

Remote Lab Management System

Bachelorarbeit

Abteilung Informatik
Hochschule für Technik Rapperswil

Frühjahrssemester 2010

Autor(en): Markus Kolb, Mirjam Schaffner
Betreuer: Prof. Beat Stettler, Rolf Schärer
Projektpartner: -
Experte: Roland Marti
Gegenleser: Prof. Peter Sommerlad

Summary Erklärung, Abstract, Management Summary	1
Projektmanagement Projektplan	2
Anforderungen Anforderungsspezifikation	3
Analyse Domainanalyse	4
Architektur Software Architecture Document	5
Anleitung Installation Guide	6
Qualitätsmassnahmen Testdokumentation	7
Ingenieurs Bericht Technischer Bericht	8
Verschiedenes Glossar, Persönliche Berichte, Literaturverzeichnis	9
Anhang Sitzungsprotokolle, ToDo's	10

Abstract / Kurzfassung

Projekt:

Remote Lab Management System

Projektmitglieder:

Markus Kolb

Mirjam Schaffner

Betreuer:

Rolf Schärer

Beat Stettler

1 Abstract / Kurzfassung

Abteilung	Informatik
Namen der Studenten	Markus Kolb , Mirjam Schaffner
Studienjahr	SS 2010
Titel der Bachelorarbeit	Remote Lab Management System
Betreuer:	Herr Prof. Beat Stettler, Herr Rolf Schärer
Gegenleser:	Herr Prof. Peter Sommerlad
Examinatorin/Examinator	Herr Roland Marti
Themengebiet	Netzwerk
Projektpartner	-
Institut	INS

Ziel der Bachelorarbeit war es, ein Control Panel zu erstellen, damit es den Benutzern möglich ist, Pods zu reservieren. Ein Pod besteht aus mehreren Geräten wie z.B. Routern, Switches, Firewalls oder Unix Servern. Um Reservationen auf Pods zu machen, muss ein Benutzer am Control Panel angemeldet sein und über genügend Credit Points verfügen. So ist es dem Administrator möglich, die Reservationen eines Benutzers zu kontrollieren und der Benutzer kann den Pod nicht die ganze Zeit reservieren. Ein weiteres Ziel war es, über das Control Panel per SNMP Ansteuerung, die Geräte neu zu starten und SSH Verbindungen trennen zu können.

Das Control Panel haben wir unter Verwendung des PHP Frameworks Symfony erstellt, da dieses Framework sehr mächtig ist und vom Betreuer vorgegeben wurde.

Bei der Entwicklung der Software wurde als Software Engineering Vorgehensmodell der Rational Unified Process (RUP) eingesetzt. Als Ergebnis haben wir ein Control Panel, das eine Ansicht für den Administrator, sowie eine eingeschränkte Ansicht für den Benutzer zeigt. Sobald sich der Benutzer am Control Panel anmeldet, wird eine Abfrage an die LDAP Datenbank gemacht, ob der Benutzername und das Passwort stimmen. Hat sich ein Benutzer angemeldet, kann er Credit Points kaufen, Reservationen verwalten, seine Benutzerangaben bearbeiten, seine History der Reservationen anschauen, Reservationen auf ein Gerät z.B. einem PDA herunterladen, Geräte der reservierten Pods neu starten und dem Administrator via Kontaktformulare Mails senden, um allfällige Probleme zu melden oder einen Kommentar zu schreiben. Der Administrator hat zudem die Möglichkeit Pods, Podkategorien, Geräte, Gerätkategorien, Labs, Gruppenrechte und Reservationen von Benutzern zu verwalten, sowie eine LDAP Gruppe oder die Benutzerdaten zu bearbeiten oder zu löschen. Der Administrator sieht alle Geräte von allen Pods, auch wenn diese von einem Benutzer reserviert sind und kann diese zu jeder Zeit neu starten.

Meldet sich ein Benutzer am Terminal Server an, wird über Radius ein XML-RPC Methodenaufruf an das Control Panel gesendet, das wissen will, ob der Benutzer zu dieser Zeit eine Reservation auf den gewünschten Pod hat. Das XML-RPC gibt als Antwort ein „ja“ oder „nein“ zurück.

Management Summary

Projekt:

Remote Lab Management System

Projektmitglieder:

Mirjam Schaffner

Markus Kolb

Betreuer:

Beat Stettler

Rolf Schärer

Inhaltsverzeichnis

1	Ausgangslage	3
2	Vorgehen	3
3	Technologien.....	4
3.1	PHP	4
3.2	MySQL.....	4
3.3	Symfony.....	4
3.4	Datatables.....	4
3.5	XML-RPC.....	4
4	Ergebnisse.....	4
5	Ausblick	5

1 Ausgangslage

Das INS Institute for Networked Solutions betreibt mehrere Remote Labs für die Ausbildung von Netzwerkspezialisten. Diese können über das Internet zu jeder Zeit durch die Studenten genutzt werden. Der Zugriff erfolgt via Terminal Server mit SSH, die Authentifizierung geschieht über RADIUS gegenüber einer LDAP Datenbank. Die Koordination der Zugriffe zwischen den Studenten erfolgt in einem gemeinsamen Kalender. Für das Löschen der Konfigurationen und das Einspielen derselben ist der Student selber verantwortlich. Ziel dieser Bachelorarbeit ist eine Umgebung zu entwickeln, welche die Reservation der Remote Labs, den Zugriff darauf sowie die Verwaltung von Konfigurationen übernimmt. Dem Studenten sollte es möglich sein, Zwischenstände einer Übung abzuspeichern, vom Lab zu entfernen und diese zu einem späteren Zeitpunkt wieder einspielen zu können. Die Benutzer sollen weiterhin in der zentralen LDAP Datenbank gespeichert werden. Das Reservationssystem soll die Möglichkeit bieten, den Labzugriff über ein Credit-System den Studenten verrechnen zu können. Das Management der Laborumgebungen (Pods) und Konfigurationen soll via zentrales Control Panel erfolgen. Für Funktionen wie Gerät einschalten oder neu starten, Konfiguration einspielen, speichern, löschen, usw. benötigt der Student eine aktive Reservation auf dem entsprechenden Lab Pod. Funktionen wie Export/Import von Konfigurationen kann der Student jederzeit ausführen.

2 Vorgehen

Bei der Entwicklung des Reservationssystems wurde gemäss dem Vorgehensmodell „Rational Unified Process“ gearbeitet. Deshalb haben wir am Anfang sehr viel Zeit in die Dokumente gesteckt und erst dann richtig mit der Implementierung begonnen.

Am Anfang hatten wir abgeklärt, was für Technologien bereits vorhanden sind, damit wir unser Control Panel in die bestehende Umgebung einbinden können. Uns wurde das PHP Symfony Framework vorgegeben, sowie XML-RPC, MySQL, Apache und Linux. Da wir noch nie mit dem Symfony Framework gearbeitet haben, mussten wir einige Zeit in das Jobeet Tutorial investieren. Dann haben wir mit der Anleitung des Jobeet Tutorials die CRUD Module implementiert. Dann begannen wir, den Reservationskalender mit dem Plugin jQuery zu implementieren und haben so viel Spass daran bekommen, dass wir die CRUD Module mit dem Plugin Datatables implementierten. Das Plugin Datatables stellte viele Features zur Verfügung, die wir sehr gut brauchen konnten und daher nicht einzeln implementieren mussten.

Aus zeitlichen Gründen haben wir uns entschieden, die Priorität auf den Reservationskalender zu legen und die Arbeit gleichzeitig eingeschränkt, sodass wir die Verwaltung der Konfigurationen leider nicht mehr implementieren konnten. Jedoch haben wir bei der Implementierung darauf geachtet, dass nachträgliche Implementierungen, wie z.B. die Verwaltung der Konfigurationen oder weitere Features, möglich sind.

Während der Arbeit haben wir unseren Code mit Unit- und Functional-Tests getestet. Zudem haben wir fortlaufend die Dokumente und den Code gegenseitig kontrolliert und in Teammeetings gemeinsam besprochen. In den Teammeetings wurde darauf geachtet, dass das Know-how ausgetauscht wird, damit beide Bachelorarbeitspartner auf dem aktuellsten Stand sind. Die Arbeiten wurden ebenfalls in den Teammeetings verteilt.

3 Technologien

3.1 PHP

Da das Symfony Framework auf PHP basiert, verwendeten wir PHP als Programmiersprache.

3.2 MySQL

Als Datenbank wurde MySQL verwendet, um unsere Daten zu speichern.

3.3 Symfony

Für das Reservationssystem haben wir das Framework Symfony verwendet, da uns dieses Framework sehr mächtig erschien und vom Betreuer vorgegeben wurde.

3.4 Datatables

Die CRUD Module haben wir mit dem Plugin Datatables implementiert. Dieses Plugin stellt sehr viele Features zur Verfügung wie dem Filter, der Suche, dem Seiten blättern, der Anzahl Objekte pro Seite anzeigen oder dem Spalten sortieren.

3.5 XML-RPC

Als Abfrage, ob ein Benutzer zurzeit eine Reservation auf den gewünschten Pod besitzt, wird der XML-RPC Methodenaufruf verwendet. Dieser Methodenaufruf fragt unser Reservationssystem die benötigten Informationen ab und gibt Radius ein „ja“ oder „nein“ als Antwort zurück. XML-RPC wurde uns vom Betreuer vorgegeben.

4 Ergebnisse

Wir haben ein Control Panel implementiert, in dem nicht nur Reservationen selbst verwaltet werden können, sondern das noch viele andere Funktionen zur Verfügung stellt. Um überhaupt eine Reservation zu machen, muss sich der Benutzer zuerst registrieren und vom Administrator in der LDAP-Datenbank erfasst werden. Nur ein Benutzer, der in der LDAP Datenbank erfasst ist und im Control Panel einer Gruppe zugewiesen wurde, kann sich am Control Panel anmelden.

Weitere Funktionen, die der Administrator ausführen kann, sind...

- Geräte und Gerätekategorien verwalten
- Geräte neu starten
- Pods und Podkategorien verwalten
- Labs verwalten
- Benutzerdaten und LDAP Gruppe bearbeiten oder löschen
- Reservationen und Credit Points verwalten
- die Reservationen auf ein Gerät z.B. einem PDA herunter zu laden
- Gruppenrechte verwalten.

Der Benutzer hat die Möglichkeit, ...

- dem Administrator mittels eines Kontaktformulars ein Mail zu senden, das einen Kommentar enthält oder ein Problem beschreibt
- Credit Points per Formular zu kaufen
- eigene Reservationen zu verwalten
- die eigene Reservationshistory anzusehen
- die Reservationen auf ein Gerät z.B. einem PDA herunter zu laden
- seine Benutzerdaten zu bearbeiten
- die Geräte seiner reservierten Pods neu zu starten.

Im Control Panel kann die Sprache Englisch oder Deutsch gewählt werden und die Applikation merkt sich bei jedem Benutzer, welche Sprache dieser gewählt hat.

5 Ausblick

Das Control Panel ist so erstellt worden, dass die Möglichkeit besteht, dieses weiter auszubauen. Als eine Erweiterung könnte(n)...

- die Konfigurationen von Geräten hinauf- und herunter geladen werden
- ein externes Kreditsystem angebunden werden
- die Eingaben des Administrators auf Plausibilitäten überprüft werden
- die Data/Time Format Eingabe als Date/Time Kalender implementiert werden
- der Benutzer mehreren LDAP Gruppen im Control Panel angehören
- eine weitere Ajax Implementierung implementiert werden, in der mehrere Administratoren gleichzeitig CRUD Objekte ändern können, ohne dass Änderungen des anderen Administrators verloren gehen
- der Verlauf der Reservationen in einem Log File gespeichert werden.
- ein Cronjob implementiert werden, welcher alle 3 Monate alten Reservationen archiviert, sodass die Performance der Datatable Reservation erhalten bleibt.

Projektplan

Version 1.4

Projekt:

Remote Lab Management System

Projektmitglieder:

Markus Kolb

Mirjam Schaffner

Betreuer:

Rolf Schärer

Beat Stettler

Revision				
Version	Status	Datum	Beschreibung/Änderung	Autor
1.0rc01	In Bearbeitung	26.02.2010	Erstellung des Projektplans, Projektübersicht erstellt, Qualitätsmassnahmen & Projektorganisation geschrieben.	MK
1.0rc02	In Bearbeitung	28.02.2010	Risikomanagement und Meilensteine erstellt, Iterationsplanung ergänzt.	MS
1.0rc03	In Bearbeitung	09.03.2010	Arbeitspakete definiert.	MS
1.0rc04	Gegenlesen	09.03.2010	Anpassung der Termine und einige Korrekturen vorgenommen.	MK
1.0rc05	In Bearbeitung	14.03.2010	Zeitplan Meilenstein erstellt und Kapitel 4.4.2 ergänzt.	MK
1.0rc06	Gegenlesen	15.03.2010	Änderungen vorgenommen.	MS
1.0rc07	In Bearbeitung	16.03.2010	Korrektur des Zeitplans (Meilenstein)	MK
1.1rc01	In Bearbeitung	23.03.2010	Zeitplan bis zum nächsten Meilenstein und Dokument mit "Dokument Versionen" ergänzt.	MK
1.1rc02	Gegenlesen	26.03.2010	Gegenlesen, kleine Verbesserungen	MS
1.1	Abschluss	27.03.2010	Dokument überarbeitet und Version 1.1 abgeschlossen.	MS
1.2	Abschluss	15.04.2010	Meilensteine angepasst	MS
1.3	Abschluss	03.05.2010	Meilensteine angepasst	MS
1.4	Abschluss	09.06.2010	Dokument ergänzt, Meilensteine angepasst	MS
1.4	Gegenlesen	16.06.2010	Review des gesamten Dokuments	MK
1.4	Close	17.06.2010	Version 1.4	MK

Inhaltsverzeichnis

1	Einführung	4
1.1	Zweck	4
1.2	Gültigkeitsbereich	4
1.3	Definitionen und Abkürzungen.....	4
1.4	Referenzen	4
2	Projekt Übersicht	5
2.1	Zweck und Ziel	5
2.2	Annahmen und Einschränkungen	5
3	Projektorganisation	6
3.1	Organisationsstruktur	6
3.2	Externe Schnittstellen.....	6
4	Management Abläufe	7
4.1	Projekt Kostenvoranschlag	7
4.2	Besprechungen	7
4.2.1	Mit Betreuer	7
4.2.2	Team-Besprechungen.....	7
4.3	Vorgegebene Dokumente bei der Bachelorarbeit	7
4.4	Projektplan.....	8
4.4.1	Zeitplan.....	8
4.4.2	Beschreibung der Dokumente und deren Inhalte	9
4.4.3	Meilensteine	11
4.4.4	Iterationsplanung	11
5	Risiko Management	13
6	Infrastruktur	14
7	Qualitätsmassnahmen.....	15
7.1	Teammeetings / Sitzungsprotokolle	15
7.1.1	Unit-Tests.....	15
7.2	Reviews	15
7.2.1	Code-Reviews.....	15
7.2.2	Dokumenten-Reviews	15
8	Arbeitspakete.....	16
9	Zeitaufwand ausgewertet.....	20
10	Anhang	22
10.1	Tabellenverzeichnis.....	22

1 Einführung

1.1 Zweck

Der Projektplan der Bachelor wird in diesem Dokument beschrieben. Mit diesem Dokument soll gemäss Vorstellung des Projektleiters weitergearbeitet werden können, falls dieser ausfällt.

1.2 Gültigkeitsbereich

Die Gültigkeit dieses Dokuments bezieht sich auf den Verlauf des ganzen Projekts. Es ist jedoch wegen des iterativen Vorgehens gut möglich, dass über die gesamte Zeit noch gewisse Änderungen vorgenommen werden.

1.3 Definitionen und Abkürzungen

Das Glossar ist im separaten Dokument (Glossar.docx) zu finden.

1.4 Referenzen

Referenz	Quelle

2 Projekt Übersicht

Das Control Panel soll ermöglichen, dass Benutzer Pods über die Webseite reservieren bzw. annullieren können. Eine Reservation bedeutet, einen Pod für eine bestimmte Zeit zu reservieren, in der kein anderer Benutzer auf denselben Pod zugreifen kann. Ein Pod beinhaltet mehrere Geräte wie z. B. Router, Switch, Server, Firewall etc.

Jeder Benutzer kann seine History der Reservationen abrufen (bereits abgelaufene Reservationen, sowie zukünftig getätigte Reservationen). Um eine Reservation zu machen, muss ein Benutzer auf seinem Konto über genügend Punkte (im weiteren Credit Points genannt) verfügen. Credit Points können beim Administrator gekauft werden. Die Kosten sowie die Zeit für einen Credit Point kann der Administrator pro Pod selbst definieren.

Wenn ein Benutzer Reservationen am Laufen hat, sieht er die Geräte seiner reservierten Pods. Diese Geräte kann er während seiner Reservation neu starten und die SSH Verbindungen aller Geräte trennen.

Sollte ein Problem auftreten, kann der Benutzer mit dem Administrator in Kontakt treten.

2.1 Zweck und Ziel

Das Ziel ist, ein Control Panel zu entwickeln, welches dem Kunden ein komfortables Lernen ermöglicht.

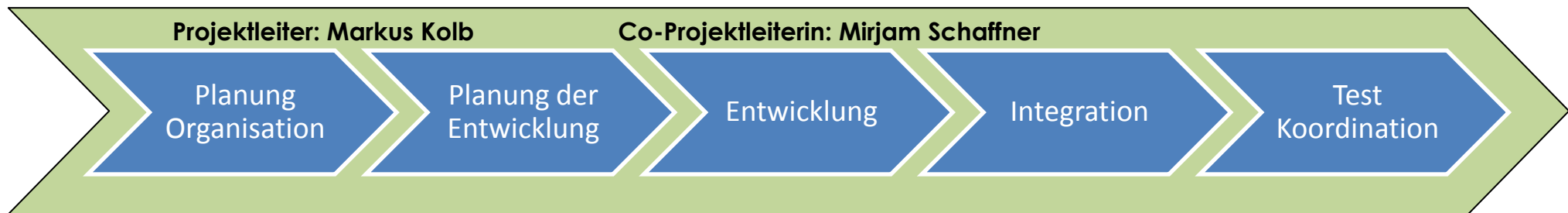
2.2 Annahmen und Einschränkungen

Die Soll-Arbeitszeit pro Projektmitarbeiter und Woche liegt bei ca. 21 Stunden. Bei unerwarteten Problemen und Aufwänden wird die Arbeitszeit soweit als nötig ausgeweitet.

3 Projektorganisation

3.1 Organisationsstruktur

Ablauforganisation:



Planung Organisation:

Prozessverantwortlicher:

Markus Kolb

Artefakte

- Projektplan
- Anforderungsspezifikation

Planung der Entwicklung

Prozessverantwortlicher:

Markus Kolb

Artefakte

- SAD (Analyse)
- Releases
- Datenmodell

Entwicklung

Prozessverantwortlicher:

Markus Kolb

Artefakte

- CODE
- Prototyp 1 & 2

Module kombinieren:

Prozessverantwortlicher:

Markus Kolb

Artefakte

- SAD (Design) Dokument

Test Koordination:

Prozessverantwortlicher:

Markus Kolb

Artefakte

- Code (Unit Tests)
- Testdokumentation
- Systemtest

3.2 Externe Schnittstellen

Rolf Schärer

Betreuung, Beratung, Kontrolle

Beat Stettler

Betreuung, Beratung, Kontrolle

4 Management Abläufe

4.1 Projekt Kostenvoranschlag

Der Umfang dieses Projekts beträgt pro Person 360 Stunden. Da das Projekt aus zwei Mitgliedern besteht, werden insgesamt ca. 720 Stunden aufgewendet. Das Projekt dauert vom 22. Februar 2010 bis 18. Juni 2010.

4.2 Besprechungen

4.2.1 Mit Betreuer

Teilnehmer	MK, MS, RS ev. BS
Datum/Zeit	Jeden Donnerstag 10 Uhr
Ort	HSR Rapperswil (Standort: neue Baracke)

4.2.2 Team-Besprechungen

Teilnehmer	MK, MS
Datum/Zeit	Wöchentlich Dienstag 8:00 Uhr und Donnerstag 9:00 Uhr
Ort	HSR Rapperswil Zimmer 1.260
Zweck	ToDo-Liste durchsehen/aktualisieren, anstehende Arbeit besprechen.

4.3 Vorgegebene Dokumente bei der Bachelorarbeit

Folgende Dokumente müssen bei jeder Bachelorarbeit erstellt werden:

Aufgrund der Vorgaben der Abteilung Informatik werden die Dokumente Abstract, Management Summary & Technischer Bericht bewusst getrennt.

Name	Inhalt	Abgabe
Abstract	Zusammenfassung der Arbeit auf einer Seite, auf dem dafür vorgegebenen Formular.	Am Ende
Management Summary	Das "Management Summary" richtet sich in der Praxis an die "Chefs des Chefs", d.h. an die Vorgesetzten des Auftraggebers (diese sind in der Regel keine Fachspezialisten).	Am Ende
Technischer Bericht	Umfasst eine Einleitung, Übersicht, Ergebnisse & Schlussfolgerungen.	Am Ende
Persönliche Berichte	Persönliche Erfahrungen bei der Arbeit.	Am Ende
Literaturverzeichnis	Im Literaturverzeichnis sind alle verwendeten Quellen (Bücher, Publikationen) aufgelistet.	Am Ende
Dokumente des Projekts	Enthält eine Übersicht über die übrigen Dokumente gemäss Aufgabenstellung bzw. Software-Engineering-Vorgehen.	Am Ende
A0-Poster	Enthält eine Übersicht über das Projekt.	14.06.2010

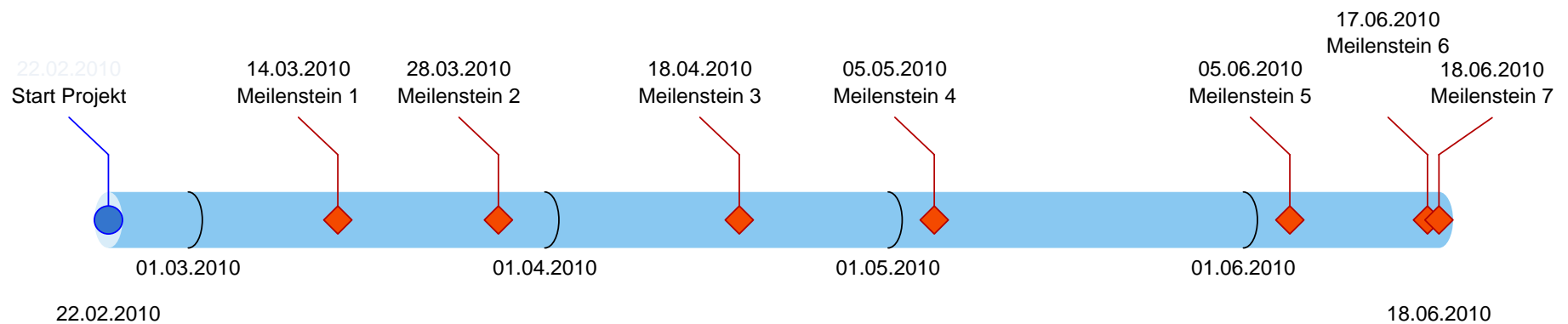
Tabelle 1: Vorgegebene Dokumente

4.4 Projektplan

4.4.1 Zeitplan

Eine detaillierte Zeiterfassung kann im Dokument 02_Projektmonitoring\Zeiterfassung\Zeitplan.xlsx nachgeschlagen werden.

Eine genaue Beschreibung der Meilensteine ist im Kapitel 4.4.3 ersichtlich.



Meilenstein 1	Meilenstein 2	Meilenstein 3	Meilenstein 4	Meilenstein 5	Meilenstein 6	Meilenstein 7
Projektplan und Anforderungsspezifikation ausgearbeitet.	Domain Analyse & SAD (Analyse)	Prototyp 1	Prototyp 2	SAD (Design) Dokumentation	Code-Freeze & Systemtests	Abgabe und Präsentation

4.4.2 Beschreibung der Dokumente und deren Inhalte

Dieses Kapitel beschreibt die Dokumentversionen und deren Inhalte, welche dem Kunden zur Verfügung gestellt werden.

Projektplan

Der Projektplan enthält die Anforderungen an das Projekt. Es enthält die Organisation des Projektes, das Risikomanagement und die Qualitätssicherungsmassnahmen.

Version	Inhalt
1.0	Der Projektplan umfasst die Planung bis zum Meilenstein 2. Alle weiteren Termine sind offen. (bis auf MS6 und MS7)
1.1	Der Projektplan umfasst die Planung bis zum Meilenstein 3. Alle weiteren Termine sind offen. (bis auf MS6 und MS7)
1.2	Der Projektplan umfasst die Planung bis zum Meilenstein 4. Alle weiteren Termine sind offen (bis auf MS6 und MS7).
1.3	Der Projektplan umfasst die Planung bis zum Meilenstein 5.
1.4	Der Projektplan umfasst die Planung bis zum Meilenstein 7. Zudem enthält er alle Versionen der verschiedenen Dokumente.

Tabelle 2: Projektplan

Anforderungsspezifikation

Enthält die Anforderungen an das Produkt. Dabei werden funktionale und nicht funktionale Anforderungen unterschieden.

Version	Inhalt
1.0	Die Anforderungsspezifikationen enthält die Anforderungen an das Produkt.

Tabelle 3: Anforderungsspezifikation

Domain Analyse

Als Erstes wird durch ein Domain Modell eine erste Übersicht über die Problem Domain verschafft. Danach werden die Use Cases der Anforderungsspezifikation als Systemsequenzdiagramme dargestellt und dessen Systemoperationen genauer identifiziert.

Version	Inhalt
1.0	Das Domain Model wird gemäss Larman erfasst.

Tabelle 4: Domain Analyse

SAD (Software Architecture Document)

Zu Beginn des Dokuments wird eine kurze Übersicht über das ganze System gegeben und die diversen architektonischen Entscheidungen & Konzepte erläutert.

Danach gliedert sich das Dokument ganz nach dem „N+1“ View Model gemäss Larman und ist in verschiedene Sichten unterteilt.

Es enthält ebenfalls die Informationen über das Design der Applikation und alle Designentscheide des Projektes.

Version	Inhalt
1.0	Enthält eine Architekturübersicht, die Unterteilung in die Subsysteme & die Schichten gemäss „N+1“ View Model.
1.1	Das Dokument wird mit der Beschreibung des Designs ergänzt.

Tabelle 5: SAD**Technischer Bericht**

Der Technische Bericht soll für einen Ingenieur irgendeiner Fachrichtung verständlich sein.

Version	Inhalt
1.0	Enthält gesamtes Vorgehen unserer Bachelorarbeit und alle Entscheide, die wir getroffen haben. Es zeigt auf, was wir während der ganzen Bachelorarbeitszeit gemacht haben und wie sich das Control Panel fortlaufend verändert hat.

Testbericht

Im Testbericht sind die Functional- und Unit-Tests enthalten. Zudem wird im Testbericht angegeben, wie die Tests ausgeführt werden.

Version	Inhalt
1.0	Enthält alle Functional- und Unit Tests, die wir gemacht haben.

Installation Guide

Der Installation Guide dient dazu, das Control Panel zu installieren und alle Konfigurationen, die benötigt sind, zu beschreiben.

Version	Inhalt
1.0	Enthält die gesamte Installation des Control Panels. Die Konfigurationsanpassungen werden beschrieben und alle Konfigurationsfiles werden mit dem Pfad angegeben.

Glossar

Das Glossar umfasst alle Abkürzungen und Erklärungen von schweren Begriffen.

Version	Inhalt
1.0	Das Glossar der Version 1.0 enthält alle Definitionen die im Projektplan, Anforderungsspezifikation, Domain Analyse und SAD definiert wurden.
1.1	Das Glossar der Version 1.1 enthält alle Definitionen die im Projektplan, Anforderungsspezifikation, Domain Analyse, SAD, Technischer Bericht, Testbericht und Management Summary definiert wurden.

Tabelle 6: Glossar

4.4.3 Meilensteine

In der folgenden Tabelle werden die einzelnen Meilensteine kurz beschrieben. Eine detaillierte Planung kann im Dokument 02_Projektmonitoring\Zeiterfassung\Zeitplan.xlsx nachgeschlagen werden.

MS	Bezeichnung	Beschreibung	Datum
MS1	Projektplan & Anforderungsspezifikation	Das Projekt ist soweit geplant und die Anforderungen soweit abgesteckt, dass für das Projekt die nötigen Informationen vorhanden sind.	14.03.10
MS2	Domain Analyse & SAD (Analyse)	Das Dokument Domain Analyse und SAD (Analyse) wird fertiggestellt.	28.03.10
MS3	Prototyp 1	Im Control Panel soll bereits die Authentifizierung möglich sein.	18.04.10
MS4	Prototyp 2	Reservierungen tätigen bzw. löschen.	05.05.10
MS5	Design	SAD (Design) erstellt.	05.06.10
MS6	Code-Freeze & Systemtest	Code für Dauer der Bachelorarbeit eingefroren. Zudem sind die Systemtests durchgeführt.	17.06.10
MS7	Final & Abgabe Release	Tests, Bugfixing & Dokumentationen abgeschlossen.	18.06.10

Tabelle 7: Meilensteinübersicht

4.4.4 Iterationsplanung

Der Unified Process definiert vier Phasen. Diese vier Phasen werden in mehreren Iterationen durchlaufen. Die folgende Tabelle zeigt eine Übersicht über die einzelnen Iterationen und beschreibt diese mit einigen Stichworten.

Iteration	Beschreibung	Ende	Dauer [W]
Inception 1	Projektplan & Anforderungsspezifikation der Version 1.0 wird geschrieben.	W02	2
Elaboration 1	In dieser Phase wird vor allem am SAD (Analyse) & Domain Analyse gearbeitet.	W05	3
Elaboration 2	Der Prototyp 1 wird in dieser Phase erstellt. Es sollte möglich sein, sich zu authentifizieren.	W08	3
Construction 1	Entwicklungen am Projekt tätigen.	W11	3
Construction 2	Der Prototyp 2 wird erstellt, mit dem Reservierungen tätigen und löschen möglich sein sollten.	W14	3

Transition 1	Verfassen der folgenden Dokumente: SAD (Design), Installationsanleitung und der vorgegebenen Dokumente gemäss Bachelorarbeit. Zudem wird intensiv getestet und eine kleine Testdokumentation geschrieben.	W16	2
Reserve	Die letzten Bugfixings und Tests werden gemacht.	W17	1

Tabelle 8: Übersicht der Iterationen

5 Risiko Management

Nr.	Risikotitel	Risikobeschreibung	max. Schaden [Std]	Wsk.	gew. Schaden [Std]	Massnahmen	Vorausgehende Massnahmen	Priorität
1	Gesamtverlust der Daten durch Hardwarefehler oder unabsichtliches löschen.	Aufwändiges Recovery oder Neuerstellung.	500	1%	5	Einsatz von SVN		3
2	Teilverlust der Daten durch unabsichtliches löschen oder gegenseitiges überschreiben.	Neuerstellung einzelner Codezeilen oder ganzer Files.	15	10%	1.5	Einsatz von SVN		2
3	Know How konzentriert auf eine Person	Know How Verlust bei Ausfall dieser Person	100	5%	5	Laufender Austausch von Informationen in Sitzungen		3
4	Software Architektur hält den Anforderungen nicht stand.	Architektur muss angepasst und Teile des Codes neu geschrieben werden.	200	5%	10	Detaillierte Anforderungsanalyse und Architekturentwurf		1

Tabelle 9: Risiko Management

Prioritäten:

1 = Hoch

2 = Mittel

3 = Niedrig

6 Infrastruktur

Hardware	Software	Organisatorisches
Persönliche Notebooks	Subversion	HSR Räumlichkeiten
Computer HSR	Microsoft Office 2007	
Druckinfrastruktur HSR	MySQL	
Server INS	Ubuntu 9.10	
	Symfony 1.4.3	
	Enterprise Architect 7.0	
	PHP 5.3	
	Apache	

Tabelle 10: Übersicht der genutzten Infrastruktur

7 Qualitätsmassnahmen

7.1 Teammeetings / Sitzungsprotokolle

Wie im Kapitel 4.2.2 beschrieben, organisiert das Team wöchentliche Meetings. Für das Meeting mit dem Betreuer muss ein Sitzungsprotokoll geschrieben werden, in dem alle Erkenntnisse und Entscheide aus dem Meeting sowie alle neu definierten Aufgaben protokolliert sind. So wird sichergestellt, dass alle Teammitglieder immer auf demselben Stand sind und auch alles mitbekommen, falls sie einmal nicht an einem Meeting teilnehmen können.

7.1.1 Unit-Tests

Jeder Entwickler ist aufgefordert an sinnvollen Stellen in seinem Source-Code fortlaufend Unit-Tests zu schreiben und gegen diese Tests zu entwickeln. Bei den Kernfunktionalitäten wird nach dem „Test-Driven-Development“ Verfahren gearbeitet. Bevor ein Entwickler seinen neuen Source Code ins SVN eincheckt, müssen alle seine Unit-Tests erfolgreich absolviert worden sein.

7.2 Reviews

7.2.1 Code-Reviews

Jeder Code muss mindestens einmal gegenseitig geprüft und wenn möglich, verbessert werden.

Der Revisor überprüft:

- Verständlichkeit des Codes (evtl. durch Kommentare ergänzen)
- Korrektheit des Codes
- Auftreten von „Code Smells“
- Effizienz von komplexen Algorithmen

7.2.2 Dokumenten-Reviews

Jedes geschriebene Dokument muss mindestens einmal von einer anderen Person durchgeschaut und wenn nötig, verbessert werden. Der Revisor trägt sich in die „Revisions History“ des entsprechenden Dokuments mit Datum und Kommentar ein.

Der Revisor überprüft:

- Rechtschreibung / Grammatik
- Einhalten des vorgegebenen Dokumenten Templates
- Konsistenz der verschiedenen Texte
- Vollständigkeit des Dokuments

8 Arbeitspakete

1	Projekt Management	Verantwortlicher	64h
1.1	Meetings abhalten	Team	17h
Beschreibung	Sitzungen mit Rolf Schärer, in denen Fragen zum Projekt geklärt und Probleme aufgezeigt werden.		
Abhängigkeiten	-		
Risiken/Probleme	-		
1.2	Projektplan erstellen	MK	3h
Beschreibung	Erstellen des Projektplan Dokuments. Projektübersicht, Projektorganisation, Managementabläufe, Risikomanagement, Arbeitspakete, Infrastruktur, Qualitätsmassnahmen grob definieren.		
Abhängigkeiten	[1.1] Meetings abhalten.		
Risiken/Probleme	-		
1.3	Projektplan erweitern	MS	8h
Beschreibung	Projektplan erweitern. Arbeitspakete definieren und Verantwortlichkeiten zuteilen, Meilensteine und Iterationen festhalten, Zeitplan erfassen, Risikomanagement erfassen.		
Abhängigkeiten	[1.2] Projektplan erstellt.		
Risiken/Probleme	-		
1.4	Projektplan verfeinern	Team	10h
Beschreibung	Projektplan verfeinern. Arbeitspakete verfeinern, Zeitplan überarbeiten.		
Abhängigkeiten	[1.3] Projektplan erweitert.		
Risiken/Probleme	-		
1.5	Projektplan gegengelesen	Team	10h
Beschreibung	Projektplan wird im Team verteilt und von beiden einzeln gelesen und Korrekturen eingebracht.		
Abhängigkeiten	[1.4] Projektplan verfeinert.		
Risiken/Probleme	-		
1.6	Projektplan Review	Team	2h
Beschreibung	Projektplan mit Projektbetreuer besprechen.		
Abhängigkeiten	[2.4] Projektplan gegengelesen.		
Risiken/Probleme	-		
1.7	Infrastruktur Aufbau & Betrieb	MK	14h
Beschreibung	SVN Repository erstellen, Accounts erstellen.		
Abhängigkeiten	-		
Risiken/Probleme	-		

Tabelle 11: Arbeitspakete der Disziplin Projekt Management

2	Requirements	Verantwortlicher	27h
2.1	Anforderungen definieren	MS	8h
Beschreibung	Definition der funktionalen und nicht funktionalen Anforderungen an das Produkt.		
Abhängigkeiten	[1.1] Meeting		
Risiken/Probleme	-		
2.2	Brief Use Cases beschreiben	MS	4h
Beschreibung	Aus funktionalen Anforderungen Use Cases erstellen.		
Abhängigkeiten	[2.1] Anforderungen definiert.		
Risiken/Probleme	-		
2.3	Use Case Diagramm erstellen	MS	3h
Beschreibung	Use Case Diagramm anhand der Use Cases erstellen.		
Abhängigkeiten	[2.2] Use Cases definiert.		
Risiken/Probleme	-		
2.4	Anforderungsspezifikation gegengelesen	Team	10h
Beschreibung	Alle erstellten Dokumente werden im Team verteilt und von beiden einzeln gegengelesen, sowie Korrekturen vorgenommen.		
Abhängigkeiten	[2.2] Use Cases definiert.		
Risiken/Probleme	-		
2.5	Anforderungsspezifikation Review	Team	2h
Beschreibung	Anforderungsspezifikation mit Rolf Schärer besprechen.		
Abhängigkeiten	[2.2] Use Cases definiert.		
Risiken/Probleme	-		

Tabelle 12: Arbeitspakete der Disziplin Requirements

3	Domain Analyse	Verantwortlicher	32h
3.1	Domain Model erstellen	MK	12h
Beschreibung	Erstellen des Domain Models mit Hilfe der Use Cases.		
Abhängigkeiten	[2.2] Use Cases definiert.		
Risiken/Probleme	-		
3.2	System Sequenzdiagramme erstellen	MS	12h
Beschreibung	Systemsequenzdiagramme der wichtigsten Interaktionen erstellen.		
Abhängigkeiten	[2.2] Use Cases definiert. [3.1] Domain Model erstellt.		
Risiken/Probleme	-		

3.3	System Contracts	MS	6h
Beschreibung	Dokumentieren der System Contracts.		
Abhängigkeiten	[2.2] Use Cases definiert.		
Risiken/Probleme	-		
3.4	Validierung Analyse gemäss Requirements	Team	2h
Beschreibung	Überprüfen, ob Analyse alle Requirements erfüllt.		
Abhängigkeiten	[3.1] Domain Model erstellt.		
Risiken/Probleme	-		

Tabelle 14: Arbeitspakete der Disziplin Domain Analyse

4	Design	Verantwortlicher	43h
4.1	Data Model	MK	8h
Beschreibung	Datenbankschema erstellen.		
Abhängigkeiten	[3.1] Domain Model erstellt.		
Risiken/Probleme	-		
4.2	Software Architecture Document	Team	22h
Beschreibung	Klassendiagramm, Objekt Interaktionsdiagramm, etc.		
Abhängigkeiten	[3.1] Domain Model erstellt.		
Risiken/Probleme	-		
4.3	Paper Prototype	MK	3h
Beschreibung	Paperprototype erstellen.		
Abhängigkeiten	[2.1] Anforderungen definiert.		
Risiken/Probleme	-		
4.4	Benutzeranleitung	Team	10h
Beschreibung	Installationsanleitung erstellen.		
Abhängigkeiten	[5.0] Development erstellt.		
Risiken/Probleme	-		

Tabelle 135: Arbeitspakete der Disziplin Design

5	Development	Verantwortlicher	440h
5.1	Authentifizierung Prototype	Team	60h
Beschreibung		Authentifizierungsprototyp erstellen.	
Abhängigkeiten			
Risiken/Problemen			
5.2	Reservation Prototype	MS	180h
Beschreibung		Reservationsprototyp erstellen.	
Abhängigkeiten		[5.1] Authentifizierungsprototyp erstellt.	
Risiken/Probleme			
5.3	Development	Team	200h
Beschreibung		Allgemeine Entwicklung der Software.	
Abhängigkeiten		[5.2] Reservationsprototyp erstellt.	
Risiken/Probleme		-	

Tabelle 146: Arbeitspakete der Disziplin Implementation

6	Test	Verantwortlicher	60h
6.1	Unit Tests	Team	30h
Beschreibung		Erstellen der Unit Tests.	
Abhängigkeiten		Hängt mit den jeweiligen Bereichen zusammen.	
Risiken/Probleme		-	
6.2	Integration Tests	Team	10h
Beschreibung		Erstellen der Funktionstests. Testen des GUIs von externen Nutzern.	
Abhängigkeiten		-	
Risiken/Probleme		Keine Testpersonen verfügbar.	
6.3	System Tests	MS	10h
Beschreibung		Erstellen der Systemtests (testen der funktionalen und nicht funktionalen Anforderungen).	
Abhängigkeiten		Hängt mit den jeweiligen Bereichen zusammen.	
Risiken/Probleme		-	
6.4	Bugfixing	Team	10h
Beschreibung		Beheben von gefundenen Fehlern bei Tests.	
Abhängigkeiten		[6.1] Unit Tests, [6.2] Integration Tests, [6.3] System Tests	
Risiken/Probleme		Bei ungenauem arbeiten werden mehr Fehler auftreten, wobei mehr Zeit in dieses Arbeitspaket investiert werden muss.	

Tabelle 157: Arbeitspakete der Disziplin Test

7	Deployment	Verantwortlicher	14h
7.1	Dokumente laut Anleitung BA	MK	10h
Beschreibung	Die Dokumente Abstract, Management Summary, Poster, Technischer Bericht, Persönliche Berichte, Glossar, Stundenauswertung erstellen.		
Abhängigkeiten	Projektabschluss		
Risiken/Probleme	-		
7.2	Enterprise Architect Synchron		4h
Beschreibung	Beheben von gefundenen Fehlern bei Tests		
Abhängigkeiten	-		
Risiken/Probleme	Bei ungenauem Arbeiten werden mehr Fehler auftreten, wobei mehr Zeit in dieses Arbeitspaket investiert werden muss.		

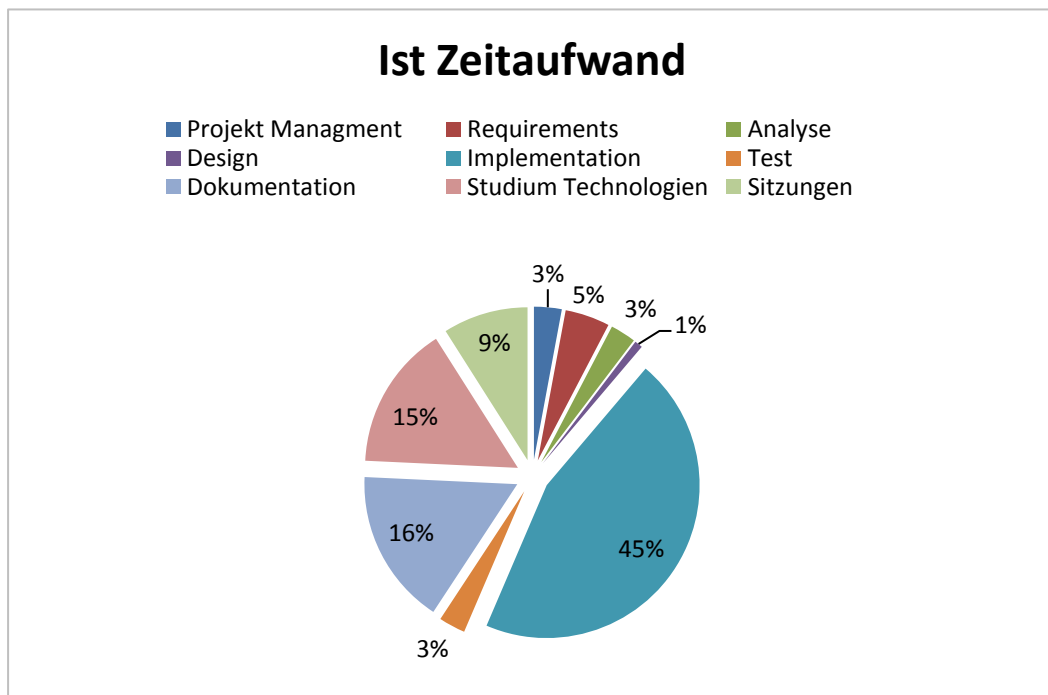
Tabelle 1816: Arbeitspakete der Disziplin Deployment

8	Reserve	Verantwortlicher	43h
---	---------	------------------	-----

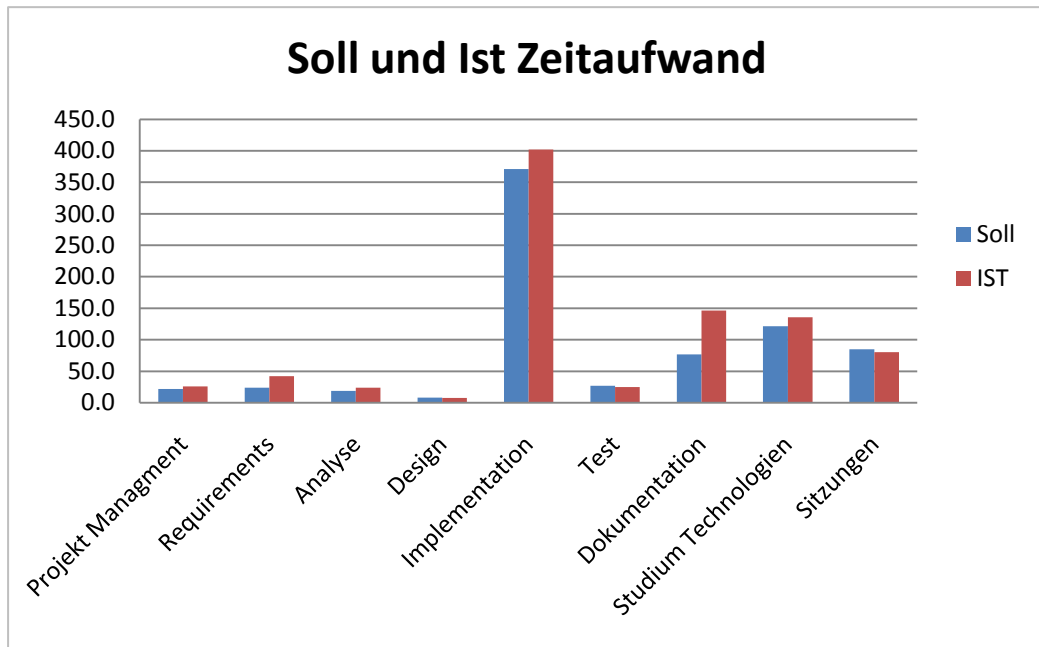
Tabelle 179: Risiko

9 Zeitaufwand ausgewertet

Wie aus dem folgenden Diagramm ersichtlich ist, haben wir fast die Hälfte der Zeit für die Implementierung gebraucht. Einen weiteren grossen Zeitaufwand gaben die Dokumentation, sowie die Einarbeitung in die Technologien.



Meistens haben wir mehr gearbeitet, als wir dafür geplant hatten. Einzig für das Design, die Tests und die Sitzungen haben wir den Soll Arbeitsaufwand nicht erreicht. Zu den Tests ist zu sagen, dass wir mehr getestet haben, als es im untenstehenden Diagramm dargestellt wird. Dies ist so, weil wir einige Zeit des Testens unter Implementierung aufgeführt haben.



10 Anhang

10.1 Tabellenverzeichnis

Tabelle 1: Vorgegebene Dokumente	7
Tabelle 2: Projektplan	9
Tabelle 3: Anforderungsspezifikation	9
Tabelle 4: Domain Analyse	9
Tabelle 5:: SAD	10
Tabelle 5: Glossar	10
Tabelle 8: Meilensteinübersicht	11
Tabelle 9: Übersicht der Iterationen	12
Tabelle 10: Risiko Management	13
Tabelle 11: Übersicht der genutzten Infrastruktur	14
Tabelle 12: Arbeitspakete der Disziplin Projekt Management	17
Tabelle 13: Arbeitspakete der Disziplin Requirements	17
Tabelle 15: Arbeitspakete der Disziplin Design	18
Tabelle 16: Arbeitspakete der Disziplin Implementation	19
Tabelle 17: Arbeitspakete der Disziplin Test	19
Tabelle 18: Arbeitspakete der Disziplin Deployment	20
Tabelle 19: Risiko	20

Anforderungsspezifikation

Version 1.0

Projekt:

Remote Lab Management System

Projektmitglieder:

Mirjam Schaffner

Markus Kolb

Betreuer:

Beat Stettler

Rolf Schärer

Revision				
Version	Status	Datum	Beschreibung/Änderung	Autor
1.0rc01	In Bearbeitung	26.02.2010	Erstellen der Anforderungsspezifikation	MS
1.0rc02	In Bearbeitung	02.03.2010	Review und Ergänzung der AS	MK
1.0rc03	In Bearbeitung	05.03.2010	Standards + UC ergänzt, einige Änderungen	MS
1.0rc04	In Bearbeitung	07.03.2010	UC Diagramm eingefügt, Einschränkungen und Ergänzungen	MS
1.0rc05	Gegenlesen	10.03.2010	Gegenlesen und einige Korrekturen vorgenommen	MK
1.0rc06	In Bearbeitung	11.03.2010	Ganzes Dokument überarbeitet	MS
1.0rc07	In Bearbeitung	12.03.2010	Ganzes Dokument überarbeitet	MK
1.0rc08	Gegenlesen	15.03.2010	Gegenlesen und einige Korrekturen anbringen	MS
1.0rc09	In Bearbeitung	16.03.2010	Use Case Tabelle erstellt, SSD eingefügt, Ergänzungen gemacht, Use Cases abgeändert, nicht funktionale Anforderungen erweitert	MS
1.0rc10	Gegenlesen	17.03.2010	Gegenlesen	MK
1.0rc11	Gegenlesen	23.03.2010	Änderungen gemäss Sitzung mit Rolf Schärer	MS
1.0rc12	In Bearbeitung	13.04.2010	Änderungen gemäss Sitzung mit Herrn Sommerlad	MS
1.0rc13	In Bearbeitung	15.06.2010	Anforderungsspezifikation gemäss Sitzung mit Herrn Stettler abgeändert.	MS
1.0	Close	17.06.2010	Version 1.0	MK

Inhaltsverzeichnis

1	Einführung	4
1.1	Zweck.....	4
1.2	Gültigkeitsbereich.....	4
1.3	Definitionen und Abkürzungen.....	4
1.4	Referenzen.....	4
2	Allgemeine Beschreibung	5
2.1	Produkt Perspektive.....	5
2.1.1	Systemschnittstellen.....	5
2.1.2	Benutzerschnittstelle.....	5
2.1.3	Hardwareschnittstellen.....	5
2.1.4	Softwareschnittstellen.....	5
2.1.5	Kommunikationsschnittstellen (Netzwerkprotokolle etc.).....	5
2.2	Produkt Funktion.....	5
2.3	Benutzer Charakteristik.....	6
2.4	Einschränkungen.....	6
2.5	Annahmen.....	6
2.6	Abhängigkeiten.....	6
3	Remote Lab Management System	7
3.1	GUI – Paperprototyp.....	7
3.2	Funktionale Anforderungen.....	8
3.2.1	Aktoren.....	8
3.2.2	User Story.....	8
3.2.3	Use Case-Diagramme.....	9
3.2.4	Use Case Tabelle.....	10
3.2.5	Use Cases brief (relevante UCs).....	11
3.2.6	Use Cases („nice-to-have“ UCs).....	12
3.3	Nichtfunktionale Anforderungen.....	13
3.3.1	Implementierung.....	13
3.3.2	Zuverlässigkeit.....	13
3.3.3	Erweiterbarkeit.....	13
3.3.4	Lizenzanforderungen.....	13

1 Einführung

1.1 Zweck

Dieses Dokument beschreibt die funktionalen und nicht funktionalen Anforderungen an das Produkt.

1.2 Gültigkeitsbereich

Die Gültigkeit dieses Dokuments bezieht sich auf den Verlauf des ganzen Projekts. Es ist jedoch wegen des iterativen Vorgehens gut möglich, dass über die gesamte Zeit noch gewisse Änderungen vorgenommen werden.

1.3 Definitionen und Abkürzungen

Das Glossar ist im separaten Dokument (Glossar.docx) zu finden.

1.4 Referenzen

Referenz	Quelle
UML 2 und Patterns angewendet – Objektorientierte Softwareentwicklung	Craig Larman Mitp-Verlag/Bonn

2 Allgemeine Beschreibung

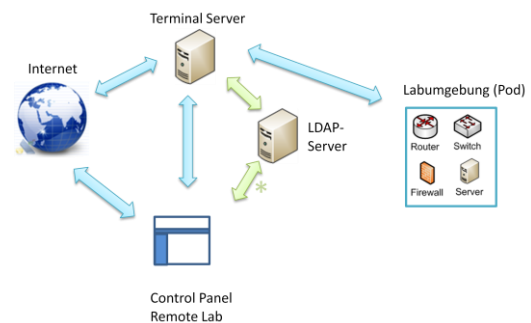
Das Control Panel soll ermöglichen, dass Benutzer Pods über die Webseite reservieren bzw. annullieren können. Eine Reservation bedeutet, einen Pod für eine bestimmte Zeit zu reservieren, in der kein anderer Benutzer auf denselben Pod zugreifen kann. Ein Pod beinhaltet mehrere Geräte wie z. B. Router, Switch, Server, Firewall etc.

Jeder Benutzer kann seine History der Reservationen abrufen (bereits abgelaufene Reservationen, sowie zukünftig getätigte Reservationen). Um eine Reservation zu machen, muss ein Benutzer auf seinem Konto über genügend Punkte (im weiteren Credit Points genannt) verfügen. Credit Points können beim Administrator gekauft werden. Die Kosten sowie die Zeit für einen Credit Point kann der Administrator selbst definieren.

Jeder Benutzer sieht während einer Reservation die Geräte seines reservierten Pods. Diese Geräte kann er während seiner Reservation neu starten und die SSH Verbindungen aller Geräte trennen.

2.1 Produkt Perspektive

Das Control Panel muss in die bereits bestehende Umgebung eingebettet werden können. Das heisst, es muss eine Kommunikation zwischen dem freeRadius Server und dem Control Panel geben, sowie eine Abhängigkeit zum Active Directory (LDAP Datenbank) und einen Zugriff über das Internet, sodass auf das Control Panel zugegriffen werden kann.

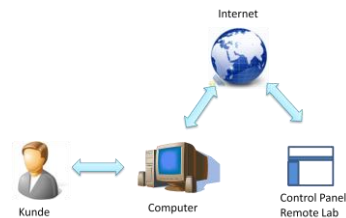


2.1.1 Systemschnittstellen

(siehe Dokument Domain Analyse)

2.1.2 Benutzerschnittstelle

Um sich am Control Panel anzumelden, muss der Benutzer via Internet auf das Control Panel zugreifen.



2.1.3 Hardwareschnittstellen

Das Control Panel muss die CISCO Geräte ansteuern können, um diese neu zu starten.

2.1.4 Softwareschnittstellen

Wenn sich ein Benutzer am Terminal Server anmelden will, macht RADIUS mittels XML-RPC eine Abfrage an das Server Script, welches Zugriff auf die Control Panel Datenbank hat.

2.1.5 Kommunikationsschnittstellen (Netzwerkprotokolle etc.)

Damit freeRadius mit dem Control Panel kommunizieren kann, müssen wir den Methodenaufruf XML-RPC implementieren.

2.2 Produkt Funktion

Über die Schnittstelle „Control Panel“ soll es möglich sein, sich zu authentifizieren. Der Benutzername sowie das Passwort müssen mit den Einträgen des Active Directories (LDAP Datenbank) überprüft werden. Der Loginname sowie die Benutzerdaten werden zudem im Control Panel gespeichert, da wir kein Schreibrecht auf der LDAP Datenbank erhalten. Zudem muss es möglich sein, dass sich ein neuer Kunde registrieren kann.

Im Control Panel muss der Benutzer auf die ihm berechtigten Pods Reservationen tätigen können. Die Reservations-History eines Benutzers muss ersichtlich sein. Ein Benutzer sieht die Reservationen, die auf Pods gemacht werden, weiss jedoch nicht, wer die Reservation gemacht hat. Seine Reservationen kann der Benutzer auf ein Gerät, z.B. seinem Computer, herunterladen.

Ein Benutzer kann beim Administrator Credit-Points kaufen. Um Credit Points zu kaufen, muss der Benutzer ein Formular mit Namen, Vornamen, E-Mail und Anzahl Credit Points, die er kaufen möchte, ausfüllen. Der Administrator erhält ein Mail mit den Angaben, die der Benutzer im Formular ausgefüllt hat. Sobald er das Geld des Benutzers erhalten hat, fügt er dem Benutzer die bezahlten Credit Points hinzu.

Um mit dem Administrator in Kontakt zu treten, kann der Benutzer Formulare ausfüllen, in denen er Kommentare schreiben oder Probleme melden kann. Per Mail empfängt der Administrator die Mitteilungen, die der Benutzer in den Formularen angegeben hat.

Wenn ein Benutzer aus der LDAP Datenbank gelöscht wird, muss der Administrator die Möglichkeit haben, diesen Benutzer im Control Panel zu löschen.

Der Administrator muss im Control Panel die Gruppen verwalten können, da er definiert, welche Gruppen auf welche Pods zugreifen dürfen.

Der Benutzer darf während einer Reservation im Action Reservation Modul nur die Pods sehen, die er zurzeit reserviert hat. Er kann während der Reservation die Geräte der reservierten Pods neu starten oder die SSH Verbindungen trennen. Sofern der Benutzer keine Reservation hat, werden ihm auch keine Pods angezeigt. Der Administrator sieht jedoch immer alle Pods und kann zu jeder Zeit die Geräte neu starten oder die SSH Verbindungen trennen.

Das Control Panel sollte zweisprachig zur Verfügung stehen. Es sollte zwischen den Sprachen Englisch und Deutsch gewechselt werden können.

2.3 Benutzer Charakteristik

Zielgruppe dieses Produkts sind Teilnehmer eines Kurses (wie z.B. dem CCIE), sowie externe Personen, die nicht am Kurs teilnehmen, die Pods für Labs benutzen möchten.

2.4 Einschränkungen

Aus zeitlichen Gründen werden wir die Funktionen Konfigurationen hinauf- bzw. herunterladen, einspielen und speichern nicht implementieren.

2.5 Annahmen

Neukunden werden ebenfalls im Active Directory erfasst. Dies muss der Administrator manuell machen, da das Control Panel kein Schreibrecht in der LDAP Datenbank hat.

2.6 Abhängigkeiten

Eine Abhängigkeit besteht zum Active Directory. Die Benutzer sind in der LDAP Datenbank erfasst und authentifizieren sich direkt beim Control Panel, welches den Benutzernamen und das Passwort mit den Einträgen des Active Directorys (LDAP DB) überprüft.

Eine weitere Abhängigkeit besteht zwischen dem freeRADIUS Server und dem Control Panel. Wenn sich ein Benutzer am Terminalserver anmeldet, muss nach der Überprüfung seines Benutzernamens und Passwort überprüft werden, ob dieser Benutzer im Control Panel eine Reservation zu diesem Zeitpunkt und auf den gewünschten Pod hat. Dazu muss RADIUS eine Kommunikation zu unserem Control Panel haben.

Um die Geräte neu zu starten, müssen wir die Geräte ansteuern können.

3 Remote Lab Management System

3.1 GUI – Paperprototyp

Login

A hand-drawn paper prototype of a login form. At the top left is the logo 'ins'. Below it is a box titled 'Control Panel Login'. Inside this box are two input fields for 'Benutzername' and 'Passwort', a 'Remember' checkbox, a 'Sign in' button, and a '>> Register' link.

Registrierung

A hand-drawn paper prototype of a registration form. At the top left is the logo 'ins'. Below it is a box titled 'Register'. The form contains input fields for 'Name', 'Surname', 'Address', 'Zip', 'Town', 'Phone', 'Mail', and a 'Comment' text area. At the bottom left is a 'Back' button and at the bottom right is a 'Send' button.

Website

A hand-drawn paper prototype of a website layout. At the top left is the logo 'ins'. Below it is a horizontal navigation bar with links: 'Home', 'Reservation', 'My Calendar', 'My Configs', 'My Profile', 'Active Reservation', 'Contact us', and 'Logout'. In the top right corner of the navigation bar is 'DE/EN'. Below the navigation bar is a 'Navigation Panel' on the left. The main content area has a 'Home' header and a large box containing the text 'Welcome to INS Reservation Center'. On the right side, there is a vertical list of menu items: 'Pool Kategorie', 'Pool', 'Device Kategorie', 'Device', 'Group Policy', 'LDAP Group', 'User', and 'Lab'.

3.2 Funktionale Anforderungen

Die Authentifizierung geschieht über das Control Panel gegenüber einer LDAP Datenbank. Das Management der Laborumgebungen (Pods) und Geräte neu starten soll via zentrales Control Panel erfolgen.

3.2.1 Aktoren

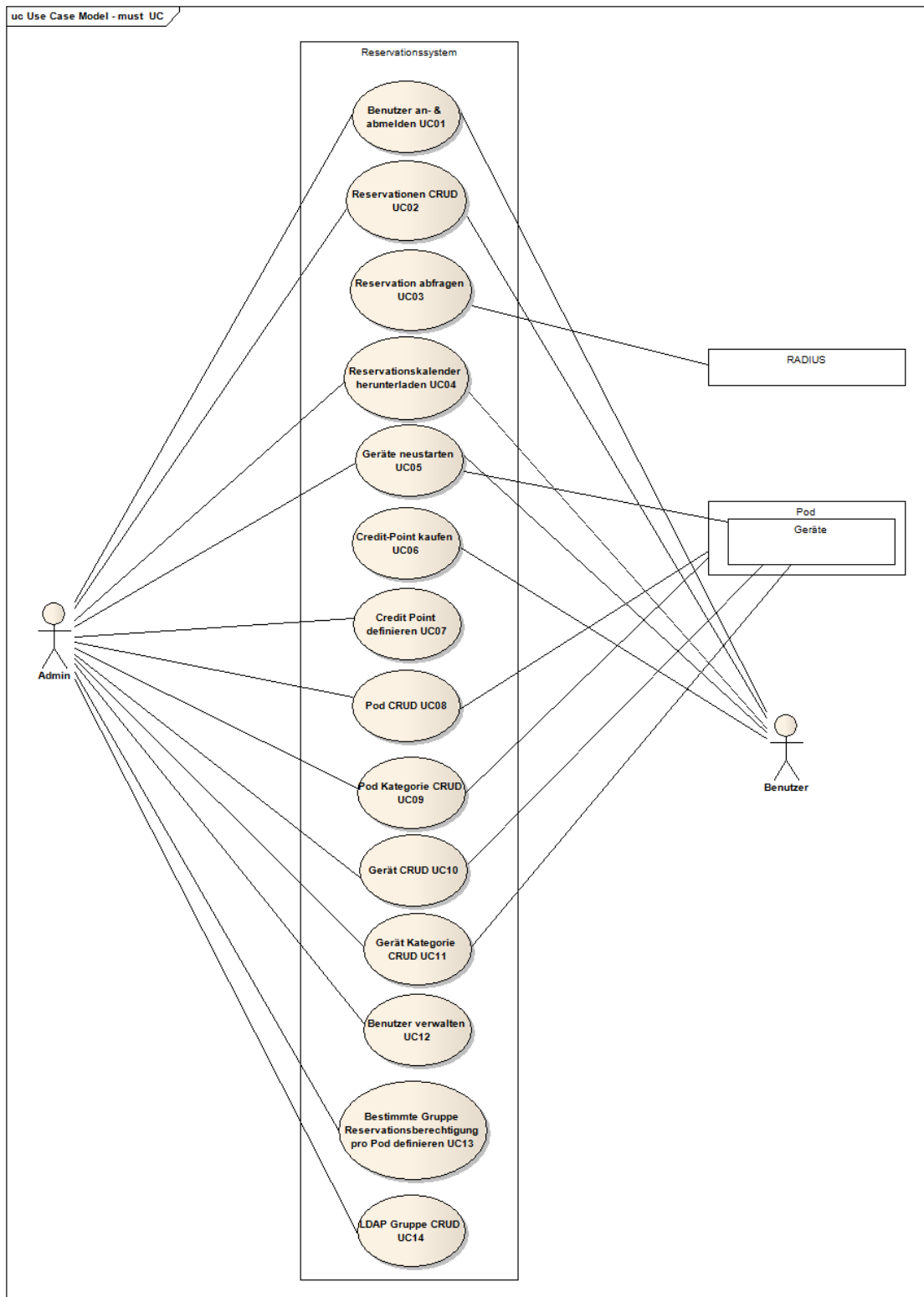
Administrator	User
<p>Der Administrator verwaltet die Geräte, Gerätekategorien, Labs, Reservationen für Benutzer, Gruppenrechte, Podkategorien und Pods. Der Administrator kann zudem die Benutzerdaten eines Benutzers oder die LDAP Gruppe bearbeiten. Zudem kann der Administrator einem Benutzer Credit Points hinzufügen oder löschen, sowie bereits abgelaufene Reservationen löschen. Wenn eine Gruppe keine Credit Points für Reservationen benötigen, kann dies der Administrator für eine Gruppe pro Pod definieren.</p>	<p>Benutzer können auf alle Pods Reservationen tätigen, auf die sie Rechte besitzen. Sie müssen dazu über genügend Credit Points verfügen.</p>

3.2.2 User Story

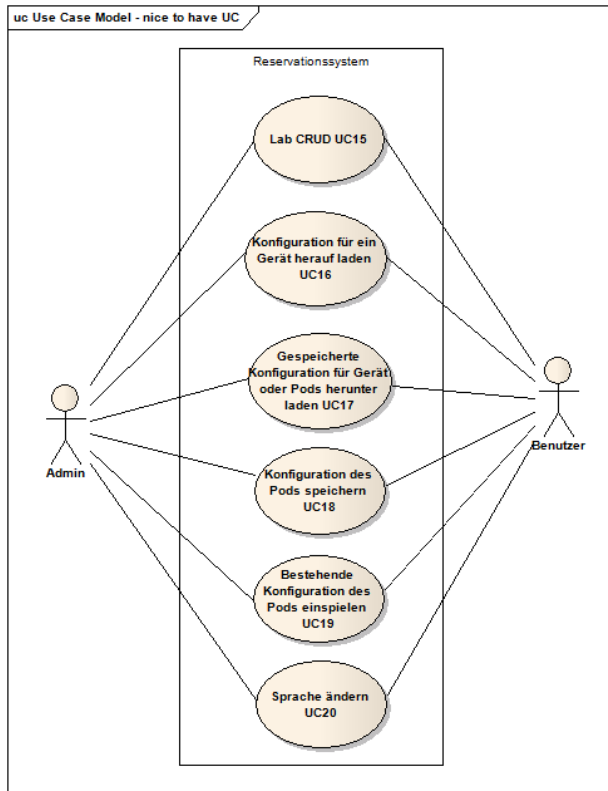
1	<p>Ein Kunde möchte einen Pod reservieren, besitzt jedoch noch kein Login. Er muss sich zuerst am Control Panel registrieren und abwarten, bis der Administrator ihm die Zugangsdaten per Mail sendet.</p>
2	<p>Ein Benutzer möchte eine Reservation auf einen Pod tätigen. Dazu muss dieser zuerst genügend Credit Points beim Administrator kaufen. Danach kann er im Reservationskalender nachsehen, wann der Pod noch nicht belegt ist und gleichzeitig eine Reservation tätigen.</p>
3	<p>Sobald ein Benutzer an einem Lab arbeitet und die Reservationszeit für diesen Pod abgelaufen ist, kann der Benutzer über das Control Panel weder die Geräte neu starten, noch die SSH Verbindung mehr trennen. Solange er jedoch am Terminal Server angemeldet ist, kann er weiterhin auf dem Pod arbeiten und wird nicht automatisch abgemeldet.</p>
4	<p>Braucht der Benutzer mehr Credit Points, kann er diese kaufen. Dazu muss der Benutzer ein Kontaktformular ausfüllen, die Anzahl Credit Points angeben und das Formular senden. Dann muss der Benutzer abwarten bis er die Credit Points erhält.</p>

3.2.3 Use Case-Diagramme

3.2.3.1 Must Use Cases



3.2.3.2 Nice to have Use Cases



3.2.4 Use Case Tabelle

Use Cases	Administrator	Benutzer
Benutzer an- & abmelden	X	X
Reservation CRUD	X	X
Reservation abfragen		
Reservationskalender herunterladen	X	X
Gerät neu starten	X	X
Credit Point kaufen		X
Credit Point definieren	X	
Pod CRUD	X	
Pod Kategorie CRUD	X	
Gerät CRUD	X	
Gerät Kategorie CRUD	X	
Benutzer verwalten	X	
Bestimmte Gruppe Reservationsberechtigung pro Pod definieren	X	
LDAP Gruppe CRUD	X	
Lab CRUD	X	X

Konfiguration für ein Gerät herauf laden	X	X
Gespeicherte Konfiguration für Geräte oder Pod's herunter laden	X	X
Konfiguration des Pods speichern	X	X
Bestehende Konfiguration des Pods einspielen	X	X
Sprache ändern	X	X

3.2.5 Use Cases brief (relevante UCs)

Die Funktionalitäten der Use Cases sind in den nicht funktionalen Anforderungen im Kapitel 3.3 definiert.

3.2.5.1 Benutzer an- & abmelden (UC01)

Benutzer melden sich über das Control Panel an oder ab. Der Administrator kann als einziger ein Benutzerkonto in der LDAP Datenbank erfassen, daher können sich zukünftige Benutzer nicht selbst registrieren.

3.2.5.2 Reservation CRUD (UC02)

Benutzer können neue Reservationen tätigen, Reservationen löschen, Reservationen bearbeiten und den Reservationskalender anschauen. Im Reservationskalender sind nur die Reservationen ersichtlich. Von wem die Reservationen getätigt wurden, wird nicht angezeigt. Um Reservationen machen zu können, muss der Benutzer über genügend Credit Points verfügen. Sobald eine Reservation gemacht wird, zieht das System dem Benutzer die Credit Points ab, die es für die Buchung des Pods benötigt. Zwischen zwei Reservationen wird eine vom Administrator festgelegte Zeit abgewartet, in der keine Reservation gemacht werden kann. Der Administrator muss von jedem Benutzer die Reservationen löschen können, auch wenn diese am Laufen sind. Der Administrator kann Reservationen für Benutzer erstellen und angeben, wie viele Credit Points diese Reservation kostet. Er kann die Reservation in der Vergangenheit erstellen oder diese ebenso löschen.

Reservationen auflisten

Ein Benutzer kann alle seine Reservationen auflisten, d.h. bereits abgelaufene Reservationen sowie zukünftige bereits getätigte Reservationen werden aufgezeigt. Diese Auflistung der Reservationen wird als History bezeichnet. Der Administrator hingegen kann von jedem einzelnen Benutzer die Reservationen auflisten und eine gesamte Auflistung der Reservationen aller Benutzer anzeigen lassen.

3.2.5.3 Reservation abfragen (UC03)

Sobald sich der Benutzer am Terminal Server anmeldet, fragt RADIUS das Reservationssystem an, ob eine Reservation zum gewünschten Pod und zu dieser Zeit besteht.

3.2.5.4 Reservationskalender herunterladen (UC04)

Benutzer lädt den Reservationskalender mit z.B. Outlook auf seinem Computer herunter. Alle seine eigenen Reservationen werden nun in seinem Outlook-Kalender angezeigt.

3.2.5.5 Gerät neustarten (UC05)

Ein Gerät wie Router, Switch, Server, Firewall etc. wird neu gestartet. Die SSH Verbindung aller Geräte kann getrennt werden. Der Benutzer sieht die Geräte der reservierten Pods nur während der Reservation selbst. Hingegen sieht der Administrator alle Geräte von allen Pods zu jeder Zeit.

3.2.5.6 Credit-Point kaufen (UC06)

Benutzer können Credit Points kaufen, um Pods zu reservieren.

3.2.5.7 Credit Point definieren (UC07)

Der Administrator definiert die Anzahl Stunden pro Credit Point, d.h. er bestimmt, wie lange ein Benutzer für einen Credit Point einen Pod reservieren darf. Somit legt der Administrator ebenfalls die mindest Reservationszeit für einen Pod fest.

3.2.5.8 Pod CRUD (UC08)

Administrator verwaltet die Pods, d.h. er kann einen Pod erstellen, löschen, bearbeiten und den Pod einer Pod-Kategorie zuweisen.

3.2.5.9 Pod Kategorie CRUD (UC09)

Administrator verwaltet die Pod Kategorie, d.h. er kann eine neue Pod Kategorie erstellen, löschen und Änderungen vornehmen. Pro Kategorie kann eine mindest Reservationszeit festgelegt werden, welche jeweils um 15 Minuten beim reservieren, erhöht werden kann. Der Administrator kann die Credit Point Kosten pro Stunde für die jeweilige Kategorie definieren.

3.2.5.10 Gerät CRUD (UC10)

Administrator verwaltet die Geräte, d.h. er kann ein neues Gerät erstellen, ein Gerät löschen und das Gerät einer Gerätekategorie zuweisen. Ein Gerät kann jeweils nur einem Pod zugewiesen werden.

3.2.5.11 Gerät Kategorie CRUD (UC11)

Administrator verwaltet die Gerätekategorie, d.h. er kann eine neue Gerätekategorie erstellen, löschen und Änderungen vornehmen.

3.2.5.12 Benutzer verwalten (UC12)

Der Administrator kann Benutzerinformationen eines Benutzerkontos löschen, bearbeiten oder anschauen. Der Benutzer kann nur seine eigenen Benutzerdaten bearbeiten.

3.2.5.13 Bestimmte Gruppen Reservationsberechtigung pro Pod definieren (UC13)

Der Administrator definiert, welche Gruppen auf welche Pods zugreifen dürfen.

3.2.5.14 LDAP Gruppe CRUD (UC14)

Der Administrator verwaltet die LDAP Gruppen, d.h. er kann Gruppen hinzufügen, bearbeiten, anschauen und löschen.

3.2.6 Use Cases („nice-to-have“ UCs)

3.2.6.1 Lab CRUD (UC15)

Bevor im Kurs mit einem Lab begonnen werden kann, müssen meist repetitive Konfigurationen auf den Geräten getätigt werden, dies erübrigt sich nun mit den Initial-Konfigurationen. Der Administrator kann Initial- oder End-Konfigurationen erstellen, löschen und aktualisieren. Dabei definiert er, für welchen Pod dieses Lab ist und welche Konfigurationen auf welches Gerät gehören. Bei der Reservation, kann der Be-

nutzer auswählen, ob er eine bereits vorkonfigurierte Initial-Konfiguration oder eine alte Konfiguration des Labs auf den Geräten vorfinden möchte.

3.2.6.2 Konfiguration für ein Gerät herauf laden (UC16)

Der Benutzer kann eine Konfiguration eines Gerätes via Control Panel auf den Server herauf laden. Dabei muss definiert werden, für welches Lab und für welches Gerät diese Konfiguration ist.

3.2.6.3 Gespeicherte Konfiguration für Geräte oder Pod's herunter laden (UC17)

Benutzer lädt seine gespeicherte Konfiguration für ein Gerät oder einen Pod via Control Panel vom Server herunter.

3.2.6.4 Konfiguration des Pods speichern (UC18)

Sobald eine Reservation zu Ende ist oder der Benutzer das aktuelle Lab speichern möchte, werden vom Control Panel alle Konfigurationen gespeichert, sodass der Benutzer später an seinem Lab weiter arbeiten kann.

3.2.6.5 Bestehende Konfiguration des Pods einspielen (UC19)

Initial- oder alte Lab Konfigurationen des Administrators, sowie gespeicherte Konfigurationen vom Benutzer selber, können vom Benutzer über das Control Panel beim reservierten Pod auf die jeweiligen Geräte wieder eingespielt werden.

3.2.6.6 Sprache ändern (UC 20)

Benutzer kann sich das Control Panel in den Sprachen Deutsch oder Englisch anzeigen lassen.

3.3 Nichtfunktionale Anforderungen

3.3.1 Implementierung

Das Remote Lab Management System muss MySQL als Applikationsinterne Datenbank verwenden. Die Server Applikation des Remote Lab Management Systems muss in PHP geschrieben sein. Die Kommunikation zwischen RADIUS und Control Panel muss über den Methodenaufruf XML-RPC geschehen.

3.3.2 Zuverlässigkeit

Nur authentifizierte Benutzer mit genügend Credit Points dürfen Reservierungen machen. Zwischen zwei Reservationen muss eine vom Administrator festgelegte Zeit (ca. 30 min.) abgewartet werden (damit Konfigurationen in dieser Zeit gespeichert werden können).

3.3.3 Erweiterbarkeit

Konfigurationen müssen auf dem Server gespeichert, eingespielt, herauf bzw. herunter geladen werden können. Es muss die Möglichkeit bestehen, ein Credit-System an das Control Panel anzubinden. Als Erweiterung könnten mehrere Administratoren zur gleichen Zeit auf den gleichen Zeitpunkt Reservationen tätigen (zurzeit FIFO).

3.3.4 Lizenzanforderungen

Lizenzen werden keine benötigt.

Domainanalyse

Version 1.0

Projekt:

Remote Lab Management System

Projektmitglieder:

Markus Kolb

Mirjam Schaffner

Betreuer:

Rolf Schärer

Beat Stettler

Revision				
Version	Status	Datum	Beschreibung/Änderung	Autor
1.0rc01	In Bearbeitung	17.03.2010	Erstellen des Domainanalyse Dokuments	MK
1.0rc02	In Bearbeitung	18.03.2010	Domain Model erstellt	MK
1.0rc03	In Bearbeitung	19.03.2010	System Sequence Diagram ergänzt, Domain Model angepasst	MS
1.0rc04	In Bearbeitung	23.03.2010	System Sequenzdiagramme erstellt und Domain Model angepasst, weitere Ergänzungen gemacht, System Contracts erstellt	MS
1.0rc05	In Bearbeitung	24.03.2010	Konzeptbeschreibung	MK
1.0rc06	Gegenlesen	24.03.2010	Kleine Änderungen	MS
1.0rc07	Gegenlesen	24.03.2010	Ganzes Dokument gegengelesen	MK
1.0rc08	In Bearbeitung	25.03.2010	Dokument anpassen gemäss Sitzung	MS
1.0rc09	Gegenlesen	30.03.2010	Ganzes Dokument gegengelesen	MK
1.0rc10	In Bearbeitung	11.06.2010	Domain Model aktualisiert und Konzeptbeschreibung überarbeitet	MS
1.0rc11	Gegenlesen	17.06.2010	Review	MK
1.0	Close	17.06.2010	Version 1.0	MK

Inhaltsverzeichnis

1	Einführung	4
1.1	Zweck	4
1.2	Gültigkeitsbereich	4
1.3	Definitionen und Abkürzungen.....	4
1.4	Referenzen	4
2	Remote Lab Management	5
2.1	Domain Model.....	5
2.1.1	Strukturdiagramm.....	5
2.1.2	Konzeptbeschreibung	6
2.2	System Sequenzdiagramme	8
2.2.1	Use Case 01: Benutzer anmelden	9
2.2.2	Use Case 02: Reservation erstellen	10
2.2.3	Reservationsüberprüfung durch Radius.....	11
2.3	Systemoperationen.....	12

1 Einführung

1.1 Zweck

In diesem Dokument wird die Problem Domain der Bachelorarbeit analysiert.

1.2 Gültigkeitsbereich

Die Gültigkeit dieses Dokuments gilt für die gesamte Projektdauer.

1.3 Definitionen und Abkürzungen

Das Glossar ist im separaten Dokument (Glossar.docx) zu finden.

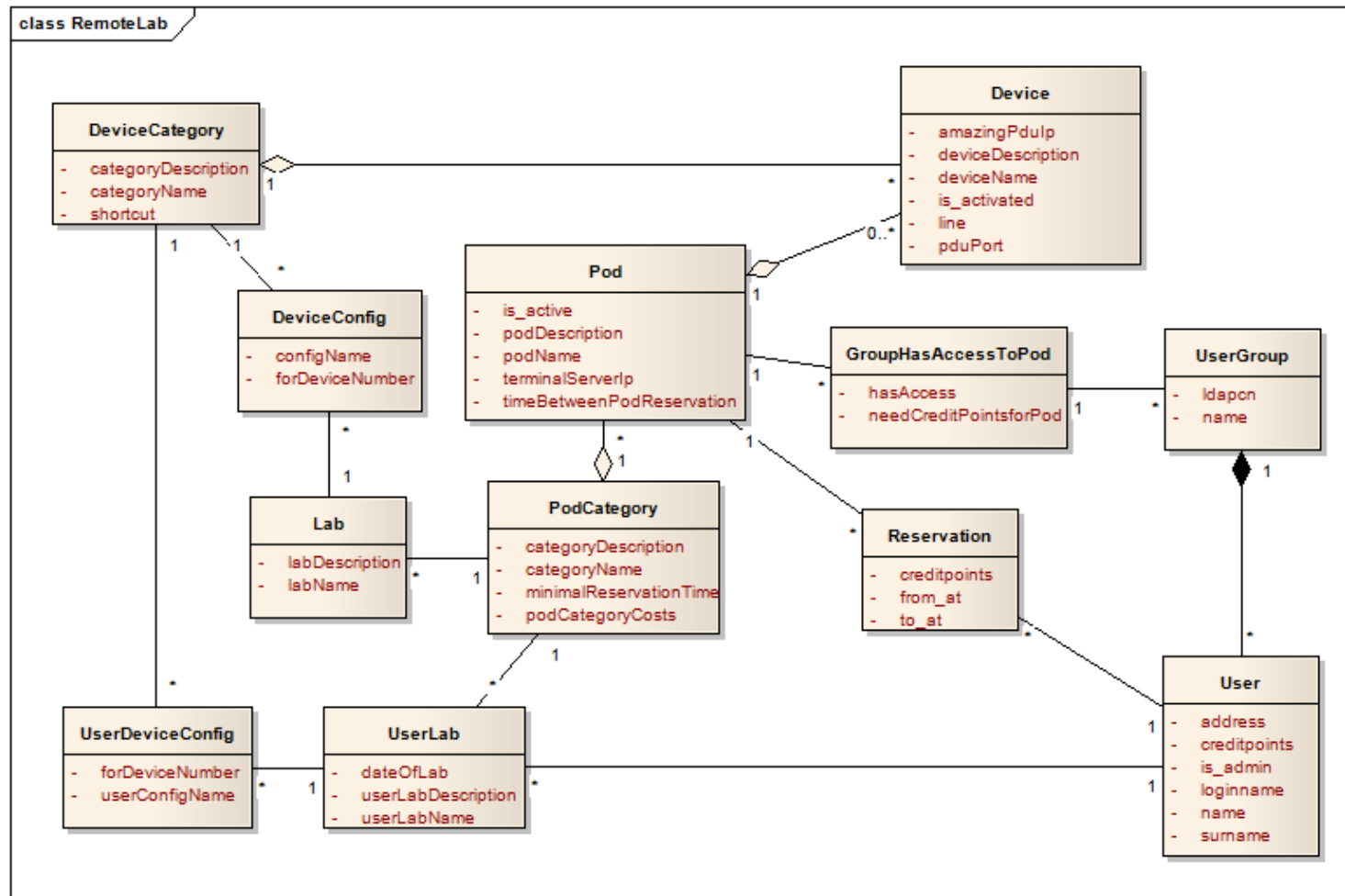
1.4 Referenzen

Referenz	Quelle
UML 2 und Patterns angewendet – Objektorientierte Softwareentwicklung	Craig Larman Mitp-Verlag/Bonn

2 Remote Lab Management

2.1 Domain Model

2.1.1 Strukturdiagramm



2.1.2 Konzeptbeschreibung

Zu beachten ist, dass selbsterklärende Attribute und Beziehungen nicht beschrieben werden.

1 Pod	
Beschreibung	Ein Pod beinhaltet mehrere Geräte wie Router, Switch, Server und Firewalls.
Attribute	is_active Der Administrator kann den Pod aktiv schalten, sodass die Benutzer Reservationen auf den Pod tätigen können. Bei einem Defekt kann so ein Pod einfach deaktiviert werden.
	timeBetweenPod-Reservation Administrator legt die Zeit fest, in der keine Reservation getätigt werden können. Während dieser Zeit werden die Konfigurationen von den Geräten geladen.
	terminalServerIp Ip des Terminalservers, über welchen der Pod erreichbar ist.
Beziehungen	An den oben aufgeführten Beziehungen ist vor allem die zwischen Device und Pod interessant. Das Control Panel braucht Informationen von den Geräten, um die Geräte neu starten zu können. Zudem sind für die Zugriffsberechtigungen die Informationen über Port und IP Adresse nötig. (siehe 2.2.3)

2 PodCategory	
Beschreibung	Jeder Pod ist einer Kategorie zugeordnet.
Attribute	minimal-ReservationTime Der Administrator kann eine Minimalreservationszeit definieren.
	podCosts Es wird definiert, wie viele Credit Points für die entsprechende Pod Kategorie pro Stunde benötigt werden.
Beziehungen	Die Beziehung zwischen PodCategory und UserLab/Lab ist trivial, jedes Lab muss einer Pod Kategorie zugewiesen werden. So wird verhindert, dass die Konfigurationen auf eine falsche Podkategorie gespielt werden können. Wieso Lab und UserLab getrennt werden, ist in der Tabelle 9 Lab / UserLab ersichtlich.

3 User	
Beschreibung	Die meisten User Informationen befinden sich in der LDAP Datenbank. Das Control Panel hat keine Schreibberechtigungen auf der LDAP Datenbank.
Attribute	creditpoints In der Userklasse ist definiert, wie viele Credit Points der Benutzer zurzeit zur Verfügung hat.
	is_admin Dieses Attribut kennzeichnet, ob ein User ein Administrator ist oder nicht.

Beziehungen	Jeder User gehört zu einer Gruppe.
--------------------	------------------------------------

4 UserGroup

Beschreibung	Die UserGroup ist eine Gruppe der LDAP Datenbank.
Attribute	ldapcn Common Name der Ldap Gruppe
Beziehungen	Die UserGroup hätte eine n zu n Beziehung zu Pod, was durch die Klasse GroupHasAccessToPod verhindert wird.

5 GroupHasAccessToPod

Beschreibung	Hier wird definiert welche Gruppe auf welchen Pod Zugriff erhältet.
Attribute	needCredit-PointsForPod Mit diesem Flag kann definiert werden, ob diese Gruppe Credit Points benötigt oder nicht.
Beziehungen	Jede GroupHasAccessToPod ist einem Pod zugewiesen.

6 Reservation

Beschreibung	Bei einer Reservation handelt sich um eine Reservation eines Pods.
Attribute	from_at Reservation von
	to_at Reservation bis
	creditpoints Kosten der Reservation
Beziehungen	Jede Reservation gehört zu einem User und zu einem Pod.

7 Device

Beschreibung	Ein Device Eintrag repräsentiert ein Switch, Router, Firewall oder ein Server.
Attribute	is_activated Um ein Device einem Pod hinzuzufügen, muss dieses Flag auf true sein. So wird verhindert, dass plötzlich ein kaputtes Device im produktiven Bereich aktiv geschaltet werden kann.
	line Interne Line, die gewählt wird.
	amazingPdulp Ip der Stromschiene.
	pduPort Steckplatz der Stromschiene (A-F).
Beziehungen	Jedes Device ist einem Pod zugewiesen.

8 DeviceCategory

Beschreibung	Eine DeviceCategory ist eine exakte Beschreibung einer Geräte Kategorie. (Bsp. Cisco Catalyst 3550 Series Switches)
---------------------	---

Attribute	CategoryName	Name der Kategorie
	shortcut	Definierte Abkürzung für das jeweilige Gerät.
Beziehungen	Eine DeviceCategory hat mehrere Geräte und mehrere Gerätkonfigurationen.	

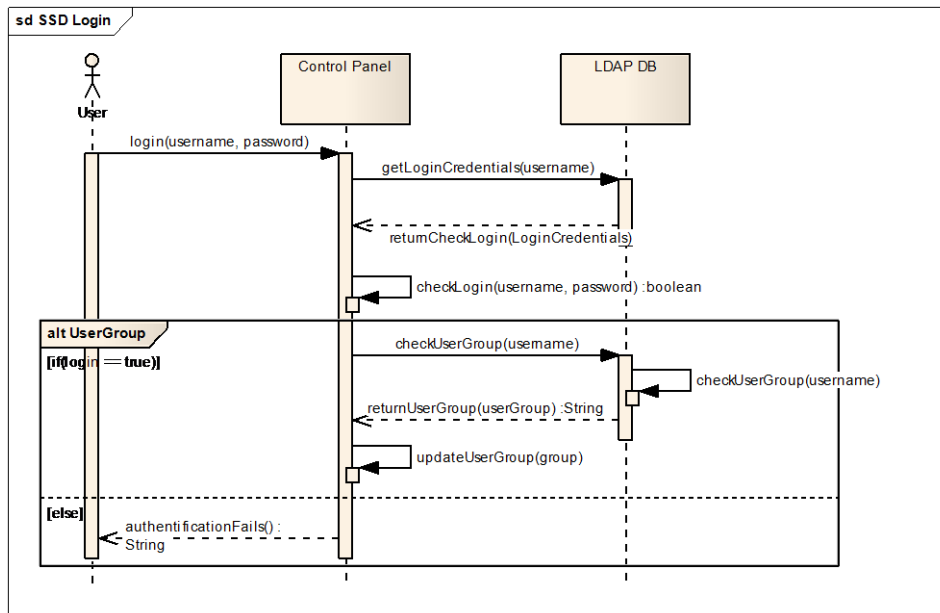
9 Lab / UserLab		
Beschreibung	Ein Lab ist ein vom Administrator definierte Initial- oder Endkonfiguration, welche von jedem User eingespielt werden kann. Ein UserLab ist beispielsweise eine Konfiguration, die automatisch gespeichert wurde, nachdem eine Reservation abgelaufen ist. Jeder User sieht seine eigenen UserLabs und kann nur auf seine eigenen UserLabs zugreifen.	
Attribute	labName userLabName	Ist die Namensgebung, kann vom Benutzer / Administrator selber ausgewählt werden.
	dateOfLab	Beschreibt, wann die Konfigurationen gespeichert wurden.
Beziehungen	Jedes Lab gehört zu einer Podkategorie und beinhaltet verschiedene Konfigurationen der Geräte.	

10 DeviceConfig / UserDeviceConfig		
Beschreibung	Eine DeviceConfig / UserDeviceConfig ist eine Konfiguration eines Gerätes.	
Attribute	forDeviceNumber	Mit der forDeviceNumber wird identifiziert, welche Konfiguration auf welches Gerät kommt.
Beziehungen	Jede Konfiguration gehört zu einem Lab und kann auf eine bestimmte DeviceCategory eingespielt werden.	

2.2 System Sequenzdiagramme

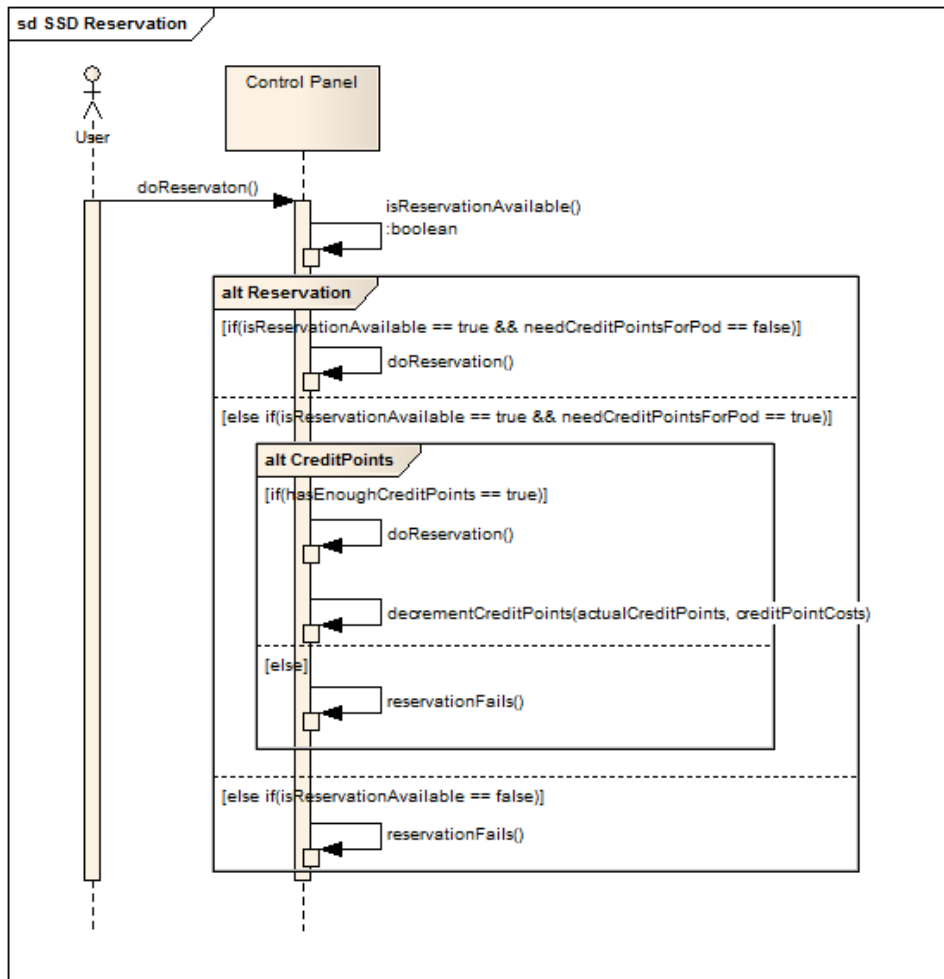
In den folgenden System Sequenzdiagrammen wurden die internen Abläufe, welche das Verständnis besser fördern, dargestellt.

2.2.1 Use Case 01: Benutzer anmelden



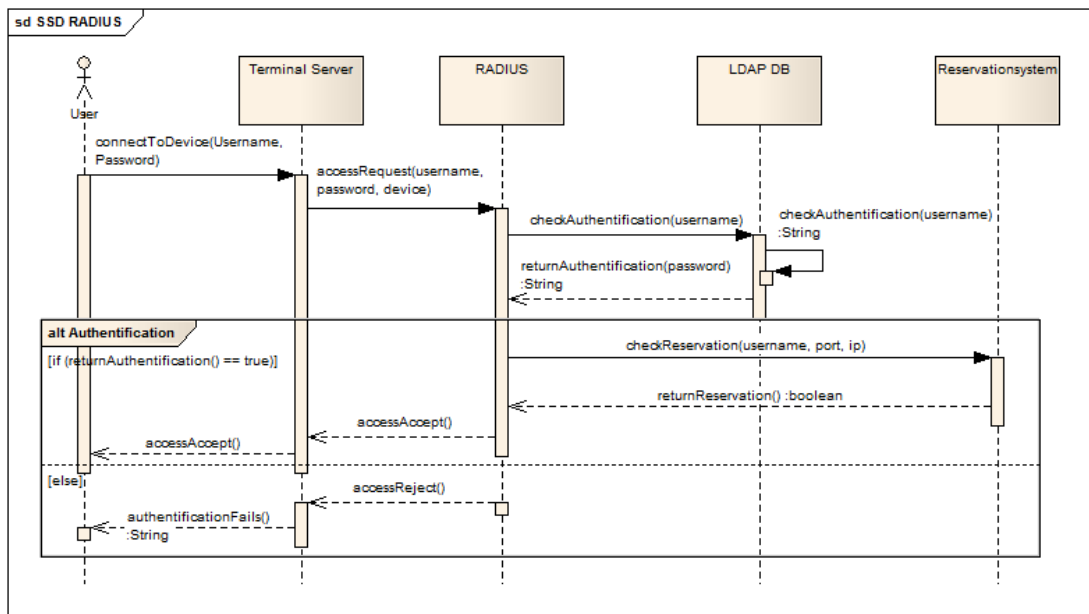
Der Benutzer meldet sich über das Control Panel an. Die Login Credentials werden von der LDAP-Datenbank abgerufen und durch das Control Panel überprüft. Sofern das Login richtig ist, wird bei der LDAP Datenbank nachgefragt, in welcher Gruppe sich der Benutzer befindet. Die LDAP Datenbank gibt dem Control Panel den Gruppennamen zurück, in der sich der Benutzer befindet. Falls der Benutzername oder das Passwort falsch eingegeben wurde, wird eine entsprechende Fehlermeldung zurück gegeben.

2.2.2 Use Case 02: Reservation erstellen



Über das Control Panel kann der Benutzer eine Reservation erstellen. Das Control Panel überprüft, ob eine Reservation des Pods zum gewünschten Zeitpunkt möglich ist und ob der Benutzer Credit Points für den Pod benötigt. Wenn der Benutzer keine Credit Points für den Pod benötigt, kann er die Reservation machen. Ist eine Reservation zum gewünschten Zeitpunkt möglich und der Benutzer braucht Credit Points, um eine Reservation zu machen, muss überprüft werden, ob er über genügend Credit Points verfügt um eine Reservation auf den gewünschten Pod zu machen. Hat der Benutzer genug Credit Points, kann er die Reservation machen. Falls der Benutzer zu wenig Credit Points hat, gibt das System eine entsprechende Fehlermeldung zurück, dass der Benutzer zu wenig Credit Points hat. Falls die gewünschte Reservationszeit nicht zur Verfügung steht, gibt das System ebenfalls eine entsprechende Fehlermeldung zurück.

2.2.3 Reservationsüberprüfung durch Radius



Wenn ein Benutzer auf einen Pod zugreifen will, muss er sich beim Terminal Server via SSH einloggen. Dabei überprüft RADIUS mit Hilfe der LDAP-Datenbank, ob der Benutzername und das Passwort stimmen. Sofern diese Angaben übereinstimmen, fragt RADIUS bei der MySQL Datenbank des Controlpanels nach, ob eine Reservation bestehe. Falls eine Reservation besteht, kann der Benutzer auf den Pod zugreifen, sonst gibt das System eine accessReject Meldung an den Benutzer zurück.

2.3 Systemoperationen

Folgend werden die wichtigsten Systemoperationen erklärt.

Contract C01: login	
Operation	login(username: String, password: String)
Cross Reference	UC01: Benutzer an- und abmelden
Preconditions	<ul style="list-style-type: none">• Benutzer ist bereits registriert.• Benutzer ist einer Gruppe zugewiesen.
Postconditions	<ul style="list-style-type: none">• Benutzer wurde am System angemeldet.

Contract C02: doReservation	
Operation	doReservation (username: String, password: String)
Cross Reference	UC02: Reservation CRUD
Preconditions	<ul style="list-style-type: none">• Benutzer ist angemeldet.• Benutzer gehört einer Gruppe an.• Nicht aktuelle Kursteilnehmer verfügen über genügend Credit Points.
Postconditions	<ul style="list-style-type: none">• Eine Reservation für diesen Benutzer wurde erstellt.• Die benötigten Credit Points für diese Reservation wurden abgezogen, sofern der Benutzer Credit Points für die Reservation gebraucht hat.

Software Architecture Document

Version 1.0

Projekt:

Remote Lab Management System

Projektmitglieder:

Markus Kolb
Mirjam Schaffner

Betreuer:

Rolf Schärer
Beat Stettler

Revision				
Version	Status	Datum	Beschreibung/Änderung	Autor
1.0rc01	In Bearbeitung	17.03.2010	Erstellen des SAD	MK
1.0rc02	In Bearbeitung	23.03.2010	Systemübersicht, Logical View, Deployment View geschrieben und hinzugefügt	MK
1.0rc03	In Bearbeitung	23.03.2010	Process View und SAD gegenseitig	MS
1.0rc04	In Bearbeitung	25.03.2010	SAD gemäss Sitzung geändert	MS
1.0rc05	Gegenlesen	30.03.2010	Gegenlesen	MK
1.0rc06	In Bearbeitung	4.05.2010	Design erstellen	MS
1.0rc07	In Bearbeitung	07.06.2010	Reservationssystem SAD erfassen	MK
1.0rc08	Gegenlesen	08.06.2010	Gegenlesen und überarbeiten	MS
1.0rc09	In Bearbeitung	08.06.2010	Datenbank Design ergänzt und LDAP Kapitel verfasst.	MK
1.0rc10	In Bearbeitung	09.06.2010	Implementierung CRUD Objekte (Database)	MK
1.0rc11	In Bearbeitung	09.06.2010	Legende Geräte erfasst, Dokument gegenseitig gelesen und überarbeitet	MS
1.0rc12	Gegenlesen	09.06.2010	Review der Dokumente	MK
1.0rc13	In Bearbeitung	15.06.2010	Erweiterung Systemübersicht und Klassendiagramm erstellt	MK
1.0rc14	Gegenlesen	16.06.2010	Gegenlesen, Process Views erstellt, Lokalisierung erweitert, Kontaktformulare, JSON ergänzt	MS
1.0	Close	17.06.2010	Version 1.0	MS

Inhaltsverzeichnis

1	Einführung	5
1.1	Zweck	5
1.2	Gültigkeitsbereich	5
1.3	Definitionen und Abkürzungen	5
1.4	Referenzen	5
2	Übersicht	6
2.1	Systemübersicht	6
2.1.1	Beschreibung	7
3	Logical View	8
3.1	Klassendiagramm	8
3.1.1	Reservation: Implementierung der Methode Save	9
3.2	Sequenzdiagramme	9
4	Physical View	10
4.1	XML-RPC	11
5	Process View	12
5.1	Process View Login	12
5.2	Process View Admin	12
5.3	Process View Reservation	13
6	Design	14
6.1	Symfony	14
6.1.1	Pattern	14
6.1.2	Design des Control Panel	15
6.1.3	Datenbank Design	16
6.1.4	Routing	18
6.1.5	Berechtigungen	18
6.1.6	Lokalisierung	19
6.1.7	Internationalisierung	19
6.1.8	Kontakt Formular	19
6.2	jQuery	21
6.3	Reservationskalender	21
6.3.1	Pod Reservation Calendar (pod-reservation-calendar.js)	22
6.3.2	Action index()	23
6.3.3	Action Jsondata()	23
6.3.4	Action Saveevent()	24
6.3.5	Action Deleteevent()	26
6.3.6	Performance	27
6.3.7	Licence	27
6.4	MyCalendar	28
6.4.1	Abweichende Konfigurationen zum Reservationskalender	28
6.4.2	Ical Informationen abrufen	29
6.5	Authentifizierung und Anbindung an LDAP Server	29
6.5.1	Konfigurationen LDAP Server	30
6.5.2	Form	30
6.5.3	Error Handling	30
6.5.4	LDAP Gruppen	31
6.6	Active Reservation Modul	32

6.7	Datatables CRUD Objekte	33
6.7.1	Datatables (dataTables-*.js)	33
6.7.2	Action Index()	34
6.7.3	Action Json()	34
6.7.4	Action Save()	34
6.7.5	Action Delete()	35
6.7.6	Performance	35
6.7.7	Licence	35
6.8	XML-RPC	36
6.8.1	Client Script	36
6.8.2	Server Script	36
6.9	Plugins & Versionen	36

1 Einführung

1.1 Zweck

Zu Beginn des Dokuments wird eine kurze Übersicht über das ganze System gegeben und die diversen architektonischen Entscheidungen & Konzepte erläutert.

Danach gliedert sich das Dokument ganz nach dem „N+1“ View Model gemäss Larman und ist in verschiedene Sichten unterteilt.

Es enthält ebenfalls die Informationen über das Design der Applikation und alle Designentscheide des Projektes.

1.2 Gültigkeitsbereich

Die Gültigkeit dieses Dokuments gilt für die gesamte Projektdauer.

1.3 Definitionen und Abkürzungen

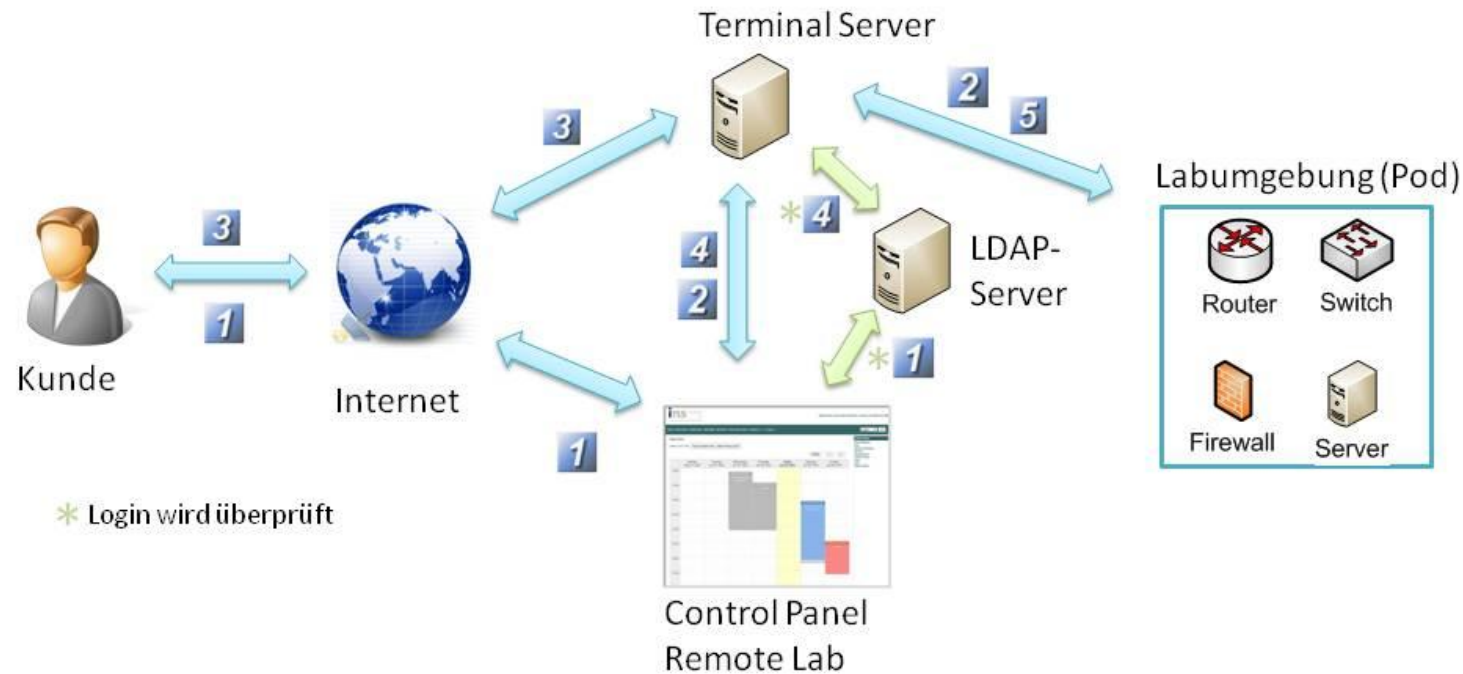
Das Glossar ist im separaten Dokument (Glossar.docx) zu finden.

1.4 Referenzen

Referenz	Quelle
UML 2 und Patterns angewendet – Objektorientierte Softwareentwicklung	Craig Larman Mitp-Verlag/Bonn
Logical View	http://www.symfony-project.org/jobeeet/1_4/Doctrine/en/04
Symfony MVC-Erklärung	http://trac.symfony-project.org/wiki/Documentation/de_DE/book/1.0/02-Exploring-Symfony-s-Code
XML-RPC Erklärung	http://de.wikipedia.org/wiki/XML-RPC
Orientation in Objects GmbH	http://www.oio.de/public/xml/xml-rpc.htm
Microsoft	http://technet.microsoft.com/dede/library/cc781821(WS.10).aspx
Ldap Bild Ressource	http://www.donotenter.com/cool/ucgraphics/large/index.htm
K. Rhodes, XML-RPC vs. SOAP	http://weblog.masukomi.org/writings/xml-rpc_vs_soap.htm

2 Übersicht

2.1 Systemübersicht



- 1** Ein Kursteilnehmer reserviert via Control Panel seine Labumgebung (Pod).
- 2** Vom Control Panel wird ein Script ausgelöst, welches die Konfigurationen via Terminal Server in die Labumgebung einspielt.
- 3** User verbindet via Terminal Server in die Labumgebung.
- 4** Via Radius wird überprüft, ob der User eine Reservation getätigt hat.
- 5** Zugriff wird gewährt

2.1.1 Beschreibung

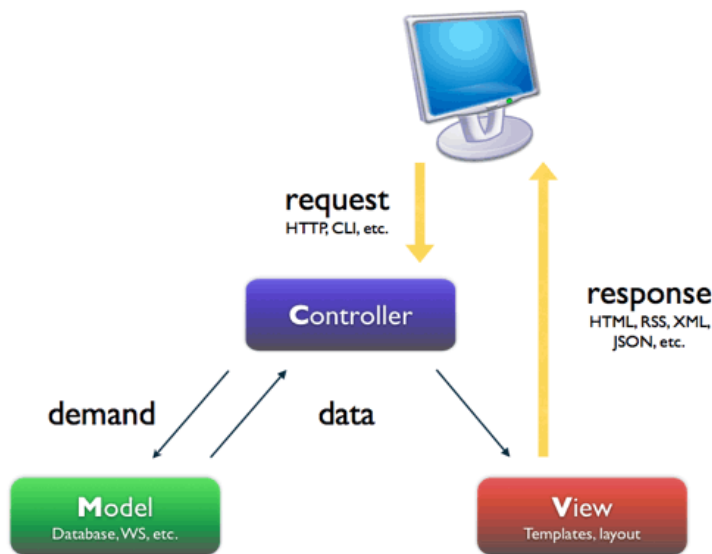
Bei Punkt (1) ist ersichtlich, dass ein Benutzer sich via Internet am Control Panel anmeldet. Während der Anmeldung am Control Panel, wird der Benutzername und das Passwort durch die Anbindung beim LDAP Server überprüft. Der Benutzer muss einer LDAP Gruppe angehören, welche im Control Panel erfasst ist. Anderenfalls wird die Anmeldung fehl schlagen. Im Control Panel kann nun eine Reservation erfasst werden, welche später ermöglicht auf den Pod zuzugreifen.

Bei Erstellung einer Reservation (2) wird zum Beispiel ein Cronjob erstellt, welcher automatisch die Start Konfigurationen in die Geräte einspielt.

(3) Während der reservierten Zeit, kann der Benutzer via SSH über den Terminal Server auf die Labumgebung zugreifen. Sobald eine Verbindung zum Terminalserver aufgebaut wird, (4) werden zuerst via LDAP Server die Logindaten überprüft. Falls diese in Ordnung sind, wird vom Terminalserver das XML-RPC Client Script aufgerufen, welches dem Server Script die IP Adresse, Benutzername und Port übergibt. Nun kann das Server Script mit diesen Informationen in der Datenbank des Control Panels (Reservationssystem) die Reservation überprüfen und dem Client Script den Zugriff bestätigen oder ablehnen.

(5) Sobald die Bestätigung des Client Scripts eingetroffen ist, gewährleistet der Terminalserver (freeRadius) den Zugriff auf das Gerät. Sobald die Reservationszeit vorüber ist, kann der Benutzer nicht mehr auf die Geräte zugreifen, trotzdem werden vom Terminalserver nicht automatisch die Verbindungen getrennt. Diese Funktion könnte nachträglich im Control Panel implementiert werden. Das Control Panel würde bei Erstellung einer Reservation einen Cronjob ausführen, welcher nach Ablauf der Reservation alle Verbindungen zu den Geräten trennt.

3 Logical View



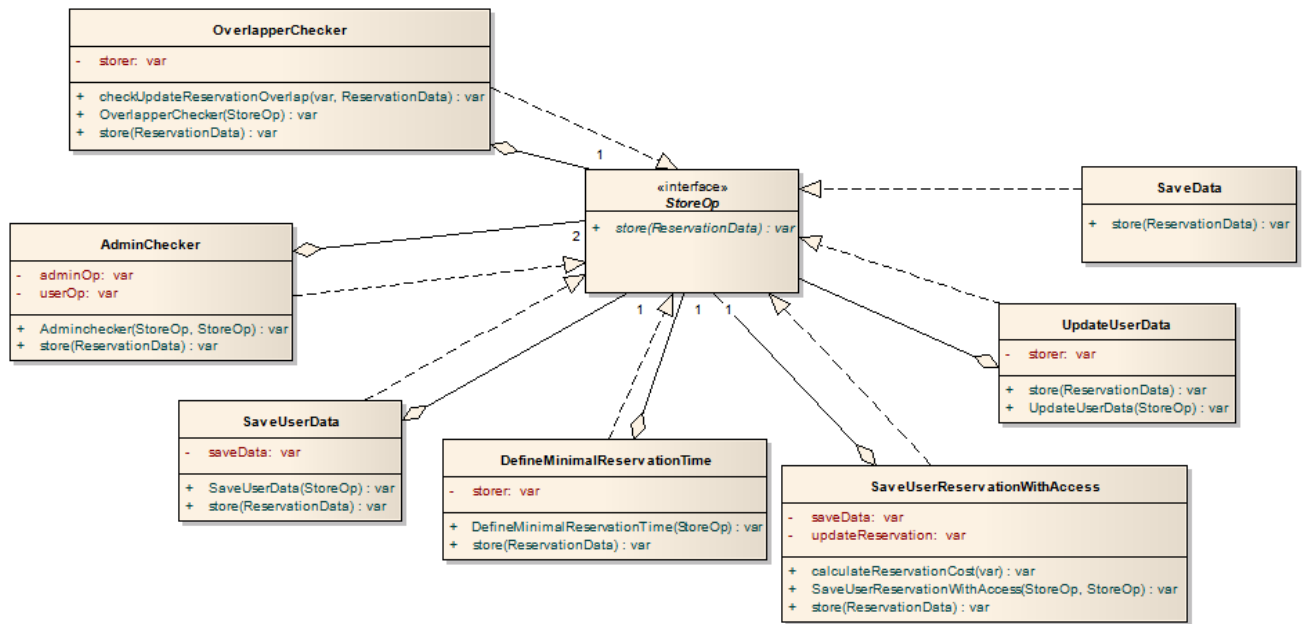
Das Control Panel, welches im Symfony geschrieben wird, basiert auf dem klassischen Webanwendungsdesign Model View Controller (MVC). Die MVC-Architektur besteht aus folgenden drei Ebenen:

- Model: Das Datenmodell enthält die Daten, welche das Control Panel präsentiert und arbeitet (Geschäftslogik).
- View: Die Präsentationsschicht ist für die Darstellung der Daten verantwortlich. Die View empfängt die Benutzerinteraktionen und ist zuständig diese weiterzuleiten, aber nicht zu verarbeiten.
- Controller: Der Controller (Steuerung) verwaltet eine oder mehrere Views und nimmt von Ihnen Benutzerinteraktionen entgegen und ist schlussendlich verantwortlich, die gewünschte Funktion zu vollziehen.

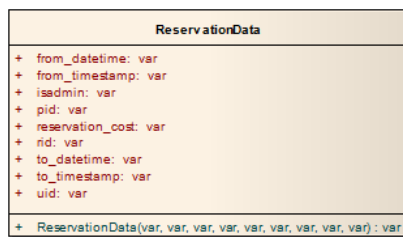
3.1 Klassendiagramm

Grundsätzlich ist ein Klassendiagramm bei Verwendung eines Frameworks nicht nötig. Trotzdem wurde die Methode Save in der Reservationsklasse kompliziert und wir haben uns somit entschieden, das Decorator Pattern zu implementieren. Dies fördert das Verständnis und die Übersichtlichkeit. Funktionalitäten können leicht hinzugefügt oder erweitert werden.

3.1.1 Reservation: Implementierung der Methode Save



Bei der Implementierung von der Save Methode in der Reservationsaction Klasse wurde das Design Pattern Decorator verwendet. Speziell an der Implementierung ist, dass die AdminChecker Klasse zwei StoreOp Objekt Referenzen enthält. Die AdminChecker Klasse entscheidet, ob ein Administrator oder ein Benutzer eine Reservation abspeichern möchte. Beim Benutzer müssen mehrere Funktionen durchlaufen werden.

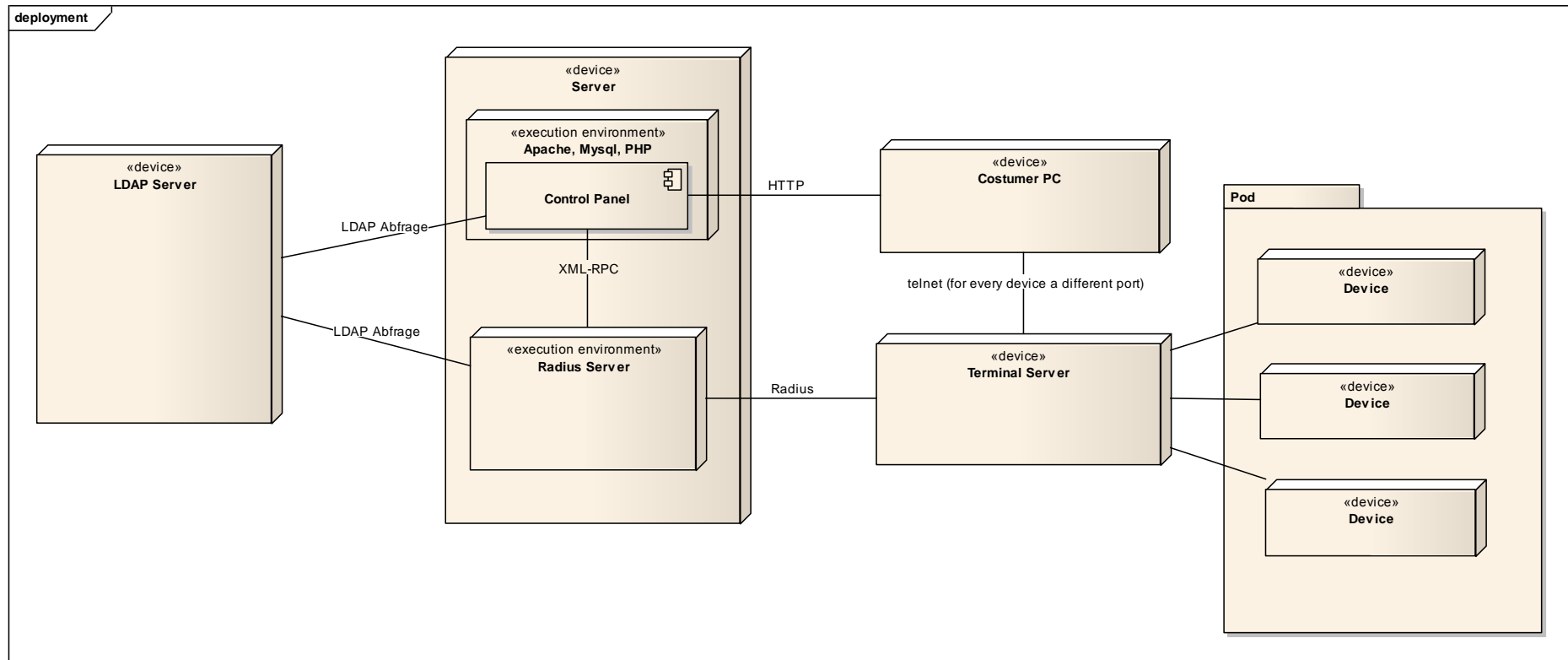


Bei einem Save Request, wird jeweils ein ReservationData Objekt erstellt, bei welchem alle Informationen gespeichert werden.

3.2 Sequenzdiagramme

Die Sequenzdiagramme sind im Dokument "Domain Analyse" unter Punkt 2.2 zu finden.

4 Physical View



4.1 XML-RPC

XML-RPC steht für Extensible Markup Language Remote Procedure Call. Es ist ein entfernter Funktionsaufruf über das Internet. XML-RPC ist sehr interessant für Mobile Web Services, da der Speicherplatz einer XML-RPC Implementierung auf wenige Kilobyte beschränkt werden kann.

Für den Transport der Daten wurde auf das HTTP Protokoll zurückgegriffen, während die Darstellung der übertragenen Daten in XML geschieht.

Ein Aufruf wird per HTTP-POST an den Server geschickt. Dieser wertet das im Body enthaltene Dokument aus und verwendet dessen Inhalt als Parameter für den Aufruf der gewünschten Funktion. Das Ergebnis wird wieder in XML gepackt und zum Client übertragen. Neben primitiven Datentypen werden auch komplexe Typen unterstützt, die aus den primitiven Typen zusammengesetzt werden können:

Typname in XML-RPC	Beschreibung
int, i4	Integer (Datentyp)
double	Gleitkommazahl
boolean	Boolesche Variable
string	Zeichenkette
dateTime.iso8601	Datum und Uhrzeit im ISO-Format
base 64	Base64-kodierte binäre Daten

Null-Werte sind mit XML-RPC nicht möglich. Methoden müssen daher immer einen konkreten Wert zurückgeben.

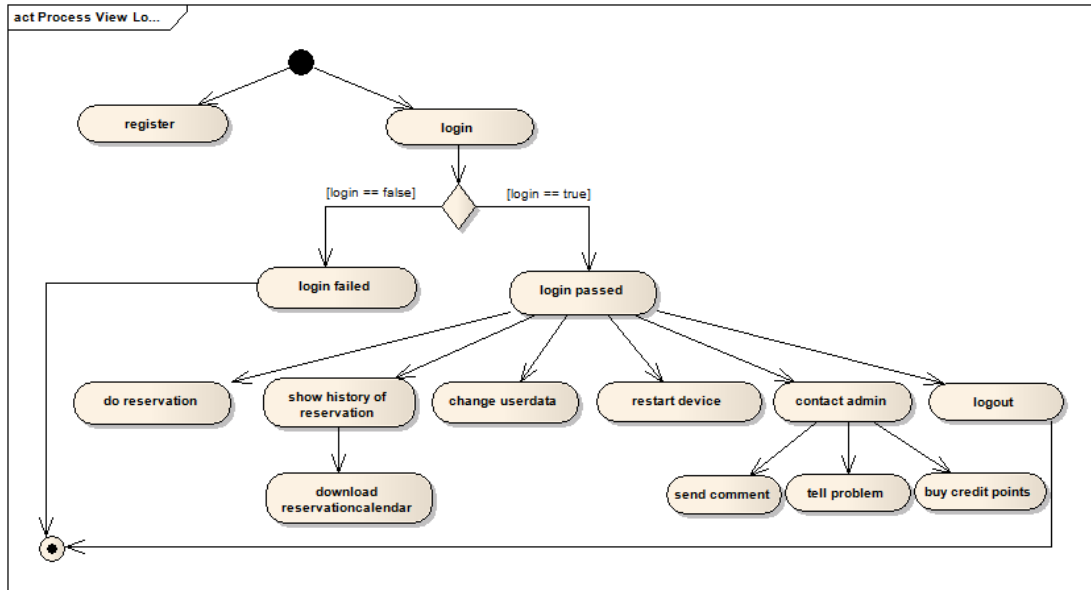
Für die Implementierung des XML-RPC haben wir die „kd_xmlrpc.php“ Bibliothek verwendet. Per Radius wird das XML-RPC Protokoll angefragt, ob der Benutzer am Terminal Server zurzeit eine Reservation auf den entsprechenden Pod hat. XML-RPC fragt das Reservationssystem an und gibt dem Radius als Antwort ein „ja“ oder „nein“ zurück. Sofern Radius ein „ja“ als Antwort erhält, kann der Benutzer auf den Pod zugreifen.

5 Process View

Parallelitäten treten in unserem System nicht auf, da das HTTP Protokoll ein zustandsloses Protokoll ist und somit jeder Request unabhängig voneinander abgearbeitet wird. Sitzungsdaten werden mittels der Session-ID implementiert.

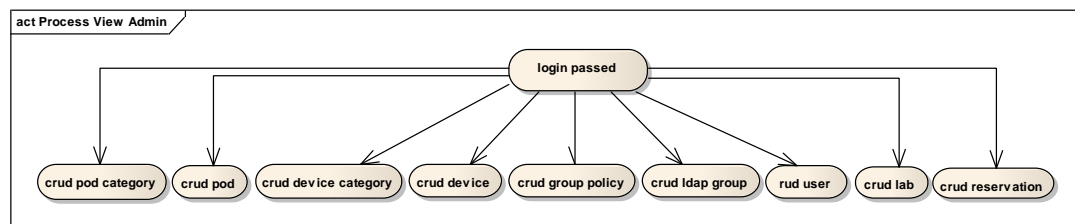
5.1 Process View Login

Ein neuer Kunde kann sich registrieren. Sobald ein Benutzer ein Benutzerkonto hat, kann er sich mit seinen Logindaten einloggen. Sofern das Login stimmt, kann der Benutzer eine Reservation machen, seine Reservationen ansehen und diese herunterladen, seine Benutzerdaten ändern, ein Gerät neu starten, dem Administrator durch Kontaktformulare Mail senden oder Credit Points kaufen oder sich wieder ausloggen.



5.2 Process View Admin

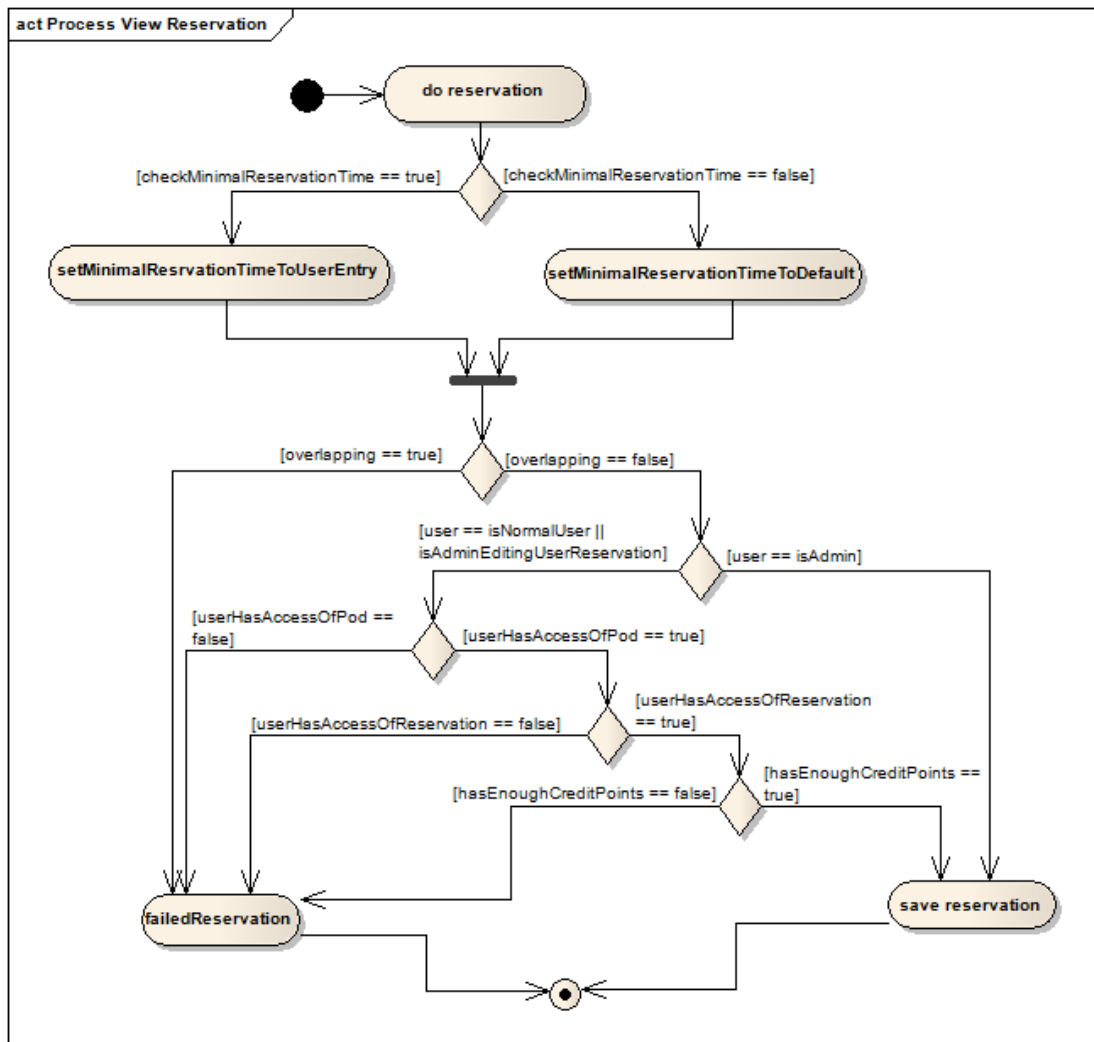
Sobald der Administrator eingeloggt ist, sieht er zusätzlich zu den Benutzer Modulen noch die CRUD Module wie Podkategorie, Pod, Gerätekategorie, Gerät, Gruppenrecht, LDAP Gruppe, Lab und Reservation für einen Benutzer, sowie das RUD Modul Benutzer.



5.3 Process View Reservation

Dieser Reservationsprozess bezieht sich auf den Reservationskalender und nicht auf das CRUD Modul Reservation!

Wenn ein Benutzer eine Reservation machen will, überprüft das System, ob die minimale Reservationszeit eingehalten wurde. Falls die minimale Reservationszeit nicht eingehalten wurde, setzt das System diese auf den Default Wert. Andernfalls setzt das System die minimale Reservationszeit auf den Wert, der vom Benutzer eingegeben wurde. Dann schaut das Control Panel, ob sich eine Reservation überlappt. Falls dies der Fall ist, gibt das System eine entsprechende Fehlermeldung aus, ansonsten wird geprüft, ob der Benutzer ein Administrator ist oder ein normaler Benutzer oder ein Administrator, der für einen Benutzer im Reservationskalender eine Reservation bearbeiten möchte. Falls der Benutzer ein Administrator ist, wird die Reservation gespeichert und sonst überprüft das System, ob der Benutzer das Recht besitzt, auf den Pod zuzugreifen. Wenn der Benutzer das Recht hat, wird geprüft, ob der Benutzer für die gewünschte Reservation genügend Credit Points besitzt. Falls dies auch der Fall ist, wird die Reservation gespeichert, ansonsten wird eine entsprechende Fehlermeldung ausgegeben.



6 Design

6.1 Symfony

Symfony ist ein Webapplikationsframework für PHP Projekte. Es gibt keine Packages an sich, sondern alles ist in Modulen angeordnet. Das besondere an Symfony ist, dass es viel Arbeit abnimmt, indem einiges generiert werden kann und dass es ein sehr mächtiges Framework ist.

controlpanel

config

web

apps

backend

config

i18n

lib

modules

auth

contactus

device

actions

templates

devicecategory

group_policy

home

lab

language

Im erstellten Symfony Projekt „controlpanel“ sind unter anderem die Unterordner „web“, „apps“, „lib“ und „test“ vorhanden.

Der Ordner „web“ ist wichtig für den Apache, um CSS Dateien, Bilder und JavaScript Dateien zu finden, die in diesem Ordner abgelegt sind.

Im Ordner „apps“ kann die Webapplikation als „Front“- oder „Backend“ unterteilt werden. Bei uns gibt es nur ein Backend, da es das Routing zwischen den verschiedenen Layouts enorm vereinfacht. Im Backend sind im „config“ Ordner die „routing.yml“ Datei sowie die „setting.yml“ Datei enthalten.

Im „i18n“ Ordner werden die für die Internationalisierung wichtigen Dateien abgelegt.

Unter dem Ordner „modules“ sind alle erstellten Module enthalten, die für die eigene Webapplikation gebraucht werden. Jedes Modul besitzt eine „action.class.php“ Klasse, in der alle Funktionen für die Aktionen implementiert sind. Weiter enthält jedes Modul Templates, in denen das Layout dieses Moduls definiert wird.

Das Layout für die ganze Applikation wird im Verzeichnis „apps/backend/templates“ definiert.

Im Verzeichnis „lib/model/doctrine“ befinden sich generierte Doctrine Dateien zu den jeweiligen Modulen. Diese werden für Abfragen auf die Datenbank benötigt.

Die Unit und Functional Tests sind im Ordner „test“ abgelegt.

6.1.1 Pattern

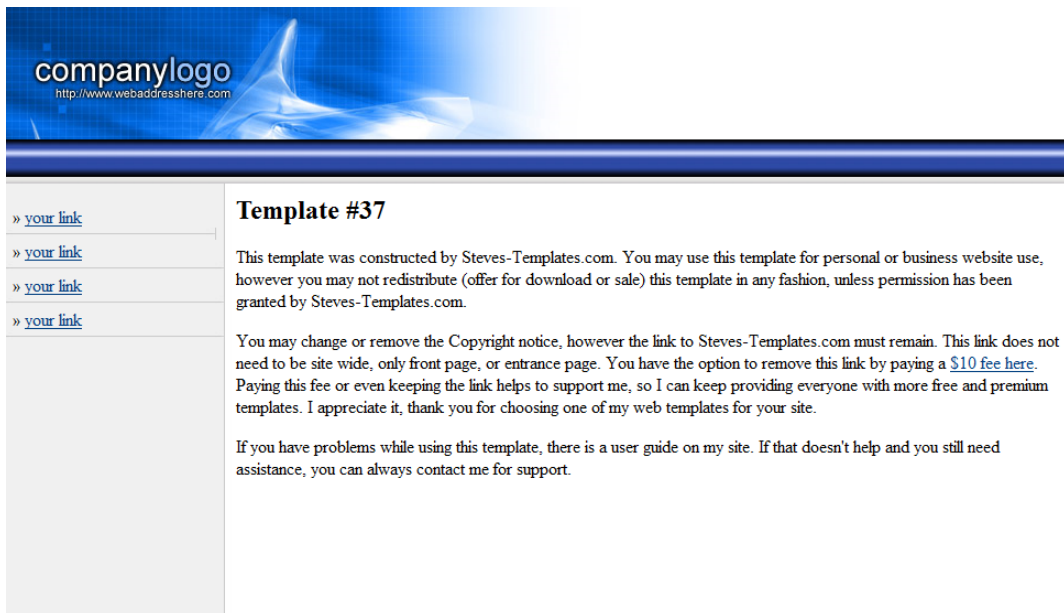
Beim Layout wird für den Content das Decorater-Pattern verwendet.



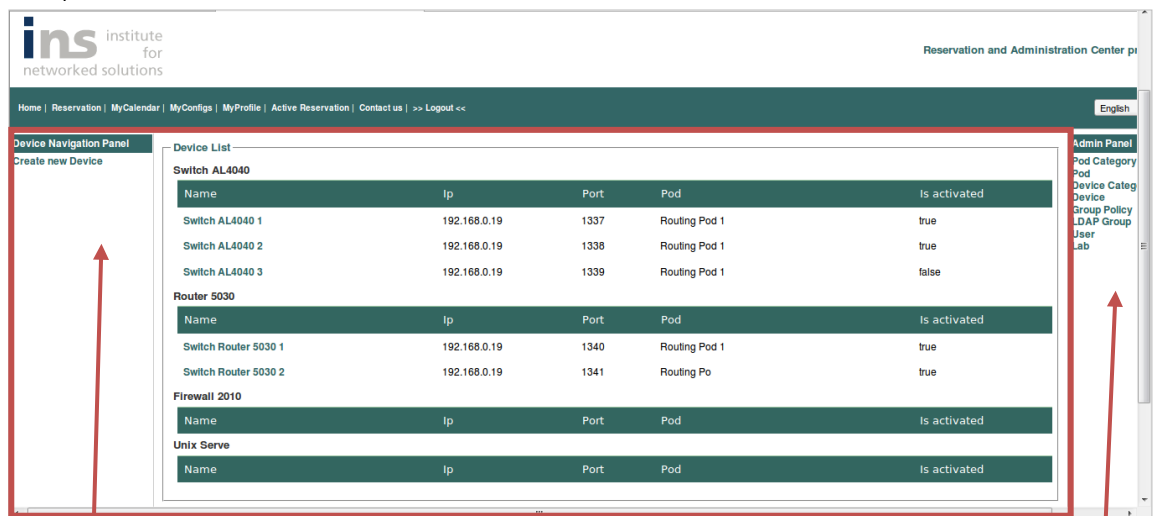
Bei Symfony ist das Routing ein Zwei-Weg-Prozess, d.h. die URLs können geändert werden, ohne dass die technische Implementierung geändert werden muss.

6.1.2 Design des Control Panel

Obwohl vom Betreuer nicht sehr viel Wert auf das Design gelegt wurde, wollten wir trotzdem etwas Anschauliches präsentieren.



Mit dem ersten Design waren wir nicht sehr zufrieden, daher entschieden wir uns, ein Template zu erweitern.

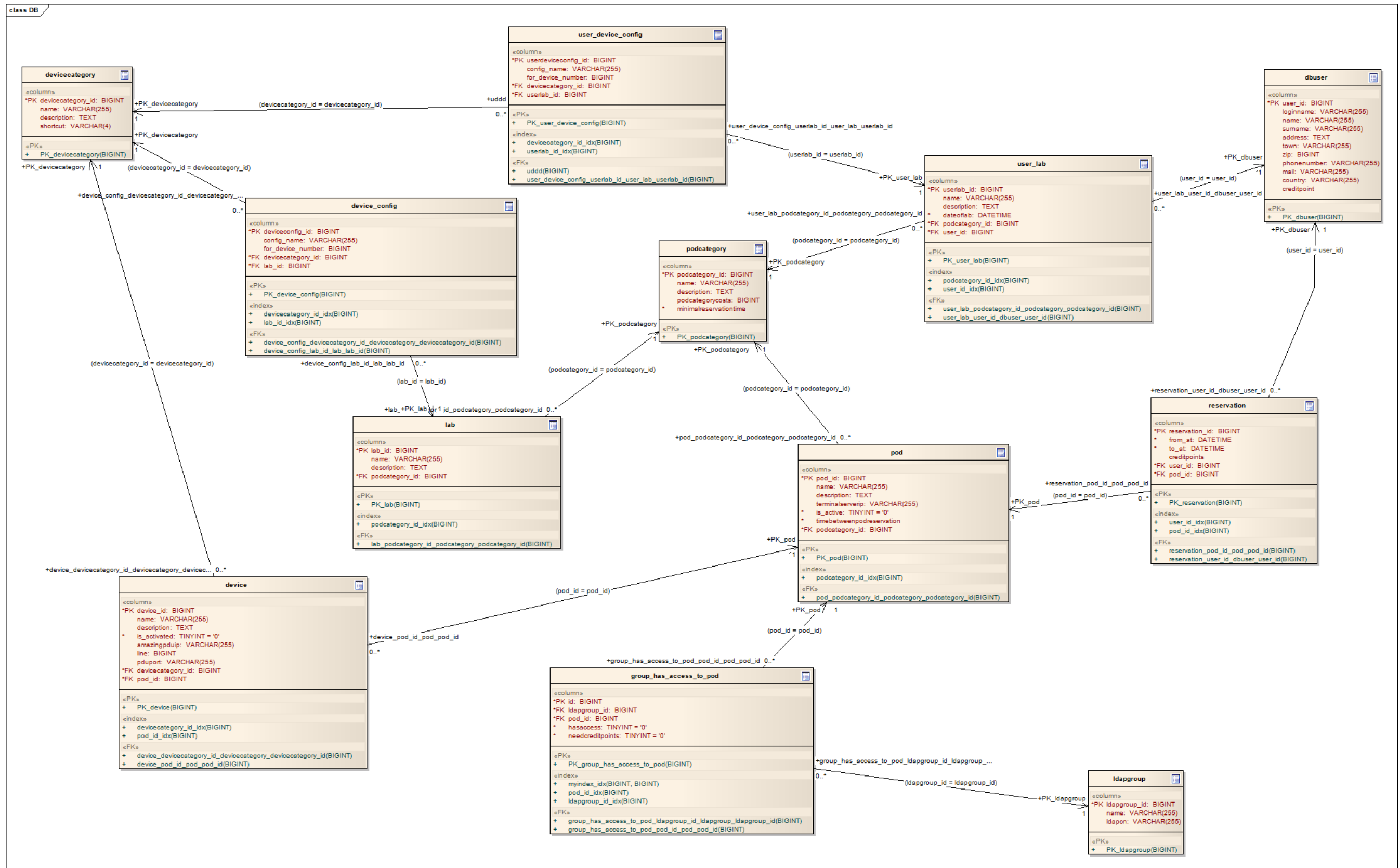


Navigation / Menü

Inhaltsanzeige pro Modul

Navigation für Administrator (CRUD Module)

6.1.3 Datenbank Design



Symfony ist ein objektorientiertes Framework. Die relationalen Datenbankinformationen müssen in ein Objekt Modell abgebildet werden. Dies kann entweder mit dem Propel ORM-Mapping gemacht werden oder mit dem Doctrine ORM-Mapping. In unserer Bachelorarbeit haben wir uns für Doctrine entschieden, da die Dokumentation von Doctrine besser ist als die bei Propel. Zudem wird Propel nur mässig weiterentwickelt. ORM generiert PHP Klassen, die Tabellen Einträge zu Objekten abbilden:

`$ php symfony doctrine:build --model`. Der `doctrine:build --model` Befehl generiert PHP Dateien im `lib/model/` Verzeichnis, die genutzt werden können, um mit der Datenbank zu interagieren.

Die Schema Datei befindet sich unter `config/schema.yml`.

Schema Beispiel

```
#config/schema.yml
Device:
  columns:
    device_id: { type: integer, notnull: true, primary: true, autoincrement: true }
    name:      { type: string(255) }
    description: { type: string(4000) }
    is_activated: { type: boolean, notnull: true, default: 0 }
    amazingpduip: { type: string(255) }
    line:      { type: integer }
    pduport:  { type: string(255) }
    devicecategory_id: { type: integer, notnull: true }
    pod_id:    { type: integer, notnull: true }
  relations:
    Devicecategory: { local: devicecategory_id, foreign: devicecategory_id, foreignAlias: Devices }
    Pod:            { local: pod_id, foreign: pod_id }
```

Um die Forms, SQL Tabellen und Validatoren für das Control Panel zu generieren, wird der Befehl `$php symfony doctrine:build --all --no-confirmation` verwendet. Danach können die Daten mit folgendem Befehl geladen werden: `$php symfony doctrine:data-load`.

Dass die Daten auch geladen werden, müssen im `data/fixture` Ordner Daten erfasst werden.

Fixtures

```
#data/fixtures/device.yml
device:
  device1:
    Devicecategory: deviceKat1
    Pod: pod1
    name: R1
    description: SSH line 2011
    is_activated: true
    amazingpduip: 10.1.1.8
    line: 33
    pduport: A
```

Je nach Schwierigkeit der SQL Abfrage wird jeweils die Funktion in die vorgesehene Datei transferiert, welche sich unter folgendem Ordner befindet: lib/model/doctrine

6.1.4 Routing

Das Routing verwaltet die internen URIs und die externen URLs. Wenn eine Anfrage ankommt, übersetzt das Routing die URL und konvertiert sie zu einer internen URI. Die Konvertierung zwischen internen URIs und externen URLs wird in der routing.yml Konfigurationsdatei gemacht. Die routing.yml Datei beschreibt die Route. Diese haben einen Namen (z.B. homepage), ein Pattern (z.B. /:module/:action/*) und Parameter (z.B. id).

Wenn eine Anfrage kommt, versucht das Routing eine Übereinstimmung zwischen einem Muster und der gegebenen URL zu finden. Die Reihenfolge in der routing.yml Datei ist sehr wichtig, da die erste Route genommen wird, die übereinstimmt.

Unser Routing sieht nach der Databales Implementierung wie folgt aus:

Routing Databales (Beispiel am Modul Device)

```
# apps/backend/config/routing.yml
device_delete:
  url: /:sf_culture/device/delete.:sf_format
  param: { module: device, action: delete, sf_format: html }
  requirements:
    sf_format: html
    sf_culture: (?:de|en)
    sf_method: [get]

device_js:
  url: /:sf_culture/device/json
  param: { module: device, action: jsondata }
  requirements:
    sf_culture: (?:de|en)

device_save:
  url: /:sf_culture/device/save
  param: { module: device, action: save }
  requirements:
    sf_culture: (?:de|en)

device:
  url: /:sf_culture/device.:sf_format
  param: { module: device, action: index, sf_format: html }
  requirements: { sf_method: get }
```

6.1.5 Berechtigungen

Der Benutzer darf nicht auf die CRUD Objekte zugreifen, dies wurde mit dem Symfony User gelöst. In den jeweiligen Modulen sind im Ordner config im File security.yml die Credentials definiert. Mit folgender Methode setzt man dem User die Credentials: `$this->getUser()->addCredential('admin');`

Apps/backend/modules/pod/config/security.yml

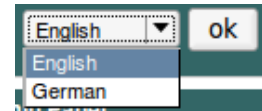
```
all:  
  credentials: admin
```

6.1.6 Lokalisierung

Bei der Lokalisierung geht es darum, dass sich die Applikation die Kultur des Benutzers merkt. D.h. wenn ein Benutzer das erste Mal auf die Website gelangt, wird die Website in der Standardsprache angezeigt z.B. Englisch. Der Benutzer wählt danach eine der Sprachen aus, die die Website anbietet und die ihm besser gefällt z.B. Deutsch. Die Applikation merkt sich die gewählte Sprache. Sobald der Benutzer ein weiteres Mal auf die Website kommt, wird ihm die beim letzten Mal gewählte Sprache z.B. Deutsch angezeigt.

Dazu muss das Routing angepasst werden, das heißt, es muss eine Localized Homepage erstellt werden und vor jede URL die Variable `/:sf_culture/` eingefügt werden.

Um die Combobox zu verwenden, muss das `sfFormExtraPlugin` installiert werden, da das Framework dies nicht unterstützt. Das Plugin hatte noch einen Bug, den wir durch Recherche im Internet beheben konnten. Um das `sfFormExtraPlugin` zu aktivieren, muss es in der `ProjectConfiguration.class.php` Klasse hinzugefügt und auf „enable“ gesetzt werden. Für die Combobox erstellten wir ein eigenes Module `Language`, damit die Action getrennt vom Layout ist, da wir die Action nicht jedes Mal aufrufen wollen, sondern nur, wenn sie gebraucht wird.



6.1.7 Internationalisierung

Symfony unterstützt für die Internationalisierung den ICU Standard `i18n`. Bei der Internationalisierung geht es darum, die Website in verschiedenen Sprachen darzustellen. In unserem Fall wäre Englisch die Standardsprache und Deutsch wird zusätzlich angeboten. Der Benutzer kann sich die Website beliebig auf Englisch oder Deutsch ansehen, es kommt nicht darauf an, welche Kultur der Benutzer hat.

Im Verzeichnis „`apps/backend/i18n/de/message.xml`“, befindet sich die deutsche Übersetzungsdatei. In dieser Datei können alle deutschen Namen angepasst werden. Wichtig ist zu beachten, dass nach jedem Ändern der Datei der Cache geleert werden muss (`php symfony cache:clear` oder `php symfony cc`).

Da wir für die CRUD und Reservationskalender Module Javascript verwenden (Plugin `Datatables` und `Weekcalendar`), wird die Internationalisierung nicht vollständig gemacht.

6.1.8 Kontakt Formular

Mit den Kontaktformularen kann ein Benutzer dem Administrator Mitteilungen senden, sowie Credit Points kaufen oder sich registrieren.

6.1.8.1 Validators

Um die Felder der Kontaktformulare zu überprüfen, braucht es Validatoren. Diese befinden sich in den Dateien unter `lib/form/`. Die Validatoren werden mittels der Methode `setValidators()` gesetzt und in der Action Klasse mit der Methode `isValid()` geprüft. In der `setValidators()` Methode können verschiedene Informationen geprüft werden, ob in einem Feld z.B. ein String oder ein Integer eingegeben werden muss, ob es eine E-Mail sein soll oder die Mindest oder Maximum Anzahl Characters des Feldes nicht über- bzw. unterschritten werden darf.

```
class ContactusForm extends BaseForm
{
    public function configure()
    {
        $this->setWidgets(array(
            'name' => new sfWidgetFormInput(),
            'surname' => new sfWidgetFormInput(),
            'email' => new sfWidgetFormInput(),
            'comment' => new sfWidgetFormTextarea(),
        ));
        $this->widgetSchema->setNameFormat('contactus[%s]');

        $this->setValidators(array(
            'name' => new sfValidatorString(),
            'surname' => new sfValidatorString(),
            'email' => new sfValidatorEmail(array(), array('invalid' => 'The email address is invalid.')),
            'comment' => new sfValidatorString(array('min_length' => '4')),
        ));
    }
}
```

6.1.8.2 Mail Funktion

Den Inhalt einer Nachricht kann in der action.php Klasse des contactus Moduls angepasst werden oder für die Registrierung in der action.php Klasse des Moduls auth. Um die Mail Funktion zu verwenden, müssen einige Konfigurationen vorgenommen werden:

Mail Konfigurationen:

```
#apps/backend/config/factories.yml
...
dev:
    mailer:
        param:
            delivery_strategy: realtime #Angabe ob und wie Mail verschickt wird

all:
    mailer:
        param:
            transport:
                class: Swift_SmtpTransport
            param:
                host: smtp.gmail.com
                port: 465
                encryption: ssl
                username: ins.bothsr@gmail.com #Benutzername des Mail Empfänger Accounts
                password: da3ruler** #Passwort des Mail Empfänger Accounts
        routing:
            class: sfPatternRouting
            param:
                generate_shortest_url: true
```

```

extra_parameters_as_query_string: true

view_cache_manager:
  class: sfViewCacheManager
  param:
    cache_key_use_vary_headers: true
    cache_key_use_host_name: true

#apps/backend/config/app.yml
...
mail:
  reciever: ins.bothsr@gmail.com #Mail des Empfängers

```

6.2 jQuery

jQuery ist ein freiverfügbare Javascript Library, welches mächtige Funktionen zur Navigation und Manipulation der DOM-Syntax zur Verfügung stellt. Bei der Implementierung des Kalenders und der CRUD Objekte haben wir vor allem von den jQuery Funktionen Ajax und Event-Handling Gebrauch gemacht.

6.3 Reservationskalender

Der Reservationskalender im Control Panel wurde mit dem Plugin Weekcalendar implementiert. Der Weekcalendar baut auf jQuery und jQuery UI auf. Das Plugin bietet keine Tages-/Monatsdarstellung. Jedoch wird dies im Reservationssystem nicht benötigt.

Das Plugin bietet folgende Funktionen:

- Anzeige von Events innerhalb einer Wochenansicht.
- Events können als Array oder als JSON File via URL geliefert werden.
- Events können mit der Maus verschoben werden. (Wurde deaktiviert, kann jedoch nachträglich implementiert werden. Bei der aktuellen 1.2.2 Version ist noch ein Fehler vorhanden, welche von uns gefixt wurde).

Der Kalender ist in hohem Mass konfigurierbar. Beispiel: Read only Events können nicht bearbeitet werden.

Folgende Funktionen wurden selber implementiert:

- Events über 2 Tage.
- Darstellung im Internet Explorer – Cache Problem gelöst.
- Farbe der Reservationen

Der Benutzer sieht seine eigenen Reservationen blau, die vergangenen Reservationen werden grau dargestellt und die Reservationen von anderen Benutzern sind rot dargestellt. Er sieht jedoch nicht, wer die anderen Benutzer sind, nur dass es zu dieser Zeit eine Reservation auf einen Pod gibt. Der Administrator sieht die vergangenen Reservationen ebenfalls in grau, jedoch werden alle sonstigen Reservationen blau angezeigt, inklusive Informationen wer diese Reservation getätigt hat.

Datei	Speicherort
Kalenderkonfigurationen	web/js/pod-reservation-calendar.js
Weekcalendar Plugin	web/js/lib/weekcalendar.min.js

Action & Template View	apps/backend/modules/reservation
Helper Class	lib/ Reservationhelper.class.php

6.3.1 Pod Reservation Calendar (pod-reservation-calendar.js)

Bei der Implementierung der Kalenderkonfigurationen haben wir uns vor allem an die Beispielimplementierung gehalten. Teilweise wurde Code übernommen und an unsere Implementierung angepasst.

Folgend werden die einzelnen Funktionen erklärt:

TimeslotsPerHour [Value=4]: Die Anzahl Zeitschlitz sind im Value definiert. Wenn der Value auf 4 gesetzt ist, bedeutet das, dass pro Stunde 4 Timeslots à 15 Minuten ersichtlich sind. Falls dieser Value geändert wird, muss beachtet werden, dass die Höhe der View fest kodiert ist. Dies kann mit nachfolgender Funktion angepasst werden.

height : function(\$calendar) { return 1850; }: Der Value ist standardmässig auf 1850 gesetzt. Falls die Anzahl Timeslots wie erwähnt verändert werden, ist es empfehlenswert, die Höhe anzupassen.

timeSlotsHeight [Value=18]: Mit dieser Funktion kann die Grösse der Timeslots konfiguriert werden.

allowCalEventOverlap [Value=false]: Diese Option ermöglicht, ob Kalendereinträge überschneiden können oder nicht. Bei der Speicherung wird auf jeden Fall nochmals überprüft, ob es ein überschneidender Eintrag ist (siehe Action Saveevent()).

overlapEventsSeperate [Value=false]: Mit dieser Option kann konfiguriert werden ob ein Event Separat gerendert wird oder überlappend. Diese Funktion muss sowieso nicht beachtet werden, da keine Überschneidungen getätigt werden können.

firstDayOfWeek [Value=1]: Value 1 ist Montag und somit der erste Tag im Kalender.

use24Hour [Value=true]: Die Zeit im Kalender wird im 24 Stunden Format dargestellt.

businessHours : {start : 8, end : 18, limitDisplay : false }: Fett markierte Zeiten im Kalender sind Businesszeiten. Die Businesszeiten können mit dieser Funktion selbst definiert werden.

07:00		
08:00		

daysToShow [Value=7]: Falls Reservationen nur unter der Woche getätigt werden dürfen, kann bei dieser Option der Value verkleinert werden.

eventRender : function(calEvent, \$event) {}: Diese Funktion wird ausgeführt, bevor ein Event dargestellt wird. In dieser Funktion wird der jeweilige Event/Reservationseintrag manipuliert. In unserer Bachelorarbeit werden Events in der Vergangenheit und „TimeBetweenReservation“ Einträge grau dargestellt. Reservationen, die von anderen Benutzern getätigt wurden, werden rot dargestellt.

draggable: function(calEvent, \$event) { return false; }: In dieser Methode muss konfiguriert werden, was passiert, wenn Events per Mausclick verschoben werden können. In unserer Implementierung wurde dieser Event deaktiviert, nach Belieben kann jedoch dieser Event nachträglich implementiert werden. Der Bug, welcher im Plugin ersichtlich ist, wurde gefixt.

resizable: function(calEvent, \$event) { return false; }: Events können auf einfache Weise vergrößert oder verkleinert werden. Dies wurde in unserer Implementierung ebenfalls deaktiviert.

eventNew : function(calEvent, \$event) {}: Jedes Mal, wenn ein Event gespeichert wird, wird diese Funktion aufgerufen. Diese sendet die Daten per Ajax an den Server, um die Speicherung vorzunehmen. Es müssen viele Operationen vom Server überprüft werden, deshalb wird der Kalender jedes Mal bei einer Speicherung neu initialisiert.

eventClick : function(calEvent, \$event) {}: Ähnlich verhältet sich die eventClick Methode, nur steht hier noch ein Delete Button zur Verfügung.

data: function(start, end, callback) {}: Mit der Data Funktion werden die Daten vom Server abgerufen. Die Reservationen werden vom Server in einem Json File bereitgestellt (siehe Action Jsondata()).

HTML Code Manipulationsfunktionen:

Folgende Funktionen sind keine direkten Kalender Funktionen, trotzdem sind sie im selben File definiert:

function resetForm(\$dialogContent) {}: Falls eine Reservation erstellt wird, öffnet sich ein Edit / New Dialog, bei welchem der vorherige Content entfernt werden muss.

function setupStartAndEndTimeFields(\$startTimeField, \$endTimeField, calEvent, timeslotTimes) {}: Da eine Reservation ebenfalls über zwei Tage gehen kann und auch in dieser Funktion Bugs vorhanden waren, musste diese Funktion angepasst werden. Nun können alle möglichen Zeiten angegeben werden.

\$("#data_source").change(function() {}: Der Kalender wird nochmals neu initialisiert, falls diese Funktion aufgerufen wird.

6.3.2 Action index()

Die executeIndex Methode ist lediglich für die Initialisierung des Kalenders verantwortlich. Sobald der Kalender initialisiert wurde, ruft der Client die executeJsondata() auf, sodass die Daten im Kalender angezeigt werden.

Bei der View des Administrators werden alle aktivierten Pods in der Combobox zur Verfügung gestellt.

Beim User werden nur Pods dargestellt, auf welche er Zugriff hat. Zudem wird angezeigt, ob der User für Reservationen Credit Points benötigt.



6.3.3 Action Jsondata()

Beim Administrator werden alle Informationen von den Benutzern dargestellt. Bei den Usern werden im Json File keine Informationen von anderen Benutzern mitgeliefert.

Abruf der Kalender Informationen:

http://domain/index.php/de/reservation/calendarevents.json?pod_id=1&start=1275861600000&end=1276466400000

Es werden nur Daten übermittelt, welche im Zeitraum von 1275861600000 bis 1276466400000 befinden. Es handelt sich hier um das Unix Timestamp Format in Millisekunden. Das übertragene Datum und die Zeit wären hier: Sun, 13 Jun 2010 24:00:00.

3 Kalender Events:

```
[{"id":"3","start":"2010-06-09T15:30:00.000+02:00","end":"2010-06-09T19:30:00.000+02:00","title":"reserved","readOnly":"true"},
{"id":"4","start":"2010-06-08T13:30:00.000+02:00","end":"2010-06-08T17:30:00.000+02:00","title":"User: mmuster Pod:Test Example Pod"},
{"id":"4","start":"2010-06-08T17:30:00.000+02:00","end":"2010-06-08T17:45:00.000+02:00","title":"timebetween","readOnly":"true"}]
```

Bei der ersten Reservation handelt es sich um eine Reservation, welche von einem anderen User getätigt wurde. Die zweite Reservation wurde vom User: mmuster getätigt. Nun ist ersichtlich, dass die dritte Reservation dieselbe ID besitzt wie die zweite Reservation. Dies ist die Zeit, welche zwischen den Reservationen mit der Option timebetween für den jeweiligen Pod definiert werden kann. Wenn eine Reservation gelöscht wird, wird nicht der ganze Kalender neu abgerufen. Beim Löschvorgang wird ganz einfach dreimal die Funktion „`$calendar.weekCalendar("removeEvent", calEvent.id);`“ aufgerufen, da es Reservationen über zwei Tage geben kann und die timebetween Zeit ebenfalls gelöscht werden muss. Es können maximal drei Events im Kalender mit derselben ID eingetragen sein (zwei Einträge für Reservation über 2 Tage und einer für den Timebetween Event).

Jedes Mal wenn die Json Daten abgerufen werden, müssen die Daten als „schon veraltet“ zurück gegeben werden. Der Grund dafür ist, dass der Internet Explorer 8 die Reservation zwischenspeichert und nicht mehr neu abrufen. Wenn somit ein anderer Benutzer eine Reservation tätigt, wird diese nach Aktualisierung der Homepage im IE8 nicht dargestellt. Darum wird nun mit folgender Funktion: `$this->getResponse()->setHTTPHeader('Expires','Sat, 1 May 2010 13:37:00 GMT');` der Inhalt als schon veraltet abgerufen.

6.3.4 Action Saveevent()

Über folgenden HTTP GET Befehl, kann eine Reservation gespeichert werden:

<http://domain/index.php/de/reservation/save?>

`rid=0&pid=1&from=1275984000000&to=1275985800000`

Parameter	Bedeutung
rid	Reservation ID: Wenn die Reservation ID null ist, handelt es sich um eine neue Reservation.
pid	Pod ID
from	Tue, 08 Jun 2010 08:00:00 GMT -> Server muss durch die Zeitzone noch eine oder zwei Stunden hinzuzählen.
to	Fri, 11 Jun 2010 08:30:00 GMT -> Server muss durch die Zeitzone noch eine oder zwei Stunden hinzuzählen.

Implementierung über 2 Tage:

New Reservation
✕

Date: Jun 12, 2010

Start Time:

10:30 pm

The Endtime is on the next Day

End Time:

03:00 am

cancel
save

Clientseitig wurde die Reservation über zwei Tage wie folgt implementiert: Sobald eine Reservation an den Server gesendet wird und die Checkbox "The Endtime is on the next Day" angewählt wurde, wird automatisch 24 Stunden zur Endzeit addiert.

Implementierung Javascript (Clientseitig)

```
if($('#input[name=endnextday]').is(':checked')){
    endTimeStamp += 86400000;
}
```

Die Daten werden folgendermassen in der Datenbank gespeichert:

```
mysql> mysql> select * from reservation;
```

reservation_id	from_at	to_at	creditpoints	user_id	pod_id
1	2010-04-27 09:00:00	2010-04-27 15:00:00	4.00	2	1
2	2010-04-03 01:00:00	2010-04-03 08:00:00	0.00	1	1
3	2010-06-08 10:00:00	2010-06-08 14:00:00	4.00	1	1
4	2010-06-08 21:30:00	2010-06-09 04:00:00	6.50	1	1
6	2010-06-10 06:30:00	2010-06-11 10:30:00	0.00	4	1

5 rows in set (0.00 sec)

Es können folgende Fehlermeldung beim Client erscheinen, welche der Server zurück gibt:

Rückgabewert vom Server	Bedeutung
0	Reservation wurde Ordnungsgemäss gespeichert.
-1	Ein überlappender Event wurde gefunden.
-2	Die Reservation startet in der Vergangenheit.
-3	Kein Zugriff auf den Pod.
-4	Nicht genügend Credit Points vorhanden.
-5	Keine Berechtigung auf die angegebene Reservation.
-6	Keine Reservation zu dieser ID gefunden.

Alle Fehlermeldung unter -1 treten nur bei Benutzern auf, beim Administrator werden diese Fehlermeldungen nicht erscheinen. Sobald der Benutzer die 'Credentials Admin' besitzt, wird die Reservation direkt gespeichert. Falls der Benutzer die, Credi-

als Admin' nicht besitzt, wird die Reservation zuerst validiert und erst dann gespeichert (siehe Process View Reservation).

Ablauf Save Reservation:

Sobald eine Reservation gespeichert wird, wird ein neues ReservationData Objekt erstellt, welches in der ReservationData Klasse zu finden ist. Pro save Request wird ein ReservationData Objekt generiert. Dieses Objekt soll vor allem als Hilfe dienen, da oft die Reservationszeiten konvertiert werden müssen. Im 'to' und 'from' Request Parameter wird die Zeit in Unixtimestamp Format in Millisekunden angegeben. Dies bedeutet, bevor ein Vergleich mit der aktuellen Zeit getätigt werden kann, muss durch 1000 dividiert werden. Die Zeiten werden in der Datenbank im Datetime Format gespeichert.

Spezialfälle für den Benutzer

Der Benutzer kann keine Reservationen löschen oder bearbeiten, welche in der Vergangenheit liegen. Reservationen welche in der Zukunft liegen, können beliebig verlängert, verkürzt oder gelöscht werden. Credit Points werden beim Erstellen einer Reservation abgezogen. Die Anzahl Credit Points, die eine Reservation kostet, wird in der Datenbank gespeichert. Dies ermöglicht eine bessere Kontrolle bei Änderungen von Pod Berechtigungen. Wenn der Pod plötzlich kostenpflichtig wird und ein Benutzer alte Reservationen löscht, erhält dieser keine Credit Points, da er zu dem damaligen Zeitpunkt nichts bezahlen musste. Für die bereits getätigten Reservationen muss er aber trotzdem nichts bezahlen, da der Pod zu dem damaligen Zeitpunkt noch kostenlos war. Durch die Speicherung der Credit Points in der Datenbank kann dieses Problem gelöst werden. Sobald der Benutzer auf den nun kostenpflichtigen Pod eine neue Reservation erstellt, muss dieser Credit Points für die Reservation bezahlen.

Ein weiterer Spezialfall ist eine laufende Reservation

Laufende Reservationen können vom Benutzer nicht mehr gelöscht werden. Auf Wunsch vom Betreuer ist es auch nicht möglich, die Reservation zu verkürzen, nur zu verlängern.

Bearbeitung einer Reservation von einem Benutzer durch einen Administrator

Eine Reservation im Reservationskalender, welche ein Benutzer erstellt hat, kann der Administrator bearbeiten. Die Credits Points werden automatisch abgezogen oder aufgeladen. Auch wenn der Administrator eine Reservation eines Benutzers löscht, werden dem Benutzer automatisch die Punkte wieder gut geschrieben. Zu beachten ist, dass ein Administrator auch vergangene Reservationen bearbeiten kann. Es kommt nicht darauf an, ob eine Reservation in der Vergangenheit ist oder nicht, die Credit Points werden automatisch angepasst (betrifft nur den Reservationskalender).

6.3.5 Action Deleteevent()

Durch einen Klick auf die Reservation, öffnet sich das Fenster, in welchem die Reservation bearbeitet oder gelöscht werden kann.

Über folgenden HTTP GET Befehl, kann eine Reservation gelöscht werden.

<http://domain/index.php/en/reservation/delete?rid=7>

```
mysql> select * from reservation;
+-----+-----+-----+-----+-----+-----+
| reservation_id | from_at          | to_at          | creditpoints | user_id | pod_id |
+-----+-----+-----+-----+-----+-----+
| 1              | 2010-04-27 09:00:00 | 2010-04-27 15:00:00 | 4.00        | 2       | 1      |
| 2              | 2010-04-03 01:00:00 | 2010-04-03 08:00:00 | 0.00        | 4       | 1      |
+-----+-----+-----+-----+-----+-----+
```

Im obigen Bild ist ersichtlich, dass pro Reservation die Anzahl Credit Points gespeichert wird, die diese Reservation gekostet hat. Dies ermöglicht, dass wenn der Administrator eine Reservation über den Reservationskalender eines Benutzers löscht, automatisch die Credit Points dem Benutzer wieder hinzugefügt werden.

Rückgabewert vom Server	Bedeutung
0	Reservation wurde Ordnungsgemäss gelöscht.
-1	Keine Berechtigung diese Reservation zu löschen.
-2	Die Reservation vom Benutzer startet schon in der Vergangenheit.

6.3.6 Performance

Jeweils beim Seitenwechsel werden die Reservationen für die View per Json heruntergeladen, dies kann mit folgendem Request getätigt werden:

http://localhost/index.php/en/reservation/calendarevents.json?pod_id=1&start=1277071200000&end=1277676000000

Durch die Angabe „start“ bis „end“ Zeit können die abgerufenen Reservation eingegrenzt werden.

6.3.7 Licence

Für den Weekcalendar sind folgende zwei Opensource Lizenzen gegeben:

Name	Bedeutung
Open Source Initiative OSI - The MIT License:Licensing	http://www.opensource.org/licenses/mit-license.php

GNU GENERAL PUBLIC LI-
CENSE

<http://www.gnu.org/licenses/gpl.html>

6.4 MyCalendar

Im Modul MyCalendar wird die History des Benutzers angezeigt. Die vergangenen und noch kommenden Reservationen des Benutzers selbst sind in dieser Ansicht dargestellt. Der Benutzer kann die Reservationen auf ein anderes Gerät z.B. einem PDA herunterladen.

6.4.1 Abweichende Konfigurationen zum Reservationskalender

Einer der wichtigsten Abweichungen ist vor allem, dass Überlappende Events angezeigt werden. Einem Benutzer ist es möglich, Reservationen zur selben Zeit auf verschiedene Pods zu machen. Dies ist so implementiert, da bei einem Kurs mit einem Benutzer für alle Kursteilnehmer Reservation auf die verschiedenen Pods gemacht werden können. Dabei ist garantiert, dass die Pods nicht von anderen Benutzern reserviert werden können und die Pods während des Kurses zur Verfügung stehen.



Jedoch werden im MyCalendar praktisch keine Funktionen behandelt. Es können keine Reservationen erstellt oder gelöscht werden. Es ist lediglich nur ein Kalender, der eine Übersicht über die Reservationen des jeweiligen Benutzers gestattet.

6.4.2 Ical Informationen abrufen

Der Benutzer kann alle seine Reservationen in einem ical Reservations-Format abrufen.

ICAL Reservation Example

```
BEGIN:VCALENDAR
VERSION:2.0
PRODID:http://www.ins.hsr.ch/reservationcenter
METHOD:PUBLISH
BEGIN:VEVENT
UID:11@reservationcenter.ins.hsr.ch
ORGANIZER:mkolb
SUMMARY:Reservation at INS Reservation Center
DESCRIPTION:Pod Test Example Pod Credit Points cost 0.00
CLASS:PUBLIC
DTSTART:20100608T080000
DTEND:20100608T120000
DTSTAMP:20100608T001816Z
END:VEVENT
BEGIN:VEVENT
UID:12@reservationcenter.ins.hsr.ch
ORGANIZER:mkolb
SUMMARY:Reservation at INS Reservation Center
DESCRIPTION:Pod Routing Pod 2 Credit Points cost 0.00
CLASS:PUBLIC
DTSTART:20100608T090000
DTEND:20100608T130000
DTSTAMP:20100608T001816Z
END:VEVENT
END:VCALENDAR
```

Die Funktionen wurden mit dem Google Kalender überprüft. Mit dem Ical File können die Reservation dem Google Kalender hinzugefügt werden. Die UID wird zusammen gestellt durch die Primary ID der Reservation und "@reservationcenter.ins.hsr.ch". Die Ical Funktion wurde gemäss RFC5545 implementiert.

6.5 Authentifizierung und Anbindung an LDAP Server

Für die Anbindung am LDAP Server wurde die adLDAP Klasse verwendet.

Konfigurationen im settings.yml:

```
all:
  .actions:
    login_module:  auth # Benutzer gelangt automatisch zur Loginseite, wenn er nicht
                      angemeldet ist. Das Login wird im Modul auth definiert.
    login_action:  login
```

Falls der User noch nicht angemeldet ist, wird er zum Modul auth und zur Action login weitergeleitet.

6.5.1 Konfigurationen LDAP Server

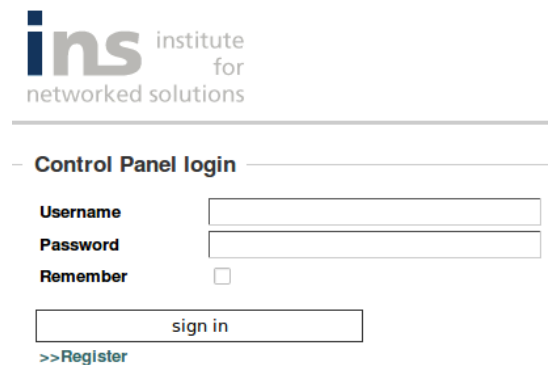
Die Konfigurationen für die Active Directory Einstellungen sind zu finden in der app.yml Konfigurationsdatei.

Beispiel: Konfigurationen für die Entwicklung

```
all:
  ad:
    options: # these options are for the adldap class
    account_suffix: '@rlm.local'
    base_dn: 'DC=rlm,DC=local'
    domain_controllers: [ '192.168.0.31' ]
    ad_username: 'ldapuser_reader'
    ad_password: '*****'
    recursive_groups: true
```

6.5.2 Form

Für das Login wurde die login.php View verwendet, welche unter folgendem Ordner zu finden ist: app/backend/templates/



Wie schnell ersichtlich ist, weicht das Design vom übrigen Control Panel ein wenig ab. Es wurde zusätzlich eine CSS Datei erstellt: login.css

6.5.3 Error Handling

Wenn kein LDAP Server gefunden wird, wird folgende Fehlermeldung ausgegeben:



Die adLDAP Klasse wirft eine adLDAPException, welche in der Action Klasse gefangen wird. Es wurde zusätzlich eine noldap View erstellt.

Falsche Logindaten:

Wenn der Benutzer einen falschen Benutzernamen oder ein falsches Passwort eingibt, folgt folgende Fehlermeldung:

The screenshot shows the 'Control Panel login' form. At the top, there is a yellow error message: '* Wrong user or password'. Below this, the form fields are: Username (containing 'mkolb'), Password (empty), and Remember (checkbox). A 'sign in' button is at the bottom, with a '>>Register' link below it.

Fehlermeldung setzen

Layout Definition:

```
<?php if ($sf_user->hasFlash('error')): ?>
  <tr>
    <td class="flash_error" colspan="2"><?php echo $sf_user->getFlash('error') ?></td>
  </tr>
<?php endif ?>
```

Definition in der Action Klasse:

```
$this->getUser()->setFlash('error', '* Wrong user or password');
```

6.5.4 LDAP Gruppen

Eine Einschränkung bei den LDAP Gruppen ist, dass ein Benutzer nur einer LDAP Gruppe angehören darf. Wie dies weiterentwickelt werden kann, ist im Technischen Bericht ersichtlich.

LDAP Gruppen werden rekursiv vom Active Directory abgefragt. Zuerst wird überprüft, ob der Benutzer ein Administrator ist. Die Administrator Gruppe kann im app.yml konfiguriert werden.

LDAP Administrator Gruppe definieren

```
#apps/backend/config/app.yml
```

```
all:
```

```
  cpadmin: 'CP-Admin'      #LDAP Admin CN Group definition
```

```
#apps/backend/modules/auth/actions/action.class.php
```

```
$user->addCredential('admin');      // Admin Credentials für den User setzen
```

Ein Benutzer kann sich anmelden, wenn er in einer Gruppe ist, welche im Control Panel erfasst ist. Wichtig ist, dass sich der Benutzer zurzeit nur einmal in einer LDAP Gruppe befinden darf.

Sobald ein User sich anmeldet, werden verschiedene Attribute gesetzt, welche vor allem im Reservationskalender verwendet werden:

```
„,$this->getUser()->setAttribute('userldapgroupid', $ldapgroup->getLdapgroupid());“
```

Attribut	Bedeutung
loginname	Login Name des Benutzers.
userid	User ID des Benutzers.
userldapgroupid	LDAP Gruppen ID: Wird nur gesetzt, wenn der Benutzer kein Administrator ist.

6.6 Active Reservation Modul

In diesem Modul können Geräte per SNMP angesteuert und neu eingeschalten bzw. ausgeschalten werden.

Folgende Dateien, welche vom INS zur Verfügung gestellt wurden, wurden teilweise weiterentwickelt und übernommen:








Datei	Änderungen
Device.php	Description Methode hinzugefügt
AmazingPDU.php	Keine
AmazingMIB_V1.02.mib	Keine
TerminalServer.php	Keine

Am Router 1 anmelden

```
ssh mkolb@10.1.1.10 -p 2011
mkolb@10.1.1.10's password: *****
Router>
```

Der Administrator sieht links im Menü alle Pods und somit alle ein- bzw. ausgeschalteten Geräte des im Menü ausgewählten Pods. Er hat die Möglichkeit, die Geräte ein- bzw. auszuschalten, auch wenn diese gerade von einem Benutzer gebraucht werden.

Test Example Pod

Device	Name	Line Status	Device Info
	R1	connected Clear Line	SSH line 2011 Terminal Server IP: 10.1.1.10
	R2	not connected	SSH line 2012 Terminal Server IP: 10.1.1.10
	R3	not connected	SSH line 2013 Terminal Server IP: 10.1.1.10
	R4	not connected	SSH line 2014 Terminal Server IP: 10.1.1.10
	R5	not connected	SSH line 2015 Terminal Server IP: 10.1.1.10
	R6	not connected	SSH line 2016 Terminal Server IP: 10.1.1.10
	R7	not connected	SSH line 2017 Terminal Server IP: 10.1.1.10

Der Benutzer hat die Möglichkeit, mehrere Reservationen auf verschiedene Pods zu tätigen. Dies wurde gemäss Wunsch des Betreuers für den Unterricht implementiert. Die reservierten Pods werden links im Menü aufgelistet. Die ein- und ausgestellten Geräte des ausgewählten Pods werden wie im Bild ersichtlich auf der linken Seite

dargestellt. Durch einen Klick auf ein Bild kann ein Gerät ein- bzw. ausgestellt werden. Unter Line Status sieht der Benutzer den Status der Verbindung des Gerätes.

Legende zu den Geräten:

	ASA		Router
	Switch		ATM
	IPS		MDS
	NETAPP		JBOD
	VM, HOST		FC

Wo und in welchen Tabellen was eingetragen werden muss, kann im Installation Guide entnommen werden.

6.7 Datatables CRUD Objekte

Die CRUD Module sind mittels Datatables implementiert worden. In dieser Datei wird folgendes definiert:

- wie die Spaltennamen heissen
- was beim doppel Klick auf eine Zeile passiert
- welche Informationen im edit Container vorhanden sind
- welche Funktionen im edit Container vorhanden sind (delete, cancel, save)
- welche Informationen es braucht, um ein Objekt neu hinzuzufügen
- welche Funktionen im new Container vorhanden sind (cancel, save).

Die Datatable Dateien befinden sich unter web/js/datatables/.

Das Plugin Datatables stellt uns mehrere Features zur Verfügung, die wir nutzen können:

- „on the fly“ filtern
- Suchen
- nach Spalten sortieren
- Seiten blättern (Paging)
- Anzahl Objekte pro Seite darstellen (z.B. 10, 25, 50, 100)

6.7.1 Datatables (dataTables-*.js)

bProcessing [Value=true]: Aktivieren der Anzeige des "Processing" Indikators, wenn die Tabelle geladen wird. Dies ist besonders nützlich bei Tabellen mit großen Datenmengen, wo es eine spürbare Ladezeit gibt.

sAjaxSource [Example Value=device/json]: Definition der Quelle der Daten.

bAutoWidth [Value=false]: Deaktiviert die automatische Berechnung der Spaltenbreite. So wird Rechenzeit erspart.

aoColumns [Value=siehe folgende Tabelle]: Definiert die Titel der Spalten.

Spaltennamen Beispiel: Device

```
[ { "sTitle" : "Device ID"}, { "sTitle" : "Name"}, { "sTitle" : "Description" }, { "sTitle" : "Is Activated", "bSortable": false}, { "sTitle" : "Amazing PDU Ip"},{ "sTitle" : "Line"}, { "sTitle" : "Devicenumber"}, { "sTitle" : "Devicecategory"},{ "sTitle" : "Pod"}]
```

\$("#devicetable tbody").dblclick(function(event): Bei einem Doppelklick auf eine Kolone, wird das „Bearbeiten Feld aufgerufen“.

function fnClickAddRow(): Diese Funktion fügt eine neue Kolone der Tabelle hinzu und ruft das Fenster „new Device“ auf.

6.7.2 Action Index()

Im indexSuccess wird die Datatables aufgerufen und ein „hidden“ Formular bereit gestellt. Ist genau gleich implementiert wie beim Reservation Kalender.

Speziell ist, wie die Fremdschlüssel übergeben werden:

Spaltennamen Beispiel: Group Policy

Hidden Formular:

```
<option label="<?php echo $ldapgroup->getName()?" value="<?php echo $ldapgroup->getLdapgroupId()?"><?php echo $ldapgroup->getName()?"</option>
```

Javascript File:

```
var ldapgrouptextField = $dialogContent.find("select[name='ldapgroup']");
var ldapgroupField = ldapgrouptextField.val($dialogContent.find("option[label='"+aData[2]+'"]').val());
```

Zuerst wird im Javascript File dataBase-grouppolicy.js der Dom Tree der gewählten LDAP Gruppe gesucht. Dabei erhält man mit folgender Funktion „ldapgroupField.val()“ den Fremdschlüssel.

Um den Text in der Tabelle anzuzeigen und nicht die ID, wird weiter im DOM Tree gesucht und kann mit folgender Methode den Text übergeben:

```
ldapgrouptextField.find("option[value='"+ldapgrouptextField.val()+"']").text()
```

6.7.3 Action Json()

Die Daten werden wie beim Reservationskalender per JSON Informationen empfangen.

<http://domain/index.php/en/pod/json>

Beispiel JSON Daten

```
{ "aData": [ ["1", "Test Example Pod", "Dies ist zur Verf\u00fchrung gestellter Test Pod.", "10.1.1.10", true, 15, "Routing Switching"], ["2", "Routing Pod 2", "normal Routing Pod", "10.1.1.12", true, 30, "Routing Switching"] ] }
```

6.7.4 Action Save()

<http://domain/index.php/en/device/save>

Per Post Request werden die Daten an den Server übermittelt. Nachfolgend ein Beispiel einer Speicherung eines Pods.

Parameter	Bedeutung
active	Beschreibt, ob ein Pod aktiviert oder deaktiviert ist.
desc	Description ist für die Beschreibung des Pods.
id	Primary ID des Pods. Wenn die ID null ist, wird ein neuer Pod erstellt, anderenfalls wird ein Pod bearbeitet.
name	Name des Pods
pc	Fremdschlüssel der PodCategory
tb	Definiert die timebetween Reservation Zeit.
ts	Terminalserver ip

Sobald einmal die Daten abgerufen werden, werden die Daten lokal bearbeitet und bei einer Änderung per Post Request an den Server übertragen. Falls aber ein anderer Administrator Änderungen vornimmt, muss die Tabelle neu geladen werden.

Folgend wird der Tabelle lokal einen neuen Pod Datensatz hinzugefügt:

Datensatz der Tabelle und auf dem Server hinzufügen
Lokal: <pre>oTable.fnAddData([data, namefield.val(), descriptionfield.val(), terminalserverip.val(), \$('input[name=isactiv]').is(':checked'), tbField.val(), pcField.find("option[value='"+pcField.val()+"']").text()]);</pre> Server: <pre>\$.post("pod/save", { id: 0, name: namefield.val(), desc: descriptionfield.val(), ts: terminalserverip.val(), active: \$('input[name=isactiv]').is(':checked'), tb: tbField.val(), pc: pcField.val() }, function(data)</pre>

6.7.5 Action Delete()

Parallel wie bei der Save Methode werden die Daten an den Server übermittelt und lokal aus der Tabelle entfernt.

Datensatz aus der Tabelle und vom Server entfernen
Lokal: <pre>oTable.fnDeleteRow(anSelected[0]);</pre> Server: <pre>\$.ajax({ url: 'pod/delete?id='+aData[0], success: function(data) { ... }</pre>

6.7.6 Performance

(Siehe separates Dokument Testdokumentation/Testdokumentation.docx)

6.7.7 Licence

Für den Weekcalendar sind folgende zwei Opensource Lizenzen gegeben:

Name	Bedeutung
GPL v2 license	http://datatables.net/license_gpl2
BSD license	http://datatables.net/license_bsd

6.8 XML-RPC

Für das vereinfachen der Implementierung des XML-RPC haben wir die Library `kd_xmlrpc.php` verwendet.

6.8.1 Client Script

Das Client Script ist auf dem Radius Server zu finden. In den folgenden zwei Variablen wird angegeben, auf welchem Server das Server Script vorzufinden ist. In der zweiten Variablen ist der Ort definiert, wo das Script aufzufinden ist. Beispielaufruf:

„192.168.223.147:8080/server.php“

```
$site = "192.168.223.147:8080";
```

```
$location = "/server.php";
```

Abruf ob eine Reservation verfügbar ist:

```
XMLRPC_request($site, $location, 'reservation.check',  
              array(XMLRPC_prepare($test, 'XML')));
```

'reservation.check' ist die Methode, welche beim Server Script aufgerufen wird. In der Variable `$test` sind die Parameter „username“, „password“ und „portIp“ vorzufinden.

6.8.2 Server Script

Im „Server.php“ File müssen folgende Konfigurationen vorgenommen werden:

Variablen für den Zugriff zur MySQL DB

```
$dbserver= "localhost"; // Hostname vom MySQL Server  
$db = "controlpanel"; // Datenbankname  
$dbuser = "root"; // DBuser  
$dbpassword = "da3ruler"; // Passwort
```

Im „web_service_api.php“ File ist die Methode definiert.

Zuerst muss die Methode, welche Remote vom Client Script aufgerufen wird, definiert werden.

```
$xmlrpc_methods['reservation.check'] = reservation_check;
```

Danach wird auf die Datenbank zugegriffen und überprüft, ob eine Reservation für das bestimmte Gerät vorhanden ist.

6.9 Plugins & Versionen

Symfony Version 1.4.3

jQuery JavaScript Library v1.4.2

jQuery WeekCalendar v1.2.2

jQuery DataTables v1.6.2

Installation Guide

Version 1.0

Projekt:

Remote Lab Management System

Projektmitglieder:

Markus Kolb

Mirjam Schaffner

Betreuer:

Rolf Schärer

Beat Stettler

Revision				
Version	Status	Datum	Beschreibung/Änderung	Autor
1.0rc01	In Bearbeitung	16.06.2010	Erstellung des Installation Guide, kapitel 1 bis 4 beschrieben	MK
1.0rc02	In Bearbeitung	16.06.2010	Gegenlesen, korrigieren, ab Gerätekategorie erstellt	MS
1.0	Close	17.06.2010	Version 1.0	MK

Inhaltsverzeichnis

1	Einführung	4
1.1	Zweck	4
1.2	Gültigkeitsbereich	4
1.3	Definitionen und Abkürzungen.....	4
1.4	Referenzen	4
2	Installation Server	5
2.1	Linux Ubuntu.....	5
2.2	Apache Webserver	6
2.3	Mysql Datenbank.....	7
2.3.1	Symfony Datenbank Konfigurationen.....	7
3	Config Files	8
3.1	LDAP Server	8
3.2	Mail Konfigurationen	8
3.2.1	Empfänger Adresse definieren.....	8
3.2.2	Mail Account Settings	9
3.3	Session Timeout.....	9
4	Control Panel Guide	10
4.1	Pod Kategorie.....	10
4.2	Pod.....	10
4.3	Geräte Kategorie	11
4.4	Gerät	11
4.5	User.....	12
4.6	Ldap Group.....	12
4.7	Group Policy.....	12
4.8	Reservation für einen Benutzer erstellen.....	12
4.9	Reservation eines Benutzers bearbeiten	13
5	XML-RPC	14
5.1	Client Script (Radius Server)	14
5.2	Server Script (Controlpanel).....	14
6	Installation Testing Guide	15

1 Einführung

1.1 Zweck

Dieses Dokument enthält die gesamte Installation des Control Panels. Die Konfigurationsanpassungen werden beschrieben und alle Konfigurationsfiles werden mit dem Pfad angegeben.

1.2 Gültigkeitsbereich

Die Gültigkeit dieses Dokuments gilt für die Abgabe des Projekts.

1.3 Definitionen und Abkürzungen

Das Glossar ist im separaten Dokument (Glossar.docx) zu finden.

1.4 Referenzen

Referenz	Quelle

2 Installation Server

2.1 Linux Ubuntu

Es muss keine spezielle Ubuntu Version verwendet werden. Ubuntu ist erhältlich unter:
<http://www.ubuntu.com>

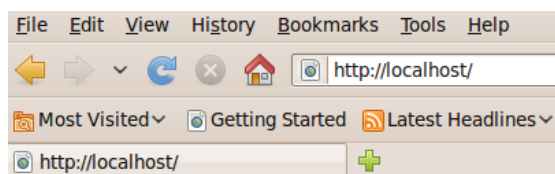
Nach der Installation des Ubuntu Systems müssen folgende Packages installiert werden:

Name	Quelle
apache2	Webserver
php5	Programmiersprache in der das Control Panel entwickelt wurde.
libapache2-mod-php5	PHP5-Modul für den Webserver Apache 2
php5-snmp	Wird benötigt, dass die Devices via SNMP angesteuert werden können.
php5-mysql	Aktiviert mysql in php
mysql-server-5.1	Datenbank Server
phpmyadmin	Wird zur Administration der Mysql Datenbanken verwendet (optional).
php5-cli	Wird für die Entwicklung benötigt (optional).
php5-ldap	Ermöglicht das Zugreifen auf die LDAP Datenbank.

Mittels folgenden Befehls können die Packages auf einer Ubuntu Distribution installiert werden:

```
root@ubuntu:~$ apt-get install apache2 php5 libapache2-mod-php5 php5-snmp  
php5-mysql mysql-server-5.1 phpmyadmin php5-cli php5-ldap
```

Nach der Installation kann mittels abrufen der localhost Webseite überprüft werden (<http://localhost/>), ob die Installation gelungen ist.



It works!

2.2 Apache Webserver

Konfigurationen, welche nach der Installation des apache2 Dienstes gemacht werden, sind in der nachfolgenden Tabelle aufgelistet. Der Standardinstallationsordner befindet sich im Verzeichnis: „/etc/apache2/“

Beim Apache muss das mod_rewrite Module aktiviert werden, was mit folgendem Shell Befehl ausgeführt wird:

```
root@ubuntu:~# a2enmod rewrite
```

Das DocumentRoot Verzeichnis muss in den Ordner „controlpanel/web“ zeigen.

sites-available.conf

```
# Document Root Definition des Control Panels
<VirtualHost *:80>
    ServerAdmin webmaster@localhost

    DocumentRoot /home/sfprojects/controlpanel/web
    <Directory />
        Options FollowSymLinks
        AllowOverride all
    </Directory>
    <Directory /home/sfprojects/controlpanel/web>
        Options Indexes FollowSymLinks MultiViews
        AllowOverride all
        Order allow,deny
        allow from all
    </Directory>
</VirtualHost *:80>

# XML-RPC Document Root Konfiguration
<VirtualHost *:8080>
    DocumentRoot "/home/sfprojects/xmlrpc"
    DirectoryIndex index.php
    <Directory "/home/sfprojects/xmlrpc">
        AllowOverride All
        Allow from All
    </Directory>
</VirtualHost>
```

Port Definitionen findet man im File ports.conf, wo konfiguriert wird, auf welche Ports der Server hört und die Homepage erreichbar ist.

ports.conf

```
ServerName 10.1.1.14
NameVirtualHost *:80
Listen *:80 # Port über welcher das Control Panel erreichbar ist
Listen *:8080 #definiert der Port für XML RPC
```

Damit phpmyadmin verwendet werden kann, muss zusätzlich noch ein VirtualHost hinzugefügt werden. Sobald die VirtualHosts angepasst wurden, muss der Apache Deamon neu gestartet werden:

```
root@ubuntu:~$ apache2ctl restart
```

2.3 Mysql Datenbank

Über die Shell wird mit folgendem Befehl die Datenbank „controlpanel“ erstellt:

```
root@ubuntu:~$ mysqladmin -uroot -p create controlpanel
```

Danach müssen die Tabellen erstellt werden, welche vom Control Panel benötigt werden:

```
root@ubuntu:~$ mysql -u root -p -h localhost controlpanel < controlpanel/data/sql/schema.sql
```

Danach kann über phpmyadmin oder über die Shell überprüft werden, ob die Tabellen sauber erstellt wurden.

```
mysql> show tables;
+-----+
| Tables_in_controlpanel |
+-----+
| dbuser                  |
| device                  |
| device_config           |
| devicecategory          |
| group_has_access_to_pod |
| lab                     |
| ldapgroup               |
| pod                     |
| podcategory             |
| reservation             |
| user_device_config      |
| user_lab                |
+-----+
12 rows in set (0.00 sec)
```

2.3.1 Symfony Datenbank Konfigurationen

Falls Externe Mysqlserver verwendet werden, können diese unter „mysql:host=“ eingetragen werden. Falls ein anderer Datenbank Name verwendet wird, muss dies ebenfalls im File controlpanel/config/database.yml geändert werden.

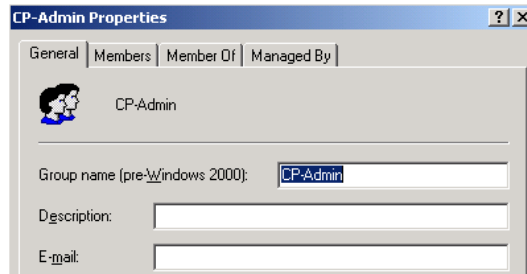
controlpanel/config/database.yml

```
all:
  doctrine:
    class: sfDoctrineDatabase
    param:
      dsn: 'mysql:host=localhost;dbname=controlpanel'
      username: root
      password: *****
test:
  doctrine:
    class: sfDoctrineDatabase
    param:
      dsn: 'mysql:host=localhost;dbname=controlpanel_test'
      username: root
      password: *****
controlpanel:
  class: sfDoctrineDatabase
  param:
    dsn: 'mysql:host=localhost; dbname=controlpanel_test'
    username: root
    password: da3ruler
```

3 Config Files

Die Konfigurationsdateien sind im Control Panel Ordner „controlpanel/apps/backend/config“ vorzufinden.

3.1 LDAP Server



Unter der Option cpadmin muss die Administrator CN Gruppe definiert werden.

app.yml

```
all:
  ad:
    options: #Diese Optionen werden für die adLDAP Klasse gebraucht, die Dokumentation ist Online vorhanden
    account_suffix: '@rlm.local'
    base_dn: 'DC=rlm,DC=local'
    domain_controllers: [ '10.1.1.12' ]
    ad_username: 'mkolb'
    ad_password: '*****'
    recursive_groups: true
    cpadmin: 'CP-Admin' #LDAP Admin CN Group Definition
```

3.2 Mail Konfigurationen

Mails werden via Mailaccount an die definierte Empfänger Adresse gesendet.

3.2.1 Empfänger Adresse definieren

app.yml

```
all:
  mail:
    reciever: reciever@example.com # definiert die Empfänger Adresse der Formulare
```

3.2.2 Mail Account Settings

factories.yml

```
all:
  mailer:
    param:
      transport:
        class: Swift_SmtpTransport
        param:
          host: smtp.gmail.com
          port: 465
          encryption: ssl
          username: ins.bothsr@gmail.com #Benutzername des Empfänger Mail Accounts
          password: da3ruler** #Passwort des Empfänger Mail Accounts
```

3.3 Session Timeout

Die Session Timeout definiert, wie lange der Benutzer ohne Tätigkeit auf der Homepage eingeloggt bleibt und ist standardmässig auf 2 Stunden gesetzt.

settings.yml

```
all:
  .settings:
    timeout: 7200 # Session Timeout: 2 hours
```

4 Control Panel Guide

Um einen Eintrag in den CRUD Modulen zu bearbeiten, kann mittels Doppelklick ein „Edit Fenster“ geöffnet werden, in welchem Änderungen des Eintrags vorgenommen werden können. Informationen, welche in den Feldern eingegeben werden, sind sehr wichtig, da bei Falscheingaben das Active Reservation Module (Ansteuerung der Devices via SNMP) zu Fehlverhalten führt.

4.1 Pod Kategorie

Name	Description	[Credit Point Cost/h]	Minimal reservation time [h]
Routing Switching	Routing Switching erlernen	1	4.00
Security	Security Rules erlernen	1	4.00
MPLS Network	erlernen von MPLS	2	6.00

Feldname	Info
Name	Bezeichnet die Pod Kategorie, welcher nur für den Administrator ersichtlich ist.
Description	Dieses Feld ist nur für den Administrator ersichtlich. Hier können auch temporäre Informationen eingetragen werden.
[Credit Points Cost/h]	Hier kann definiert werden, wie viele Credit Points eine Pod Kategorie pro Stunde kostet.
Minimal reservation time [h]	Hier kann die minimale Reservationszeit für Reservationen definiert werden.

4.2 Pod

Name	Description	Terminalserver	Is Active	Time between Reservation	Pod Category
Test Example Pod	Dies ist zur Verfügung gestellter Test Pod.	10.1.1.10	true	15	Routing Switching
Routing Pod 2	normal Routing Pod	10.1.1.12	true	30	Routing Switching
Security Pod 1	FW, alles was benötigt wird	10.1.1.13	true	30	Security

Feldname	Info
Name	Bezeichnet den Namen des Pods, welcher in der Combobox im Reservationskalender beim Administrator und Benutzer ersichtlich ist.
Description	Dieses Feld ist nur für den Administrator ersichtlich. Hier können auch temporäre Informationen eingetragen werden.
Terminalserver	Definiert den Terminalserver für den jeweiligen Pod.
Is Active	Falls der Pod deaktiviert wird, ist er nicht mehr für den User in der Reservation Combobox ersichtlich.
Time between Reservation	Definiert die Zeit, welche zwischen zwei Reservationen gewartet werden muss, damit in dieser Zeit Konfigurationen gespeichert werden können.
Pod Category	Definiert, von welcher Pod Kategorie der Pod abhängig ist.

4.3 Geräte Kategorie

Name	Description	Shortcut
ROUTER	r	r
L3SWITCH	s	s
ATMSWITCH	b	b

Feldname	Info
Name	Bezeichnet den Namen der Gerätekategorie.
Description	Dieses Feld ist nur für den Administrator ersichtlich. Hier können auch temporäre Informationen eingetragen werden.
Shortcut	Definiert, welches Bild für das Gerät gebraucht wird. Die Abkürzungen sind wie folgt: r = Router s = Switch a = ASA b = ATM i = IPS m = MDS p = FC h = Host n = Netapp Wenn keine Abkürzung gewählt wurde, wird das JBOD Bild gewählt.

4.4 Gerät

Name	Description	Is Activated	Amazing PDU Ip	Line	Pdu Port	Devicecategory	Pod
R1	SSH line 2011	true	10.1.1.8	33	A	ROUTER	Test Example Pod
R2	SSH line 2012	true	10.1.1.9	34	A	ROUTER	Test Example Pod
R3	SSH line 2013	true	10.1.1.8	35	B	ROUTER	Test Example Pod

Feldname	Info
Name	Bezeichnet den Namen des Geräts.
Description	In diesem Feld werden Informationen eingetragen, welche der Benutzer in der Description sieht. Vorzugsweise wird hier der Port des Gerätes eingetragen, worüber zugegriffen werden kann.
Is Activated	Um ein Device einem Pod hinzuzufügen, muss dieses Flag auf true sein. So wird verhindert, dass plötzlich ein kaputtes Device im produktiven Bereich aktiv geschaltet werden kann.
Amazing PDU Ip	Ip der Stromschiene.
Line	Interne Line, die gewählt wird.
Pdu Port	Steckplatz der Stromschiene (A-F).
Devicecategory	Definiert, von welcher Gerätekategorie das Gerät abhängig ist.

Pod	Bezeichnet den Namen des Pods
-----	-------------------------------

4.5 User

Wenn ein User gelöscht wird, werden automatisch alle Reservationen des Users mit gelöscht. Der Benutzer kann sich jedoch weiterhin am Control Panel anmelden, da er immer noch in der LDAP Datenbank vorhanden ist. Durch Doppelklick auf einen Benutzer kann der Administrator dem Benutzer Credit Points hinzufügen bzw. löschen.

Loginname	Name	Surname	Address	Zip	Town	Country	Phonenumber	Email	Credit Points
mkolb	kolb	Markus	Musterstrasse 4	88111	Wädenswil	switzerland	12412312	mkolb@hsr.ch	0.00
mmuster	muster	max	Musterstrasse 4	8001	Thalwil	switzerland	12412312	max@example.ch	2.75
jttest									16.00

Feldname	Info
Loginname	Bezeichnet den Loginnamen des Benutzers, welcher durch die LDAP DB definiert wird.
Credit Points	Dort wird angezeigt, wieviel Credit Points ein Benutzer besitzt.

4.6 Ldap Group

Name	LDAP CN
CCIE-RS	GRP-EDU-CCIE-RS
CCIE-Aktuell	GRP-EDU-CCIE-Aktuell
CCNA	GRP-EDU-CCNA

Feldname	Info
Name	Bezeichnet den Namen der LDAP Gruppe.
LDAP CN	Definiert den Common Name der Ldap Gruppe.

4.7 Group Policy

Pod Name	Ldap Group	has access	need Credit Points
Test Example Pod	CCIE-RS	true	true
Test Example Pod	CCIE-Aktuell	true	false
Test Example Pod	CCNA	false	false

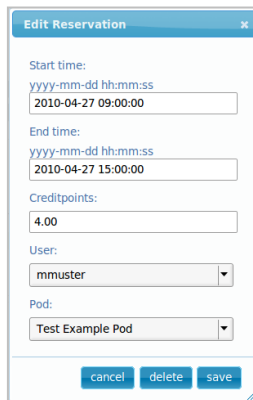
Feldname	Info
Pod Name	Bezeichnet den Namen des Pods.
Ldap Gruppe	Bezeichnet den Namen der Ldap Gruppe.
has access	Definiert, welche Gruppe auf welchen Pod Zugriff hat.
need Credit Points	Definiert, ob diese Gruppe Credit Points benötigt oder nicht.

4.8 Reservation für einen Benutzer erstellen

Start time	End time	Creditpoints	User	Pod
2010-04-27 09:00:00	2010-04-27 15:00:00	4.00	mmuster	Test Example Pod
2010-04-03 01:00:00	2010-04-03 08:00:00	0.00	mkolb	Test Example Pod

Feldname	Info
Start time	Wann eine Reservation beginnt (Startzeit).
End time	Bis wann eine Reservation dauert (Endzeit).
Creditpoints	Zeigt die Anzahl Credit Points eines Benutzers.
User	Bezeichnet den Benutzernamen.
Pod	Bezeichnet den Namen des Pods.

4.9 Reservation eines Benutzers bearbeiten



The screenshot shows a dialog box titled "Edit Reservation" with the following fields and values:

- Start time: 2010-04-27 09:00:00
- End time: 2010-04-27 15:00:00
- Creditpoints: 4.00
- User: mmuster
- Pod: Test Example Pod

Buttons: cancel, delete, save

Der Administrator muss die Startzeit und die Endzeit genau nach dem Date / Time Format yyyy-mm-dd hh:mm:ss angeben. Ansonsten werden die folgenden Default Werte verwendet: Start time / End time: 0000-00-00 00:00:00 und Credit Points: 0.00.

Die Credit Points müssen vom Administrator selbst erfasst und entsprechend beim Benutzer manuell hinzugezählt oder gelöscht werden.

Der Administrator kann hier einem oder mehreren Benutzern zur selben Zeit und auf den gleichen Pod mehrere Reservationen erstellen. Der Administrator ist für seine Angaben selber verantwortlich! Es wird nicht überprüft, ob die Angaben stimmen.

5 XML-RPC

5.1 Client Script (Radius Server)

Benötigte Files, welche im selben Ordner sein müssen:

- client.php
- kd_xmlrpc.php

Das Client Script ist auf dem Radius Server unter „/var/local/client.php“ zu finden. In den folgenden zwei Variablen wird angegeben, auf welchem Server das Server Script vorzufinden ist. In der zweiten Variablen ist der Ort definiert, wo das Script aufzufinden ist. Beispielaufruf: „192.168.223.147:8080/server.php“

```
$site = "192.168.223.147:8080";
```

```
$location = "/server.php";
```

5.2 Server Script (Controlpanel)

Benötigte Files, welche im selben Ordner sein müssen:

- server.php
- kd_xmlrpc.php
- web_service_api.php

Auf dem Server muss lediglich die MySQL Konfigurationen getätigt werden, welche im server.php File vorzufinden sind.

Variablen für den Zugriff zur MySQL DB

```
$dbserver= "localhost"; // Hostname vom MySQL Server
```

```
$db = "controlpanel"; // Datenbankname
```

```
$dbuser = "root"; // DBuser
```

```
$dbpassword = "da3ruler"; // Passwort
```

6 Installation Testing Guide

In Symfony gibt es Doctrine Unit Tests, die mit einer Anbindung an eine Testdatenbank getestet werden.

Konfiguration für die Testdatenbank

```
$php symfony configure:database --name=doctrine --class=sfDoctrineDatabase --env=test "mysql:host=localhost;dbname=controlpanel_test" root da3ruler
```

Die Option "env=test" besagt, dass für die Tests nur die Testumgebung gebraucht wird und die Daten der eigentlichen Datenbank nicht verändert werden.

SQL Files laden

```
$ mysqladmin -uroot -pda3ruler create controlpanel_test
$ php symfony doctrine:insert-sql --env=test
```

Test Dokumentation

Version 1.0

Projekt:

Remote Lab Management System

Projektmitglieder:

Markus Kolb

Mirjam Schaffner

Betreuer:

Rolf Schärer

Beat Stettler

Revision

Version	Status	Datum	Beschreibung/Änderung	Autor
1.0rc01	In Bearbeitung	14.05.2010	Erstellen des Testdokuments	MS
1.0rc02	In Bearbeitung	18.05.2010	Eigene Tests erfasst	MS
1.0rc03	In Bearbeitung	20.05.2010	Alle Tests überarbeitet und ergänzt	MS
1.0rc04	In Bearbeitung	10.06.2010	System Tests hinzugefügt	MS
1.0rc05	In Bearbeitung	17.06.2010	Eigene Tests erfasst	MK
1.0rc06	Gegenlesen	17.06.2010	Review	MK
1.0	Close	17.06.2010	Version 1.0	MS

Inhaltsverzeichnis

1	Einführung	4
1.1	Zweck	4
1.2	Gültigkeitsbereich	4
1.3	Definitionen und Abkürzungen.....	4
1.4	Referenzen	4
2	Testdurchführung in Symfony	5
2.1	Functional Test - Allgemein	5
2.2	Unit Test - Allgemein.....	6
3	Functional Tests	8
3.1	contactusActionsTest	8
3.2	deviceActionsTest	8
3.3	devicecategoryActionsTest	8
3.4	homeActionsTest	8
3.5	labActionsTest	8
3.6	myprofileActionsTest	8
3.7	podActionsTest	8
3.8	podcategoryActionsTest	8
3.9	reservationActionsTest	9
4	Unit Tests.....	10
4.1	DevicecategoryTest	10
4.2	DeviceTest	10
4.3	LabTest	10
4.4	PodcategoryTest	10
4.5	PodTest	10
4.6	MyCalendarTest	10
4.7	ReservationAdminCheckerTest	10
4.8	ReservationGenerateJsonTest	11
4.9	ReservationMinimalTimeTest	11
4.10	ReservationOverlapperCheckerTest	11
4.11	ReservationTest	11
4.12	ReservationUserSaveTest	11
5	System Tests.....	12
6	Performance Test	21

1 Einführung

1.1 Zweck

Zuerst wird in diesem Dokument aufgezeigt, wie in Symfony Tests durchgeführt werden. Dann werden einige eigene Functional- und Unit-Tests aufgeführt und erläutert.

1.2 Gültigkeitsbereich

Die Gültigkeit dieses Dokuments gilt für die gesamte Projektdauer.

1.3 Definitionen und Abkürzungen

Das Glossar ist im separaten Dokument (Glossar.docx) zu finden.

1.4 Referenzen

Referenz	Quelle
UML 2 und Patterns angewendet – Objektorientierte Softwareentwicklung	Craig Larman Mitp-Verlag/Bonn

2 Testdurchführung in Symfony

Um in Symfony alle Tests (Unit- sowie Functional-Tests) automatisiert laufen zu lassen, müssen die Tests mit folgendem Befehl aufgerufen werden:

```
$ php symfony test:all
```

2.1 Functional Test - Allgemein

In Symfony werden Funktionale Test durch einen speziellen Browser ausgeführt. Dieser wird durch die `sfBrowser` Klasse implementiert. Es wird kein WebServer benötigt, da dieser Browser direkt mit der Applikation verbunden ist. Somit ist es möglich, auf alle Symfony Objekte vor und nach einer Anfrage zuzugreifen.

Methode	Beschreibung
<code>get()</code>	Holt eine URL.
<code>post()</code>	Etwas zu einer URL aufgeben.
<code>call()</code>	Ruft eine URL auf (gebraucht für PUT und DELETE Methoden):
<code>back()</code>	Geht eine Seite in der History zurück.
<code>forward()</code>	Geht eine Seite in der History vorwärts.
<code>reload()</code>	Aktualisiert die aktuelle Seite.
<code>click()</code>	Klickt auf einen Link oder einen Button.
<code>select()</code>	Selektiert einen Radiobutton oder eine Checkbox.
<code>deselect()</code>	Deselektiert einen Radiobutton oder eine Checkbox.
<code>restart()</code>	Startet den Browser neu.

Beispiel

```
$browser = new sfBrowser();
$browser->
  get('/')->
  click('Design')->
  get('/category/programming?page=2')->
  get('/category/programming', array('page' => 2))->
  post('search', array('keywords' => 'php'))
;
```

Tests laufen innerhalb eines Test-Block-Inhalts ab. Dieser Test-Block-Inhalt beginnt mit: `with('TESTER NAME')->begin()` und endet mit: `end()`:

Beispiel

```
$browser->
  with('request')->begin()->
  isParameter('module', 'category')->
  isParameter('action', 'index')->
  end()
;
```


Der Request Tester

Methode	Beschreibung
<code>isParameter()</code>	Überprüft einen Request-Parameter-Wert.
<code>isFormat()</code>	Überprüft das Format eines Requests.
<code>isMethod()</code>	Überprüft die Methode.
<code>hasCookie()</code>	Überprüft, ob der Request ein Cookie mit dem gegebenen Namen hat.
<code>isCookie()</code>	Überprüft den Wert eines Cookies.

Der Response Tester

Methode	Beschreibung
<code>checkElement()</code>	Überprüft, ob ein Response CSS Selektor mit den Kriterien übereinstimmt.
<code>checkForm()</code>	Überprüft ein sfForm Form-Objekt.
<code>debug()</code>	Prints the response output to ease debug
<code>matches()</code>	Testet eine Response gegen eine Regular Expression.
<code>isHeader()</code>	Überprüft den Wert eines Headers.
<code>isStatusCode()</code>	Überprüft den Response-Status-Code.
<code>isRedirected()</code>	Überprüft, ob die aktuelle Response eine Umleitung ist.
<code>isValid()</code>	Überprüft, ob eine Response eine gut gestaltete XML ist.

Functional Tests ausführen

```
$ php test/functional/backend/categoryActionsTest.php
```

oder:

```
$ php symfony test:functional backend categoryActions
```

Automatisiert:

```
$ php symfony test:functional backend
```

Debugging Functional Tests

```
$browser->with('response')->debug();
```

2.2 Unit Test - Allgemein

Symfony benützt die lime Library, jedoch kann ebenso eine andere Library wie PHPUnit verwendet werden.

Lime: Die Tests beginnen immer mit demselben Code. Zuerst wird das unit.php bootstrap File eingebunden, das einige Dinge initialisiert. Dann wird ein neues lime_test Objekt erstellt und die Anzahl Tests, die geplant sind, als Argument übergeben.

Methode	Beschreibung
<code>ok(\$test)</code>	Testet eine Bedingung und besteht den Test, wenn „true“ zurück gegeben wird.

<code>is(\$value1, \$value2)</code>	Vergleicht zwei Werte und der Test ist richtig, wenn diese beiden Werte gleich sind (==).
<code>isnt(\$value1, \$value2)</code>	Vergleicht zwei Werte und der Test ist richtig, wenn diese beiden Werte ungleich sind.
<code>like(\$string, \$regex)</code>	Testet einen String gegen eine Regular Expression.
<code>unlike(\$string, \$regex)</code>	Überprüft, dass ein String nicht mit einer Regular Expression übereinstimmt.
<code>is_deeply(\$array1, \$array2)</code>	Überprüft, dass zwei Arrays dieselben Werte haben.

Es gibt mehrere verschiedene Methoden um etwas zu testen, da die Fehlermeldungen bei einigen Methoden aussagekräftiger sind. Sonst könnte auch nur die „ok“ Methode verwendet werden.

Weitere Methoden sind:

Methoden	Beschreibung
<code>fail()</code>	Schlägt immer fehl; ist gut, um Exceptions zu testen.
<code>pass()</code>	Besteht den Test immer; ist gut, um Exceptions zu testen.
<code>skip(\$msg, \$nb_tests)</code>	Zählt als <code>\$nb_tests</code> Test; ist gut für Zustand-Tests zu überprüfen.
<code>todo()</code>	Zählt als einen Test; gut, um auf Tests hinzuweisen, die noch geschrieben werden müssen.

Alle Unit Tests werden im Verzeichnis `test/unit` gespeichert. Die Tests werden zuerst mit dem Namen der Klasse benannt und das Wort `Test` wird diesem Namen angefügt.

Unit Tests ausführen

`$ php symfony test:unit Testname`

oder:

`$ php test/unit/model/Testname.php`

Automatisiert

`$ php symfony test:unit`

3 Functional Tests

3.1 contactusActionsTest

Testbeschreibung	Resultat
Testet die Site „Contact us“, ob die URL mit dem Modul „contactus“ und der Action „index“ übereinstimmen.	OK

3.2 deviceActionsTest

Testbeschreibung	Resultat
Testet die Site „Device“, ob die URL mit dem Modul „device“ und der Action „index“ übereinstimmen.	OK

3.3 devicecategoryActionsTest

Testbeschreibung	Resultat
Testet die Site „Devicecategory“, ob die URL mit dem Modul „devicecategory“ und der Action „index“ übereinstimmen.	OK

3.4 homeActionsTest

Testbeschreibung	Resultat
Testet die Homepage, ob die URL mit dem Modul „home“ und der Action „index“ übereinstimmen.	OK

3.5 labActionsTest

Testbeschreibung	Resultat
Testet die Site „Lab“, ob die URL mit dem Modul „lab“ und der Action „index“ übereinstimmen.	OK

3.6 myprofileActionsTest

Testbeschreibung	Resultat
Testet die Site „My Profile“, ob die URL mit dem Modul „myprofile“ und der Action „index“ übereinstimmen.	OK

3.7 podActionsTest

Testbeschreibung	Resultat
Testet die Site „Pod“, ob die URL mit dem Modul „pod“ und der Action „index“ übereinstimmen.	OK

3.8 podcategoryActionsTest

Testbeschreibung	Resultat
Testet die Site „Podcategory“, ob die URL mit dem Modul „podcategory“ und der Action „index“ übereinstimmen.	OK

3.9 reservationActionsTest

Testbeschreibung	Resultat
Testet die Site „Reservation“, ob die URL mit dem Modul „reservation“ und der Action „index“ übereinstimmen.	OK

reservationDatatableActionsTest

Testbeschreibung	Resultat
Testet die Site „Reservation“ des Administrators, ob die URL mit dem Modul „reservationDatatable“ und der Action „index“ übereinstimmen.	OK

```

contactusActionsTest.....ok
deviceActionsTest.....ok
devicecategoryActionsTest.....ok
homeActionsTest.....ok
labActionsTest.....ok
podActionsTest.....ok
podcategoryActionsTest.....ok
reservationActionsTest.....ok
reservationDatatableActionsTest.....ok
All tests successful.
Files=9, Tests=18
root@ubuntu:/home/sfprojects/controlpanelV2# s
    
```

Die Functional Tests haben wir vor der LDAP Anbindung implementiert und alle Tests liefen ohne Fehler ab. Da wir später das Login implementierten und eine LDAP Anbindung haben, konnten wir die Functional Test nicht weiter führen. Aus diesem Grund haben wir uns entschieden, anstelle der Functional Tests System Tests durchzuführen.

4 Unit Tests

4.1 DevicecategoryTest

Testbeschreibung	Resultat
Testet ob eine Gerätekategorie in die Testdatenbank gespeichert wird. Es wird überprüft, ob die ID, der Name und die Beschreibung übereinstimmen.	OK

4.2 DeviceTest

Testbeschreibung	Resultat
Testet ob ein Gerät in die Testdatenbank gespeichert wird. Es wird überprüft, ob die ID, der Name und die Beschreibung übereinstimmen.	OK

4.3 LabTest

Testbeschreibung	Resultat
Testet ob ein Lab in die Testdatenbank gespeichert wird. Es wird überprüft, ob die ID, der Name und die Beschreibung übereinstimmen. Anschliessend wird das Lab wieder gelöscht.	OK

4.4 PodcategoryTest

Testbeschreibung	Resultat
Testet ob eine Podkategorie in die Testdatenbank gespeichert wird. Es wird überprüft, ob die ID, der Name und die Beschreibung übereinstimmen.	OK

4.5 PodTest

Testbeschreibung	Resultat
Testet ob ein Pod in die Testdatenbank gespeichert wird. Es wird überprüft, ob die ID, der Name und die Beschreibung übereinstimmen.	OK

4.6 MyCalendarTest

Testbeschreibung	Resultat
Testet, ob die Generierung der Reservationen des iCal File funktioniert.	OK

4.7 ReservationAdminCheckerTest

Testbeschreibung	Resultat
Testet, ob die Reservation von einem Benutzer oder Administrator ist und korrekt abgespeichert wird.	OK

4.8 ReservationGenerateJsonTest

Testbeschreibung	Resultat
Testet die Generierung der Reservationen, welche als JSON Format erstellt werden.	OK

4.9 ReservationMinimalTimeTest

Testbeschreibung	Resultat
Überprüft, ob die minimale Reservationszeit korrekt eingetragen wird.	OK

4.10 ReservationOverlapperCheckerTest

Testbeschreibung	Resultat
Überprüft, ob bereits eine Reservation in der angegebenen Zeit vorhanden ist.	OK

4.11 ReservationTest

Testbeschreibung	Resultat
Überprüft allgemeine Reservationstests.	OK

4.12 ReservationUserSaveTest

Testbeschreibung	Resultat
Überprüft die Speicherung von Benutzer Reservationen.	OK

```

root@ubuntu:/home/sfprojects/controlpanelV2# php symfony test:unit
model/DeviceTest.....ok
model/DevicecategoryTest.....ok
model/LabTest.....ok
model/MyCalendarTest.....ok
model/PodTest.....ok
model/PodcategoryTest.....ok
model/ReservationAdminCheckerTest.....ok
model/ReservationGenerateJsonTest.....ok
model/ReservationMinimalTimeTest.....ok
model/ReservationOverlapperCheckerTest.....ok
model/ReservationTest.....ok
model/ReservationUserSaveTest.....ok
All tests successful.
Files=12, Tests=32
root@ubuntu:/home/sfprojects/controlpanelV2#
    
```

5 System Tests

Durch die System Tests wird unser Control Panel auf verschiedene Funktionalitäten getestet. Wir zählen verschiedene Testfälle eines Benutzers und des Administrators auf.

Name	ST 1 Benutzer registrieren (schlägt fehl)
Beschreibung	Benutzer registriert sich am Control Panel. Es wird keine richtige E-Mail angegeben.
Erwartetes Resultat	Eine Meldung wird über dem E-Mail Feld angezeigt: „E-Mail is inva- lide“. Alle benötigten Felder, die leer gelassen werden, werden mit „Required“ gekennzeichnet.
Testdaten	
Resultat	
Kommentar	OK
Testdatum	11.06.2010

Name	ST 2 Benutzer registrieren
Beschreibung	Benutzer registriert sich am Control Panel. Alle Felder werden kor- rekt ausgefüllt.
Erwartetes Resultat	Administrator erhält ein E-Mail mit den Angaben des Kunden.
Testdaten	
Resultat	OK
Kommentar	
Testdatum	11.06.2010

Name	ST3 Benutzer anmelden schlägt fehl
Beschreibung	Benutzer meldet sich auf der Login-Seite des Control Panels an.
Erwartetes Resultat	Fehlermeldung wird angezeigt, dass Benutzername oder Passwort nicht stimmen.
Testdaten	Benutzername: peter Passwort: 5678
Resultat	OK
Kommentar	
Testdatum	11.06.2010

Name	ST 4 Benutzer anmelden
Beschreibung	Benutzer meldet sich auf der Login-Seite des Control Panels an.
Erwartetes Resultat	Benutzer gelangt auf die Startseite des Control Panels.

Testdaten	Benutzername: pmuster Passwort: 1234
Resultat	OK
Kommentar	
Testdatum	11.06.2010

Name	ST 5 Reservation zum selben Zeitpunkt wie die eines anderen Benutzers erstellen
Beschreibung	Benutzer erstellt eine Reservation zu der Zeit, wo es schon eine vorhandene Reservation dieses Pods eines anderen Benutzers hat.
Erwartetes Resultat	Fehlermeldung wird angegeben: „Error occurred: there was a overlapping reservation your reservation was not saved!“
Testdaten	20 Credit Points
Resultat	OK
Kommentar	
Testdatum	11.06.2010

Name	ST 6 Reservation in der Vergangenheit erstellen
Beschreibung	Benutzer erstellt eine Reservation in der Vergangenheit.
Erwartetes Resultat	Fehlermeldung wird angegeben: „reservation starts in the past“
Testdaten	20 Credit Points
Resultat	OK
Kommentar	
Testdatum	11.06.2010

Name	ST 7 Reservation mit ungenügenden Credit Points erstellen
Beschreibung	Benutzer erstellt eine Reservation, hat jedoch zu wenig Credit Points.
Erwartetes Resultat	Fehlermeldung wird ausgegeben: „not enough credit points“
Testdaten	0 Credit Points
Resultat	OK
Kommentar	
Testdatum	11.06.2010

Name	ST 8 Reservation auf unberechtigten Pod erstellen
Beschreibung	Benutzer will eine Reservation erstellen. In dieser Zeit entnimmt ihm der Administrator die Berechtigung auf den Pod.
Erwartetes Resultat	Fehlermeldung wird ausgegeben „no access to pod“.
Testdaten	20 Credit Points
Resultat	OK
Kommentar	
Testdatum	11.06.2010

Name	ST 9 Vorhandene Reservation eines anderen Benutzers bearbeiten
Beschreibung	Benutzer will eine Reservation eines anderen Benutzers mittels URL-Befehl bearbeiten.
Erwartetes Resultat	Fehlermeldung wird ausgegeben " no permission to this reservation".
Testdaten	rid: 4, pid: 2, fromat:1275984000000, toat:1275985800000
Resultat	OK
Kommentar	
Testdatum	11.06.2010

Name	ST 10 Reservation kann nicht mehr bearbeitet werden
Beschreibung	Benutzer will seine Reservation bearbeiten, jedoch hat der Administrator in derselben Zeit die Reservation dieses Benutzers gelöscht.
Erwartetes Resultat	Fehlermeldung wird ausgegeben „no reservation found“
Testdaten	
Resultat	OK
Kommentar	
Testdatum	11.06.2010

Name	ST 11 Reservation erstellen
Beschreibung	Benutzer erstellt eine Reservation.
Erwartetes Resultat	Reservation wird erstellt.
Testdaten	20 Credit Points
Resultat	OK
Kommentar	

Testdatum	11.06.2010
-----------	------------

Name	ST 12 Eigne Reservation ansehen
Beschreibung	Benutzer geht zum Modul My Calendar.
Erwartetes Resultat	Sein Reservationskalender wird ihm mit den aktuellen Reservationen angezeigt.
Testdaten	-
Resultat	OK
Kommentar	
Testdatum	11.06.2010

Name	ST 13 Eigne Reservationen herunterladen
Beschreibung	Benutzer geht zum Modul My Calendar und lädt die Einträge seines Reservationskalenders herunter.
Erwartetes Resultat	Seine Reservationseinträge werden auf einem anderen Kalender auf einem anderen Gerät angezeigt.
Testdaten	-
Resultat	OK
Kommentar	
Testdatum	11.06.2010

Name	ST 14 Benutzerinformationen bearbeiten
Beschreibung	Benutzer geht zum Modul My Profile und ändert seine Daten.
Erwartetes Resultat	Die Daten werden mit den aktuellen, geänderten Informationen in der Datenbank gespeichert.
Testdaten	-
Resultat	OK
Kommentar	
Testdatum	11.06.2010

Name	ST 15 Gerät einschalten
Beschreibung	Benutzer geht zum Modul Active Reservation und schaltet ein Gerät ein. Dazu klickt der Benutzer auf das Bild des Geräts.
Erwartetes Resultat	Das Bild wird aktiv und ist connected.
Testdaten	-

Resultat	OK
Kommentar	
Testdatum	11.06.2010

Name	ST 16 Line clear
Beschreibung	Benutzer geht zum Modul Active Reservation und trennt die Verbindung.
Erwartetes Resultat	Das Bild wird deaktiviert und die Verbindung ist getrennt.
Testdaten	-
Resultat	OK
Kommentar	
Testdatum	11.06.2010

Name	ST 17 Kontaktformular mit einem Kommentar senden
Beschreibung	Benutzer geht zum Modul Contact us und sendet dem Administrator einen Kommentar.
Erwartetes Resultat	Der Administrator erhält ein E-Mail mit den Angaben des Benutzers
Testdaten	-
Resultat	OK
Kommentar	
Testdatum	11.06.2010

Name	ST 18 Problem beschreiben und senden
Beschreibung	Benutzer geht zum Modul Contact us und sendet dem Administrator das Formular mit dem beschriebenen Problem.
Erwartetes Resultat	Der Administrator erhält ein E-Mail mit dem beschriebenen Problem des Benutzers und dessen Angaben.
Testdaten	
Resultat	OK
Kommentar	
Testdatum	11.06.2010

Name	ST 19 Credit Points kaufen
Beschreibung	Benutzer geht zum Modul Contact us und fügt eine Anzahl Credit Points ein.

Erwartetes Resultat	Der Administrator erhält ein E-Mail mit der Anzahl Credit Points, die der Benutzer kaufen möchte und dessen Angaben.
Testdaten	
Resultat	OK
Kommentar	
Testdatum	11.06.2010

Name	ST 20 Kontaktformular ohne Angaben senden
Beschreibung	Benutzer geht zum Modul Contact us und sendet dem Administrator ein leeres Formular.
Erwartetes Resultat	Alle benötigten Felder werden mit „Required“ gekennzeichnet.
Testdaten	
Resultat	OK
Kommentar	
Testdatum	11.06.2010

Name	ST 21 Benutzer abmelden
Beschreibung	Der Benutzer meldet sich vom Control Panel ab.
Erwartetes Resultat	Der Benutzer gelangt auf die Login Seite des Control Panels.
Testdaten	
Resultat	OK
Kommentar	
Testdatum	11.06.2010

Name	ST 22 Pod verwalten
Beschreibung	Der Administrator kann einen neuen Pod erstellen, einen vorhandenen Pod bearbeiten oder löschen.
Erwartetes Resultat	Pod hinzufügen: Ein Pod wird in der Datenbank gespeichert und im Control Panel richtig angezeigt. Pod bearbeiten: Die aktualisierten Daten werden in der Datenbank erfasst. Pod löschen: Der Pod wird in der Datenbank gelöscht und im Control Panel nicht mehr angezeigt.
Testdaten	
Resultat	OK
Kommentar	

Testdatum	11.06.2010
-----------	------------

Name	ST 23 Podkategorie verwalten
Beschreibung	Der Administrator kann eine neue Podkategorie erstellen, eine vorhandene Podkategorie bearbeiten oder löschen.
Erwartetes Resultat	Podkategorie hinzufügen: Eine Podkategorie wird in der Datenbank gespeichert und im Control Panel richtig angezeigt. Podkategorie bearbeiten: Die aktualisierten Daten werden in der Datenbank erfasst. Podkategorie löschen: Die Podkategorie wird in der Datenbank gelöscht und im Control Panel nicht mehr angezeigt.
Testdaten	
Resultat	OK
Kommentar	
Testdatum	11.06.2010

Name	ST 24 Gerät verwalten
Beschreibung	Der Administrator kann ein neues Gerät erstellen, ein vorhandenes Gerät bearbeiten oder löschen.
Erwartetes Resultat	Gerät hinzufügen: Ein Gerät wird in der Datenbank gespeichert und im Control Panel richtig angezeigt. Gerät bearbeiten: Die aktualisierten Daten werden in der Datenbank erfasst. Gerät löschen: Das Gerät wird in der Datenbank gelöscht und im Control Panel nicht mehr angezeigt.
Testdaten	
Resultat	OK
Kommentar	
Testdatum	11.06.2010

Name	ST 25 Gerätekatgorie verwalten
Beschreibung	Der Administrator kann eine neue Gerätekatgorie erstellen, eine vorhandene Gerätekatgorie bearbeiten oder löschen.
Erwartetes Resultat	Gerätekatgorie hinzufügen: Eine Gerätekatgorie wird in der Datenbank gespeichert und im Control Panel richtig angezeigt. Gerätekatgorie bearbeiten: Die aktualisierten Daten werden in der Datenbank erfasst. Gerätekatgorie löschen: Die Gerätekatgorie wird in der Datenbank gelöscht und im Control Panel nicht mehr angezeigt.
Testdaten	

Resultat	OK
Kommentar	
Testdatum	11.06.2010

Name	ST 26 Lab verwalten
Beschreibung	Der Administrator kann ein neues Lab erstellen, ein vorhandenes Lab bearbeiten oder löschen.
Erwartetes Resultat	Lab hinzufügen: Ein Lab wird in der Datenbank gespeichert und im Control Panel richtig angezeigt. Lab bearbeiten: Die aktualisierten Daten werden in der Datenbank erfasst. Lab löschen: Das Lab wird in der Datenbank gelöscht und im Control Panel nicht mehr angezeigt.
Testdaten	
Resultat	OK
Kommentar	
Testdatum	11.06.2010

Name	ST 27 Gruppenrecht verwalten
Beschreibung	Der Administrator kann ein neues Gruppenrecht erstellen, ein vorhandenes Gruppenrecht bearbeiten oder löschen.
Erwartetes Resultat	Gruppenrecht hinzufügen: Ein Gruppenrecht wird in der Datenbank gespeichert und im Control Panel richtig angezeigt. Gruppenrecht bearbeiten: Die aktualisierten Daten werden in der Datenbank erfasst. Gruppenrecht löschen: Das Gruppenrecht wird in der Datenbank gelöscht und im Control Panel nicht mehr angezeigt.
Testdaten	
Resultat	OK
Kommentar	
Testdatum	11.06.2010

Name	ST 28 Reservation verwalten
Beschreibung	Der Administrator kann eine neue Reservation für einen Benutzer erstellen, eine vorhandene Reservation bearbeiten oder löschen.

Erwartetes Resultat	Reservation hinzufügen: Eine Reservation wird in der Datenbank gespeichert und im Control Panel richtig angezeigt. Reservation bearbeiten: Die aktualisierten Daten werden in der Datenbank erfasst. Reservation löschen: Die Reservation wird in der Datenbank gelöscht und im Control Panel nicht mehr angezeigt. Die Anzahl Credit Points wird nicht verändert.
Testdaten	
Resultat	OK
Kommentar	
Testdatum	11.06.2010

Name	ST 29 LDAP Gruppe verwalten
Beschreibung	Der Administrator kann eine LDAP Gruppe bearbeiten oder löschen.
Erwartetes Resultat	LDAP Gruppe bearbeiten: Die aktualisierten Daten werden in der Datenbank erfasst. LDAP Gruppe löschen: Die LDAP Gruppe wird in der Datenbank gelöscht und im Control Panel nicht mehr angezeigt.
Testdaten	
Resultat	OK
Kommentar	
Testdatum	11.06.2010

Name	ST 30 Benutzer verwalten
Beschreibung	Der Administrator kann einen vorhandenen Benutzer bearbeiten oder löschen.
Erwartetes Resultat	Benutzer bearbeiten: Die aktualisierten Daten werden in der Datenbank erfasst. Benutzer löschen: Der Benutzer wird in der Datenbank gelöscht und im Control Panel nicht mehr angezeigt.
Testdaten	
Resultat	OK
Kommentar	
Testdatum	11.06.2010

6 Performance Test

Bei 1'000 Reservationen braucht das Control Panel 5,9 Sekunden und es werden 2,9 kByte Daten übertragen. Für 6'000 Reservationen werden 16,7 kByte Daten übertragen und es braucht 40,8 Sekunden. Wir haben damit gerechnet, dass das JSON Format etwas länger braucht, jedoch nicht mit einer so hohen Latenzzeit. Damit die 6'000 Reservationen abgerufen werden können, muss das php.ini File angepasst werden (`memory_limit=150mb`) und in der Action Klasse muss die Limite ebenfalls angepasst werden (`ini_set(„memory_limit“, “256M“)`). Um 10'000 Reservationen zu testen, müssten wir eine bessere Leistung unserer Hardware haben und das `memory_limit` nochmals erhöhen.

Technischer Bericht

Version 1.0

Projekt:

Remote Lab Management System

Projektmitglieder:

Markus Kolb

Mirjam Schaffner

Betreuer:

Rolf Schärer

Beat Stettler

Revision

Version	Status	Datum	Beschreibung/Änderung	Autor
1.0rc01	In Bearbeitung	15.06.2010	Erstellen des Technischen Berichts	MS
1.0rc02	Gegenlesen	16.06.2010	Review des Technischen Berichts	MK
1.0rc03	In Bearbeitung	17.06.2010	Ergänzung XML-RPC	MK
1.0rc04	In Bearbeitung	17.06.2010	Ergänzung Mock Objekt	MS
1.0rc05	In Bearbeitung	17.06.2010	Reservierungen ergänzt	MK
1.0	Close	17.06.2010	Version 1.0	MK

Inhaltsverzeichnis

1	Einführung	4
1.1	Zweck	4
1.2	Gültigkeitsbereich	4
1.3	Definitionen und Abkürzungen.....	4
1.4	Referenzen	4
2	Einführung und Übersicht.....	5
3	Ereignisse	6
3.1	Implementierung des Control Panels.....	6
3.1.1	Design des Control Panels.....	6
3.1.2	Plugin Databases.....	6
3.1.3	Routing	7
3.1.4	Authentifizierung am Control Panel	8
3.1.5	Reservationsverwaltung.....	8
3.1.6	MyCalendar	10
3.1.7	Lokalisierung und Internationalisierung.....	10
3.1.8	Kontaktformulare:	10
3.1.9	Active Reservation	10
3.2	XML-RPC.....	11
3.2.1	Radius Server	11
3.3	Unit- und Functionaltests.....	11
3.4	Erkenntnisse	11
4	Schlussfolgerung	12
4.1	Erreichte Ziele.....	12
4.2	Nicht erreichte Ziele.....	12
4.3	Ursachen.....	12
4.4	Vergleich mit anderen Lösungen	12
4.5	Ausblick	13

1 Einführung

1.1 Zweck

Dieses Dokument soll für einen Ingenieur irgendeiner Fachrichtung verständlich sein.

1.2 Gültigkeitsbereich

Die Gültigkeit dieses Dokuments bezieht sich auf den Verlauf des ganzen Projekts. Es ist jedoch wegen des iterativen Vorgehens gut möglich, dass über die gesamte Zeit noch gewisse Änderungen vorgenommen werden.

1.3 Definitionen und Abkürzungen

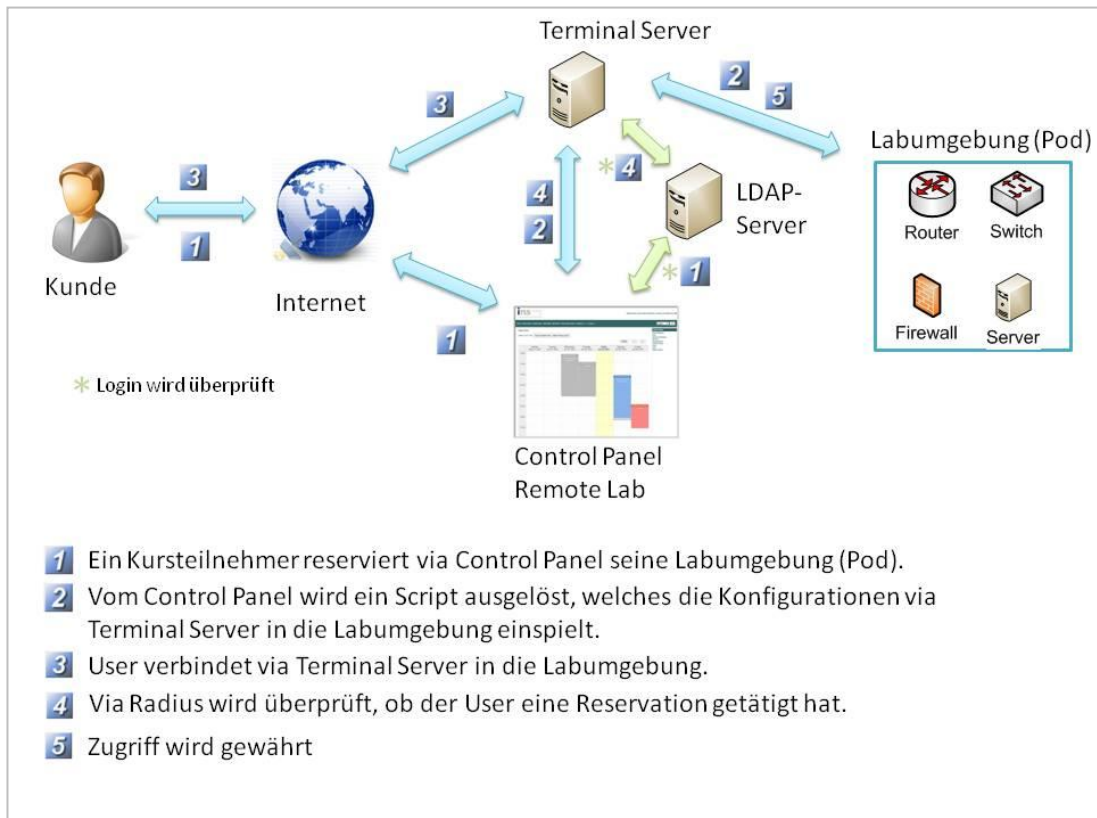
Das Glossar ist im separaten Dokument (Glossar.docx) zu finden.

1.4 Referenzen

Referenz	Quelle
Symfony	http://www.symfony-project.org/
ApPHP Calendar	http://www.apphp.com/php-calendar/index.php
Fullcalendar	http://arshaw.com/fullcalendar/
Weekcalendar	http://wiki.github.com/robmonie/jquery-week-calendar/
Steve templates	http://www.steves-templates.com/
Jobeet Tutorial	http://www.symfony-project.org/jobeeet/

2 Einführung und Übersicht

Ziel der Bachelorarbeit war es, ein Control Panel zu erstellen, damit es den Benutzern möglich ist, Pods zu reservieren. Ein Pod besteht aus mehreren Geräten wie z.B. Routern, Switches, Firewalls oder Unix Servern. Um Reservationen auf Pods zu machen, muss ein Benutzer am Control Panel angemeldet sein und über genügend Credit Points verfügen. So ist es dem Administrator möglich, die Reservationen eines Benutzers zu kontrollieren und der Benutzer kann den Pod nicht die ganze Zeit reservieren. Ein weiteres Ziel war es, über das Control Panel per SNMP Ansteuerung, die Geräte neu zu starten und SSH Verbindungen trennen zu können.



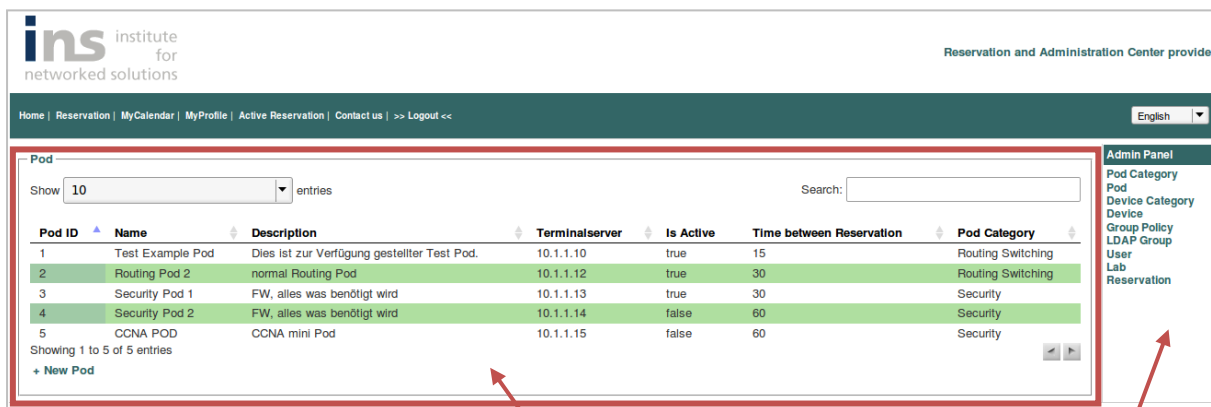
Das Control Panel haben wir unter Verwendung des PHP Frameworks Symfony erstellt, da dieses Framework sehr mächtig ist und vom Betreuer vorgegeben wurde.

3 Ereignisse

3.1 Implementierung des Control Panels

3.1.1 Design des Control Panels

Der Administrator sieht rechts ein Admin Panel, in dem die CRUD Module aufgeführt sind. In der Content View (Inhaltsanzeige) wird das ausgewählte CRUD Modul angezeigt. In einigen Modulen wie dem Active Reservation oder Contact us wird links für alle Benutzer zusätzlich noch ein Navigation Panel erscheinen. Dieses Navigation Panel wird jedoch nicht für alle Module verwendet und wird deshalb nicht in jeder Ansicht angezeigt.



Inhaltsanzeige pro Modul

Navigation für Administrator (CRUD Module)

3.1.2 Plugin Datatables

Da wir die Handhabung für den Administrator vereinfachen wollten und viel dynamischer implementiert haben wollten, haben wir uns entschieden, ein Plugin zu installieren. Wir haben die zwei Plugins jqGrid und Datatables verglichen:

Datatables	jqGrid
<p>Vorteile:</p> <ul style="list-style-type: none"> - Übersichtlich - Gut dokumentiert - Beinhaltet mehrere Funktionen - Viel schöner - Einfachere Handhabung - Nach Einarbeitung in jQuery war der Aufwand geringer für die Implementierung <p>Nachteile:</p> <ul style="list-style-type: none"> - Viel neu programmieren, was schon implementiert war - Einige Dateien können nicht mehr verwendet werden 	<p>Vorteile:</p> <ul style="list-style-type: none"> - Sieht gut aus <p>Nachteile:</p> <ul style="list-style-type: none"> - Sehr schlecht dokumentiert

Da das Plugin Datatables viel besser dokumentiert war, haben wir uns für dieses Plugin entschieden. Features wie Filterung, Suche, Anzahl Objekte pro Seite, nach Spalten sortieren und Seiten blättern stellte dieses Plugin schon zur Verfügung. Wir haben für die Implementierung des ersten CRUD Moduls ca. 25 Stunden gebraucht und für jedes weitere CRUD Modul ca. 3 Stunden.

3.1.3 Routing

Wir haben unterschiedliche Routings implementiert, eines für die Datatables, eines für die lokalisierte Homepage und ein weiteres für die restlichen Module. Das Routing für die restlichen Module generiert sieben sfDoctrineRoute Routen für: myprofile, myprofile_new, myprofile_create, myprofile_edit, myprofile_update, myprofile_delete und myprofile_show. Für die Lokalisierung braucht es die Variable `/:sf_culture` vor jeder URL.

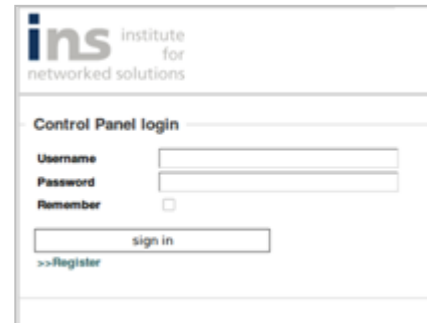
Routing für Lokalisierung	Lokalisierte Homepage Route
<pre># apps/backend/config/routing.yml myprofile: url: /:sf_culture/myprofile class: sfDoctrineRouteCollection options: { model: Myprofile, type: object } requirements: id: \d+ sf_method: [get]</pre>	<pre># apps/backend/config/routing.yml localized_homepage: url: /:sf_culture/ param: { module: home, action: index } requirements: sf_culture: (?:(fr en))</pre>

Das Routing für die CRUD Module ist statisch implementiert worden, das heisst der Routingname hinter dem Objektname (z.B: delete) muss mit dem Namen in der `action.class.php` Klasse des jeweiligen Moduls übereinstimmen.

Routing Datatables
<pre># apps/backend/config/routing.yml device_delete: url: /:sf_culture/device/delete.:sf_format param: { module: device, action: delete, sf_format: html } requirements: sf_format: html sf_culture: (?:(de en)) sf_method: [get] device_js: url: /:sf_culture/device/json param: { module: device, action: jsondata } requirements: sf_culture: (?:(de en)) device_save: url: /:sf_culture/device/save param: { module: device, action: save } requirements: sf_culture: (?:(de en)) device: url: /:sf_culture/device.:sf_format param: { module: device, action: index, sf_format: html } requirements: { sf_method: get }</pre>

3.1.4 Authentifizierung am Control Panel

Ein Benutzer kann sich am Control Panel nur anmelden, wenn er in einer Gruppe im Control Panel erfasst ist und im Active Directory existiert. Das Passwort wird nicht im Control Panel gespeichert, sondern nur in der LDAP Datenbank. Benutzerinformationen wie Loginname, E-Mail, Credit Points etc. werden im Control Panel gespeichert, da das Control Panel gemäss Wunsch kein Schreibrecht in der LDAP Datenbank erhält. Ist ein Kunde neu und hat noch kein Benutzerkonto, kann sich dieser registrieren.



Einschränkung der Implementierung:

Der Benutzer kann zurzeit nur einer LDAP Gruppe angehören. Im auth Modul wird in der action Klasse überprüft, welcher LDAP Gruppe der Benutzer angehört und wird als User Attribut gesetzt (siehe Ausblick).

```
public function setUserLdapgroupAndRedirect($ldapusergroups, $user){
    $ldapgroups = Doctrine::getTable('Ldapgroup')
    ->createQuery('a')
    ->execute();
    foreach ($ldapgroups as $ldapgroup){
        foreach ($ldapusergroups as $ldapusergroup){
            if($ldapusergroup == $ldapgroup->getLdapcn()){
                $user->setAttribute('userldapgroupid', $ldapgroup->getLdapgroupid());
                $user->setAuthenticated(true);
                $this->redirect('@homepage');
            }
        }
    }
}
```

3.1.5 Reservationsverwaltung

Der Kalender für den Benutzer ist interaktiv gestaltet und somit mit JavaScript implementiert worden.

Evaluation des Reservationskalenders

Kalender	Vor- und Nachteile
ApPHP Cal	Pro: - Statistiken sind verfügbar Contra: - Die Weekview und viele Funktionen sind kostenpflichtig
Fullcalendar	Pro: - Opensource - Viele Funktionen - Gute Dokumentation Contra: - Design nicht sehr ansprechend - Events über zwei Tage sind nicht übersichtlich
Weekcalendar	Pro: - Opensource - Dokumentation ok

	<ul style="list-style-type: none"> - Absolut überzeugendes Design - Gute Demo vorhanden <p>Contra:</p> <ul style="list-style-type: none"> - Events über 2 Tage muss selber implementiert werden.
Selbst Implementierung	<p>Pro:</p> <ul style="list-style-type: none"> - Kann gemäss Wunsch designed werden - Bugs werden schneller gefunden <p>Contra:</p> <ul style="list-style-type: none"> - Zeitaufwand gross - Massive Einarbeitung in jQuery nötig

Wir haben uns für den Weekcalendar entschieden, weil uns das Design am besten zugesagt hat und eine gute Demo vorhanden war. Der Weekcalendar setzt die jQuery Library voraus. Beim Speichern des Reservationskalenders gilt das Prinzip First Come First Serve (FIFO). Wenn zwei Benutzer gleichzeitig eine Reservation auf den selben Zeitpunkt erstellen möchten, wird die Reservation gespeichert, welche zuerst vom Server verarbeitet wird.

Der Weekcalendar wurde nicht weiterentwickelt und somit haben wir uns entschieden, die benötigten Features selbst zu implementieren. Wenn der zweite Teil der Reservation angewählt wird, kann nur noch der aktuelle Tag bearbeitet werden. Dies liegt daran, dass die ID der Reservation dieselbe ist, wie die dazugehörige Reservation am Vortag. Bei der Bearbeitung der Reservation wird die ID der zugehörigen Reservation des Vortags überschrieben. Reservationen, welche über zwei Tage dauern, haben zwei Reservationseinträge im Kalender, weil dies vom Plugin nicht unterstützt wird.

Im Reservationskalender werden nur Reservationen für die angezeigte Woche abgerufen. Das Positive daran ist, dass weniger Daten übertragen werden müssen.

The screenshot shows the 'Reservation and Administration Center provided by INS' interface. At the top, there is a navigation bar with links: Home | Reservation | MyCalendar | MyConfigs | MyProfile | Active Reservation | Contact us | >> Logout <<. A language dropdown is set to 'English' with an 'ok' button. Below the navigation, the 'Reservation' section has a dropdown menu for 'Select your Pod:' with the selected option 'Test Example Pod - Credit Points nee'. To the right is an 'Admin Panel' with a list of links: Pod Category, Pod, Device Category, Device, Group Policy, LDAP Group, User, Lab, and Reservation. The main area is a weekly calendar for the week of June 01, 2010, to June 06, 2010. The calendar grid shows time slots from 00:00 to 07:00. Reservations are shown as colored blocks: a grey block on Wednesday (Jun 02) from 12:15 am to 04:15 am; a grey block on Thursday (Jun 03) from 01:00 am to 04:15 am; a blue block on Saturday (Jun 05) from 02:15 am to 06:15 am; and a red block on Sunday (Jun 06) from 05:00 am to 07:15 am. A 'today' button and navigation arrows are located above the calendar grid.

3.1.6 MyCalendar

Im MyCalendar besteht die Möglichkeit, die eigenen Reservationen auf ein Gerät (z.B. einem PDA) herunterzuladen. Die iCal Funktion wurde gemäss RFC5545 implementiert. Die Implementierung ist im SAD Dokument beschrieben.

3.1.7 Lokalisierung und Internationalisierung




Das Control Panel wird in Deutsch und Englisch bereit gestellt. Symfony unterstützt für die Internationalisierung den ICU Standard i18n. Die Sprache kann per Combobox geändert werden. Dafür mussten wir ein sfFormExtraPlugin installieren. Die Übersetzungsdatei befindet sich unter apps/backend/i18n/de/messages.xml. (Weitere Informationen siehe SAD.)

3.1.8 Kontaktformulare:

Auf Wunsch des Betreuers haben wir kein externes Creditsystem angebungten, sondern die Credit Points können per Kontaktformular beim Administrator gekauft werden. Der Administrator erhält eine E-Mail und fügt dem Benutzer manuell die Credit Points hinzu. Für die Kosten kann eine Rechnung gestellt werden. Die Formulare für Probleme und Kommentare sind ebenfalls im Navigations Panel zu finden. Sind zu wenig Informationen eingegeben worden, wird das Feld mit „Required“ gekennzeichnet. (Implementation siehe SAD.)

3.1.9 Active Reservation

Im Active Reservation Modul kann ein Gerät mittels SNMP angesteuert werden und die SSH Verbindung getrennt werden. Der Administrator sieht alle Pods und alle Geräte dieser Pods. Der Benutzer sieht nur die Geräte der reservierten Pods während einer Reservation. (Implementation siehe SAD.)

Device	Name	Line Status	Device Info
	R1	not connected	SSH line 2011 Terminal Server IP: 10.1.1.10
	R2	not connected	SSH line 2012 Terminal Server IP: 10.1.1.10
	R3	not connected	SSH line 2013 Terminal Server IP: 10.1.1.10

Zurzeit wird jedes Mal, wenn ein Gerät gestartet/beendet wird, die Geräte Informationen neu generiert und herunter geladen. Dies wurde aus zeitlichen Gründen nicht perfekt implementiert. Beim Beenden eines Gerätes wird jedes Mal eine Sekunde gewartet, diese Verzögerung musste für die AmazingPdu (Stromschiene) eingebaut werden.

3.2 XML-RPC

Die Kommunikation zwischen RADIUS und unserem Control Panel findet mittels XML-RPC statt. RADIUS fragt per XML-RPC unser Reservationssystem ab, ob der Benutzer eine Reservation auf den gewünschten Pod zum momentanen Zeitpunkt hat. XML-RPC antwortet RADIUS mittels „ja“ oder „nein“. Das client.php Script befindet sich auf dem freeRadius Server.

3.2.1 Radius Server

In der Radius Konfiguration muss das Client.php Script eingebunden werden.

radiusd.conf

```
exec authphp {
    wait = yes
    program = "/var/local/client.php"
    input_pairs = request
    output_pairs = request
}
```

Im Client Script kann mittels der Funktion „trim(\$_SERVER['USER_NAME'], '\\")“ auf die benötigten Variablen zugegriffen werden.

Client.php

```
$username = trim($_SERVER['USER_NAME'], '\\");
$nas_ip = trim($_SERVER['NAS_IP_ADDRESS'], '\\");
$nas_port = trim($_SERVER['NAS_PORT'], '\\");
```

(Weitere Implementierung siehe SAD)

3.3 Unit- und Functionaltests

Um unseren Code zu testen, implementierten wir Unit- und Functionaltests. Im separaten Dokument Test/Testbericht.docx haben wir die Tests dokumentiert.

Unter test/fixtures sind die Fixtures enthalten, welche die Testdaten darstellen. In der Originaldatenbank unter data/fixtures gibt es auch Fixtures, welche in die Testumgebung kopiert werden können.

Daten laden

```
#test/bootstrap/Doctrine.php
Doctrine_Core::loadData(sfConfig::get('sf_test_dir').'/fixtures');
```

Wir haben eine Testdatenbank angebunden, da gemäss Symfony die Lime Library verwendet wird und in dieser Version die Mock Objekte noch nicht unterstützt werden.

3.4 Erkenntnisse

Wir würden von Anfang an die LDAP Implementierung anders angehen und erlauben, dass ein Benutzer mehreren LDAP Gruppen angehören kann. Einem Benutzer macht es viel mehr Spass mit interaktiven Objekten zu arbeiten als mit statischen. Daher haben wir mehr mit jQuery implementiert und uns nicht mehr zu fest auf PHP fixiert.

4 Schlussfolgerung

4.1 Erreichte Ziele

In der Bachelorarbeit haben wir alle „Must“-Use Cases erreicht und sogar noch einige „Nice to have“-Use Cases implementiert:

- Benutzer an- & abmelden
- Geräte verwalten
- Gerätekatgorie verwalten
- Pod verwalten
- Podkatgorie verwalten
- Reservationen verwalten
- Reservationen abfragen
- Reservationskalender herunterladen
- Benutzer verwalten
- Geräte neu starten
- Credit Points kaufen
- Credit Points definieren
- Bestimmte Gruppe Reservationsberechtigung pro Pod definieren
- Lab verwalten
- Internationalisierung
- Kontakt
- Registrierung

4.2 Nicht erreichte Ziele

- Gespeicherte Konfiguration hinauf- bzw. hinunter laden
- Konfigurationen des Pods speichern
- Bestehende Konfigurationen des Pods einspielen

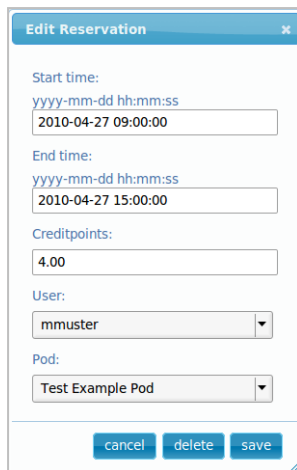
4.3 Ursachen

Hauptursache für die Nicht erreichten Ziele ist der Mangel an Zeit und die Priorität des Reservationssystems an sich, wie z.B. der Reservationskalender.

4.4 Vergleich mit anderen Lösungen

Im INS Institute for Networked Solutions ist bereits ein Reservationssystem in Betrieb. Das vorhandene Reservationssystem ist jedoch nicht zufrieden stellend, da bei einer Änderung verschiedene Komponenten angepasst werden müssen. Daher haben wir ein neues Reservationssystem implementiert, indem Änderungen mit weniger Aufwand gemacht werden können.

4.5 Ausblick



Das Reservationssystem kann mit weiteren Features erweitert werden. Im Modul Reservation, das nur der Administrator sieht, könnte für die Datenfelder „Startzeit“ und „Endzeit“ als Kalender dargestellt werden. Momentan muss der Administrator die Start- und Endzeit genau nach dem Format „yyyy-mm-dd hh:mm:ss“ eingeben. Durch den Kalender könnte dem Administrator die Arbeit erleichtert werden. Im selben Modul Reservation ist es möglich, dass der Administrator zum gleichen Zeitpunkt denselben Pod einem oder verschiedenen Benutzern gleichzeitig zur Verfügung stellt. In einer Erweiterung könnte dies mit einer entsprechenden Fehlermeldung behoben werden. Zudem werden die Credit Points beim Erstellen der Reservation dem Benutzer nicht abgezogen bzw. beim Löschen einer Reservation nicht hinzugefügt. Dies muss der Administrator manuell vornehmen,

was bei einer Erweiterung automatisch implementiert werden könnte.

Der Benutzer soll in einer Erweiterung im Control Panel in mehreren LDAP Gruppen hinzugefügt werden können.

Die Use Cases „Konfiguration für ein Gerät herauf laden“, „gespeicherte Konfiguration für Geräte oder Pod's herunter laden“, „Konfiguration des Pods speichern“ und „bestehende Konfiguration des Pods einspielen“ können zusätzlich implementiert werden.

Der Verlauf der Reservationen soll in einem Log-File gespeichert werden.

Beim Aufruf eines CRUD Objektes werden jedes Mal die Daten neu abgerufen und lokal zwischengespeichert. Wenn nun gleichzeitig ein anderer Administrator Datensätze ändert, werden die lokalen Datensätze des ersten Administrators nicht aktualisiert. Dafür müsste eine weitere Ajax Implementierung implementiert werden.

Ein User soll zu mehreren LDAP Gruppen im Control Panel angehören und so auf die jeweilige definierten Pods für die Gruppe zugreifen können.

Bei einer Erweiterung könnte ein externes Kreditsystem angebunden werden.

Dass die Performance bei vielen Reservationseinträgen in der Datatable Reservation erhalten bleibt, könnte ein Cronjob implementiert werden, welcher alle 3 Monate alten Reservationen archiviert.

Initial- und End-Konfigurationen könnten im Modul Lab erstellt und definiert werden. Dabei müssen die Konfigurationen auf den Server geladen werden.

Im Active Reservation Modul kann ein Caching implementiert werden.

Glossar

Version 1.1

Projekt:

Remote Lab Management System

Projektmitglieder:

Markus Kolb

Mirjam Schaffner

Betreuer:

Rolf Schärer

Beat Stettler

Revision				
Version	Status	Datum	Beschreibung/Änderung	Autor
1.0rc01	In Bearbeitung	26.02.2010	Erstellen des Glossars inkl. Dokument Abkürzungen	MK
1.0rc02	In Bearbeitung	11.03.2010	Abkürzungen ergänzt, ganzes Dokument erweitert	MS
1.0rc03	In Bearbeitung	16.03.2010	Weitere Abkürzungen ergänzt und einige Korrekturen vorgenommen.	MK
1.0rc04	Gegenlesen	30.03.2010	Ganzes Dokument gegengelesen	MK
1.0rc05	In Bearbeitung	09.06.2010	Weitere Abkürzungen ergänzt	MS
1.1	Close	18.06.2010	Version 1.1	MK

Inhaltsverzeichnis

1	Einführung	3
1.1	Zweck	3
1.2	Gültigkeitsbereich	3
1.3	Definitionen und Abkürzungen.....	3
2	Glossar	4

1 Einführung

1.1 Zweck

1.2 Gültigkeitsbereich

Die Gültigkeit dieses Dokuments gilt für die gesamte Projektdauer.

1.3 Definitionen und Abkürzungen

Name	Kürzel
Markus Kolb	MK
Mirjam Schaffner	MS

2 Glossar

Begriff	Beschreibung
AB	Dokument: Abstract / Kurzfassung
Ajax	Asynchronous JavaScript and XML: Bezeichnet ein Konzept der asynchronen Datenübertragung zwischen einem Browser und dem Server. ¹
Aktor	Kunde, Benutzer oder technische Einrichtung.
Aktueller Kursteilnehmer	Der Benutzer nimmt zurzeit an einem Kurs teil.
ApPHP Cal	ApPHP Cal ist ein Plugin für einen Kalender.
AS	Dokument: Anforderungsspezifikation
BS	Beat Stettler
Code Smells	Übel riechender Code → schlecht programmierter Code
Code-Freeze	Das Einfrieren des Quellcodes bedeutet innerhalb eines Software Projekts den Zeitpunkt, ab dem sich der Quellcode bis zur Abgabe nicht mehr ändern soll.
Config	Konfiguration eines Gerätes oder eines Pods.
Control Panel	Ein Control Panel ist ein Visualisierungsgerät, welches verschiedene Funktionen bereitstellt. In diesem Projekt sind die Hauptfunktionen, die Reservationen und die Verwaltung von Pods.
Credit Point	Punkte, die beim Administrator gekauft werden können. Der Benutzer kann im Gegenzug zu den Punkten einen Pod reservieren.
DA	Dokument: Domain Analyse
Datatables	Datatables ist ein Plugin, dass viele Features zur Verfügung stellt wie eine Suchfunktion, einen Filter und vieles mehr.
DBD	Dokument: Datenbank Design
DD	Dokument: Design Dokumentation
DMS	Dokument Management Summary
Domain Model	Das Domain Model bietet einen Überblick über alle Klassen und Beziehungen der Problem Domain.
DOM-Scripting	DHTML (Dynamisches HTML): Bezeichnung von Websites, die gegenüber normalen, statischen Seiten erweiterte Funktionalität oder Anzeigeeffekte aufweisen. ²
DRY- Konzept	„Don't repeat yourself“ Keinen doppelten Text/Code erfassen

¹ [http://de.wikipedia.org/wiki/Ajax_\(Programmierung\)](http://de.wikipedia.org/wiki/Ajax_(Programmierung))

² http://de.wikipedia.org/wiki/Dynamic_HTML

EEA	Dokument: Erklärung über eigenständige Arbeit
End-Konfiguration	Eine End-Konfiguration des Labs (Lösung), die vom Administrator bereitgestellt wird.
Funktionale Anforderungen	Funktionale Anforderungen beschreiben gewünschte Funktionalitäten eines Systems bzw. Produkts, dessen Daten oder Verhalten. ³
Gerät	Ist z.B. ein Router, Switch, Firewall, Server etc.
Geräte-Kategorie / Device Kategorie	Die Geräte Kategorie fasst eine Gruppe von Geräten zusammen. Bsp. Gruppe: Cisco Router 3600
Gespeicherte Konfiguration	Eine vom Benutzer gespeicherte Konfiguration des Labs.
GL	Dokument: Glossar
History	Auflistung abgelaufener und bereits zukünftig getätigter Reservationen eines Benutzers.
iCal	iCal ist ein Plugin, um Einträge eines Kalenders herunter zu laden.
IG	Dokument: Installation Guide
Initial-Konfiguration	Start Konfiguration des Labs, die der Administrator bereitstellt.
IV	Dokument: Inhaltsverzeichnis
Javascript	Skriptsprache, die hauptsächlich für das DOM-Scripting in Web-Browsern eingesetzt wird.
Jobeet Tutorial	Eine Anleitung um in Symfony ein Projekt zu machen.
JSON	Java Script Object Notation: Datenformat, zum Austausch von Daten zwischen Anwendungen.
Lab	Ein Lab ist eine praxisnahe Laborumgebung, welches ermöglicht, Weiterbildung nicht an produktiven Umgebungen zu erlernen.
LAMP	Kombinierter Einsatz von: Linux, Apache, MySQL und PHP.
Larman, Craig	Autor des Buches: UML 2 und Patterns angewendet – Objektorientierte Softwareentwicklung
LDAP	Lightweight Directory Access Protocol ist ein Anwendungsprotokoll. Es erlaubt Abfragen und die Modifikation von Informationen eines Verzeichnisdienstes über ein IP-Netzwerk. (Ist ähnlich zu einem Telefonbuch: Name suchen -> Telefonnummer herausfinden.) ⁴

3

http://www.anforderungsmanagement.ch/in_depth_vertiefung/funktionale_nicht_funktionale_anforderungen/

4

www.wikipedia.ch

LV	Dokument: Literaturverzeichnis
MK	Markus Kolb
MS	Mirjam Schaffner
My SQL	MySQL Server ist ein relationales Datenbankverwaltungssystem.
Nicht funktionale Anforderungen	Dies sind Anforderungen an die Qualität, in welcher die geforderte Funktionalität zu erbringen ist. ⁵
Nicht Kursteilnehmer	Externer Benutzer (Kunde) oder ehemaliger Kursteilnehmer, der einen Pod reservieren will, um ein Lab zu machen.
PBMK	Dokument: Persönlicher Bericht von Markus Kolb
PBMS	Dokument: Persönlicher Bericht von Mirjam Schaffner
PHP	Skriptsprache mit einer an C angelehnten Syntax, die hauptsächlich zur Erstellung dynamischer Websites verwendet wird.
POD	Ein Pod beinhaltet mehrere Geräte wie z. B. Router, Switch, Server, Firewall.
Pod-Kategorie	Eine Kategorie fasst eine exakt gleiche Gruppe von Pods zusammen.
PP	Dokument: Projektplan
RADIUS	Remote Authentication Dial-In User Service: Authentifizierungsdienst für sich einwählende Benutzer.
Reservation	Der Benutzer kann Pods reservieren.
Reservationskalender	Kalender, in dem alle Reservationen enthalten sind.
SAD	Dokument: Software Architecture Document
SSD	System Sequenz Diagramm: Dies ist ein Interaktionsdiagramm, das das Szenario eines Use Cases aufzeigt.
Symfony Framework	Web-Framework in PHP 5 geschrieben.
System Contracts / Systemoperationen	Dies sind Operationen, die das als Blackbox aufgefasste System über seine öffentliche Schnittstelle anbietet. ⁶
TB	Dokument: Technischer Bericht
TD	Dokument: Testdokumentation

5

http://www.anforderungsmanagement.ch/in_depth_vertiefung/funktionale_nicht_funktionale_anforderungen/

6

Larman, C. (2005). *UML 2 und Patterns angewendet*. mitp-Verlag.
Larman, C. *UML2 Patterns and Framework*.

ToDo	Arbeiten aufzählen, die anfallen. Dies wird durch das Sitzungsprotokoll aufgezeigt.
UC	Use Case: Anwendungsfall
Unified Process	Vorgehensmodell zur Softwareentwicklung
Use Case	Anwendungsfall: Alle möglichen Szenarien, die eintreten können, wenn ein Akteur versucht, ein bestimmtes Ziel zu erreichen.
User Story	Benutzergeschichte: Dies ist eine Software Anforderung an das Produkt.
Weekcalendar	Plugin, um einen Kalender zu implementieren.
XML-RPC	Extensible Markup Language Remote Procedure Call: Definition zum Methodenaufruf durch verteilte Systeme. ⁷
ZE	Dokument: Zeiterfassung
Credentials	Bei den Credentials handelt es sich um den Benutzernamen und das Passwort (Berechtigungsnachweis).

⁷ <http://de.wikipedia.org/wiki/XML-RPC>

Persönlicher Bericht

Markus Kolb

1 Einleitung

Da ich mich im Bereich Netzwerk weiterbilden und arbeiten möchte, war ich sofort von der Arbeit (Remote lab Management) begeistert. Seit 2 1/2 Jahren arbeite ich nun bei Hostpoint im 1st, 2nd, Abuse und in den Ferien in der Administrator Abteilung, darum konnte ich bereits mit Linux Grundkenntnisse die Arbeit beginnen.

2 Ablauf der Arbeit

Anfangs der Arbeit wurde viel Zeit in die Einarbeitung des PHP Framework Symfony investiert, da ich im Webentwicklungsbereich noch nie tätig war. Während dem Verlauf der Arbeit wurde mir schnell klar, wenn wir einen interaktiven Kalender gestalten möchten, müssen wir Javascript verwenden. Bei der Suche nach passenden Kalendern stiess ich auf verschiedene Kalender Implementierungen. Da das Design des Weekcalendars mir am meisten zugesagt hat und vor allem auch die meisten Features mit sich brachte, haben wir uns für das Plugin Weekcalendar entschieden. Worauf ich mich auch noch in die jQuery Javascript Library einarbeiten durfte. Die Implementierung des Kalenders war doch nicht ganz so einfach wie anfangs gedacht, es mussten Bugs gefixt und Weiterentwicklungen gemacht werden. Nach der Fertigstellung des Kalenders war ich doch ein wenig stolz auf meine Arbeit. Trotzdem war der Code sehr unübersichtlich, darum habe ich nach einem Design Pattern Ausschau gehalten, wonach ich nach kleinen Recherchen auf das Decorator Pattern gestossen bin. Nach der Implementierung ist nun sogar der Code übersichtlich.

Anfangs implementierten wir die CRUD Module gemäss Symfony, dadurch konnten wir viel generieren. Trotzdem war ich nicht zufrieden und schaute mich im Internet nach einem Tabellen Plugin um. Nach der Evaluation von 3 Plugins war ich völlig überzeugt vom Datatables Plugin, von welchem ich nach wenigen Stunden bereits einen Prototyp hatte. Beim Präsentieren war der Betreuer begeistert, worauf wir uns entschlossen, das Plugin für alle CRUD Module einzubauen.

Den Testpod erhielten wir in der Woche 15, worauf ich mit der Implementierung der Ansteuerung der Geräte anfangen konnte. Nach zwei Tagen verzweifelt versuchen beim Apache das SNMP Module zu aktivieren, haben wir ein Meeting mit unserem Betreuer vereinbart. Beim Meeting kam heraus, dass wir ein Missverständnis in der dritten oder vierten Woche hatten. Er teilte uns mündlich mit, dass wir lamp (linux, apache, mysql und php) verwenden sollen, worauf wir Lampp (Xampp for Linux) verstanden haben. Fazit war schlussendlich, dass Lampp nicht gleich lamp ist und bei Xampp for Linux kein SNMP Modul aktiviert werden kann. Somit konnte ich leider nicht mehr sehr viel Zeit in die Implementierung stecken, da wir viel Dokumentationsarbeit nachholen mussten.

3 Nachforschungen

Es wurden viele Nachforschungen betrieben. Ich habe mich in den Bereich PHP, Symfony Framework, Doctrine ORM Mapping, Javascript, jQuery Library eingearbeitet. Beim Designen der Datenbank und Anwendung des Apache, konnte ich meinen Horizont erweitern. Leider sind wir nicht mehr dazu gekommen, was eigentlich meine grösste Freude an der Arbeit gewesen wäre, ist das herunter-/herauf laden der Konfigurationen. Jedoch erhoffe ich, dass ich weiterhin Weiterentwicklungen am Tool tätigen darf.

4 Teamarbeit

Da ich dieses Semester nur ein Modul besucht habe, konnte ich viel Zeit in die Arbeit stecken. Die Teamharmonie war nicht immer perfekt, trotzdem konnten wir beide über unser Schatten springen und unsere Arbeit erledigen. Was mir sehr wichtig ist und am Herzen liegt, ist wenn

eine Person an einem Problem ansteht, zuerst mal eine richtige Fehleranalyse tätigt, bevor eine Frage gestellt wird. Dies musste ich anfangs meiner Bachelorpartnerin erst beibringen, was schlussendlich auch hervorragend geklappt hat.

Persönlicher Bericht

Mirjam Schaffner

1 Einleitung

Als Bachelorarbeit hatten wir drei Themen zur Auswahl. Unsere Wahl fiel auf das Thema, das Markus Kolb am meisten interessierte. Dieses Thema haben wir dann schlussendlich auch erhalten. Das Thema, welches mich am meisten interessierte, war unsere zweite Wahl.

Da er den CCNA Kurs bereits gemacht hatte, war er vom Thema sehr überzeugt und hatte bereits Vorkenntnisse. Ebenso kannte er die Betreuer etwas besser. Als ich die Aufgabenstellung las, war ich vom Thema nicht sehr überzeugt. Dies jedoch änderte sich schnell, als ich mehr über die Arbeit erfahren habe.

2 Ablauf der Arbeit

Zu aller erst habe ich sehr viel Zeit in die Einarbeitung des PHP Symfony Frameworks gesteckt, da ich dieses Framework noch nie gebraucht habe. Auch in der Programmiersprache PHP hatte ich bisher kaum Kenntnisse. Es machte mir sehr viel Spass und ich fand es interessant mit Symfony zu arbeiten. Ich habe mir dann ein Test Projekt aufgesetzt und selbst mit Symfony experimentiert und lernte so etwas schneller mit Symfony zu arbeiten. Neben dem Erlernen des Symfony Frameworks habe ich noch einige Zeit in die Dokumente gesteckt, da ich aus der Semesterarbeit gelernt hatte und wir uns damals zu wenig Zeit für die Dokumente nahmen.

Da Symfony so gut dokumentiert ist und Schritt für Schritt ein Projekt aufzeigt, war es sehr leicht, sich ein Projekt zu erstellen und die Module generieren zu lassen, sowie die Datenbank anzubinden. Es wurde schnell klar, wie ein Modul gegliedert ist und so haben wir am Anfang mit den CRUD Modulen begonnen. Es war sehr interessant zu sehen, wie unser Projekt von Tag zu Tag wächst. Aufgrund keinerlei Vorkenntnisse mit Symfony, hatte ich mit dem Routing ein wenig Mühe, da dies ein wenig komplex ist. Dadurch, dass ich mich mit dem Routing beschäftigt hatte, wurden mir die Struktur und die Zusammenhänge von Symfony klarer.

Leider war während der Bachelorarbeit mein Laptop für eine Woche in der Reparatur, da die Tastatur und das Touchpad defekt waren. So konnte ich an einem anderen Computer mehr an der Dokumentation arbeiten und informierte mich über den Methodenaufruf XML RPC. Ich habe dann die Library `kd_xmlrpc.php` gefunden und ein Tutorial, das ein Projekt aufzeigte. Ich versuchte dann, den Methodenaufruf zu implementieren und hatte noch einen Fehler, den Markus Kolb beheben konnte. Später holte ich die verloren gegangene Zeit wieder mit der Implementierung weiterer Punkte auf.

Ich informierte mich zudem über die Internationalisierung und implementiere diese sogleich. Im entsprechenden Plugin hatte es noch einen Fehler, den ich mit richtiger Recherche im Internet beheben konnte. Es machte mir sehr viel Spass am Projekt zu arbeiten und war mitten in der Implementierung des „Seiten blättern“ und habe dieses Feature in einem Modul fast fertig implementiert, als Markus Kolb das Plugin Datatables im Internet fand. Da er nicht zu stoppen war und das Plugin sehr viel Vorteile brachte (viele Features), haben wir das Plugin Datatables implementiert. Markus Kolb hat mit einem Modul angefangen und ich habe über ein Wochenende alle Module bis auf Group Policy implementiert. Zusätzlich auf Wunsch von Rolf Schärer habe ich das Modul `reservationDatatables` auch noch implementiert. JSON hatte ich schon in meiner Semesterarbeit verwendet und daher kannte ich es schon ein wenig und wusste, wie anfällig es war, wenn man z.B. eine Klammer vergisst.

Damit wir unseren Code auch testen können, habe ich mich in die Themen Functional- und Unit-Tests in Symfony eingearbeitet. Von mir wurde eine Test-Datenbank erstellt und die ersten Tests implementiert, damit wir auch einige Tests aufweisen konnten.

Während Markus Kolb den Reservationskalender implementierte und danach den Code „refactored“, implementierte ich das Modul „Benutzerinformationen bearbeiten“ und informierte mich mehr über die Kontaktformulare und das Mailen. Danach musste ich diese Klassen ein wenig „aufräumen“ und übersichtlicher gestalten.

Die VMWare bereitete mir sehr viele Probleme, da ich auf meinem Laptop Windows 7 installiert habe und ich dann plötzlich nicht mehr auf dem Windowssystem schreiben konnte. Ich verlor immer wieder Zeit, um den Laptop neu zu starten. Die VMWare machte uns allgemein noch zu schaffen, da diese plötzlich nicht mehr richtig funktionierte und wir das Projekt erneut vom SVN Server herunterladen mussten. Einmal übernahm ich dann die VMWare von Markus Kolb und habe leider zu spät bemerkt, dass ich unter seinem Namen meine Implementierung im SVN eingchecked habe (2-3 Mal passiert).

In der Woche 15 erhielten wir dann den Testpod und erfuhren, dass es am Anfang ein Missverständnis zwischen LAMP und LAMPP gab. Dieses Missverständnis kostete uns zwei Tage und da wir mit der Dokumentierung anfangen mussten, konnten wir nicht mehr sehr viel Zeit in die Implementierung des Moduls Active Reservation stecken.

In der letzten Woche hatten wir noch eine Sitzung mit Herrn Stettler, in der heraus kam, dass wir die Dokumentation überarbeiten mussten. Wir waren schon viel zu vertieft in unserer Arbeit und haben so viele Funktionen implementiert, dass wir z.T. einige Dinge vergessen hatten zu erwähnen, die unser Control Panel auch noch konnte. Das Produkt konnte in Wirklichkeit viel mehr, als wir in den Dokumenten festgehalten hatten. Für uns war alles so klar, dass wir nicht daran dachten, dass es für jemanden Externes nicht klar sein könnte.

3 Nachforschungen

Wir haben während unserer Bachelorarbeit sehr viele Nachforschungen gemacht. Ich habe mich im Bereich PHP, Symfony Framework, Doctrine ORM Mapping, LAMPP, JavaScript, XML-RPC, Internationalisierung, Mailing in Symfony, Databases, Functional- und Unit-Tests etc. eingearbeitet.

Leider hatten wir nicht mehr genug Zeit um weitere Features zu implementieren. Ich hätte gerne noch die Internationalisierung in den CRUD Modulen gemacht und die Konfigurationen hinauf- bzw. herunter zu laden implementiert. Es gibt noch einige Dinge mehr, die ich noch gerne implementiert hätte. Sobald man mit Nachforschungen anfängt, sieht man so viele Features, die man noch näher betrachten möchte und in das Projekt einbringen will.

4 Teamarbeit

Nebst der Bachelorarbeit habe ich noch zwei weitere Module besucht und musste einen Tag in der Woche arbeiten. Deshalb habe ich anfangs nicht mehr Zeit in die Bachelorarbeit investieren können und habe in den letzten zwei Wochen nur noch am Projekt gearbeitet. Da Markus Kolb nur von zu Hause aus arbeiten wollte, musste ich anfangs fast jeden Dienstag und nachher noch etwas mehr von Chur aus nach Wädenswil fahren. Er holte mich dafür vom Bahnhof ab und kochte uns zu Mittag. Die letzten drei Wochen arbeiteten wir dann in der Schule, was mir sehr entgegen kam. Ich wollte schon von Anfang an in der HSR arbeiten, kam aber Markus Kolb entgegen, da er mir einige persönliche Gründe aufzählte, warum er nicht an der Schule arbeiten möchte. Dass wir nicht auf der gleichen Wellenlänge waren, haben wir schon früh herausgefunden. Dies machte uns jedoch nicht sehr viel aus, da wir beide wussten, dass wir die Bachelorarbeit nur als Team meistern können.

Literaturverzeichnis

Version 1.0

Projekt:

Remote Lab Management System

Projektmitglieder:

Markus Kolb

Mirjam Schaffner

Betreuer:

Rolf Schärer

Beat Stettler

Revision				
Version	Status	Datum	Beschreibung/Änderung	Autor
1.0rc01	In Bearbeitung	26.02.2010	Erstellen des Literaturverzeichnisses	MS
1.0	Close	18.06.2010	Version 1.0	MK

Inhaltsverzeichnis

1	Einführung	3
1.1	Zweck	3
1.2	Gültigkeitsbereich	3
1.3	Definitionen und Abkürzungen.....	3
2	Literaturverzeichnis	4

1 Einführung

1.1 Zweck

Dieses Dokument soll dazu dienen, von allen Dokumenten der Bachelorarbeit die Literaturverzeichnisse aufzulisten und für Interessierte mehr Informationsquellen anzugeben.

1.2 Gültigkeitsbereich

Die Gültigkeit dieses Dokuments gilt für die gesamte Projektdauer.

1.3 Definitionen und Abkürzungen

Name	Kürzel
Markus Kolb	MK
Mirjam Schaffner	MS

2 Literaturverzeichnis

Thema	Beschreibung	Quelle
ApPHP Calendar	Plugin	http://www.apphp.com/php-calendar/index.php
Craig Larman	Mitp-Verlag/Bonn	UML 2 und Patterns angewendet – Objektorientierte Softwareentwicklung
Datatables	Plugin	http://datatables.net/
Free Radius		http://freeradius.org/
Fullcalendar	Plugin	http://arshaw.com/fullcalendar/
jQuery		http://jquery.com/
Ldap Bild Ressource		http://www.donotenter.com/cool/ucgraphics/large/index.htm
Logical View		http://www.symfony-project.org/jobeeet/1_4/Doctrine/en/04
Microsoft		http://technet.microsoft.com/dede/library/cc781821(WS.10).aspx
Orientation in Objects GmbH		http://www.oio.de/public/xml/xml-rpc.htm
PHP	PHP lernen	http://www.selfphp.de/
Steve templates		http://www.steves-templates.com/
Symfony	Framework MVC-Erklärung	http://www.symfony-project.org/ http://trac.symfony-project.org/wiki/Documentation/de_DE/book/1.0/02-Exploring-Symfony-s-Code
	Jobeeet Tutorial	http://www.symfony-project.org/jobeeet/1_4/
Weekcalendar	Plugin	http://wiki.github.com/robmonie/jquery-week-calendar/
XML-RPC	XML-RPC Beschreibung	http://www.oio.de/public/xml/xml-rpc.htm http://articles.sitepoint.com/article/own-web-service-php-xml-rpc
	Tutorial Wikipedia	http://de.wikipedia.org/wiki/XML-RPC
	K. Rhodes, XML-RPC vs. SOAP	http://weblog.masukomi.org/writings/xml-rpc-vs-soap.htm

Sitzungsprotokoll

Zeit: 10:00 – 11:00 Uhr
 Woche: SW01
 Ort: Baracke
 Anwesende: Markus Kolb, Mirjam Schaffner, Rolf Schärer

Traktanden

- Rückblick
- Fragen bezüglich Projekt
- Ziel vom Projekt
- Wünsche, Anliegen von Seite INS
- Ausblick: ToDo nächste Woche

Diskussionen / Beschlüsse

Thema	Resultat
Rückblick	<ul style="list-style-type: none"> - Einarbeiten Symfony - Zeitplan / Zeiterfassung erstellt - Architektur-Entwurf und Funktionen erstellt - SVN und Ordnerstruktur erstellt - Symfony Projekt erstellt - Dokumentvorlagen erstellt
Fragen bezüglich Projekt	<ul style="list-style-type: none"> - Vorgehensmodell ist RUP: Dokumente möglichst kurz und aussagekräftig halten, keine unnötigen Informationen erwähnen - Arbeitspakete und Zeitplan gemäss eigenem Ermessen definieren - Wichtige Funktionen des Projekts ist das Reservationensystem und ein „nice to have“ wären die Konfigurationen der Geräte herunter bzw. herauf zuladen - zur Zeit sind alle Benutzer im LDAP Verzeichnis -> für die Entwicklung noch nicht definiert, ob externe Benutzer in LDAP registriert werden oder nicht - mehrere Äste vom LDAP Verzeichnis (Admins sind in einem anderen Ast verzeichnet als Benutzer) - Zwischen den Reservationen genug Zeit einplanen (ca. 30 min), damit Schluss-Konfigurationen gespeichert werden können und so für neue Benutzer wieder freigeschalten werden - mehr Zeit in DB-Modell investieren als in GUI-Design (keine Wertlegung in GUI-Design, mehr in Funktionen)

Funktionen User	<ul style="list-style-type: none">- Reservationen verwalten- Reservationskalender- Historie der Benutzer anzeigen z.B. abgelaufene Reservationen, kommende Reservationen- Geräte neustarten, „reseten“- Konfiguration verwalten, herauf bzw. herunter laden- Über Credit-System „Credit-Points“ kaufen
Funktionen Admin	<ul style="list-style-type: none">- Pods verwalten- Geräte verwalten- Geräte an Pod hinzufügen / löschen- Reservationen verwalten- alle Reservationen auflisten lassen- Kategorien der Geräte festlegen (Standardkonfiguration pro Gerät)- Credits für Benutzer definieren- Kosten der Credits pro Pod pro Stunde festlegen
Funktionen Visitor	<ul style="list-style-type: none">- Homepage Info abrufen- Tutorial ansehen- Registrationsanfrage tätigen
Funktionen System	<ul style="list-style-type: none">- „Passwort - Recovery“ (Ursprungzustand des Systems wiederherstellen, wenn ein Passwort gesetzt wurde)
Ausblick	<ul style="list-style-type: none">- Wöchentlich ab 9. März 2010 Sitzung um 10.00 Uhr- Dokumente immer einen Tag vor Sitzung an Betreuer senden- Betreuer sendet uns Architekturübersicht über die Pods
ToDo	<ul style="list-style-type: none">- an Symfony Tutorial weiter arbeiten- Dokumente erstellen (Projektplan, Anforderungsspezifikation)- Meilensteine festlegen

Sitzungsprotokoll

Zeit: 14.30 – 15.20
 Woche: SW02
 Ort: Gebäude 1, Cafeteria
 Anwesende: Markus Kolb, Mirjam Schaffner, Rolf Schärer

Traktanden

- Rückblick
- Fragen bezüglich Projekt
- Ausblick: ToDo nächste Woche

Diskussionen / Beschlüsse

Thema	Resultat
Rückblick	<ul style="list-style-type: none"> - Einarbeiten Symfony - Projektplan & Anforderungsspezifikation - UseCase-Diagramm - Datenbank Schema
Fragen bezüglich Projekt	<ul style="list-style-type: none"> - Wichtig: Werden neue Kunden nun in der LDAP Datenbank erfasst? - Einschränkungen für das Projekt müssen wir mal noch genauer betrachten? (-> Welche Use Cases sind "must" Use Cases und welche "nice to have") - Code welcher schon vorhanden ist, bsp. für die Geräte Verwaltung etc... und die Authentifizierung bei der LDAP Datenbank. - Die Kursteilnehmer müssen keine Credit Points zahlen für die Pods? Macht es nicht mehr Sinn, wenn die Benutzer bereits Credit Points zur Verfügung haben und der Administrator falls nötig weitere freischalten kann? - Login / Passwort zum Testen - können Benutzer Configs der Pods und der Geräte herauf- bzw. herunter laden? - Credit System als Externer Actor: Über wen, wie wird dies gemacht? (Paypal, Saverpay, Google Checkout...)
Credits Points	<ul style="list-style-type: none"> - Credit Points müssen manuell vom Benutzer zugewiesen werden können. - Es muss ein Flag zur Verfügung stehen, dass der Benutzer keine Credit Points benötigt, um Pods zu reservieren. - Im Control Panel wird angezeigt, wieviel Credit Points ein Benutzer zur Verfügung hat und falls das Flag auf true ist, dies korrekt gekennzeichnet wird.

LDAP-DB	<ul style="list-style-type: none"> - In der LDAP-DB gibt es verschiedene Gruppen. Die aktuelle Gruppe (Teilnehmer am Kurs) hat Zugriff auf Pods. Sobald der Kurs vorbei ist, wird diese Kurs-Gruppe aus einer Obergruppe „aktueller Kurs“ entfernt. - Es gibt verschiedene Gruppen, die aktiv sind und Zugriff auf verschiedene Pods benötigen. Zudem benötigt jede Gruppe andere Pods (-> muss zugeteilt werden können pro Gruppe). Der Zugriff der Pods sind zudem für mehrere Gruppen möglich. - Authentisierung bei der LDAP DB kann über ein Symfony Module gelöst werden, welches wir erhalten. - Die Radius Schnittstelle fragt das "Reservationssystem" ab, ob der Benutzer auf das Device zugreifen darf oder nicht. (vermutlich wird dies über HTML realisiert, da Sicherstellung der Authentifizierung über die LDAP Datenbank geschieht.) Antwort JA/NEIN
User	<ul style="list-style-type: none"> - Benutzer muss alles können in Bezug auf Configs. Er muss Configs von Geräten sowie von Pods herunter bzw. herauf laden und speichern können. -> Diese Anforderung ist jedoch "ein nice to have", da das Reservationssystem (Reservierungen und Authentisierung) mehr Bedeutung hat für die Bachelorarbeit. Wichtig ist, dass vor allem noch der Arbeit, zusätzliche diese Funktionen eingebaut werden können.
Interface	<ul style="list-style-type: none"> - Login mittels XML-RPC Abfragen implementieren. - Interfacezustand merken (shut_down / no_shut_down). Interface-Namen können mittels Variablen ersetzt werden. Es sollen noch mehr Replacements getätigt werden können. - Abfrage der IP-Adresse -> Security Feature
Config	<p>Wenn eine Config herunter geladen wird, muss das System sich merken, von welchem Pod sie herunter geladen wurde.</p>
Ausblick	<ul style="list-style-type: none"> - Sitzungsprotokolle werden vom Betreuer bestätigt - erhalten Test-Login auf Lab - User in LDAP-DB oder lokal -> wird bis zur nächsten Sitzung geklärt - Installation in der HSR ist Teil der Bachelorarbeit. Es genügt nicht, nur eine CD abzugeben. Für die Installation erstellen wir eine Anleitung. - Die Applikation wird auf verschiedene Sprachen zur Verfügung gestellt. - Radius Schnittstelle zu Reservationssystem muss in dieser Arbeit beinhaltet sein.
ToDo	<ul style="list-style-type: none"> - Am Symfony Tutorial weiter arbeiten - Datenbank Schema verbessern / anpassen - zweites UseCase Diagramm erstellen, nur mit den wirklich benötigten UseCases. Es muss ersichtlich sein, welche Use Case must sind und welche „nice to have“. - Anforderungsspezifikation und Projektplan erweitern

Sitzungsprotokoll

Zeit: 10:00 – 11:00 Uhr
 Woche: SW03
 Ort: Baracke
 Anwesende: Markus Kolb, Mirjam Schaffner, Rolf Schärer

Traktanden

- Rückblick und aufgetauchte Fragen
- Besprechung der Dokumente Projektplan und Anforderungsspezifikation

Diskussionen / Beschlüsse

Thema	Resultat
Rückblick	<ul style="list-style-type: none"> - Einarbeiten Symfony Tutorial - Projektplan - Anforderungsspezifikation - Datenbank Schema
Fragen bezüglich Projekt	<ul style="list-style-type: none"> - Genauer Login Prozess? - Registration?
Authentifizierung	<ul style="list-style-type: none"> - Die Authentifizierung findet direkt über die LDAP-DB statt. - Benutzer können sich nicht selbst registrieren. Dies wird über den Admin erledigt (-> Anfrage an Admin bezüglich Kontoeinrichtung)
Anforderungs-Spezifikation	<ul style="list-style-type: none"> - Die Anforderungsspezifikation muss verständlicher geschrieben werden (Wörterklärungen), auch wenn das DRY-Prinzip dabei verletzt wird. - Use Cases ausarbeiten, erweitern
Reservation	<ul style="list-style-type: none"> - eine mindest Reservationszeit pro Pod Kategorie muss festgelegt werden können - die Reservation kann im Stunden-Schritt erweitert werden - im Reservationskalender darf nur die Reservation gesehen werden, nicht wer die Reservation getätigt hat
Admin	<ul style="list-style-type: none"> - muss die History der Reservationen jedes einzelnen Benutzers sehen, sowie eine gesamt Übersicht über alle Reservationen der Benutzer - muss die Puffer Zeit nach der Reservationen der jeweiligen Pod Kategorie selbst definieren, sodass alle Konfigurationen sauber gespeichert werden können.
Ausblick	<ul style="list-style-type: none"> - Einarbeiten Symfony Tutorial - Anforderungsspezifikation überarbeiten - Projektplan überarbeiten - Glossar vor nächster Sitzung mitschicken - Sitzungen jeweils donnerstags 10.00 Uhr - Linux VM auf Windows installieren und Projekt mit Linux entwickeln

Sitzungsprotokoll

Zeit: 10:00 – 11:00 Uhr
 Woche: SW04
 Ort: Baracke
 Anwesende: Markus Kolb, Mirjam Schaffner, Rolf Schärer

Traktanden

- Rückblick und aufgetauchte Fragen
- Besprechung der Dokumente Projektplan und Anforderungsspezifikation

Diskussionen / Beschlüsse

Thema	Resultat
Rückblick	<ul style="list-style-type: none"> - Einarbeiten Symfony Tutorial - Projektplan - Anforderungsspezifikation
Fragen bezüglich Projekt	- Woher wissen wir, welche die aktuelle CCIE Kursgruppe ist? Gibt es da eine genaue Definition?
Fragen allgemein	<ul style="list-style-type: none"> - Wer ist der Gegenleser unserer Bachelorarbeit? - Authentisierung bei der LDAP DB kann über ein Symfony Module gelöst werden, welches wir erhalten. -> wann erhalten wir dies?
Projektplan	<ul style="list-style-type: none"> - Dokument ist in Ordnung - Dokumentänderungen müssen offensichtlich sein z.B. wenn sich ein Meilenstein nach hinten verschiebt -> begründen
Anforderungsspezifikation	<ul style="list-style-type: none"> - CCIE-Kurs nicht erwähnen, soll für alle Kurse gelten - klar definieren, dass Credit Points abgezogen werden, wenn eine Reservation gemacht wird. - Administrator muss von jedem Benutzer die Reservationen löschen können, auch wenn diese am Laufen sind. - UC06: Kursteilnehmer können dies auch. - Der Administrator kann definieren, ob ein Benutzer das externe Credit-system sehen kann oder nicht. - UC12: Additional Information ergänzen. - Infrastruktur: Versionen der Komponenten aufführen - Aktuelle Kursgruppe bleibt immer die gleiche. Es ändert sich für das Reservationssystem nichts.
Glossar	- Ist soweit in Ordnung.
Ausblick	<ul style="list-style-type: none"> - XML-RPC in der Anforderungsspezifikation erfassen - Fokus wird aufs Produkt gesetzt, nicht auf die Doku - Gegenleser wird in den nächsten zwei Wochen bekannt sein

Sitzungsprotokoll

Zeit: 10:00 – 10:30 Uhr
Woche: SW05
Ort: Baracke
Anwesende: Markus Kolb, Mirjam Schaffner, Rolf Schärer

Traktanden

- Rückblick Domain Model
- Frage bezüglich Projekt

Diskussionen / Beschlüsse

Thema	Resultat
Rückblick	<ul style="list-style-type: none">- Domain Model- Dokument: SAD und Domain Analyse- erste Layout-Versuche- CSS Tutorial- VM komplett aufgesetzt
Fragen bezüglich Projekt	<ul style="list-style-type: none">- Wie arbeitet LDAP bezüglich der Überprüfung des Logins?- Layout: Gibt es Wünsche zum Layout? (Sind Grafiken schon vorhanden)
Layout	<ul style="list-style-type: none">- INS Logo verwenden
SAD	<ul style="list-style-type: none">- Systemübersicht anpassen- Activity Diagramm anpassen
Domain Analyse	<ul style="list-style-type: none">- Konzeptbeschreibung: PodCategory und Lab / UserLab nochmals überarbeiten,- Sequenzdiagramm anpassen
Ausblick	<ul style="list-style-type: none">- Herr Sommerlad ist als Gegenleser bestimmt- Logo INS ist vorhanden und können wir verwenden- nächste Sitzung 1. April 2010 13.00 Uhr- Dokumentation bereits Dienstagabend schicken
ToDo	<ul style="list-style-type: none">- Dokumente fertig stellen gemäss Meilensteine- An Prototyp weiter arbeiten

Sitzungsprotokoll

Zeit: 13:00 – 14:00 Uhr
 Woche: SW06
 Ort: Baracke
 Anwesende: Markus Kolb, Mirjam Schaffner, Rolf Schärer

Traktanden

- Rückblick
- Frage bezüglich Projekt

Diskussionen / Beschlüsse

Thema	Resultat
Rückblick	<ul style="list-style-type: none"> - Dokumente - Projekt Stand
Fragen bezüglich Projekt	<ul style="list-style-type: none"> - Möchten Sie zwei verschiedene Backend? oder eines mit einem Zusatzpanel? - Reservations Template? Design von Reservationskalender - Geräte ein und ausschalten -> wie ist dies realisiert? - request / response (Fehlermeldungen) bezüglich Radius (wenn wirklich nötig, Einführung/Schulung in RADIUS) - XML-RPC: wie sieht ein request aus (Parameter, Methodennamen)
Gruppe	<ul style="list-style-type: none"> - Gruppenzugehörigkeit ist dynamisch -> wird mit Roman noch abgeklärt, ob Caching benötigt wird, oder nicht.
RADIUS	<ul style="list-style-type: none"> - INS verwendet FreeRadius. - Es gibt keine entsprechende Fehlermeldung zurück. - Username, Port und IP wird mitgegeben(NAS).
Gerät	<ul style="list-style-type: none"> - einschalten/ausschalten: wird via SNMP realisiert. PHP Funktionen, ein- & ausschalten sind bereits vorhanden -> erhalten wir noch.
XML-RPC	<ul style="list-style-type: none"> - Es sollte möglich sein, dass das Skript nicht im Nachhinein abgeändert werden muss, weil sich z.B. die DB-Struktur geändert hat. - Wir erhalten als Beispiel das PHP-Skript test.php. - PHP Skript gibt 0 oder 1 zurück.
Reservationskalender	<ul style="list-style-type: none"> - Reservationskalender ist uns überlassen zu implementieren, können Template verwenden. Evtl. Java Script implementieren, um die Reservationsen zu machen (Mouse-Over Event).
Projektplan	<ul style="list-style-type: none"> - Seite 7 Satz einfügen: Vorgabe von Abteilung Informatik -> getrennte Dokumente abgeben. - Satz mit DRY-Prinzip herausnehmen.
Ausblick	<ul style="list-style-type: none"> - An nächster Sitzung nimmt Herr Peter Sommerlad und Herr Beat Stettler teil. Die Zeit ist noch nicht bekannt. - Ausflug nach Bern, um den Experten Herr Roland Marti zu besuchen.

ToDo

- Internationalisierung
- Alle Views fertig stellen
- XML-RPC Script schreiben
- Routing
- Definitives Projekt inkl. SVN einrichten (Namensgebung)
- Template Reservation implementieren
- Login mit LDAP & Gruppe

Sitzungsprotokoll

Zeit: 14:00 – 15:20 Uhr

Woche: SW07

Ort: Zimmer 1.223

Anwesende: Markus Kolb, Mirjam Schaffner, Rolf Schärer, Beat Stettler, Peter Sommerlad

Traktanden

- Rückblick
- Frage bezüglich Projekt

Diskussionen / Beschlüsse

Thema	Resultat
Rückblick	<ul style="list-style-type: none">- Bachelorarbeit (Thema) erklären- Status BA: Am Implementieren
Fragen bezüglich Projekt	<ul style="list-style-type: none">- Reservation View: Wie wird dies gewünscht? (pro Pod ein Kalender?)
Anforderungsspezifikation	<ul style="list-style-type: none">- UC Model: Gerät fehlt als Actor- UC XML-RPC fehlt- UC Tabelle: Anstatt UseCase-Nummer den Namen angeben
Symfony	<ul style="list-style-type: none">- Pod darstellen, welches Gerät dieser hat.- Link generieren, um Pod anzusehen- Unit Tests schreiben -> diese an Herrn Sommerlad senden
Projektplan	<ul style="list-style-type: none">- Herr Sommerlad findet, dass der Projektplan zu viele Template Informationen enthält
Ausblick	<ul style="list-style-type: none">- In ca. 2 Wochen am Mittwochmorgen ab 9.00 Uhr Herrn Sommerlad unseren bisherigen Code zeigen und Unit Tests (Zimmer 1.202)
ToDo	<ul style="list-style-type: none">- Internationalisierung- XML-RPC Script schreiben- Template Reservation implementieren- Login mit LDAP & Gruppe

Sitzungsprotokoll

Zeit: 9:00 – 9:30 Uhr
Woche: SW08
Ort: Baracke
Anwesende: Markus Kolb, Mirjam Schaffner, Rolf Schärer

Traktanden

- Rückblick
- Frage bezüglich Projekt

Diskussionen / Beschlüsse

Thema	Resultat
Rückblick	- Status BA: Am Implementieren
Fragen bezüglich Projekt	- Server VM wird zum lokalen Betrieb transferiert, da die HSR die ldap & rdc Ports nicht frei gibt.
Ausblick	- Nächstes Meeting ist am Freitag, 13.30 Uhr
ToDo	- Routing, Internationalization, Localization - JQuery und Kalender Implementation - Views optimieren

Sitzungsprotokoll

Zeit: 13:30 – 14:00 Uhr
Woche: SW09
Ort: Baracke
Anwesende: Markus Kolb, Mirjam Schaffner, Rolf Schärer

Traktanden

- Rückblick
- Frage bezüglich Projekt

Diskussionen / Beschlüsse

Thema	Resultat
Rückblick	- Status BA: Implementieren
Kommentar zum Reservationsmodul	- Mehr als erwartet
Pod	- Ein Test-Pod wird eingerichtet
Routing	- Sieht gut aus.
Functional Tests	- Functionale Tests sind ok.
Ausblick	-
ToDo	- Internationalization, Localization - Unit Tests schreiben - User Login und LDAP Anbindung

Sitzungsprotokoll

Zeit: 11:15 – 11:45 Uhr
 Woche: SW10
 Ort: Cafeteria Gebäude 1
 Anwesende: Markus Kolb, Mirjam Schaffner, Rolf Schärer

Traktanden

- Rückblick
- Frage bezüglich Projekt

Diskussionen / Beschlüsse

Thema	Resultat
Rückblick	- Status BA: Implementieren
LDAP Anbindung	- Ist gut.
Kommentare im Code	- Wird noch abgeklärt, voraussichtlich guter Mix.
Gruppe / Pods	- ca 100 Devices, 5 Pods und 3 Service Provider sind im INS vorhanden (ca. 13 Geräte pro Pod) - Im Modul Gruppe kann angegeben werden, auf welche Pods diese Gruppe Zugriff hat (via Checkboxen, die untereinander angeordnet sind -> Pod1, Pod2...).
Administrator definieren im Control Panel	In einem Config File wird anhand der CN definiert, welche Gruppe Administratoren Rechte besitzen.
LDAP Gruppen	Jeweils für den User werden die LDAP Gruppen rekursiv abgerufen.
Localization	- Ist ok.
Unit Tests	- Unit Tests sind ok.
Layout	- z.B. die Devices auf mehreren Seiten anordnen (Seite <u>1</u> , <u>2</u> , <u>3</u> ...) und eine Suche implementieren
ToDo	- Internationalization in Form machen - Unit Tests schreiben - User Login und LDAP Anbindung weiter implementieren - Routing Reservationskalender anpassen - Design dokumentieren - Kontaktformulare erstellen

Sitzungsprotokoll

Zeit: 11:50 – 12:20 Uhr
Woche: SW11
Ort: Cafeteria Gebäude 1
Anwesende: Markus Kolb, Mirjam Schaffner, Rolf Schärer

Traktanden

- Rückblick
- Frage bezüglich Projekt

Diskussionen / Beschlüsse

Thema	Resultat
Rückblick	- Status BA: Implementieren
Login	- Sieht gut aus.
Lokalisierung / Internationalisierung	- Ist ok.
Scripts	- Scripts erstellt Rolf Schärer, wir implementieren nur die Links, die die Scripts aufrufen.
Expert	- Meeting mit dem Experten wird in die Wege geleitet und uns wird der Termin bekannt gegeben, sobald dieser feststeht.
LDAP	-
Kontaktformular	- Aufteilung in Themengebiete ist ok.
Sitzungsprotokolle	- Sitzungsprotokolle, die wir erstellt haben, werden noch bestätigt.
ToDo	<ul style="list-style-type: none">- Unit Tests schreiben- User Login und LDAP Anbindung weiter implementieren- Routing Reservationskalender anpassen- Design dokumentieren- Kontaktformulare erstellen

Sitzungsprotokoll

Zeit: 14:00 – 15:10 Uhr
 Woche: SW13
 Ort: Cafeteria Gebäude 1
 Anwesende: Markus Kolb, Mirjam Schaffner, Rolf Schärer

Traktanden

- Rückblick
- Frage bezüglich Projekt

Diskussionen / Beschlüsse

Thema	Resultat
Rückblick	<ul style="list-style-type: none"> - Status BA: Implementieren - Demo der bisherigen Implementation zeigen - Bugfixing des Weekcalendar Plugin - Implementation Python Programm / Cronjob wie dies implementiert werden kann. - Reservationskalender Feature implementiert (über 2 Tagen Reservationen tätigen, timebetween, minimal Reservation time) - MyCalendar implementiert - Mailer Problem gelöst, Mails können nun via Mail Account verschickt werden. - Demonstration des JQuery Plugins Datatables - Kontaktformulare implementiert - MyProfile implementiert - Register auf Login Seite implementiert
Fragen bezüglich Projekt	<ul style="list-style-type: none"> - Homepage: wie sollen wir die Eingangsseite gestalten? - Wie werden Reservationen in der Vergangenheit gehandhabt? - Wie sollen das Seiten blättern vom Layout her implementiert werden (Variationen zeigen) - Mit welcher bestehenden Technologie werden die Reservationen mit z.B. einem PDA synchronisiert?
Pod	<ul style="list-style-type: none"> - Ein Pod wird uns zur Verfügung gestellt. Aus technischen Gründen wird voraussichtlich an der Schule gearbeitet.
Kreditsystem	<ul style="list-style-type: none"> - Gemäss Wunsch wurde anstatt einer Anbindung an ein Kredit System verschiedene Formulare implementiert, über welche Credit Points per Mail angefordert werden können. Die Credit Points werden vom Administrator manuell aufgeladen.
Homepage	<ul style="list-style-type: none"> - Auf der Startseite belassen wir den Standard-Text, es muss nichts Zusätzliches auf der Startseite implementiert werden.
Expert	<ul style="list-style-type: none"> - Meeting mit dem Experten wird vermutlich nicht mehr stattfinden -> wird abgeklärt.

Seiten blättern	- Es wird ein neues Plugin installiert, das ebenfalls die Suche beinhaltet, sowie weitere Features.
Reservationen	- Reservationen in der Vergangenheit können von Benutzern nicht manipuliert werden, ausser vom Administrator selbst. - Vor einer Reservation gibt es keine Wartezeit, nach einer Reservation wird eine Wartezeit angerechnet, die der Benutzer bezahlen muss. - Während einer Reservation kann der Benutzer die Reservation verlängern, aber nicht verkürzen.
Präsentationstermin	- Wunschtermin auf den 23. Juni 2010 festgelegt -> wird noch abgeklärt
Sitzungsprotokolle	- Sitzungsprotokolle, die wir erstellt haben, werden noch bestätigt.
ToDo	- Unit Tests schreiben - Testdokumentation erstellen - Mailing Funktion implementieren - Design dokumentieren - Management Summary erstellen - Abstract erstellen - Datables Plugin fertig implementieren und so alle CRUD Objekte neu erstellen. - JQuery Plugin Link an Rolf Schärer schicken - Reservationskalender gewünschte Änderungen anpassen und Abzüge für Credit Points implementieren. - Zugriff ssh & Geräte neustarten implementieren

Sitzungsprotokoll

Datum: 29.05.2010
 Zeit: 14:30 – 15:30 Uhr
 Woche: SW14
 Ort: 1.212a
 Anwesende: Markus Kolb, Mirjam Schaffner, Rolf Schärer

Traktanden

- Rückblick
- Frage bezüglich Projekt

Diskussionen / Beschlüsse

Thema	Resultat
Rückblick	- Status BA: Implementieren
Fragen zum Projekt	- Anzahl Benutzerdaten, die der Administrator bearbeiten kann - Reservationen, die der Admin für den Benutzer machen kann
Gerät neu starten	10.1.1.1 - Default Gateway 10.1.1.8 - Amazing PDU IP Stromschiene 1 ins/ins-remote (http) 10.1.1.9 - Amazing PDU IP Stromschiene 2 ins/ins-remote (http) 10.1.1.10 - Terminal Server, ins / ins-remote / secret: ins-remote 10.1.1.11 - FreeRADIUS, ins / ins-remote 10.1.1.12 - Microsoft Active Directory (RDP funktioniert) Device/Port Zuordnung Stromschiene 1: R1 - Port A - Terminal Server line 33 R3 - Port B - Terminal Server line 35 R5 - Port C - Terminal Server line 37 R6 - Port E - Terminal Server line 38 R7 - Port D - Terminal Server line 39 ATM - Port G - Terminal Server line 45 BB2 - Port F - Terminal Server line 43 Device/Port Zuordnung Stromschiene 2: R2 - Port A - Terminal Server line 34 R4 - Port B - Terminal Server line 36 R9 - Port C - Terminal Server line 41 R8 - Port D - Terminal Server line 40 BB1 - Port E - Terminal Server line 42 BB3 - Port F - Terminal Server line 44 - Rolf Schärer wird sich am Sonntag Zeit nehmen und den Terminal Server und den FreeRADIUS aufsetzen und an das AD anbinden.
Präsentation	- Termin ist noch offen
User	- Bearbeitungsdaten sind ok, es müssen nicht alle Daten bearbeitet werden können, nur die nötigsten.

Reservation	- Der Administrator soll für Benutzer Reservationen erfassen können.
ToDo	<ul style="list-style-type: none">- Administrator kann Reservation für Benutzer erfassen- Geräte neu starten implementieren- Nötige Dokumente erstellen- iCal implementieren- Unit- und Functional Tests machen

Sitzungsprotokoll

Zeit: 14:00 – 15:00 Uhr
 Woche: SW16
 Ort: Container
 Anwesende: Markus Kolb, Mirjam Schaffner, Rolf Schärer

Traktanden

- Rückblick
- Frage bezüglich Projekt

Diskussionen / Beschlüsse

Thema	Resultat
Rückblick	- Demo: Geräte neustarten, Reservation verwalten Administrator-sicht
Fragen	- Soll der Benutzer überhaupt mehrere Reservationen tätigen können? - Design der Device Table ein / ausschalten ok? - Administrator sieht alle Pods und kann alle bearbeiten etc...? - Kommentar des Codes
Kommentar	- Nicht allzu viel kommentieren, gemäss Wunsch von Herr Sommerlad, nur wirklich da wo wir es für nötig und sinnvoll empfinden.
Modul Active Reservation	- Jeweils eine Sekunde Delay einbauen, sobald Aktionen auf den Geräten ausgeführt werden, da die AmazingPDU's andernfalls nicht nachkommen. - Design sieht gut aus, ist logisch aufgebaut.
Terminalserver	- Pro Pod gibt es einen Terminalserver.
Erweiterungen:	- Benutzer im Control Panel in mehrere Gruppen hinzufügen zu können. - Plausibilitäten des Admins besser überprüfen: Kalender beim Modul reservationDatatables beim Format, es dürfen nicht mehrere Reservationen zur gleichen Zeit auf den gleichen Pod gemacht werden, Credit Points automatisch beim Benutzer abziehen bzw. hinzufügen.
ToDo	- Kontaktformular noch ganz fertig stellen - Kommentar im Code hinzufügen - Dokumente fertig stellen - Dokumente den Betreuern schicken - Code refactoring

Sitzungsprotokoll

Zeit: 17:00 – 18:20 Uhr
 Woche: SW17
 Ort: Container
 Anwesende: Markus Kolb, Mirjam Schaffner, Beat Stettler

Traktanden

- Rückblick
- Dokumentbesprechung

Diskussionen / Beschlüsse

Thema	Resultat
Rückblick	<ul style="list-style-type: none"> - Demo des Control Panels - Dokumente: Anforderungsspezifikation, Projektplan, SAD, Technischer Bericht
Anforderungsspezifikation	<ul style="list-style-type: none"> - Zu mager - viele Sachen sind im Design Dokument erklärt, die in der Anforderungsspezifikation auch erklärt werden sollten - Einschränkungen beschreiben, nicht was vorhanden ist - Produkt beschreiben, damit externe Person schnell alles versteht - User Story erweitern: z.B: Was passiert, wenn Lab zu Ende ist? - Funktionalität: sind Funktionale Anforderungen beschrieben - Anforderungsspezifikation an neue Situation anpassen
Projektplan	<ul style="list-style-type: none"> - Schreibfehler korrigieren - Code Review gegenseitig machen -> aus Text nicht richtig verständlich - Wie viel geplant, wie viel erreicht -> festhalten (-> steht eigentlich im Zeitplan, auf den verwiesen wurde)
SAD	<ul style="list-style-type: none"> - Systemansicht mehr beschreiben - Process View ganzes System aufzeigen, mehrere Prozesse aufzeigen - Reservation beschreiben
Technischer Bericht	<ul style="list-style-type: none"> - nicht zu gebrauchen - nicht im erzähl-Style schreiben, sondern technischer - Schnittstelle zu den CISCO Geräten aufzeigen - aufzeigen, Einbettung von Client Script auf freeRadius Server erklären - andere Software erklären, die gebraucht werden (wie freeRadius)

Ausblick / Erweiterungen	<ul style="list-style-type: none">- History der Reservationsvorgänge (Reservation löschen etc.) in Log File speichern- Reservationseinträge herunterladen => 10*000 Einträge => wie sieht die Performance aus?)- mehrere Administratoren können gleichzeitig Reservationen tätigen (ReservationDatatables)
ToDo	<ul style="list-style-type: none">- Dokumente überarbeiten- Technischer Bericht neu schreiben- Installation Guide

TODO: Remote Lab Management System

Version: 1.0

Woche: 1

Name	Kürzel	E-Mail
Markus Kolb	MK	mkolb@hsr.ch
Mirjam Schaffner	MS	m1schaff@hsr.ch

Aufgaben

Was muss gemacht werden	Priorität	Wer	Bis WANN	OK
einarbeiten Symfony bis Thema 4	1	MK/MS	23.03.2010	OK
Zeitplan erstellen und planen	1	MS	25.02.2010	OK
Architektur & Funktionen vom System überlegen und aufzeichnen	2	MK/MS	25.02.2010	OK
Vorbereitung Meeting	2	MK/MS	25.02.2010	OK
Einrichten SVN und Ordnerstrukturen	1	MK	25.02.2010	OK

TODO: Remote Lab Management System

Version: 1.0

Woche: 2

Name	Kürzel	E-Mail
Markus Kolb	MK	mkolb@hsr.ch
Mirjam Schaffner	MS	m1schaff@hsr.ch

Aufgaben

Was muss gemacht werden	Priorität	Wer	Bis WANN	OK
Projektplan erstellen	1	MK	02.03.2010	OK
Anforderungsspezifikation erstellen	1	MS	02.03.2010	OK
Projektplan gegenlesen	2	MS	02.03.2010	OK
Anforderungsspezifikation gegenlesen	2	MK	02.03.2010	OK
Symfony einarbeiten bis Thema Kapitel 8	3	MK	07.03.2010	OK
Symfony einarbeiten bis Thema Kapitel 8	3	MS	07.03.2010	OK
Datenmodell und Schema erstellen	4	MK	03.03.2010	OK
1. Entwurf Gui Prototyp	5	MK	05.03.2010	OK

TODO: Remote Lab Management System

Version: 1.0

Woche: SW03

Name	Kürzel	E-Mail
Markus Kolb	MK	mkolb@hsr.ch
Mirjam Schaffner	MS	m1schaff@hsr.ch

Aufgaben

Was muss gemacht werden	Priorität	Wer	Bis WANN	OK
Symfony Tutorial weiter arbeiten bis Kapitel 10	2	MS/MK	13.03.2010	OK
Anforderungsspezifikation anpassen / erweitern	1	MS	10.03.2010	OK
Projektplan anpassen / erweitern	1	MS/MK	10.03.2010	OK
Datenbank-Schema anpassen	1	MK	10.03.2010	OK
Sitzungsprotokoll erfassen	1	MS	7.03.2010	OK
Sitzungsprotokoll gegenlesen	1	MK	10.03.2010	OK
ToDo Woche 3 erstellen	1	MS/MK	8.03.2010	OK
UseCase Diagramm erstellen mit den wirklichen UCs	1	MS	10.03.2010	OK
Zeitplan anpassen	2	MK/MS	10.03.2010	OK
Schema Anpassen und Datenmodell generieren	1	MK	10.03.2010	OK
Erweiterungen in der Datenbank überlegen und implementieren.	1	MK	10.03.2010	OK

TODO: Remote Lab Management System

Version: 1.0

Woche: SW04

Name	Kürzel	E-Mail
Markus Kolb	MK	mkolb@hsr.ch
Mirjam Schaffner	MS	m1schaff@hsr.ch

Aufgaben

Was muss gemacht werden	Priorität	Wer	Bis WANN	OK
Einarbeiten Symfony Tutorial bis Kapitel 13	2	MS/MK	18.03.2010	OK
System Sequenz Diagramm	2	MS	16.03.2010	OK
System Sequenz Diagramm gegenlesen	2	MK	17.03.2010	OK
ToDo Woche 4 erstellen	2	MS	12.03.2010	OK
Sitzungsprotokoll erstellen	1	MS	11.03.2010	OK
Sitzungsprotokoll gegenlesen	1	MK	11.03.2010	OK
Anforderungsspezifikation so weit als möglich fertig stellen	1	MS/MK	16.03.2010	OK
Anforderungsspezifikation gegenlesen	1	MK	17.03.2010	OK
Projektplan überarbeiten	1	MK	16.03.2010	OK
Projektplan gegenlesen	1	MS	17.03.2010	OK
Glossar erweitern, überarbeiten	1	MK	17.03.2010	OK
Glossar gegenlesen	1	MS	17.03.2010	OK
Linux VM auf Windows installieren	2	MS/MK	Open	OK

TODO: Remote Lab Management System

Version: 1.0

Woche: SW05

Name	Kürzel	E-Mail
Markus Kolb	MK	mkolb@hsr.ch
Mirjam Schaffner	MS	m1schaff@hsr.ch

Aufgaben

Was muss gemacht werden	Priorität	Wer	Bis WANN	OK
Domain Model	1	MK	24.03.10	OK
System Sequenz Diagramme	1	MS	24.03.10	OK
SAD Systemübersicht	2	MK	24.03.10	OK
VM	1	MK/MS	21.03.10	OK
SAD (Analyse)	2	MS/MK	24.03.10	OK
Domain Analyse erweitern	2	MS/MK	24.03.10	OK

TODO: Remote Lab Management System

Version: 1.0

Woche: SW06

Name	Kürzel	E-Mail
Markus Kolb	MK	mkolb@hsr.ch
Mirjam Schaffner	MS	m1schaff@hsr.ch

Aufgaben

Was muss gemacht werden	Priorität	Wer	Bis WANN	OK
Symfony bis Kapitel 13 weiter machen	1	MS	27.03.2010	Ok
Sitzungsprotokoll Woche 5 erstellen	1	MS	27.03.2010	Ok
ToDo Woche 6 erstellen	1	MS	27.03.2010	Ok
Prototyp 1 weiter arbeiten	1	MK/MS	31.03.2010	Ok
Glossar gegenlesen	2	MK	30.03.2010	Ok
Dokumente gemäss Sitzung ändern	1	MS/MK	30.03.2010	Ok
Klassendiagramm	2	MK	30.03.2010	Ok
Packagediagramm	3	MK/MS	30.03.2010	
Protokollbeschreibungen	3	MK/MS	30.03.2010	Ok

TODO: Remote Lab Management System

Version: 1.0

Woche: SW07

Name	Kürzel	E-Mail
Markus Kolb	MK	mkolb@hsr.ch
Mirjam Schaffner	MS	m1schaff@hsr.ch

Aufgaben

Was muss gemacht werden	Priorität	Wer	Bis WANN	OK
Alle Views	1	MK		OK
Definitives Projekt inkl. SVN einrichten (Namensgebung)	1	MK		Ok
XML-RPC	1	MS		Ok
Internationalisierung	3	MS		OK
Template Reservation?				Ok
Routing	2	MK/MS		Ok
Login mit LDAP & Gruppe	2	MK		OK

TODO: Remote Lab Management System

Version: 1.0

Woche: SW08

Name	Kürzel	E-Mail
Markus Kolb	MK	mkolb@hsr.ch
Mirjam Schaffner	MS	m1schaff@hsr.ch

Aufgaben

Was muss gemacht werden	Priorität	Wer	Bis WANN	OK
Anforderungsspezifikation anpassen gemäss Sitzung	1	MS	12.4.2010	Ok
XML-RPC	1	MS	12.4.2010	Ok
Internationalisierung	3	MS	12.4.2010	OK
Template Reservation	1	MK	12.4.2010	Ok
Login mit LDAP & Gruppe	2	MK	12.4.2010	OK
Routing	1	MS/MK	12.4.2010	OK

TODO: Remote Lab Management System

Version: 1.0

Woche: SW09

Name	Kürzel	E-Mail
Markus Kolb	MK	mkolb@hsr.ch
Mirjam Schaffner	MS	m1schaff@hsr.ch

Aufgaben

Was muss gemacht werden	Priorität	Wer	Bis WANN	OK
Internationalisierung	3	MS		OK
Localization	2	MS		OK
Login mit LDAP & Gruppe	2	MK		OK
Functional Tests	1	MS		OK
Unit Tests	1	MS		OK
Reservationskalender	1	MK		OK
Routing	1	MS/MK		OK

TODO: Remote Lab Management System

Version: 1.0

Woche: SW10

Name	Kürzel	E-Mail
Markus Kolb	MK	mkolb@hsr.ch
Mirjam Schaffner	MS	m1schaff@hsr.ch

Aufgaben

Was muss gemacht werden	Priorität	Wer	Bis WANN	OK
Internationalisierung	3	MS		OK
Login mit LDAP & Gruppe	2	MK		OK
Functional Tests	1	MS		OK
Unit Tests	1	MS		OK

TODO: Remote Lab Management System

Version: 1.0

Woche: SW11

Name	Kürzel	E-Mail
Markus Kolb	MK	mkolb@hsr.ch
Mirjam Schaffner	MS	m1schaff@hsr.ch

Aufgaben

Was muss gemacht werden	Priorität	Wer	Bis WANN	OK
Kontaktformulare	1	MS		OK
Reservationskalender anpassen	1	MK		OK
LDAP Gruppen und LDAP Admin	1	MK		OK
Functional Tests	1	MK	23.05.10	OK
Unit Tests	1	MK	23.05.10	OK
Internationalisierung dokumentieren	1	MS	16.05.10	OK

TODO: Remote Lab Management System

Version: 1.0

Woche: SW12

Name	Kürzel	E-Mail
Markus Kolb	MK	mkolb@hsr.ch
Mirjam Schaffner	MS	m1schaff@hsr.ch

Aufgaben

Was muss gemacht werden	Priorität	Wer	Bis WANN	OK
Validation, Mail	3	MS	12.06.10	
Reservationen an 2 Tagen	1	MK	16.05.10	OK
Bugfixing Reservationskalender	1	MK	16.05.10	OK
User erfassen	1	MK	16.05.10	OK
Unit Tests	2	MK	15.06.10	
MyProfile erstellen	1	MS	15.06.10	OK
Kontaktformulare, CSS, XSS	1	MS	15.06.10	OK
Testdokument erstellen	2	MS	15.06.10	OK
Link Paging implementieren	2	MS	18.06.10	OK

TODO: Remote Lab Management System

Version: 1.0

Woche: SW13

Name	Kürzel	E-Mail
Markus Kolb	MK	mkolb@hsr.ch
Mirjam Schaffner	MS	m1schaff@hsr.ch

Aufgaben

Was muss gemacht werden	Priorität	Wer	Bis WANN	OK
Kontaktformulare Register implementieren	1	MS	21.05.10	OK
Abstract / Kurzfassung erstellen	2	MS	21.05.10	OK
Refactoring	1	MK	21.05.10	OK
Functional Tests	1	MK	21.05.10	
Unit Tests	1	MK	21.05.10	
Testdokumentation erweitern	2	MS	21.05.10	OK
CRUD Module neu implementieren mit Javascript / Plugin	1	MS/MK	26.05.10	OK

TODO: Remote Lab Management System

Version: 1.0

Woche: SW14

Name	Kürzel	E-Mail
Markus Kolb	MK	mkolb@hsr.ch
Mirjam Schaffner	MS	m1schaff@hsr.ch

Aufgaben

Was muss gemacht werden	Priorität	Wer	Bis WANN	OK
Validation	1	MS	30.05.10	OK
iCal implementieren	2	MK	30.05.10	OK
Refactoring Reservation	1	MK	30.05.10	OK
Functional Tests	1	MK	30.05.10	
Unit Tests	1	MK	30.05.10	OK
Reservation Datatable implementieren	1	MS	30.05.10	OK
Technischer Bericht erstellen	1	MS	30.05.10	OK
Management Summery erstellen	2	MS	30.05.10	OK
Mehrere Forms erstellen für Kontaktformulare	1	MS	30.05.10	OK
Testbericht erstellen	1	MS	30.05.10	OK

TODO: Remote Lab Management System

Version: 1.0

Woche: SW15

Name	Kürzel	E-Mail
Markus Kolb	MK	mkolb@hsr.ch
Mirjam Schaffner	MS	m1schaff@hsr.ch

Aufgaben

Was muss gemacht werden	Priorität	Wer	Bis WANN	OK
Validator	1	MS	11.06.10	OK
Implementierung SNMP Module	1	MK	6.06.10	OK
Aktive Reservation Modul implementieren	2	MK	11.06.10	OK
Functional Tests	1	MK	11.06.10	OK
Unit Tests	1	MK	11.06.10	OK
Dokumente SAD, Technischerbericht erweitern	2	MS	11.06.10	OK
Poster A0 erstellen	1	MS	11.06.10	OK
Dokumente Management Summery, Technischer Bericht, Abstract erweitern	1	MS	11.06.10	OK
Lamp anstelle Lampp installieren	1	MK/MS	4.06.10	OK

TODO: Remote Lab Management System

Version: 1.0

Woche: SW16

Name	Kürzel	E-Mail
Markus Kolb	MK	mkolb@hsr.ch
Mirjam Schaffner	MS	m1schaff@hsr.ch

Aufgaben

Was muss gemacht werden	Priorität	Wer	Bis WANN	OK
Dokument SAD erweitern	1	MS/MK	11.06.10	OK
Dokument Technischer Bericht erweitern	1	MS/MK	6.06.10	OK
Dokument Management Summery erweitern	2	MS/MK	11.06.10	
Dokument Installationsanleitung erweitern	2	MK	11.06.10	
Abstract erstellen	2	MS	11.06.10	OK
Kurzfassung für Diplombroschüre fertig stellen	1	MS	11.06.10	OK
Active Reservation Modul fertig stellen	1	MK	11.06.10	OK
XML-RPC implementieren	2	MK	11.06.10	OK
Register Formular fertig stellen	1	MS	11.06.10	OK
Unit Tests implementieren	1	MK	13.06.10	OK
Functional Tests implementieren	1	MS	13.06.10	