Deutsches Krebsforschungszentrum  
Im Neuenheimer Feld: 580  
69120 Heidelberg, Deutschland

Yannic Abou-Taleb

Projektdokumentation





**with Python**

**Web Application**

Entwicklung eines webbasierten Buchungsverwaltungssystems

von Arbeits- und Laborgeräten

Meisenweg: 6

68623 Lampertheim

Fachinformatiker

In Anwendungsentwicklung

Oliver Pelz

Wohnort:

Ausbildungsberuf:

Prüflingsnummer:

Projektzeitraum:

Projektbetreuer:

Inhaltsverzeichnis

1. Ausgangssituation 1

1.1. Ausbildungsbetrieb 1

1.2. Projektumfeld 1

1.3. IST-Zustand 1

1.4. SOLL-Zustand 1

1.5. Projektauftrag 2

2. Projektplanung 2

2.1. Projektschnittstellen/ Kontaktpersonen 2

2.2. Kostenplanung 2

2.3. Aufgabendefinition 3

2.4. Erfassung der Anforderungen 3

2.5. Konzept entwickeln 4

3. Projektdurchführung 4

3.1. Installation des Web Frameworks 4

3.2. Apache Webserver Konfiguration 5

3.3. Übersichtskalender 6

3.4. Benutzerbereich 8

# Ausgangssituation

## Ausbildungsbetrieb

Das Deutsche Krebsforschungszentrum (DKFZ) wurde 1964 in Heidelberg gegründet. Am DKFZ begann ich am 1.9.2008 meine Ausbildung zum Fachinformatiker in Anwendungsentwicklung. Dieses Projekt wurde während meiner Ausbildung geplant, umgesetzt und dokumentiert.

Das Deutsche Krebsforschungszentrum ist eine Einrichtung des öffentlichen Rechts, was mit seinen rund 2500 Mitarbeiter eines der größten Forschungszentren innerhalb der Helmholtz-Gemeinschaft.

Das DKFZ beschäftigt sich mit der Ergründung und Bekämpfung von Krebs. Dabei wird die Forschung darauf ausgelegt durch experimentelle Gewinnung von Versuchs- und Patientendaten mehr über das menschliche Genom zu erfahren und so neue Möglichkeiten der Vorbeugung und der Therapie von Krebs zu entwickeln. Durch die Gewinnung von Versuchs- und Patientendaten fallen eine große Menge Daten an, die dann mit Computern weiterverarbeitet werden.

## Projektumfeld

Die Abteilung Signalwege und funktionelle Genomik (B110) am DKFZ versucht die Eigenschaften einzelner Erbanlagen mit Hilfe von Hochdurchsatz-Verfahren und RNA-Interferenz-Methoden in verschiedenen zellulären Abläufen aufzuklären. Bei der Erforschung dieser unbekannten biologischen Funktionen wird eine sehr große Menge an Messwerten produziert.

Die Informatik-Untergruppe der Abteilung beschäftigt sich hierbei mit der Speicherung und der statistischen Auswertung dieser Daten. Daneben gehören sowohl die Entwicklung von Web- als auch das Erstellen von Desktop-Anwendungen zu ihren Aufgaben, um den Forschern die Arbeit zu erleichtern.

## IST-Zustand

In der Abteilung gibt es eine Sammlung gemeinschaftlich genutzter Labor- und Arbeitsgeräte, die zur Durchführung bestimmter biologischer Versuche benötigt werden. Zur Verwaltung dieser Ressourcen gibt es momentan eine auf PHP basierte Webanwendung.

Diese ist jedoch sehr unübersichtlich, zu statisch, unsicher und teilweise fehlerhaft. Aufgrund dieser Probleme wird diese Anwendung wenig genutzt und Geräte werden meist mündlich reserviert.

Dies ist natürlich nicht sehr praktikabel, da man keine zuverlässige Übersicht über die verwendeten oder benötigten Geräte bekommt: So kann es leicht zu Doppelreservierungen kommen, und die Projektplanung wird erschwert, da es keine Informationen über die Auslastung von Ressourcen gibt

## SOLL-Zustand

Da die Verbesserung der PHP Webanwendung zu zeitaufwändig wäre, um die erwähnten Probleme zu beseitigen, soll eine komplett neue Webanwendung erstellt werden. Aus Kostengründen soll ausschließlich Open Source Software eingesetzt werden.

Als Datenbankverwaltungssystem wurde MySQL ausgewählt. Die Wahl der Programmiersprache fiel auf Python und auf das moderne, vielseitig einsetzbare und sehr gut dokumentierte Web Framework Django, das sehr schnelle Entwicklungszyklen und ein einfaches und sauberes Design ermöglicht. Die fertige Webanwendung soll auf einem Apache-Webserver ausgeführt werden und somit den Mitarbeitern der Abteilung über Standard Webbrowser zur Verfügung stehen.

Die Anwendung soll in zwei Grundbereiche unterteilt werden. Es soll einen allgemein zugänglichen Bereich (Frontend) geben, in dem man sich anhand eines Kalenders eine Übersicht über die momentan aktiven Reservierungen anzeigen lassen kann. Weiterhin soll es einen internen Bereich (Backend) geben, in dem man sich authentifizieren muss, um Geräte reservieren zu können. In diesem Bereich soll es Unterschiede zwischen normalen Benutzern (Usern) und den Administratoren der Webanwendung geben. Der administrative Bereich dient der Verwaltung und Konfiguration der Webanwendung (Anlegen, Editieren, Löschen von Geräten und Reservierungen). Daneben soll die normale Gerätereservierung mit einem Standard DKFZ "Active Directory" Benutzerkonto (LDAP-Authentifizierung) der Arbeitsgruppe B110 möglich sein. Diese normalen Benutzer können nur Reservierungen anlegen und ihre eigenen verwalten.

## Projektauftrag

Ziel des Projektes ist es, eine benutzerfreundliche und praktische Webanwendung zu erstellen, die von den Mitarbeitern der Abteilung B110 genutzt wird, um die tägliche Arbeit mit gemeinsam genutzten Arbeitsmitteln zu erleichtern. Das Projekt soll bei Beendigung als freie Software zur Verfügung gestellt werden.

# Projektplanung

## Projektschnittstellen/ Kontaktpersonen

|  |  |  |
| --- | --- | --- |
| Name | Abteilung | Funktion |
| Michael Boutros | Molekulare Genomanalyse | Auftraggeber |
| OIiver Pelz | Molekulare Genomanalyse | Projektleiter und Projekbetreuer |

## Kostenplanung

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Beschreibung | Kosten pro | | Jahr | Stunde | Projekt |
| **Personalkosten** | |  | |  |  |
| Yannic Abou-Taleb | | 10591,80 € | | 5,16 € | 360,09 € |
| **Sachmittel** | |  | |  |  |
| **Lizenzkosten** | |  | |  |  |
| MS Office 2011 für Mac | | 29,88 € | | 0,01 € | 0,70 € |
| **Arbeitsplatz** | |  | |  |  |
| DKFZ Standard iMac | | 240,00 € | |  |  |
| Strom und Büromaterial | |  | | 0,30 € | 21,00 € |
| **Summe** | |  | |  |  |

## Aufgabendefinition

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Bezeichnung | | Beschreibung | | |
| **Planung** |  | |  |  |
| Kundenwünsche erfassen | | Hier sollen die Kundenwünsche erfasst werden. | | |
| Konzept entwickeln | | Hier soll ein Konzept entworfen werden das den Kundenwünschen gerecht wird. | | |
| **Durchführung** | |  | | |
| Einrichtung des Web Frameworks | | Installation des Web Frameworks.  Anbindung an die Datenbank. | | |
| Apache Webserver konfigurieren | | Den Apache Webserver konfigurieren, das die Webanwendung aufgesetzt werden kann. | | |
| Übersichtskalender | | Erstellung eines Kalenders, der eine Übersicht über die reservierten Geräte bietet. | | |
| Backend | | Erstellung des Backend Bereich.  Benutzer und Administrativen Bereich. | | |
| Reservierung von Geräten | | Geräte-Reservierung Modul umsetzen und implementieren. | | |
| Authentifikation | | LDAP-Login realisieren | | |

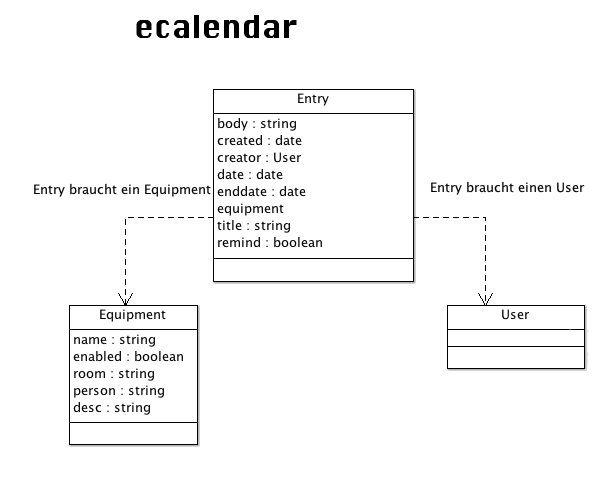
## Erfassung der Anforderungen

Vor der Umsetzung des Projektes wurden die Richtlinien vom Auftraggeber festgelegt. Die Erfassung der Anforderungen an die Webanwendung ergab folgende Kriterien:

* Zuverlässigkeit
* einfache Bedienbarkeit für den Benutzer
* schlichtes Design
* “Active Directory“ Authentifikation mittels LDAP
* Trennung von Benutzer und administrativen Bereich
* Bearbeitung eigener Reservierungen

## Konzept entwickeln

Aus den oben erwähnten Anforderungen wurde ein Konzept entwickelt. Die Grundstruktur wurde in Form eines UML-Diagrammes dargestellt:



ecalendar

Die Applikation soll “ecalendar“ heißen und wird in das Projekt “bookingCal“ eingebunden.

Während die Klassen dieser Applikation noch programmiert werden müssen ist die Klasse “User“ schon von Django aus mitgeliefert. Diese Klasse ist in der in Django implementierten Benutzerverwaltung enthalten. Diese muss man lediglich genau wie eigene Applikationen in das Projekt einbinden. Durch die vorhandene Benutzerverwaltung kann einfach auf die User Klasse zugegriffen werden und muss nicht erst noch realisiert werden. Da die Datenbankstruktur von Django organisiert wird fällt dieser Punkt bei der Konzeptentwicklung weg.

# Projektdurchführung

## Installation des Web Frameworks

Bei der Installation des Web Frameworks Django gibt es drei wesentliche Schritt:

* Python installieren
  + LDAP Modul für Python installieren
* Django installieren
* Datenbank installieren

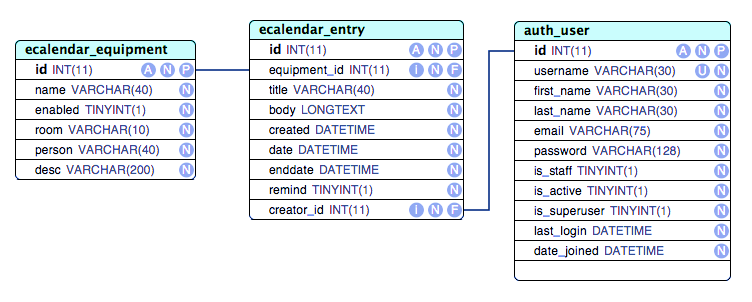
Es wurde folge dessen Python 2.5 und Django 1.2 installiert und eine vorhandene MySQL Datenbank benutzt. In MySQL wurde ein neuer Benutzer und eine neue Datenbank angelegt.

So ergaben sich folgende Daten für die Verbindung zur Datenbank:

|  |  |
| --- | --- |
| Benutzername | django |
| Passwort | geheim |
| Datenbank | booking |

Nachdem die Datenbank installiert wurde, wurde ein neues Projekt namens “bookingCal“ angelegt.

In der Konfigurationsdatei “settings.py“ wurde die Datenbankverbindung eingetragen. Danach wurde die Applikation “ecalendar“ erstellt und ebenfalls in der Konfigurationsdatei und somit in das Prokekt eingebunden. Die Applikation beinhaltet jedoch nur die Variablen und noch keinerlei Funktion.

Da die Datenbank zu diesem Zeitpunkt noch leer ist, wurde auf Kommandozeilenebene der Befehl “django-admin syncdb“ eingetippt und somit wurde die Datenbankstruktur automatisch von Django erstellt. Da Django viele Tabellen automatisch anlegt hier nur ein Ausschnitt der wichtigsten Tabellen:

## Apache Webserver Konfiguration

Der Apache Webserver läuft bereits auf einem Ubuntu System. Um die Webapplikation auf einem Apache Webserver laufen zu lassen musste eine neue Dateistruktur angelegt werden, so wurde alles in den Ordner “django“ gelegt in dem Verzeichniss “/var/www/“.

Dieser Ordner hat folgende weiterführende Dateistruktur:

Applikations Ordner

Applikations Templates

Admin Templates

Template Ordner

Danach musste nur noch die Apache Konfigurationsdatei “httpd.conf“ bearbeitet werden.

MaxRequestsPerChild 1

<location "/">

SetHandler python-program

PythonHandler django.core.handlers.modpython

SetEnv DJANGO\_SETTINGS\_MODULE bookingCal.settings

PythonOption django.root /bookingCal

PythonDebug On

PythonPath "['/var/www/django/django\_projects'] +

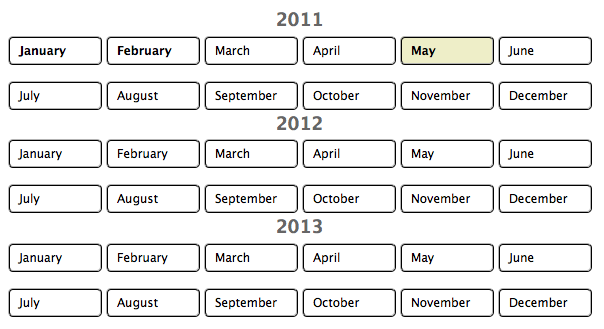
sys.path"

</location>

Damit ist die Konfiguration des Apache Webserver schon fertig und nach einem Neustart ist die Webapplikation auf dem Server über einen Webbrowser erreichbar.

## Übersichtskalender

Der Kalender dient zur Übersicht über die Reservierungen, in einer Monats- und Tagesansicht. Python besitzt standardmäßig eine Datums Klasse auf der sich der Kalender aufbaut.

Es werden Listen für jedes Jahr erstellt die wiederum in einer Liste zusammengefasst werde. Diese List wird im Template ausgegeben.

Der Aktuelle Monat ist farbig hinterlegt. Jeder Monat hat ein eigenes HTML div-tag in dem der Link zu dem jeweiligen Monat zu finden ist. Bereits angeklickte Monate werden hervorgehoben.

Die Links zu der Monatsansicht werden mit einer Django Funktion generiert, hier ein Beispiel:

Die Klasse month erstellt Listen zur Darstellung des Monates. Sie kann außerdem den nächsten oder vorherigen Monat generieren.

Hier für wurde im Template für den Monat 2 Links eingefügt. Diese benutzen auch die URL Generation von Djnago und übergeben zusätzlich den Parameter “change“, der angibt ob der nächste oder der vorherige Monat angezeigt werden soll:

{% url ecalendar.views.month year month "prev" %}

In der Klasse month wird dann abgefragt was in dem Parameter “change“ steht und dann wird der Monat eins hoch oder, wie in diesem Fall, herunter gezählt.

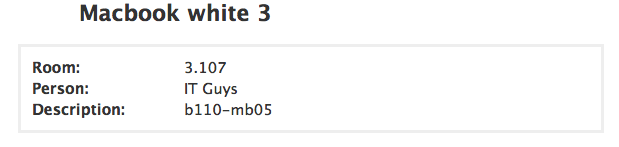
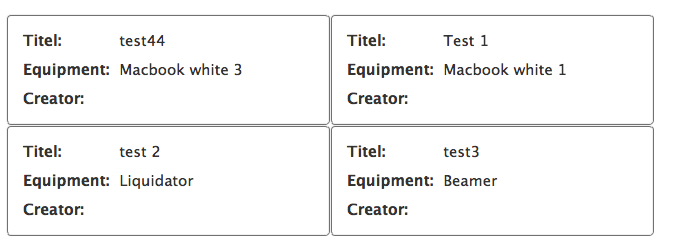
In allen Fällen wird der month Klasse das Jahr und der Monat übergeben. Je nachdem ob der Parameter “change“ gesetzt ist wird der Monat ausgewählt. Entweder wird der übergebene Monat verwendet oder der Monat wird erst neu gesetzt.

Danach wird jeder Tag des Monats durchlaufen und es wird überprüft ob an diesem Tag eine Reservierung aktiv ist und ob das der aktuelle Tag ist. Diese Informationen werden wochenweise in eine Liste gespeichert. Die Liste hat so insgesamt immer 4 Einträge, für jede Woche des Monats einen. Jeder Wocheneintrag ist wiederum auch eine Liste, in der jeder Tag 3 Informationen hat.

Hier ein Beispiel für die Liste der ersten Woche eines beliebigen Monats:

Das Template generiert mit diesen Informationen die Monatsansicht. Mit dem 3. Parameter, der den aktuellen Tag angibt, wird der aktuelle Tag farbig hinterlegt. Der zweite Parameter liefert ggf. eine Liste der Reservierungen, diese wir im Template ausgelesen und dargestellt.

So ähnlich wird auch die Tagesansicht generiert nur wird dabei dem Template eine Liste mit einem Eintrag übergeben. Bei dieser Ansicht kann man nun per klick auf der entsprechende Reservierung sich diese komplett anzeigen lassen.

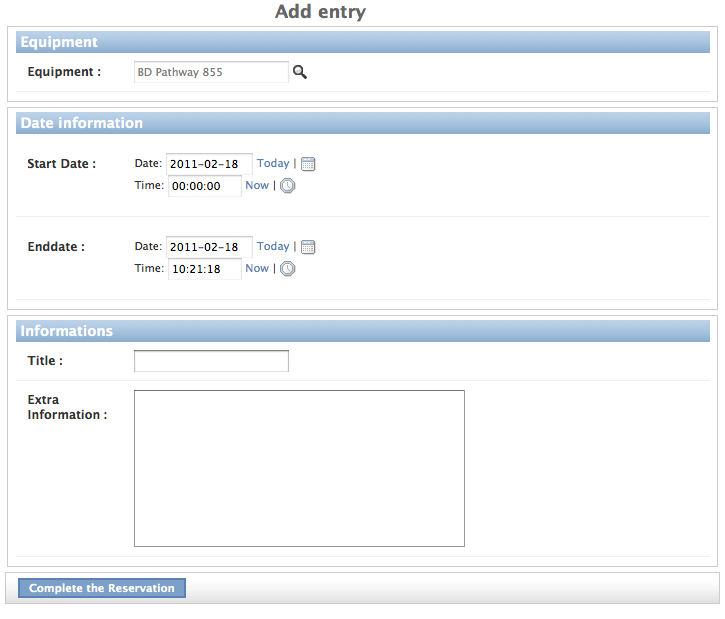


Bei der Reservierungsansicht werden alle Informationen aufgelistet die bei der Reservierung angegeben wurden. Klickt man auf das Gerät erscheint unterhalb der Reservierung Informationen zu diesem Gerät.

## Benutzerbereich

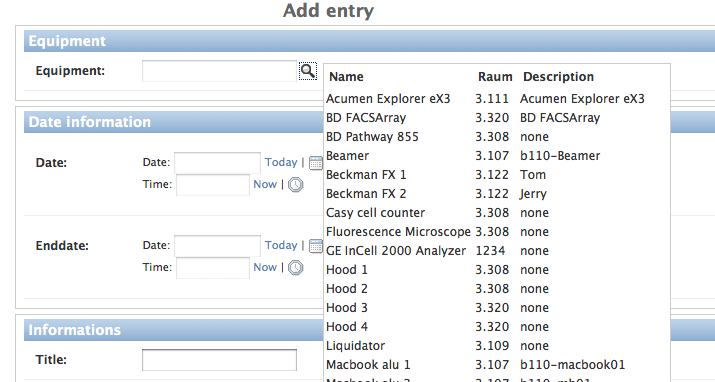
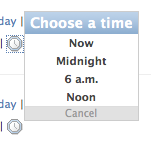
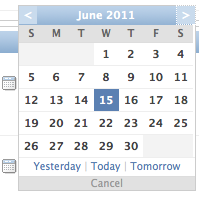
Für den Benutzerbereich wurde zunächst eine neues Template für die Reservierung angelegt. Dieses Template beinhaltet ein Formular um Informationen für die Reservierung einzugeben. Das Formular beinhaltet folgende Eingabe Felder:

* Gerät
* Startseitpunkt
* Endzeitpunkt
* Name der Reservierung
* Extra Information



Über die Symbole werden das Gerät und die Zeitpunkten ausgewählt. Die manuelle Eingabe der Felder wurde gesperrt um mögliche Eingabe Fehler auszuschließen.

Das Gerät lässt sich nur über das Klicken auf das Symbol auswählen, während die Auswahl des Tages und der Uhrzeit auch über den neben den Symbolen stehenden links getroffen werden kann. Diese tragen den aktuellen Tag oder die aktuelle Uhrzeit ein.



Werden die Symbole angeklickt wird ein OnClick-Event ausgelöst und das entsprechende Overlay wird mittels Javascript angezeigt.

Nach dem Abschicken des Formulars werden die Daten verarbeitet. So wird aus den Informationen von dem Datum und der Uhrzeit der Start- und Endzeitpunkte ein Datetime Format erstellt. Aus den Ausgewählten Zeitpunkten wird eine Datetime Variable nach folgendem Format erstellt: "%Y-%m-%d %H:%M:%S".

Dazu muss zunächst das Datum und die Uhrzeit zusammengeführt werden und dann diesen String in ein Datumsformat umgewandelt werden.

Startdate = Entrydate1 +“ “+ Entrytime1

sDT = datetime.fromtimestamp(time.mktime(

time.strptime(Startdate, time\_format)))

Da nun die 2 Zeitpunkte auch programmiertechnisch einen Zeitpunkt definieren kann damit auch die Prüfung gemacht werden, ob das Gerät zum ausgewählten Zeitraum frei ist.

Um diese Prüfung zu veranschaulichen wurde die Grafik erstellt:

Zeitraum der zu überprüfende Reservierung

Möglicher Zeitraum

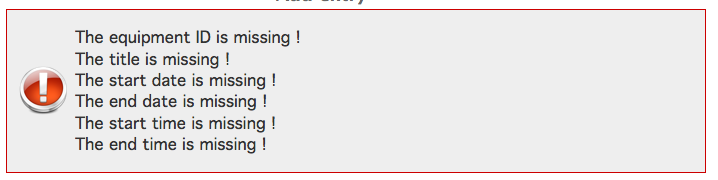
Möglicher Zeitraum

Möglicher Zeitraum

Möglicher Zeitraum

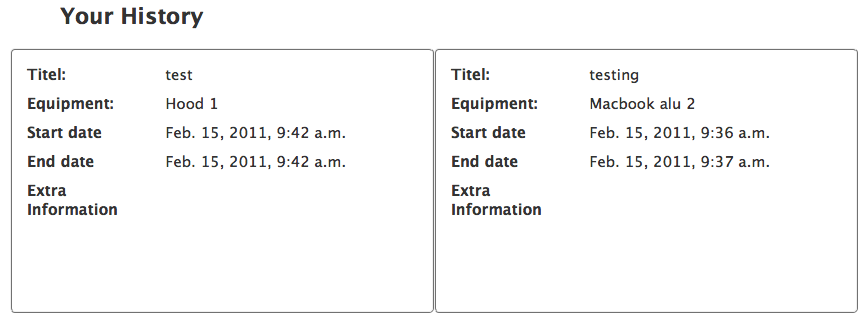
So ergeben sich vier Möglichkeiten bei denen das ausgewählte Gerät nicht frei ist. Diese Möglichkeiten werden bei der Überprüfung abgefangen.

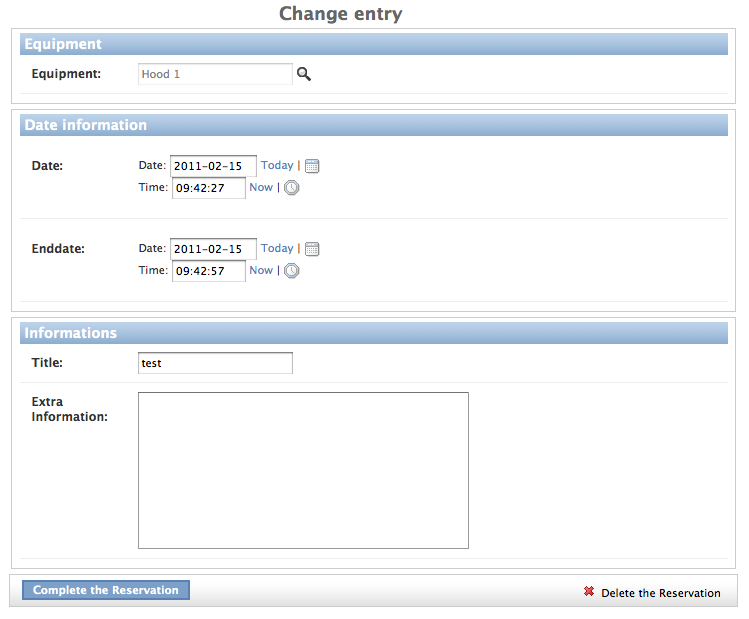
Des Weiteren werden fehlende Eingaben oder falsch Eingabe bei den Zeitpunkten abgefangen, z.B. ob der Startzeitpunkt vor dem Endzeitpunkt ist. Bei einem Fehler wird eine Fehlermeldung über dem Formular ausgegeben:



Wenn die Reservierung erfolgreich war erscheint eine Bestätigung, dass die Reservierung erfolgreich war.

Im Benutzerbereich gibt es außer der Reservierung noch den Bereich der eigenen Reservierungen. Dieser ist im Menü über “Your History“ zu erreichen und es wird, ähnlich wie bei der Tagesansicht im Kalender, die Reservierungen angezeigt. Den Ersteller der Reservierungen anzuzeigen würde hier keinen Sinn machen, so wird diese Information ausgeblendet. Da es keine Einzel Ansicht der Reservierungen gibt, werden die restlichen Information angezeigt. Klickt man auf eine Reservierung kommt man, anders wie bei der Tagesansicht, zu dem Bearbeitungs-Formular, das genauso funktioniert wie das Erstell-Formular.





Das Bearbeitungs-Formular hat noch einen Zusatz nämlich die Möglichkeit die Reservierung zu löschen.