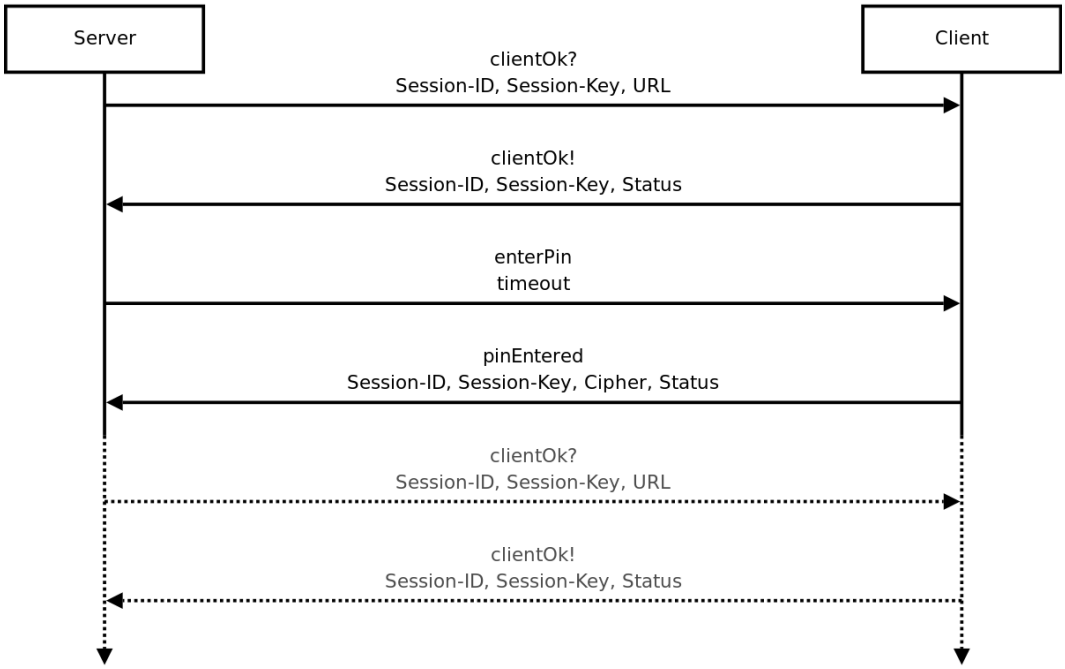


Abbildung 4: Ablaufdiagramm Login

3.2 Protokollierung

Ein Protokoll erleichtert während der Entwicklung die Fehlersuche und überwacht im Produktivbetrieb das System.

Vom Datenmodell lehnt sich die Implementierung an das bewährte Syslog unixoider Betriebssysteme an. Gespeichert werden folgende Daten:

- **timestamp** – Datum und Uhrzeit
- **facility** – auslösendes Modul
- **level** – Wichtigkeit der Meldung
- **text** – Inhalt
- **object** – Objekttyp, falls vorhanden
- **oid** – Objekt-Referenz, sofern ein Objekttyp angegeben wurde

Mittels „timstamp“, „facility“ und „text“ können die wichtigsten Daten erfasst werden: Wann und wo welches Ereignis im System aufgetreten ist. Mit „level“ kann zusätzlich nach der Ausführlichkeit der Nachrichten gefiltert werden.

Dies geschieht anhand folgender Tabelle:

Level	**Art der Meldung**
1	Fehler - wichtig
2	Warnung
3	Information
4	Debugging – alle Ausgaben

Mittels „object“ und „oid“ kann ein Objekt mit dem Logeintrag referenziert werden. Dies ist dafür gedacht, in der Weboberfläche Benutzern oder Administratoren die Möglichkeit zu geben zu Dongles, laufenden Sessions oder Berechtigungen die jeweils protokollierten Daten einzusehen. Diese Angaben sind fakultativ.

Das Logging-Modul besteht aus einer Funktion, welche im Datenbank-Modul definiert wird. Sie nimmt die Parameter entgegen und überprüft bei Angabe eines Objekttyps ob das entsprechende Objekt gemäss „oid“ in der Datenbank wirklich existiert. Ist dies nicht der Fall wird der Eintrag ohne Objektreferenz gespeichert.

3.3 Berechtigungssystem

Um die Verwaltung der Benutzer, Firmen, Sessions und Logeinräge nicht in eine externe Webapplikation auszulagern, wurde ein Berechtigungssystem entworfen.

Für jede Verwaltungstätigkeit benötigt man ein bestimmtes Recht („privilege“). Besitzt man dieses Recht nicht, ist diese Aktion implizit verboten. Bei einigen Aktionen wird zwischen Anzeige („get“) und Veränderung („set“) unterschieden. Ebenso wird unterschieden zwischen Berechtigungen auf die Objekte der Firma, der man selbst angehört oder dem Recht alle Objekte einer Art zu verwalten. Letztere Rechte sind für einzelne Mitarbeiter der Firma HES vorgesehen, welche damit eine Administratoren-Funktion übernehmen.

Rechte:

- **getOrgCalcs** – erlaubt den Zugriff auf Berechnungen aller Benutzer der Firma, der man selbst angehört
- **getAllCalcs** – erlaubt den Zugriff auf alle Berechnungen
- **setOrgDongles** – erlaubt das Verwalten der Dongles der eigenen Firma
- **setAllDongles** – erlaubt das Verwalten aller Dongles
- **setOrgRights** – erlaubt das Bearbeiten der Berechtigungen der eigenen Firma. Es können nur Berechtigungen verliehen werden, die man als Benutzer selbst besitzt.
- **setAllRights** – erlaubt das Bearbeiten aller Firmen. Es können nur Berechtigungen verliehen werden, welche man als Benutzer selbst besitzt.
- **getOrgSessions** – erlaubt das Einsehen der laufenden Sessions der

eigenen Firma

- **getAllSessions** – erlaubt das Einsehen aller laufenden Sessions
- **setOrgSessions** – erlaubt das Bearbeiten (Beenden) von Sessions der eigenen Firma
- **setAllSessions** – erlaubt das Bearbeiten (Beenden) aller Sessions
- **getLog** – erlaubt das Einsehen des Protokolls

Die Berechtigungen werden im Modul Session-Management nach der erfolgreichen Validierung einer Session in die Benutzerdatenstruktur als boolesche Werte eingelesen, so dass diese Daten später in jedem Modul zur Verfügung stehen.

Im Laufe der Entwicklung wurden für einzelne Module weitere Berechtigungsstufen notwendig. So verwendet der Auslegungsalgorithmus Stammdaten bezüglich der Materialfestigkeit und der Stoffeigenschaften der eingesetzten Medien. Um auch hier den selben Komfort wie bei der Benutzer- und Sessionverwaltung bieten zu können wurden zwei weitere Berechtigungen eingeführt:

- **setMaterial** – erlaubt das Bearbeiten von Werkstoffen, das heisst das Ändern und Hinzufügen von Materialien und deren Belastbarkeitskennwerten
- **setMedium** – erlaubt das Bearbeiten und hinzufügen von Stoffen und deren Stoffdaten wie Dichte, dynamische Viskosität, spezifische Wärmekapazität und Wärmeleitfähigkeit.

3.4 Datenbankmodell

Folgende Entitäten werden gemäss der Auftragsbeschreibung benötigt:

- **Benutzer / Dongles:**
  - Seriennummer (Primärschlüssel, 10stelliger String)
  - Firma (Fremdschlüssel, Integer)
  - PIN (5stelliger String)
  - Kryptoschlüssel (32stelliger String)
  - Name des Benutzers (255stelliger String)
  - Aktiv (Boolean)
- **Firmen:**
  - ID (Primärschlüssel, Integer)
  - Name (255stelliger String)
  - Adresse (255stelliger String)
  - ZIP-Code (20stelliger String)
  - Ort (255stelliger String)
  - Land (255stelliger String)
- **Session:**
  - Session ID (Primärschlüssel, 10stelliger String)
  - Seriennummer (Fremdschlüssel, 10stelliger String)
  - Sessionkey (32stelliger String)

  - Zeitstempel (datetime)
  - IP-Adresse (15stelliger String)
  - Seite (20stelliger String)
- **Berechtigungen:**
  - Seriennummer (Primärschlüssel, Fremdschlüssel, 10stelliger String)
  - Berechtigung (Primärschlüssel, 15stelliger String)
- **Protokoll:**
  - ID (Primärschlüssel, Integer)
  - Zeitstempel (datetime)
  - Quelle (20stelliger String)
  - Level (5stelliger String)
  - Text (255stelliger String)
  - Objekttyp (10stelliger String)
  - Objekt-ID (Fremdschlüssel, Integer)
- **Berechnung:**
  - Angebotsnummer (Primärschlüssel, 10stelliger String)
  - Revision (Primärschlüssel, Integer)
  - Datum (date)
  - Anfragenummer (255stelliger String)
  - Positionsnummer (10stelliger String)

  - Kunde (255stelliger String)
  - Ansprechpartner (255stelliger String)
  - Autor (Fremdschlüssel, 10stelliger String)
  - ... weitere Materialdaten
- **Materialkenndaten:**
  - Name (Primärschlüssel, 25stelliger String)
  - Temperatur (Primärschlüssel, Integer)
  - Belastbarkeit (Fliesskommazahl)
- **Stoffdaten:**
  - Name (Primärschlüssel, 100stelliger String)
  - Typ (Primärschlüssel, 3stelliger String)
  - Temperatur (Primärschlüssel, Fliesskommazahl)
  - Wert (Fliesskommazahl)
- **Spline:**
  - Name (Primärschlüssel, Fremdschlüssel, 100stelliger String)
  - Intervall Start (Primärschlüssel, Fliesskommazahl)
  - Faktor a (Fliesskommazahl)
  - Faktor b (Fliesskommazahl)
  - Faktor c (Fliesskommazahl)
  - Faktor d (Fliesskommazahl)

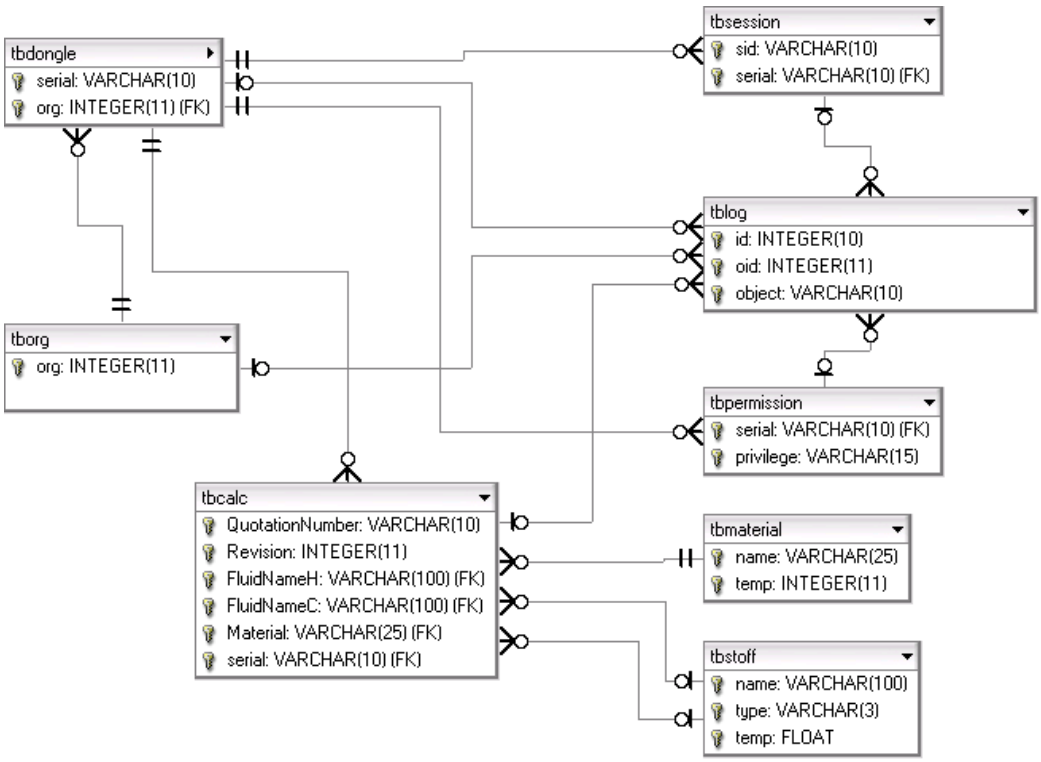


Abbildung 5: ER-Modell

3.5 Programmierstil

Um die Übersichtlichkeit und Wartbarkeit des Codes sicherzustellen wurden folgende Regeln aufgestellt:

Der Einrückstil orientiert sich an Allman. Dabei werden öffnende Klammern jeweils auf der selben Zeilen wie Schleifen- oder Funktionsköpfe geschrieben, schliessende Klammern auf in der selben Spalte wie Funktions- oder Schleifenkopf.

Bei runden Klammern (Parameterlisten) befindet sich zwischen Klammer und Inhalt jeweils ein Leerzeichen. Die öffnende Klammer befindet sich jeweils ohne Leerzeichen direkt hinter dem Funktionsnamen. Für Parameterlisten gelten die üblichen Regeln der Typographie.

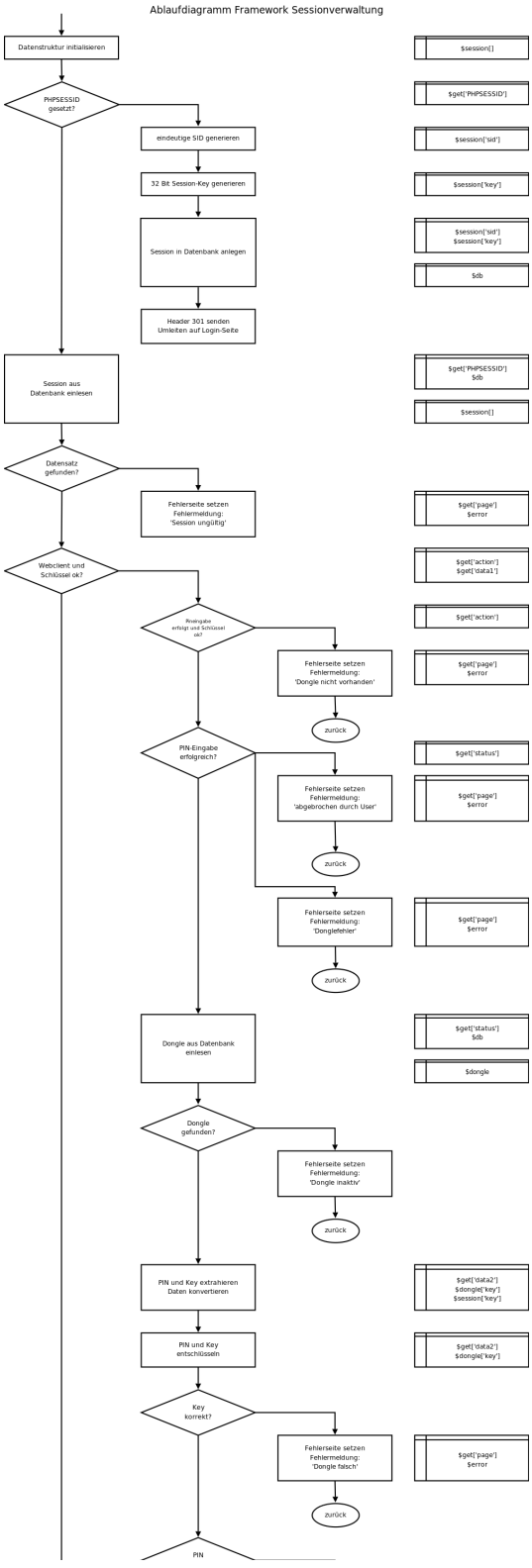
In Gleichungen werden die Elemente durch Leerzeichen getrennt, dies gilt auch für mathematische Symbole.

Der Code ist mit Kommentaren versehen. Diese entsprechen textgenau den Beschreibungen in den im Anhang dieser Diplomarbeit befindlichen Ablaufdiagrammen.

Bei Ausgaben wurde darauf geachtet wo möglich auf doppelte Anführungszeichen zu verzichten und einfache zu verwenden. Das Argument liefert die Performanz: Eine mit doppelten Anführungszeichen umschriebene Zeichenketten wird vom Interpreter vollständig geparst, da Variablen und ANSI-Steuerzeichen eingesetzt beziehungsweise umgesetzt werden müssen. Bei einer in einfachen Anführungszeichen umschriebenen Zeichenkette geschieht dies nicht. Zur Ausgabe von Variablen müssen diese daher explizit mit dem Punkt-Operator mit der Zeichenkette verknüpft werden. Die Geschwindigkeit der Ausgaben steigt dadurch erheblich.

```
  # Datenstruktur initialisieren #
  unset( $session );
  sql( 'DELETE FROM tbsession WHERE NOW()
- timestamp > 300 AND serial LIKE
NULL;' );

  # PHPSESSID gesetzt?
  if( $get['PHPSESSID'] ) {
    # PHPSESSID ist gesetzt
    logger( 'Session', 4,
$get['PHPSESSID'].' PHPSESSID found', '',
0 );
    # Session aus Datenbank einlesen
    $r = sql( 'SELECT * FROM tbsession
WHERE sid = '.$get['PHPSESSID'].';' );
    # Datensatz gefunden?
    if( $session = mysql_fetch_assoc( $r )
) {
      # Session gefunden
      logger( 'Session', 4,
$get['PHPSESSID'].' session found', '',
0 );
      # Webclient und Schlüssel ok?
      if( $get['action'] == 'clientOk' &&
$get['data1'] == $session['key'] ) {
        # Webclient ist ok
        logger( 'Session', 4,
$get['PHPSESSID'].' clientOk', '', 0 );
        # Seriennummer bekannt?
        sql( 'SELECT serial FROM tbdongle
WHERE serial = "'.$session['serial'].'"
AND active = 1;' );
        if( $session['serial'] &&
mysql_affected_rows( $db ) ) {
          # Seriennummer ist bekannt
          logger( 'Session', 4,
$get['PHPSESSID'].' serial known', '',
0 );
          ## Fall 4 - Session in Ordnung
          # Sessiondaten in Array
aktualisieren
          $session['login'] = true;
          $session['link'] =
'http://127.0.0.1:8096/?site=secure.hes-
kapp.de/&ssid='.$session['sid'].'&data1='.
```



Beispiel: Auszug von Code und Ablaufdiagramm der Sessionverwaltung

4 Framework und Bibliotheken

Wie bei der Evaluation der Programmiersprachen erwähnt, baut die Webapplikation auf ein selbstgeschriebenes Framework auf. Dieses Framework trennt einerseits die Programmierung vom Design der Seite, stellt aber auch ein System zur Modularisierung der Routine-Aufgaben der Webapplikation dar. So sind Funktionalitäten wie Datenbankverbindungen, das Einlesen und Säubern von Parametern, das Session-Management sowie die mehrsprachige Benutzeroberfläche in einzelne Module ausgelagert und werden vom Framework bei jedem Seitenaufruf entsprechend eingebunden, ohne dass man als Entwickler eines einzelnen Moduls dafür Sorge tragen muss.

Das Framework sorgt für einen zuverlässigen Modul-Kontext: Eine bestehende Datenbankverbindung, die mit einer internen Funktion genutzt werden kann. Über verschiedene Datenstrukturen liegen Informationen über Benutzer, Session und Parameter in einer Form vor, wie sie direkt zum Beispiel auch in SQL-Abfragen verwendet werden können, ohne dass die Gefahr von Cross-Site-Scripting-Attacken (XSS) oder dem Einschleusen von Code in SQL-Abfragen (SQL-Injections) besteht.

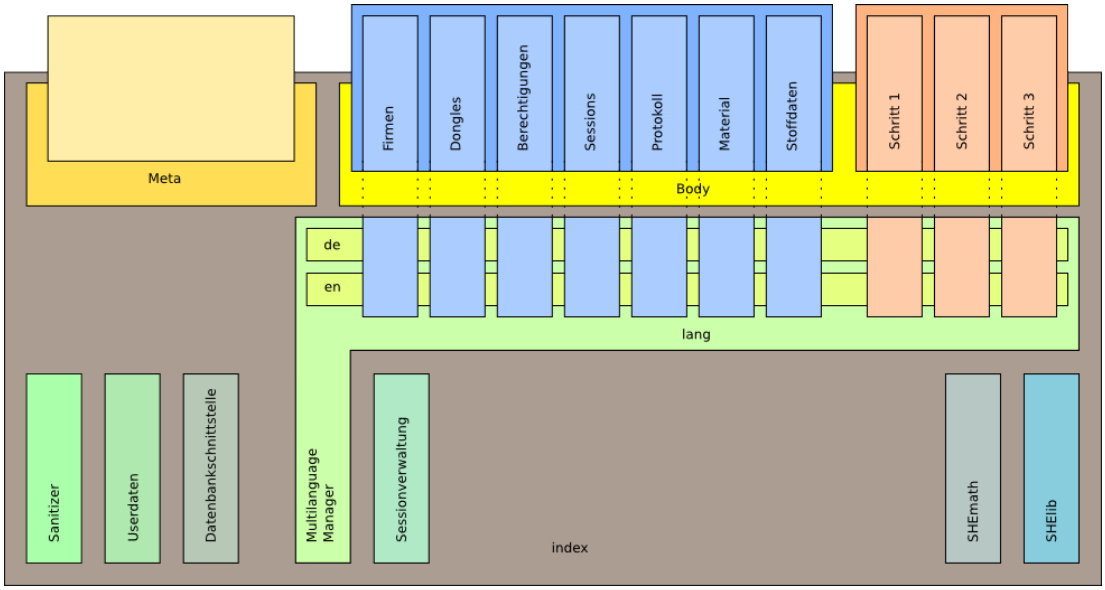


Abbildung 6: Aufbau der Webapplikation

4.1 Struktur des Frameworks

- **body/** – Verzeichnis welches alle adressierbaren Module enthält.
  - **body/index.php** – Ausgabe der Designelemente und Einbindung des Menüs sowie des angefragten Moduls (und gegebenenfalls weitere Nebenmodule wie Sidebars) an der im Design vorgesehenen Stelle.
- **images/** – Verzeichnis mit statischem Inhalt (Bildern, Grafiken)
- **lang/***xx*/ – Verzeichnis mit Unterverzeichnissen gemäss Sprachkürzel für alle unterstützen Sprachen. Die Sprachverzeichnisse enthalten die selben Dateien wie body, in denen sich die Übersetzungen für die in den jeweiligen Modulen verwendeten Zeichenketten befinden.
- **meta/** – Enthält für die Module Vorverarbeitungscode. Wird hier nicht eingesetzt, dient ansonsten beispielsweise zur Suchmaschinenoptimierung (Setzen von Meta-Daten im HTML-Kopf) oder dem Sessionmanagement (Validierung eines Logins / Zerstören einer Session bevor das Sessionmanagement selbst aktiv wird).
  - **meta/index.php** – Einbindung der Meta-Datei des angefragten Moduls und Ausgabe der HTML-Meta-Daten.
- **scripts/** – Enthält die in diesem Kapitel beschriebenen Framework-Dateien und Bibliotheken.
- **index.php** – Zentrale Datei, durch welche die Hauptfunktionen des Frameworks angestossen werden:
  1. Setzen der HTTP-Header (zum Beispiel Zeichensatz)
  2. Einbinden des *Sanitizers* zum Einlesen und Säubern der Parameter
  3. Einbinden der *Userdaten* zum Ermitteln und Bereitstellen von Benutzerdaten (zum Beispiel Spracheinstellungen des

Browsers, verwendeter VHost)

4. Einbinden des *Datenbankschnittstelle* zum Aufbau der Datenbankverbindung und Bereitstellung einer Schnittstelle
5. Prüfen ob eine Modul angefragt wurde, ansonsten Setzen der Startseite
6. Prüfen des angefragten Moduls auf seine Existenz, ansonsten Fehlerbehandlung HTTP-Fehler 404.
7. Einbinden des *Multilanguage-Managers* zur Ermittlung der Sprachversion und Laden der betreffenden Dateien
8. Ausgabe HTML-Header-Daten
9. Einbinden der Meta-Dateien zur Vorverarbeitung der Module und Ausgabe der Meta-Daten.
10. Einbinden der *Sessionverwaltung*
11. Absicherung, dass Module nur bei erfolgtem Login geladen werden, ansonsten Fehlerbehandlung HTTP-Fehler 401.
12. Einbindung des Ausgabescripts zur Ausgabe des Designs unter Einbindung des angefragten Moduls und weiterer Daten wie zum Beispiel das Menü.
13. Schliessen der Datenbankverbindung
14. Ausgabe des HTML-Fusses

– **.htaccess** – Steuerdatei für den Apache-Webserver. Wird genutzt um mit mod_rewrite eine Erweiterung des Webservers zu konfigurieren, mit welcher alle Zugriffe auf den VHost im Parameter-Format auf die Datei index.php umgeleitet werden. Diese verarbeitet dann im oben genannten Punkt 2 die Parameter intern.

4.2 Komponenten

Die Komponenten des Frameworks im Detail, in der Reihenfolge wie sie anhand der oben genannten Beschreibung verwendet und erwähnt werden.

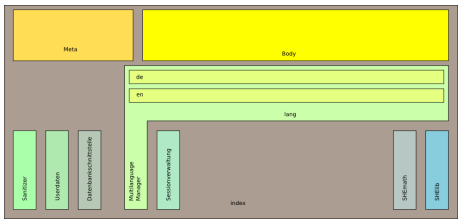


Abbildung 7: Schema Komponenten

4.2.1 Sanitizer

Die Sanitizer-Komponente zerlegt die per URL übergebenen Parameter. Die resultierenden Name-Wert-Paare werden bereinigt indem PHP- und HTML-Tags aus dem String entfernt und gefährliche Zeichen – wie Steuerzeichen, ANSI-Codes oder Anführungsstrichen – mit einem Backslash unwirksam gemacht werden. Gleichzeitig werden alle Sonderzeichen in HTML-Entitäten umgewandelt um Zeichensatz-Probleme zu vermeiden.

Derselbe Vorgang wird auf alle Elemente der globalen Arrays $_GET und $_POST angewandt, welche die GET- und POST-Parameter der HTTP-Anfrage vom Webserver erhalten haben. Alle Name-Wert-Paare werden dabei in eine interne Datenstruktur kopiert, sodass die Originalwerte nach wie vor in den Quellvariablen zur Verfügung stehen, ein Entwickler auf diese also auf eigene Gefahr nach wie vor Zugriff hätte.

4.2.2 Userdaten

Im Laufe der Verarbeitung der HTTP-Anfrage durch das Framework werden diverse Daten des Benutzers, welcher die Anfrage gestellt hat, benötigt. Dabei handelt es sich zu diesem Zeitpunkt der Verarbeitung noch nicht um einen dem System bekannten Benutzer sondern bezieht sich auf den Benutzer im Sinne des Verursachers des HTTP-Requests.

Das Userdaten-Script ist eine im Laufe der Entwicklung gewachsene Komponente und stellt alle bisher benötigten Daten in einer eigenen

Datenstruktur bereit. Dies sind bislang:

- HTTP-Host zerlegt nach
  - TLD – Toplevel-Domain
  - Domain
  - Host
- IP-Adresse
- GET-String
- URL – aufgerufener URL, intern anhand der oben genannten Daten neu generiert
- bevorzugte Sprache, Array von Sprachen sortiert nach Präferenz

Diese Daten werden zum Beispiel bei der Protokollierung (IP), beim Sessionmanagement (IP) oder vom Multilanguage-Manager (Host, TLD) verwendet. Denkbar ist auch die Erweiterung auf Referer, verwendeter Browser und so weiter.

4.2.3 Datenbankschnittstelle

Diese relativ kleine Komponente enthält die Datenbankzugangsdaten und versucht eine Verbindung zu dieser herzustellen und das resultierende Datenbank-Handle in einer Datenstruktur zu hinterlegen. Im Falle eines Fehlers wird eine entsprechende Fehlermeldung im Klartext zurückgegeben um die Fehlersuche zu vereinfachen.

Die Komponente enthält ausserdem eine Funktion um Zugriffe auf die Datenbank zu standardisieren, so dass der Entwickler keine Kenntnis über die zu verwendende Datenbank haben muss – die Funktion liefert gegen ein gültiges SQL-Statement das Handle zum Resultat zurück. Im Falle eines Fehlers wird auch hier eine Meldung ausgegeben unter Angabe der betreffenden SQL-Abfrage im Klartext um SQL-Fehler

transparenter zu machen.

Die Komponente wurde im Rahmen der Diplomarbeit ausserdem um eine Logger-Funktion erweitert, welche als Schnittstelle zur Protokollierung dient und die übergebenen Daten wie Objekttyp und -ID aus Plausibilität überprüft um den Logeintrag entsprechend in der Datenbank zu speichern, sofern der global definierte Detailgrad (Verbosity) dies erwünscht.

4.2.4 Multilanguage-Manager

Um eine mehrsprachige Benutzeroberfläche zu implementieren muss vor der Ausgabe die gewünschte Sprache ermittelt werden. Dabei gibt es mehrere Strategien mit unterschiedlichen Prioritäten.

So kann die gewünschte Sprache durch die HTTP-Anfrage bereits definiert sein, zum Beispiel weil der Anwender in der Sprachwahl der Benutzeroberfläche eine Sprache ausgewählt hat. In diesem Fall ist der betreffende Parameter zu prüfen und alle weiteren Möglichkeiten zu ignorieren.

Ist keine Sprache gesetzt, so muss diese automatisch ermittelt werden. Dazu eignen sich die Sprachpräferenzen, welche der Benutzer in seinem Browser eingestellt hat. Weiter oben wurden diese bereits eingelesen und stehen als Array bereit, absteigend nach deren Priorität sortiert. Dieses Array wird elementweise durchlaufen und jeweils geprüft, ob im Sprachverzeichnis die betreffende Sprache vorhanden ist. Wurde eine Sprache gefunden, ist die Suche beendet. Im anderen Fall kann die vom Benutzer aufgerufene Topleveldomain verwendet werden. Es kann angenommen werden, dass ein Besucher, der die Webseite unter der Topleveldomain .de aufruft, sich für die deutsche Sprache interessiert, während .com eher auf englische Sprachkenntnisse schliessen lässt. Insgesamt lässt sich bei unklaren Verhältnissen eine Default-Sprache im Modul (hier: Englisch) festlegen.

Im Rahmen der Diplomarbeit wurde der Multilanguage-Manager etwas verändert: So wird der Parameter zur Übermittlung der gewählten Sprache nicht verwendet (üblicherweise der dem Domainnamen vorangestellten Hostname als Sprachkürzel), sondern ein temporärer Parameter eingeführt, welcher nur zur Umschaltung zwischen den Sprachen dient. Die Einstellung wird dann in der Benutzerdatenbank fest gespeichert. Ist dort kein Wert eingetragen (üblicherweise nur bei neu angelegten Benutzern), so wird die Sprache wie oben beschrieben automatisch ermittelt und dort gespeichert. Allerdings kann im vorliegenden Fall auf die Entscheidung anhand der Topleveldomain nicht zurückgegriffen werden, da die Webapplikation nur unter einer Adresse erreichbar ist.

Wurde die Sprache ermittelt so werden die notwendigen Sprachdateien aus dem Sprachverzeichnis mit den Übersetzungen geladen - sowohl die globalen Dateien mit Lokalisierungseinstellungen (Währungen, Datumsformat) und dem Menü als auch für das angeforderte Modul.

4.2.5 Sessionverwaltung

In der Komponente Sessionverwaltung ist implementiert, was weiter oben unter dem selben Stichwort bereits im Entwurf ausgearbeitet und dargelegt wurde. Die gesamte Anmeldeprozedur ist hier abgebildet, inklusive der Auswertung der Parameter, Ausgabe von Fehlermeldungen sowie die notwendigen Umleitungen und Datenbank-Updates. Nach erfolgreichem Ablauf des Sessionmanagements steht eine Datenstruktur zur Verfügung, welche alle Daten den angemeldeten Benutzers beinhaltet. Somit entfallen spätere Abfragen der Datenbank. Es kann ausserdem nachgeprüft werden, ob die Anmeldung erfolgreich war, auch wenn das Sessionmanagement ausser Fehlermeldungen keine weitere Verarbeitung einer Anfrage von nicht angemeldeten Benutzer zulässt.

Zusätzlich zum Session- und Benutzerdatensatz werden ausserdem die Berechtigungen in eine zweite Dimension der Sessiondatenstruktur

eingelesen, so dass diese Daten für die Prüfung von Berechtigungen den Modulen bereits zur Verfügung stehen.

4.2.6 Bibliothek/SHEmath

Bei der Auslegung werden einige mathematische Funktionen, hauptsächlich zur Interpolation von Stoffdaten und Festigkeitskennwerten verwendet. Um den Code sauberer zu gestalten wurden diese allgemeinen Funktionen getrennt in dieser Bibliothek implementiert und ausgelagert.

Diese Bibliothek wird nur von der Bibliothek SHElib eingebunden und verwendet. Sie enthält unter anderem die extern entwickelten Funktionen zur Cubic-Spline-Interpolation.

4.2.7 Bibliothek/SHElib

In dieser Bibliothek sind alle in *Auslegung/Schritt 3* benötigten Funktionen enthalten. Sie enthält im Kern die Kernkompetenz der Firma HES als Anbieter dieser Auslegungssoftware und unterliegt daher der Geheimhaltung.

Die in SHElib enthaltenen Funktionen zur Berechnung der zum Bau und zur Kalkulation von Spiralwärmetauschern benötigten Parametern verwenden ihrerseits zum Teil allgemeine mathematische Funktionen, welche in der Bibliothek *SHEmath* ausgelagert wurden.

Diese Bibliothek wird nur vom Modul *Auslegung/Schritt 3* eingebunden und verwendet.

5 Module

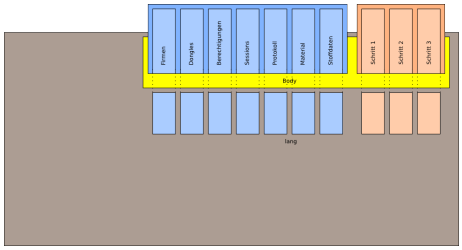


Abbildung 8: Schema Module

Die Funktionalität der Webapplikation ist in Modulen gegliedert, welche durch das Framework gestartet und dargestellt werden. Durch die typische Struktur von Webapplikationen – die Bindung einzelner Funktionalitäten an dynamische Seiten – entspricht ein Modul jeweils einer Seite, welche wiederum im Dateisystem als eine Programmdatei vorhanden ist.

Die Module sind jeweils autark voneinander, tauschen aber über Datenbankeinträgen Daten untereinander aus oder kommunizieren über GET- und POST-Parameter.

Dank des oben beschriebenen Webframeworks kann die Entwicklung der Module auch an einzelne Entwickler ausgelagert werden, da das Framework alle relevanten Daten in eigenen Datenstrukturen bereitstellt – es ist kein weiteres Wissen über andere Komponenten und Module notwendig ausser dem über die die Datenbasis an sich.

Die Beschreibung der Module erfolgt in der Reihenfolge, in der diese entwickelt wurde, da zuerst Grundlagen geschaffen werden mussten auf denen aufbauend dann weitere Funktionalität zur Applikation hinzugefügt werden konnte.

5.1 Verwaltung

Dieses Modul listet die dem Benutzer zur Verfügung stehenden Verwaltungsmöglichkeiten auf. Diese sind dann per Weblink aufrufbar.

Dazu verwendet das Modul Verwaltung die Benutzerberechtigungen. Es werden nur solche Module angezeigt, welche der betreffende Nutzer verwenden darf. Dies stellt kein grundlegender Schutz gegen Missbrauch dar, sind die Module bei Kenntnis des URLs trotzdem

aufrufbar. Die entsprechende Sicherheit wurde daher in den betreffenden Modulen selbst zusätzlich implementiert.

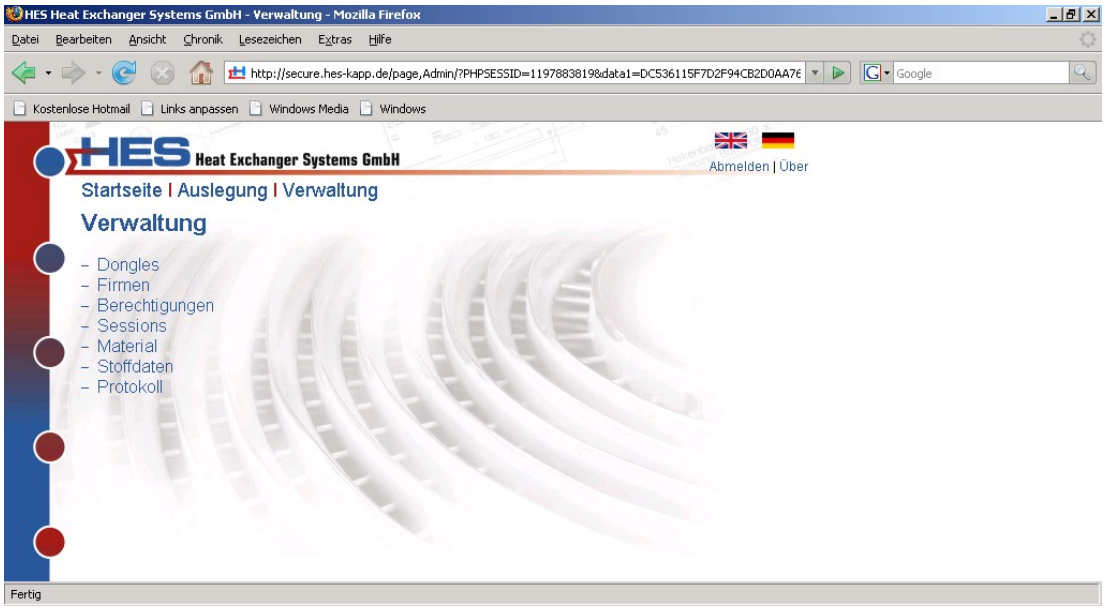


Abbildung 9: Screenshot Modul Verwaltung

5.2 Verwaltung/Firmen

Das Module Verwaltung/Firmen darf von jedem Benutzer, mit setAllDongles-Recht verwendet werden. Jeder Dongle ist einer Firma (Organisation) zugeordnet. Wer neue Dongles anlegen darf (setAllDongles), muss daher auch zuerst betreffende Firmen anlegen können.

Dieses Modul liefert eine Liste aller Firmen. Diese können jeweils als Formular bearbeitet, gespeichert und gelöscht werden. Ausserdem wird im Seitenkopf ein Formular zu Erstellung einer neuen Firma angeboten.

Wird ein Unternehmen gelöscht, so wird geprüft, ob diesem noch Dongles zugeordnet sind. In diesem Fall wird dann eine Fehlermeldung ausgegeben und der Vorgang unterbunden.

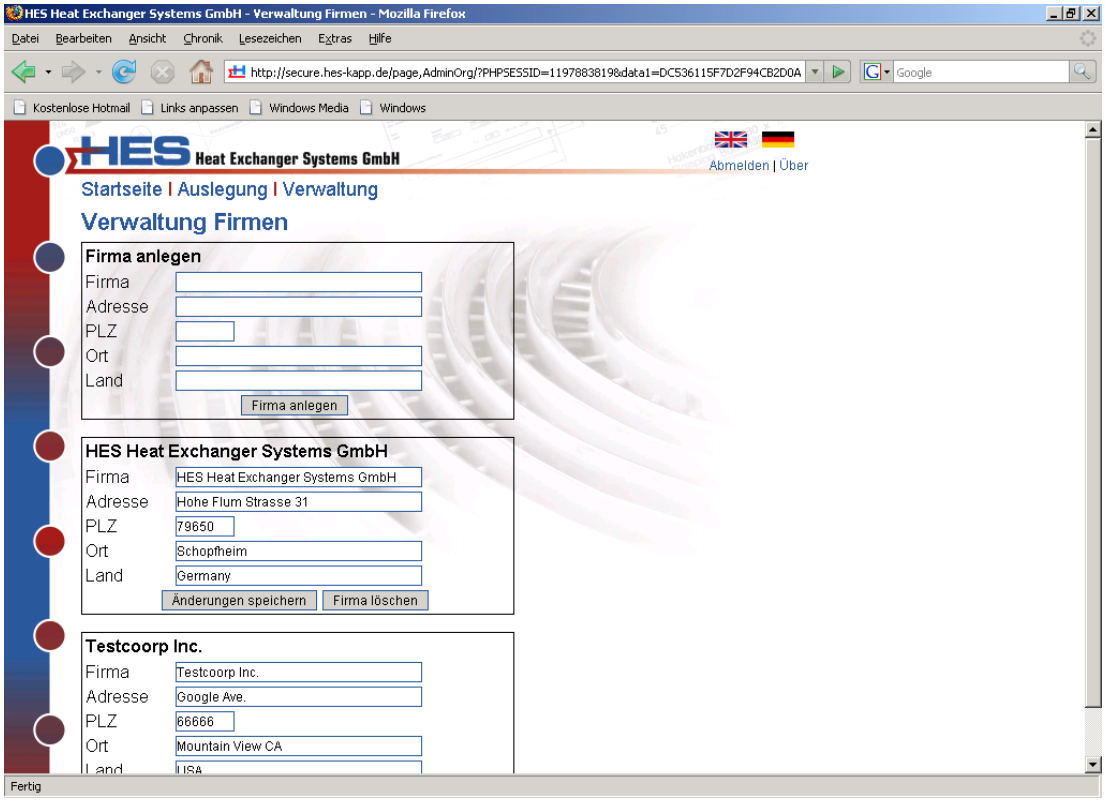


Abbildung 10: Screenshot Modul Verwaltung/Firmen

5.3 Verwaltung/Dongles

Die Verwaltung der Dongles – somit der effektiven Benutzerzugängen – erfolgt auf drei Ebenen. Zum einen gibt es Benutzer, welche das setAllDongles-Recht besitzen. Diese können über dieses Modul alle Dongles einsehen, bearbeiten und löschen sowie neue Dongles anlegen.

Benutzer mit dem setOrgDongles-Recht erhalten eine Liste der Dongles der Organisation, der sie selbst als Benutzer angehören. Sie können diese Dongles aber nur begrenzt bearbeiten, so ist es ihnen erlaubt die Namen der Benutzer, dessen PIN oder den Status (aktiv / inaktiv) des Dongles zu ändern. Daten wie Seriennummer, Dongle-Key und Firma werden nicht angezeigt oder sind nicht veränderbar.

Um hier möglichen Missbrauch zu verhindern werden nicht nur bei der

Ausgabe, sondern auch beim Speichern der Änderungen die Benutzerberechtigungen überprüft.

Benutzer, welche über keines der genannten Rechte verfügen können dieses Modul unter dem Titel „PIN ändern“ ebenfalls aufrufen, erhalten aber nur ein Formular zur Änderung ihrer PIN.

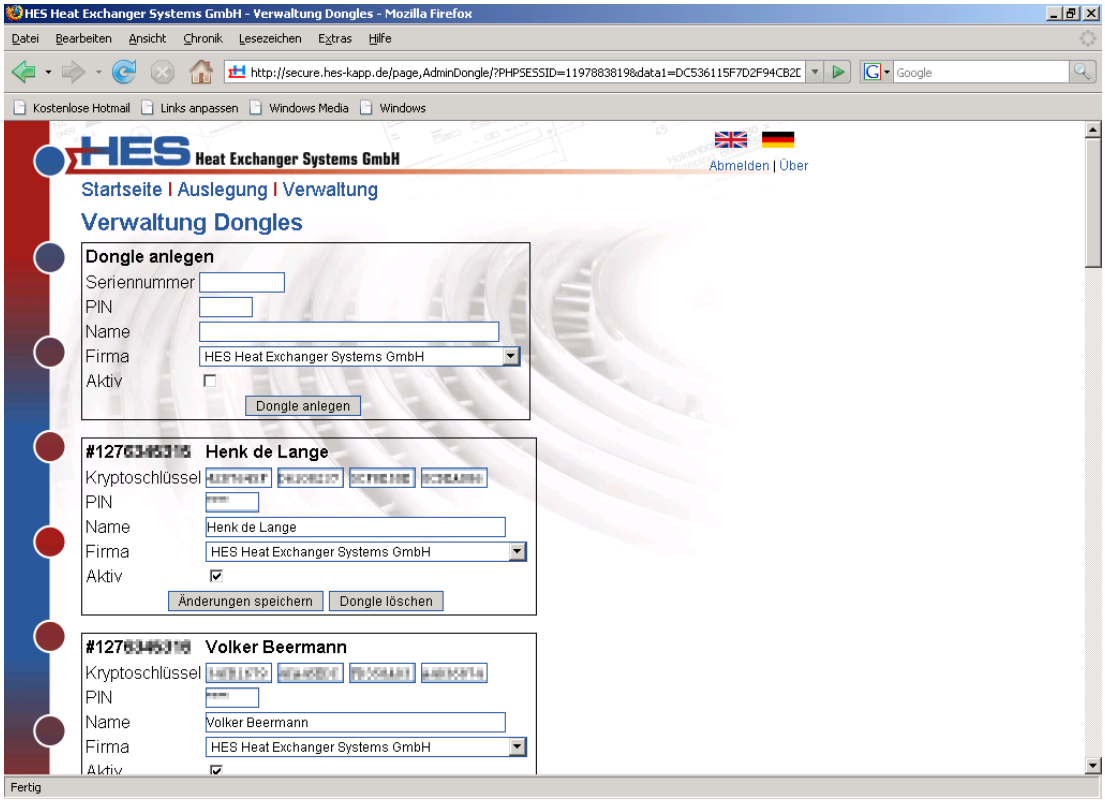


Abbildung 11: Screenshot Modul Verwaltung/Dongles

5.4 Verwaltung/Berechtigungen

Jeder Benutzer mit setAllRights-Recht darf die Berechtigungen sämtlicher Dongles verändern. In einer Liste werden alle Dongles angezeigt, daneben zu jedem Dongle die Liste aller existierender Berechtigungen. Über ein Häkchen (Checkbox) können diese ein- und ausgeschaltet werden. Berechtigungen, welche der ausführende Benutzer selbst nicht besitzt kann dieser zwar sehen, das Feld ist

allerdings inaktiv und wird beim Speichern der Änderungen ignoriert um Missbrauch zu verhindern.

Benutzer mit setOrgRights-Berechtigung erhalten nur eine Liste der Benutzer, die der selben Firma zugeordnet sind, der sie selbst als Benutzer angehören. Sie dürfen ebenfalls nur jene Berechtigungen bearbeiten, die sie selbst besitzen.

Die Änderung der eigenen Benutzerberechtigungen ist nicht möglich um zu verhindern, dass man sich selbst relevante Rechte entzieht.

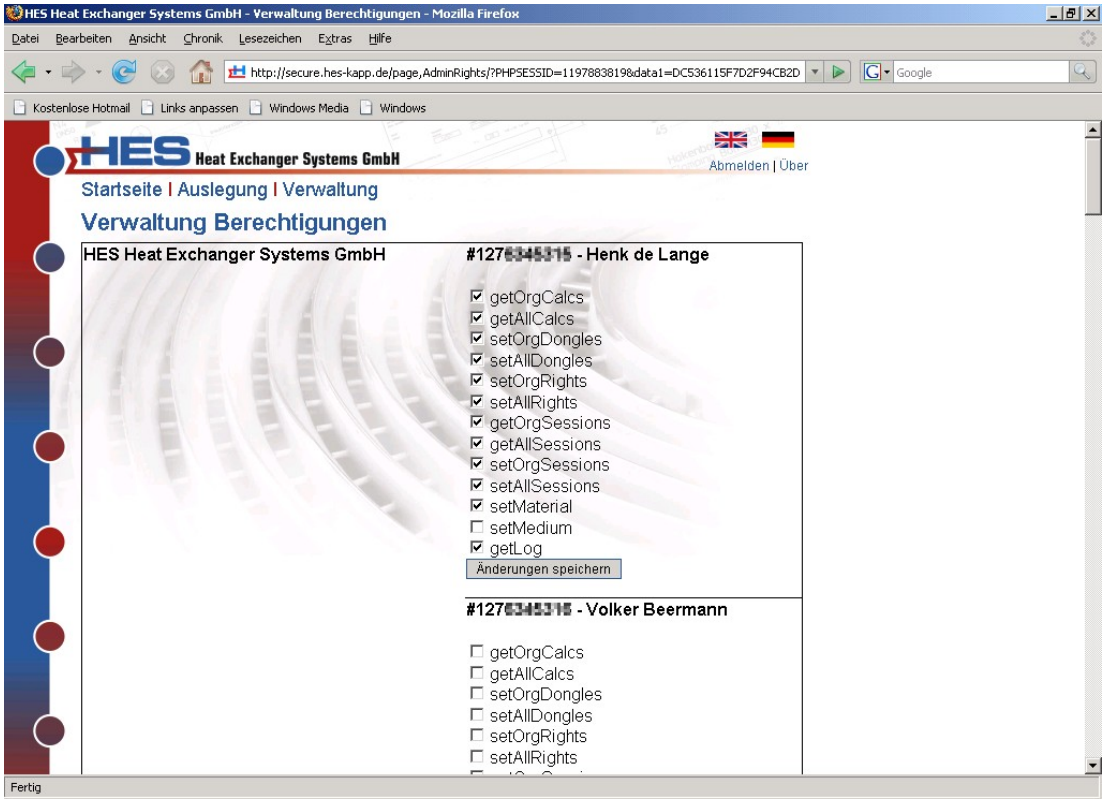


Abbildung 12: Screenshot Verwaltung/Berechtigungen

5.5 Verwaltung/Sessions

Benutzer mit setOrgSessions-Rechten können hier die aktuellen Sessions ihrer Kollegen einsehen und bei Bedarf diese beenden. Mit setAllSessions-Recht erhält man alle laufenden Sessions zur Ansicht.

Als Information erhält man neben der Seriennummer die zugehörige Firma und den Namen des Benutzers angezeigt. Ausserdem wird der Zeitpunkt des letzten Seitenaufrufs (in der Zeitzone des Servers) sowie die IP-Adresse, von welcher die Session gestartet wurde und der Namen des zuletzt aufgerufenen Moduls angezeigt.

Beim Klick auf „löschen“ wird die Session in der Datenbank gelöscht, somit erhält der betreffende Benutzer beim nächsten Seitenaufruf die Fehlermeldung „Timeout“ und muss sich neu am System anmelden.

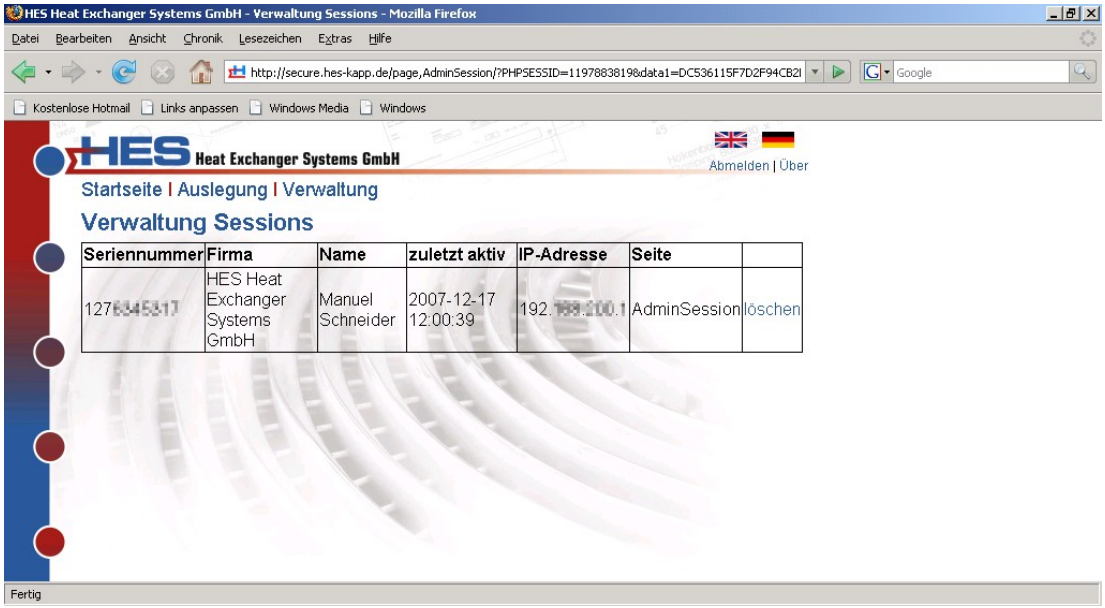


Abbildung 13: Screenshot Modul Verwaltung/Session

5.6 Verwaltung/Protokoll

Benutzer mit dem Recht getLog können hier einen Auszug der letzten Protokolleinträgen sehen.

Für spätere Erweiterungen der Webapplikation ist vorgesehen, in den anderen Verwaltungsmodulen den Link zum Protokoll bezüglich einzelner Objekte wie Sessions, Dongles oder Firmen anzubieten. Über eine Filterfunktion würde dann die Ausgabe des Protokolls entsprechend

eingeschränkt.

5.7 Verwaltung/Material

Zur Auslegung der Spiralwärmetauschern benötigt man die Eigenschaften der verwendeten Materialien, insbesondere der Belastbarkeitskennwerten. Diese sind temperaturabhängig und nicht linear. Da verschiedene Materialien zum Einsatz kommen können, wurde eine Datenbank mit den Belastbarkeitskennwerten der jeweiligen Materialien in Bezug auf unterschiedliche Temperaturen eingeführt. Jeder Nutzer mit setMaterial-Recht kann hier eine Liste aller hinterlegten Materialien einsehen, neue Materialien einfügen oder die Tabelle der Temperaturen und korrespondierenden Belastbarkeitskennwerten bearbeiten.

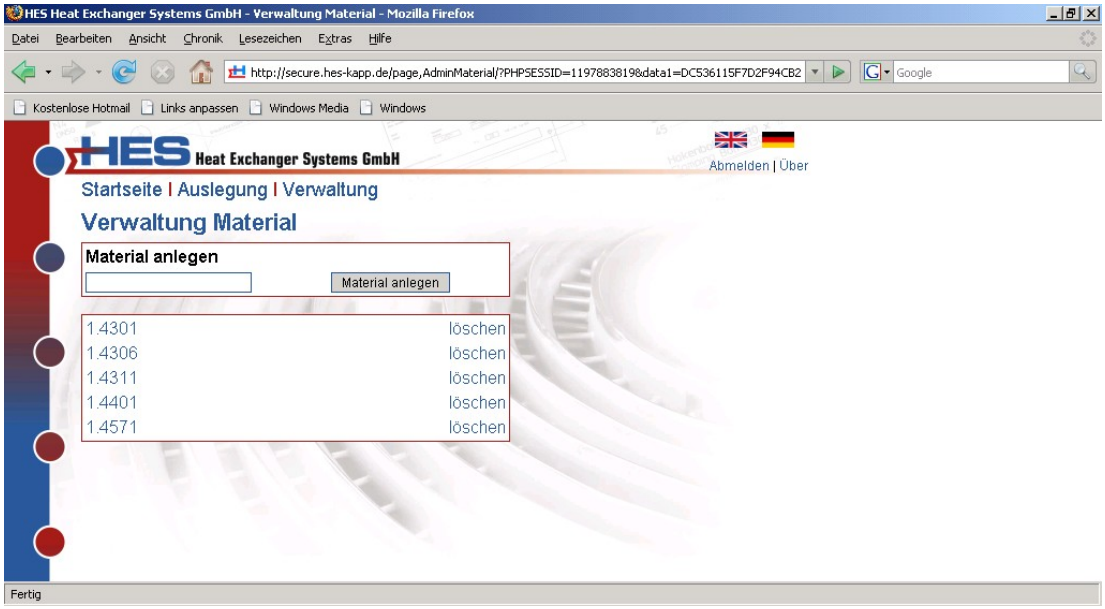


Abbildung 14: Screenshot Modul Verwaltung/Material, Übersicht

Wird eine Tabelle bearbeitet, welche noch keine Kennwerte erhält, so werden die von den TÜV-Datenblätter üblichen Temperaturen bereits vorgegeben. Somit ist es für einen Benutzer meist sehr einfach die betreffenden Kennwerte einzutragen und abzuspeichern.

Fehlende oder zu löschende Werte können in der Tabelle einfach ausgelassen werden. Beim Abspeichern werden alle unvollständigen Datensätze gelöscht. Soll ein neuer Datensatz eingefügt werden, so bietet das Modul jeweils einen Leerdatensatz am Fuss der Tabelle an. Die dort eingetragenen Werte werden abgespeichert und beim erneuten Aufruf in sortierter Reihenfolge – mit einem weiteren Leerdatensatz – angezeigt. Somit können beliebig viele Datensätze je Material verwendet werden.

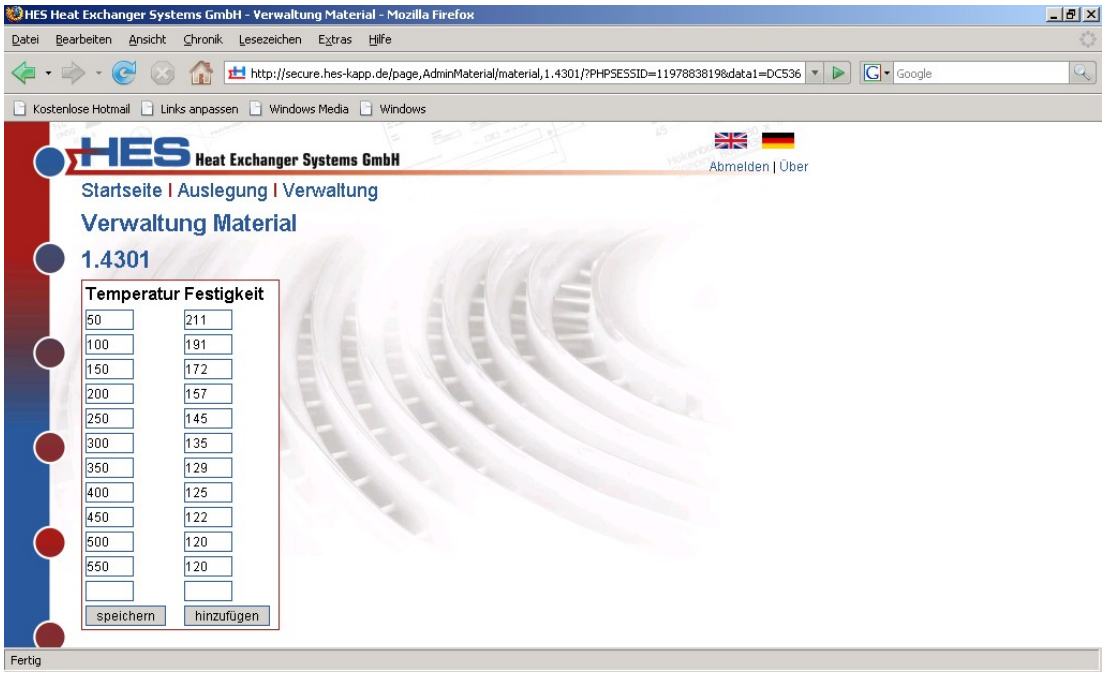


Abbildung 15: Screenshot Modul Verwaltung/Material, Eingabeformular

5.8 Verwaltung/Stoffdaten

Analog zu den Materialien gibt es verschiedene Medien, welche in einem Wärmetauscher aufgewärmt oder abgekühlt werden sollen. Die Kapazität und damit die Bauart eines Wärmetauschers ist dabei von den spezifischen Eigenschaften der eingesetzten Medien abhängig. Wie bei den Materialien steht hier Benutzern mit dem Recht setMedium eine Liste von Stoffen zur Verfügung. Es können neue Stoffe eingegeben oder

bestehende bearbeitet werden. Hierbei existieren zu jedem Stoff vier Tabellen:

1. Dichte
2. spezifische Wärmekapazität
3. dynamische Viskosität
4. Wärmeleitfähigkeit

Wiederum sind diese Werte stoff- und temperaturabhängig und verhalten sich bezüglich der Temperatur nicht linear. Daher werden je Stoff experimentell ermittelte Werte bezüglich individueller Temperaturen eingetragen. Wie bei den Materialkennwerten werden auch hier die Tabellen dynamisch verwaltet und bei noch nicht vorhandenen Daten wird ein Temperaturraster als Vorlage angeboten. Um die Verwaltung der Stoffdaten zu erleichtern wird die Anzahl der jeweils verfügbaren Stoffdaten in der Übersicht des Moduls mit angezeigt, nicht vorhandenen Daten werden durch eine rot hervorgehobenen Null gekennzeichnet.

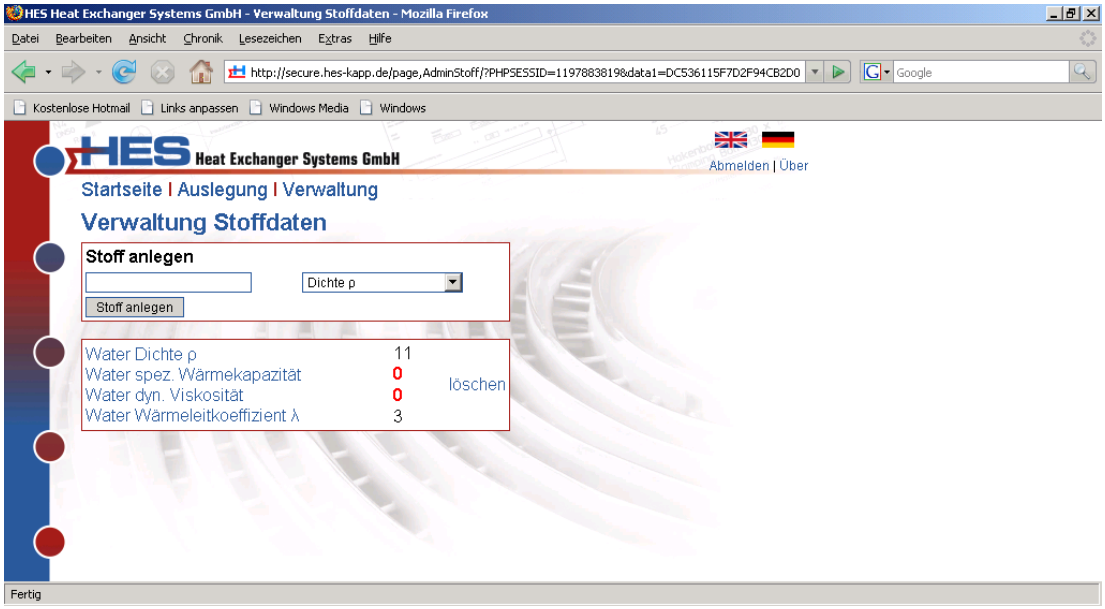


Abbildung 16: Screenshot Modul Verwaltung/Stoffdaten, Überblick

In den bereits vorhandenen Implementierungen der Auslegungssoftware in Excel wurden manuell Stoffdaten eingepflegt. Um Werte abseits der bekannten Daten zu gewinnen wurden die experimentell ermittelten Daten in Matlab eingetragen und dort interpoliert. Bei den meisten Stoffen war eine binomiale Interpolation ausreichend genau, bei manchen Stoffen musste aber auf andere Interpolationsmethoden zurückgegriffen werden. Die so gewonnenen Funktionen wurden dann in Excel als Gleichung hinterlegt.

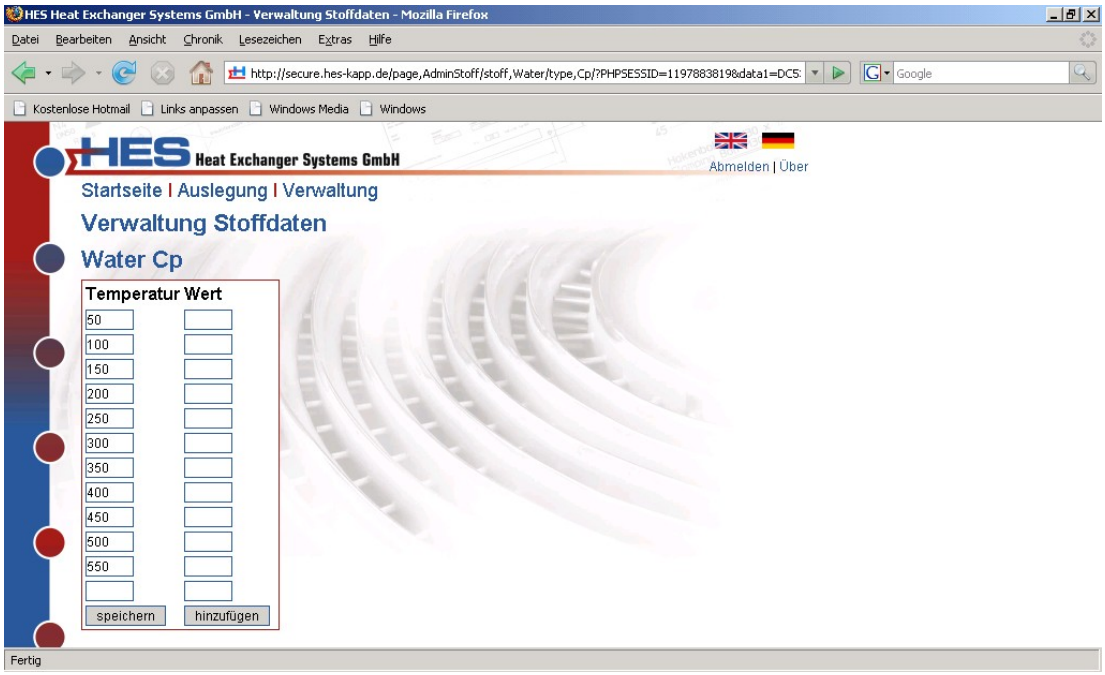


Abbildung 17: Screenshot Modul Verwaltung/Stoffdaten, Eingabeformular

Bei diesem Ansatz wird ein anderer Weg gewählt: Um eine möglichst genauer Annäherung an die Realität zu erhalten setzt man die allgemein als zwar komplex, aber genau anerkannte Interpolationsmethode, den Cubic Spline ein. Dabei werden jeweils zwischen zwei bekannten Werten – den Stützpunkten der Funktion – eine kubische Funktion gesucht. Diese ist dann nur im betreffenden Intervall definiert.

Die Implementierung dieser Tabellen und die weitere Verarbeitung und Bereitstellung dieser Daten geht über den Umfang dieser Diplomarbeit

hinaus. Da externer Code zur Implementierung des Cubic Splines angeboten wurde, erschien der Aufwand vertretbar, so dass diese als Erweiterung der Diplomarbeit vorgesehen wurde.

Dabei werden beim Abspeichern einer der vier Tabellen die Werte einerseits wie gewohnt gespeichert. Es wird dann die Interpolation auf den Daten angewandt wobei die Temperaturen als x-Werte und die jeweiligen Messdaten als y-Werte beziehungsweise Stützpunkte für die zu berechnende Funktion verwendet werden. Die aus der Cubic-Spline-Approximation resultierenden Parameter werden getrennt in einer weiteren Tabelle gespeichert. Mit deren Hilfe kann später eine Funktion angesetzt werden, welche durch die Messwerte führt und anhand derer fehlende Messwerte interpoliert werden können.

5.9 Auslegung

Die Auslegung von Spiralwärmetauschern stellt die Kernaufgabe der Webapplikation dar. Diese Aufgabe unterteilt sich in das Erfassen, verarbeiten und ausgeben von Daten. Ausserdem sollen die Auslegungen verwaltet werden können, so dass Berechnungen verändert, angepasst oder auch von anderen Mitarbeitern überprüft werden können. Ausserdem ist es notwendig, dass von einer Berechnung mehrere Revisionen gespeichert werden können. Diese Funktionalitäten und Aufgaben wurden in drei Schritte – jeweils eigene Module – aufgegliedert. Jedes Modul übergibt dem nachfolgenden relevante Daten per POST.

5.10 Auslegung/Schritt 1

Zuerst werden hier neue Auslegungen angelegt und bereits gespeicherte Berechnungen zur Ansicht oder Weiterverarbeitung aufgelistet. Dafür werden die Berechtigungen setOrgCalcs und setAllCalcs benötigt. Erstere erlaubt die Verwendung aller Berechnungen, die von Mitarbeitern des selben Unternehmens erstellt

wurden, Letzteres erlaubt die Bearbeitung sämtlicher Berechnungen.

Neue Berechnungen erstellen darf grundsätzlich jeder Benutzer, da dies die Kernfunktion der Software ist. Dazu müssen nur wenige Kundendaten eingetragen werden – Kunde, Ansprechpartner, Anfrage-, Positions- und Angebotsnummer. Diese werden zum Anlegen der neuen Berechnung an das Folgemodul gesandt.

Soll eine bestehende Kalkulation verwendet werden, so kann diese aus der Liste gewählt werden. Neuste Berechnungen werden zuerst aufgelistet, zu jedem Angebot dazu eine Liste aller bestehenden Revisionen. Wird eine Revision ausgewählt, so wird diese ebenfalls an das Folgemodul übergeben.

Bei Bedarf können alte Berechnungen gelöscht werden.

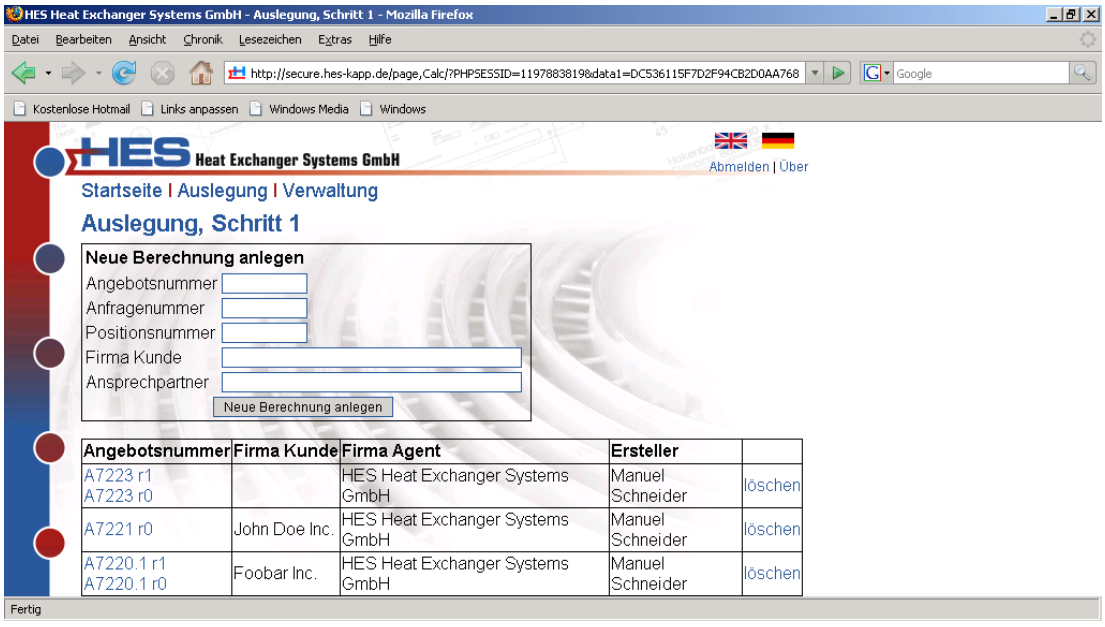


Abbildung 18: Screenshot Module Auslegung, Schritt 1

5.11 Auslegung/Schritt 2

Von *Auslegung/Schritt 1* erhält dieses Modul entweder die Daten einer neuen Anfrage oder die Nummer und Revision einer bestehenden

Kalkulation. Bei einer neuen Anfrage werden deren Daten gespeichert und die Revision intern auf 0 gesetzt, die Anfragenummer wird beibehalten.

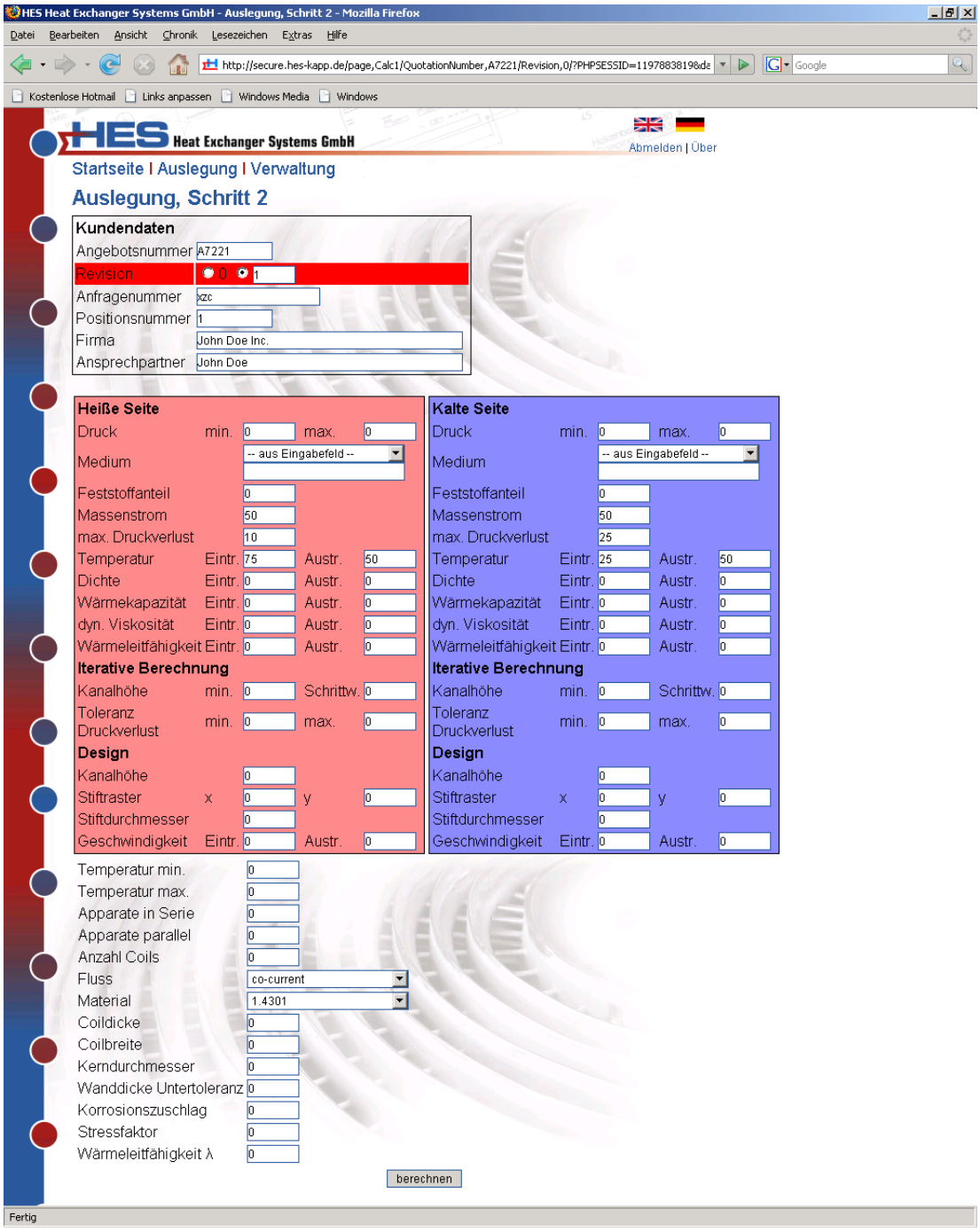


Abbildung 19: Screenshot Modul Auslegung, Schritt 2

Anschliessend wird anhand der Angebots- und Revisionsnummer der betreffende Datensatz aus der Datenbank gelesen und in einem Formular dargestellt. In diesem Formular können alle relevanten Eingabedaten geändert oder eingetragen werden. Anhand der Standardwerte in der Datenbank sind diese bei neuen Anfrage bereits im Formular enthalten.

Es wird automatisch ermittelt, wieviele Revisionen einer Berechnung bereits bestehen. Dem Benutzer bleibt dann die Wahl die Änderungen unter der nächsthöheren Revisionsnummer abzuspeichern (Standard) oder die bestehende Revision zu ändern. Die Änderungen an diesem Formular werden dem Folgemodul per POST-Parameter übergeben.

Speziell ist der Einsatz der Stoffdatenbank an dieser Stelle. Wie im Modul *Verwaltung/Stoffe* bereits beschrieben existieren in der Datenbank teilweise Daten für Medien. Die bestehenden Medien können aus einer Drop-Down-Liste ausgewählt werden. Per Ajax werden dann die benötigten Daten vom Webserver abgefragt, welcher zuvor die beschriebene Cubic-Spline-Interpolation durchführt und die Daten für die eingetragene Prozesstemperaturen zurück liefert. Die gewonnenen Daten werden dann von einem Java Script in die betreffenden Felder eingetragen, an denen dann bei Bedarf noch manuelle Änderungen vorgenommen werden können. Ist das Medium in der Datenbank nicht verfügbar, so müssen die betreffenden Werte manuell eingetragen werden.

5.12 Auslegung/Schritt 3

Die gesammelten Werte von *Auslegung/Schritt 2* werden – je nach Einstellung der Radiobox „Revision“ - entweder als neue Revision oder in die bestehende gespeichert. Anschliessend werden die in der Bibliothek SHElib hinterlegten Funktionen genutzt um die notwendigen Berechnungen durchzuführen.

Anschliessend wird das Ergebnis in tabellarischer Form dargestellt. Eine mit PHPlot erstellte Grafik zeigt ausserdem den Temperaturverlauf beider Medien im Prozess an. Die Ergebnisse werden auch als Plain-Text-Datei zum Download angeboten, so dass sie in anderen Programmen weiterverarbeitet werden können.

6 Deployment

6.1 Voraussetzungen

Um eine Zugriffsberechtigung auf das System zu erhalten, benötigen die Anwender drei Komponenten:

- den Webclient, die Software, welche als Proxyserver den Zugriff der Webapplikation auf den Dongle regelt
- den Dongle, dessen Seriennummer und 32-Bit Kryptoschlüssel im System gespeichert worden sind
- die PIN, eine Geheimzahl, welche vom Webclient abgefragt und im Dongle verschlüsselt wird.

Der Webclient ist ein einzelnes Windows-Binary, welches fertig vorkonfiguriert als einzelne Datei ausgeliefert wird.

Die USB-Dongles werden von Windows als Human Interface Device (HID) erkannt und benötigen daher keinen Treiber. Der Webclient kann direkt auf die Hardware zugreifen und den Dongle als kryptographisches Gerät zum verschlüsseln der PIN verwenden.

Die Dongles sind einzeln registriert – die Seriennummer und der auf dem Dongle gespeicherte Schlüssel sind auf dem einzelnen USB-Stick fixiert. Die PIN, welche vom Benutzer im Login-Prozess abgefragt wird, ist fest mit der Kombination Seriennummer und dem 32-Bit-Schlüssel verbunden.

6.2 Vorgehensweise

6.2.1 Anlegen der Berechtigungen

1. Ein Mitarbeiter mit setAllDongles- und setAllRights-Berechtigung meldet sich an der Webapplikation mit seinem Dongle an.

2. Er kann dort falls notwendig im Modul Verwaltung/Firmen eine neue Firma anlegen.

3. Im Modul Verwaltung/Dongles wird ein neuer Dongle auf den Namen des neuen Benutzers und dessen Firma angelegt. Dazu muss dessen Seriennummer angegeben werden. Diese wird von der vom Hersteller gelieferten Software angezeigt.

4. Nach Absenden der Daten speichert die Webapplikation den neuen Dongle in seiner Benutzerdatenbank und versieht ihn mit einem 32 Bit langen zufällig generierten Dongle-Key. Dieser wird in vier Blöcken zu je acht Byte (16 Hexadezimal-Zeichen) angezeigt.

5. Mit Hilfe der vom Hersteller gelieferten Software können die vier Schlüsselblöcke per Copy & Paste übertragen und im Dongle gespeichert werden.

6.2.2 Auslieferung der Software

Im Rahmen dieser Arbeit wurde eine Dokumentvorlage mit Aufkleber für die Dongles entworfen, so dass diese eindeutig gekennzeichnet werden können. Es existieren drei Grössen, für die Bauformen „kurz“ und „lang“ der USB-Dongles sowie für LPT-Dongles.

Für jeden Dongle wurden zwei Aufkleber vorgesehen – einer enthält die Seriennummer und ggf. in der darunter liegenden Zeile den Namen des Benutzers. Der zweite Aufkleber für die Vorderseite des Dongles

enthält das Kürzel „HES“ um den Benutzern eine Orientierung zum Zweck des Dongles zu geben. Die Aufkleber können am PC ausgefüllt und ausgedruckt und anschliessend mit einem Klebestift auf den vorgesehenen Flächen auf dem Dongle angebracht werden.

Der Dongle kann dann gefahrlos mit der Post transportiert werden.

Die benötigte Webclient-Software kann per CD-ROM, aber auch per E-Mail zugestellt werden. Sie ist so klein, dass sie auch bei langsamer Internetanbindung problemlos zu übertragen ist. Da sie ohne Dongle unbrauchbar ist, kann sie auch auf unsicherem Weg verteilt werden wie beispielsweise per Download oder E-Mail.

6.2.3 Anwenderschulung

Vorgehensweise zur Inbetriebnahme der Webapplikation:

- Webclient starten – eine Installation ist nicht notwendig, der Webclient kann und sollte in den Autostart des System eingetragen werden

- USB-Stick einstecken – das Gerät wird vom System erkannt, anschliessend erkennt der Webclient den Dongle und öffnet im Standardbrowser den vorkonfigurierten URL der Webapplikation

- PIN eingeben – nachdem die Webapplikation den Webclient initialisiert hat, wird über diesen die PIN des Benutzers abgefragt, im Dongle verschlüsselt und zusammen mit der Seriennummer und Prüfdaten übertragen.

- PIN ändern – nach der ersten Anmeldung sollte der Anwender die vom vorgegebene PIN auf eine persönliche ändern.

6.2.4 Handbuch

Installation:

Kopieren Sie den Webclient auf Ihren Desktop. Doppelklicken Sie auf die Datei. Im Systray (rechts unten neben der Uhr) erscheint ein Schlüssel-Symbol.

Anmeldung am System:

Stecken Sie Ihren Dongle in den USB-Anschluss. Ihr Standard-Browser sollte sich automatisch öffnen und die Webseite „http://secure.hes-kapp.de/“ öffnen.

Sollte dies nicht geschehen, können Sie sich alternativ per Rechtsklick auf das Schlüsselsymbol und der Auswahl von „Verbinden / Logon“ oder dem der manuellen Eingabe der Webadresse in Ihren Webbrowser am System anmelden.

Die Webseite lädt mehrfach neu. Innerhalb weniger Sekunden erscheint ein Dialogfeld zur Eingabe Ihrer PIN. Geben Sie hier die Ihnen zugewiesene PIN ein und bestätigen Sie die Eingabe mit Enter.

Wenn Sie auf der nun erscheinenden Webseite Willkommen geheissen werden, sind Sie erfolgreich am System angemeldet. Sollten Sie stattdessen aber eine Fehlermeldung erhalten, so lesen Sie diese durch und probieren es noch einmal. Sollten Sie wider Erwarten erneut keinen Erfolg haben, markieren Sie bitte den Text der Fehlermeldung mit der Maus und kopieren Sie diesen in ein E-Mail an HES. Wir helfen Ihnen dann weiter.

PIN ändern:

Nachdem Sie sich zum ersten Mal angemeldet haben, sollten Sie Ihre PIN ändern.

Wählen Sie dazu im Menü den Punkt „Verwaltung“ aus.

Klicken Sie in der Verwaltungsübersicht auf „PIN ändern“ oder „Dongle“.

Auf der Seite „PIN ändern“ können Sie eine neue PIN eintragen. Diese kann bis zu fünf Zeichen lang sein und auch Buchstaben beinhalten.

Sollten Sie die Berechtigung besitzen mehrere Dongles zu bearbeiten, etwa jene Ihrer Firma, dann wählen Sie auf der Seite „Dongles“ in der Übersicht den Ihnen zugeordneten aus. Tragen Sie im betreffenden Formular eine neue PIN ein und klicken Sie auf „Änderungen speichern“.

Berechnungen durchführen:

Wählen Sie im Menü „Auslegung“.

Tragen Sie im Formular die Daten bezüglich der Anfrage ein und legen Sie diese mit dem Klick auf „Neue Berechnung anlegen“ an.

Sollten Sie bereits eine Berechnung getätigt haben und Sie möchten diese weiterverwenden, überarbeiten oder eine neue Revision anlegen, so wählen Sie die gewünschte Anfrage und Revision aus der angebotenen Liste aus.

Nach Auswahl der Anfrage beziehungsweise nach Anlegen einer neuen Anfrage können Sie die Prozessparameter im folgenden Formular festlegen. Haben Sie eine bestehende Anfrage gewählt, so wird automatisch eine neue Revision der Anfrage erzeugt. Möchten Sie dies nicht, so ändern Sie die Revisionsnummer auf jene Revision, welche Sie überschreiben möchten oder wählen Sie die alte Revision aus. Klicken Sie anschliessend auf „Berechnen“.

Die Daten werden nun gespeichert und das Ergebnis berechnet. Sie können die Seite nun drucken oder die Berechnung als Text herunterladen.

7 Probleme und Lösungen

7.1 UTF-8-Zeichensatz und Sanitizer-Funktion

Da alle Eingaben im Modul „getvars“ gefiltert werden, kann es zu Problemen mit Sonderzeichen kommen.

Da sich UTF-8 als internationaler Standard durchsetzt, wird für die Webapplikation dieser Zeichensatz eingesetzt. Damit können alle Sonderzeichen verschiedener Sprachen abgebildet werden, was insbesondere für die Mehrspachigkeit unerlässlich ist.

Die Sanitizer-Funktion prüft alle Eingaben, entfernt ausführbaren Code, entschärft Anführungszeichen durch Backslashes (Escaping) und wandelt Sonderzeichen in HTML-Entitäten um.

Da dies byteweise geschieht, erhält man beim Einsatz von UTF-8 ein Problem, da bei diesem Zeichensatz Sonderzeichen mit zwei Byte adressiert werden. So werden dann das erste und zweite Byte einzeln in HTML-Entitäten umgewandelt, wobei die eigentliche Information – das Sonderzeichen – verfälscht wird.

Abhilfe schafft hier die Tatsache, dass die Funktion „htmlentities()“ den Zeichensatz berücksichtigen kann und im Voraus auf Sonderzeichen prüft. Diese werden dann als Ganzes betrachtet und so in die richtigen HTML-Entitäten überführt.

7.2 Absichern von Array-Eingaben

An einigen Stellen werden eine unbekannte Mengen gleicher Daten – gleich Datensätzen – per Formular übertragen. Dies geschieht beispielsweise überall dort, wo Listen wie die Stoffdaten in dynamischer Länge bearbeitet werden sollen.

Dazu lassen sich die Feldnamen im HTML-Code wie Arrays in PHP mit einer laufenden Nummer als Indizes verwenden. Beim Absender der

Formulardaten werden diese als PHP-Array eingelesen und in einer zweiten Dimension in den globalen $_GET- oder $_POST-Arrays bereitgestellt. Der Sanitizer in seiner ursprünglichen Version stiess hier an seine Grenzen, da alle Elemente der genannten Arrays einzeln als Skalare eingelesen wurden. Dadurch standen die Arrays nicht zur Verfügung. Die implizierte Typenumwandlung PHPs sorgte dafür, dass das Element mit dem Namen des Arrays dann den String „Array“ gespeichert hatte anstelle der Elemente.

Dieses Problem wurde behoben, indem vor der Säuberung eines Elements der globalen Arrays erst dessen Typ geprüft wird. Bei Bedarf eine liesst eine zweite Foreach-Schleife die Elemente in der zweiten Dimension ein, säubert und legt diese in einer zweiten Dimension der Zieldatenstruktur ab.

7.3 Zahlenformate

Für Fliesskommazahlen gibt es verschiedene Schreibweisen. In Datenbank-Systemen und in der Berechnung wird die amerikanische (internationale) Syntax erwartet.

Bei der deutschen Syntax werden Komma als Dezimaltrennzeichen verwendet.

Um Benutzer nicht an das System umgewöhnen und Fehler abfangen zu müssen, werden in den betreffenden Modulen vor Verarbeitung der Zahlen diese durch eine eigene Sanitizer-Funktion bereinigt.

Dabei werden Kommata in Punkte umgewandelt, anschliessend werden alle Punkte bis auf den letzten im String auftretenden entfernt. Danach kann die Zahl mit der in PHP vorhandenen Funktion floatval() in eine Fliesskommazahl umgewandelt werden. Die Zahl wird entsprechend dann gespeichert und anschliessend weiterverarbeitet.

Mit dieser Methode ist es möglich, auch Tausenderpunkte und Einheiten einzugeben ohne mit Fehlern konfrontiert zu werden. Die

Eingabe von Zahlen im wissenschaftlichen Format ist ebenfalls möglich.

Die Vorverarbeitung der Variable bevor floatval() verwendet wird dient dazu, die Fehler bzw. Unzulänglichkeiten der PHP-Funktion auszugleichen. So führen beispielsweise mehrere Kommata oder Punkte zu einem unerwarteten Ergebnis. Andererseits wäre es zuviel Aufwand bzw. unökonomisch, die Bereinigung von Zeichen und vor allem die wissenschaftliche Darstellung ebenfalls selbst zu implementieren.

8 Schlussbemerkungen

Die Diplomarbeit wurde gemäss den Vorgaben abgeschlossen. Die vom Auftraggeber erwünschte Basis ist geschaffen worden, auf welcher Weiterentwicklungen der Software möglich sind. Wünsche diesbezüglich sind von der Firma HES bereits im Voraus geäussert worden (wie Kostenrechnung), andere ergaben sich im Rahmen dieser Arbeit, auf diese wird weiter unten noch eingegangen. Insgesamt wurde bei der Umsetzung der Aufgabe darauf geachtet eine leistungsfähige Infrastruktur für all diese Ideen zu schaffen, daher lag der Fokus nicht darauf die Anwendung aus Benutzersicht zu optimieren. Dies kann ebenso Ziel der Weiterentwicklung sein.

Das Ergebnis wurde – soweit funktionsfähig – getestet. Die Auslegung ist noch in Zusammenarbeit mit den Verfahrenstechnik-Ingenieuren der HES zu überarbeiten und vervollständigen. Einem Feldtest der Software kann dann zuversichtlich entgegen geblickt werden.

8.1 gewonnene Erfahrungen und Erkenntnisse

Die Entscheidung, das PHP-Framework einzusetzen, hat sich als richtig erwiesen. Der Arbeitsfortschritt zu Beginn verlief sehr zügig, geriet aber im Laufe der Arbeit unter Druck. Die zehn Wochen liessen durchaus nicht zuviel Zeit übrig, so dass ein Mehraufwand in der Programmierung den Zeitplan unhaltbar gemacht hätte. Ausserdem zeigte sich das Framework als ausreichend leistungsfähig und es war interessant, den im Kapitel „Problemlösungen“ angesprochenen Problemen im Produktiveinsatz zu begegnen und diese zu lösen.

Techniken, welche im Laufe des Studiums diskutiert und trainiert wurden, konnten erfolgreich eingesetzt werden. Dadurch gab es keine Überraschungen in der Programmierung oder dem Zeitplan. Der eigene Arbeitsstil hat sich dadurch zum Positiven verbessert und optimiert. Es ist somit nicht nur durch das Studium, sondern auch durch dieses

Projekt eine Basis entstanden, welches sich durch die zukünftige Arbeit weiter verbessern lässt.

8.2 Empfehlungen an den Hersteller der Sicherheitslösung

Im Rahmen des Einsatzes der Dongle-Lösung und des Webclients sind Verbesserungsvorschläge aufgetaucht, welche nach Abschluss dieser Arbeit der Herstellerfirma zur Verfügung gestellt werden:

Bei der Initialisierung, das heisst beim Zugriff auf den Webproxy ohne PIN-Abfrage wird das Vorhandensein des Dongles nicht geprüft. Dies erfolgt nur bei der PIN-Abfrage oder beim Lesen und Schreiben auf den Dongle, wobei letztere Funktionen nicht verwendet wurden. Dadurch kann sich ein Nutzer einloggen und anschliessend den Dongle von seinem System entfernen. Dies ist nicht zwingend ein Sicherheitsproblem, da er den Dongle trotzdem besitzen muss, es ist aber unprofessionell. Daher die Empfehlung sämtliche Zugriffe auf den Webproxy nur mit eingestecktem Dongle zu ermöglichen.

Ursprünglich war geplant, den Zugriff auf die Webapplikation mittels SSL zu sichern. Leider unterstützt dies die Webclient-Software nicht. Es werden zwar keine geheimen Daten übertragen, da die eingegebene PIN mit dem Dongle-Key verschlüsselt übertragen wird, trotzdem könnte es ein Sicherheitsproblem darstellen, wenn im Laufe der Sitzung sensitive Daten übertragen werden oder der Verschlüsselungsalgorithmus des Dongles gebrochen werden sollte. Die SSL-Funktionalität liesse sich über freie Bibliotheken nachrüsten, da sie völlig transparent implementiert ist entstünde für die Entwicklung des Webclients kaum Mehraufwand. Ein weitergehender Vorschlag wäre dann, dem Kunden bei der Erstellung des Webclients zu ermöglichen ein Zertifikat einer CA zu hinterlegen mit dem das vom Server vorgezeigte Zertifikat geprüft werden kann, so dass eine Man-in-the-Middle-Attacke ebenfalls ausgeschlossen werden kann.

Die Dokumentation des Produkts, speziell der Webclient-Lösung, ist mangelhaft. Notwendige Informationen konnten Lesen des Codes der Beispielwebapplikation des Herstellers gewonnen werden. Dieser Code musste explizit angefordert werden und wurde elektronisch zur Verfügung gestellt. Die darin enthaltene Klasse zur Entschlüsselung der Dongle-Daten – ein proprietärer Algorithmus – ist nicht im Lieferumfang des Produkts enthalten, der Beispielcode unnötig kompliziert und umfangreich. An dieser Stelle hätten im Projekt drei Manntage eingespart werden können. Die Dokumentation enthält zwar die notwendigen Parameter zur Ansteuerung der dem Webclient bekannten Aktionen und die Beschreibung der Antworten, allerdings wurde vergessen zu erwähnen, unter welchem Port und Adresse sich die Webclient-Software auf dem Clientsystem erreichen lässt. Auch werden die notwendigen Typ-Konvertierungen (Antwortdaten liegen als HEX-String vor, die Entschlüsselung basiert aber auf byteweise umgerechnete Integerzahlen) nirgends dokumentiert. Diese Informationen konnten nur anhand des Quelltextes und Versuchen ermittelt werden.

8.3 eigene Bewertung

Insgesamt bin ich mit dem Verlauf der Diplomarbeit zufrieden. Der Anfangs sehr grosszügige Zeitplan und den daraus gewonnenen Zeitvorsprung erwies sich am Ende doch als recht knapp, die anfangs gewonnene Zeit wurde benötigt um die Arbeit rechtzeitig abzuschliessen.

Wie im Abschnitt „Erfahrungen“ beschrieben war die Arbeit interessant und vermittelte im Vergleich zur Arbeit im Laufe des Studiums und Projekten vor dem Studium ein anderes Arbeitsgefühl.

Insgesamt bleibt die Erkenntnis, ein interessantes Projekt umgesetzt zu haben und die Hoffnung, dass es der Firma HES als Auftraggeber noch viele Jahre hilfreich sein wird.

8.4 Zukünftige Weiterentwicklung

Neben den technischen Verbesserungen des Webclients gibt es sowohl weitere Vorschläge sowohl zur Erweiterung der Funktionalität der Software, als auch zur Verbesserung der Handhabung durch Änderungen an der Benutzeroberfläche.

Die gewählten Technologien wie PHP und MySQL sind zukunftssicher, die Komponenten lassen sich unabhängig voneinander aktualisieren. Da auf offene Standards aufgebaut wurde, können auch andere Serverbetriebssysteme wie Windows eingesetzt werden. Die Programmiersprache PHP ist ausserdem von jeder Webserver-Software einsetzbar, welche eine CGI-Schnittstelle besitzt. Die Datenbank lässt sich durch jedes SQL-konformes Datenbanksystem ersetzen.

8.4.1 Technische Änderungen

- Verwendung von SSL
- bessere Webclient-Software (evtl. Linux-Port)
- Aktualisierung der Serverhardware
- Verbesserung der Protokollansicht

8.4.2 Verbesserung der Benutzbarkeit

- Auslegungsformulare im Tab-Design

8.4.3 Erweiterung der Funktionalität

- Kostenrechnung
- Projektabwicklung
- Ausbau und Test der Stoffdaten-Interpolation

9 Eidesstattliche Erklärung

Ich versichere, dass ich diese Arbeit selbständig verfasst, keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Es wurden keine Textstellen wörtlich oder sinngemäss übernommenen.

Dies gilt auch für die in der Arbeit enthaltenen Zeichnungen, Skizzen und graphischen Darstellungen.

Die Arbeit ist in gleicher oder ähnlicher Form oder auszugsweise im Rahmen einer anderen Prüfung noch nicht vorgelegt worden.

Schopfheim, den 21. Dezember 2007 Manuel Schneider