

# Pflichtenheft

Gruppe SWP08-7

20. Mai 2008

## Inhaltsverzeichnis

<b>1 Zielbestimmung</b>	<b>3</b>
1.1 Musskriterien . . . . .	3
1.2 Wunschkriterien . . . . .	3
1.3 Abgrenzungskriterien . . . . .	3
<b>2 Produkteinsatz</b>	<b>3</b>
2.1 Anwendungsbereiche . . . . .	3
2.2 Zielgruppen . . . . .	3
2.3 Betriebsbedinungen . . . . .	3
<b>3 Produktübersicht</b>	<b>4</b>
3.1 Muss-Funktionen . . . . .	4
3.2 Kann-Funktionen . . . . .	8
3.3 USE-Case . . . . .	9
<b>4 Produktdaten</b>	<b>9</b>
<b>5 Produktleistungen</b>	<b>9</b>
<b>6 Qualitätsanforderungen</b>	<b>10</b>
<b>7 Benutzeroberfläche</b>	<b>10</b>
<b>8 Nichtfunktionale Anforderungen</b>	<b>11</b>
<b>9 Technische Produktumgebung</b>	<b>11</b>
9.1 Software . . . . .	11
9.2 Hardware . . . . .	11
9.3 Orgware . . . . .	11
9.4 Produkt-Schnittstellen . . . . .	11

<b>10 Spezielle Anforderungen an die Entwicklungsumgebung</b>	<b>11</b>
10.1 Software . . . . .	11
10.2 Hardware . . . . .	11
10.3 Orgware . . . . .	11
<b>11 Gliederung in Teilprodukte</b>	<b>11</b>
<b>12 Ergänzung</b>	<b>12</b>

## 1 Zielbestimmung

Das bestehende OLAT-System soll durch eine neue Schnittstelle so verändert werden, dass neue Lernressourcen-Arten auf kontrollierte Weise (XML-Konfigurations-Datei) zum OLAT-System hinzugefügt werden können. Als “proof-of-concept” werden einige der bestehenden Lernressourcen in das neue Format transformiert sowie eine weitere Lernressource neu entwickelt, bei deren Einbindung das Integrationskonzept prototypisch umgesetzt wird.

### 1.1 Musskriterien

- RepositoryMainController erweitern
- RepositoryExtension erstellen, importieren
- “Schwarzes Brett” als Lernressourcen-Erweiterung umsetzen
- Bestehende Lernressourcen-Arten transformieren

### 1.2 Wunschkriterien

Optional ist die Möglichkeit des konsistenten Abzugs von Lernressourcen-Arten umzusetzen, wobei insbesondere zu klären ist, wie mit dem Entfernen solcher Lernressourcen aus aktuellen Kursen umzugehen ist.

### 1.3 Abgrenzungskriterien

-

## 2 Produkteinsatz

Das Produkt dient zur Erweiterung der Funktionalität des Tabs Lernressourcen. Es richtet sich also an Erweiterungsentwickler um diesen die Einbindung von neuen Lernressourcen zu erleichtern.

### 2.1 Anwendungsbereiche

Das Produkt kann in jedem OLAT-System eingefügt werden um dieses um neue Lernressourcen zu erweitern.

### 2.2 Zielgruppen

Studenen, Dozenten, OLAT-Systemverwalter

### 2.3 Betriebsbedingungen

Die Anwendung soll auf jedem internetfähigen Rechner über einen Browsers lauffähig sein.

### 3 Produktübersicht

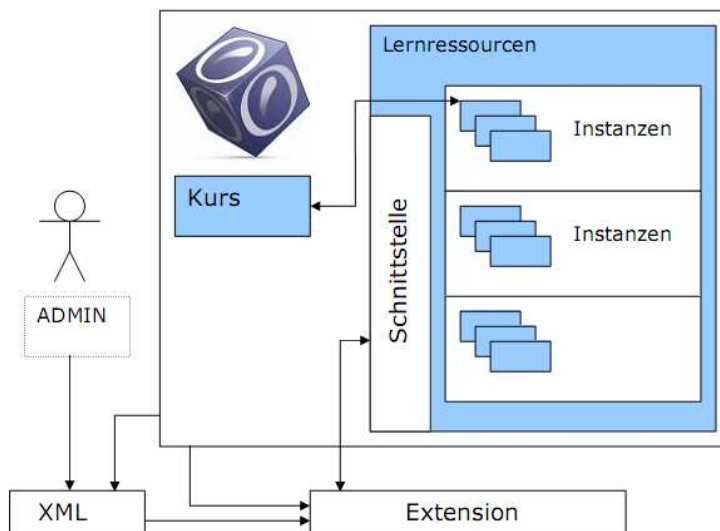


Abbildung 1: Produktuebersicht

#### 3.1 Muss-Funktionen

##### /F 010/ - RepositoryMainController erweitern

**Geschäftsprozess** RepositoryMainController erweitern

**Ziel** Vorhandene Klasse erweitern um Lernressourcen hinzufügen zu können

**Kategorie** primär

**Vorbedingung** -

**Nachbedingung Erfolg** -

**Nachbedingung Fehlschlag** -

**Akteure** -

**Auslösendes Ereignis** -

**Beschreibung** Erweitern der Methode "buildTreeModel". Hier werden zusätzlich zu den statischen Inhalten alle verfügbaren Erweiterungen überprüft und nach möglichen Menüeinträgen abgefragt. Hierzu werden Description, Actiontext und Controller abgefragt. Beim Anklicken des Links (Actiontext wird als Text angezeigt, Description als Mouse-Over-Text) wird der Controller als Element eines neuen Panels eingefügt.

**Erweiterungen** -

**Alternativen** -

**/F 020/ - RepositoryExtension erstellen**

**Geschäftsprozess** RepositoryExtension erstellen

**Ziel** Neuen Erweiterungspunkt erstellen

**Kategorie** primär

**Vorbedingung** -

**Nachbedingung Erfolg** -

**Nachbedingung Fehlschlag** -

**Akteure** -

**Auslösendes Ereignis** -

**Beschreibung** Beschreibung Auf Basis der ActionExtension wird eine Klasse erstellt mit den nachfolgenden Funktionalitäten:

**Erweiterungen** -

**Alternativen** -

**/F 021/ - Import**

**Geschäftsprozess** Import

**Ziel** Abfragen einer Import-Funktion der Extension

**Kategorie** primär

**Vorbedingung** -

**Nachbedingung Erfolg** -

**Nachbedingung Fehlschlag** -

**Akteur** RepositoryMainController

**Auslösendes Ereignis** -

**Beschreibung** Prüfen der Extension mithilfe von Funktionen der RepositoryExtension. Hierzu werden Description, Actiontext und Controller für ein mögliches neues Element im Toolbaum-Untermenü Import abgefragt. Beim Anklicken des Links (Actiontext wird als Text angezeigt, Description als Mouse-Over-Text) wird der Controller als ClosableModalController geöffnet.

**Erweiterungen** -

**Alternativen** -

**/F 022/ - Erstellen**

**Geschäftsprozess** Erstellen

**Ziel** Abfragen einer Erstellen-Funktion der Extension

**Kategorie** -

**Vorbedingung** -

**Nachbedingung Erfolg** -

**Nachbedingung Fehlschlag** -

**Akteur** RepositoryMainController

**Auslösendes Ereignis** -

**Beschreibung** Prüfen der Extension mithilfe der RepositoryExtension auf eine Erstellfunktion. Hierzu werden Description, Actiontext und Controller für ein mögliches neues Element im Toolbaum-Untermenü Erstellen abgefragt. Beim Anklicken des Links (Actiontext wird als Text angezeigt, Description als Mouse-Over-Text) wird der Controller als ClosableModalController geöffnet.

**Erweiterungen** -

**Alternativen** -

**/F 030/ - Beispielerweiterung "Schwarzes Brett"**

**Geschäftsprozess** Beispielerweiterung "Schwarzes Brett"

**Ziel** Prototypische Umsetzung einer Erweiterung für die Lernressourcen-Schnittstelle

**Kategorie**

**Vorbedingung** -

**Nachbedingung Erfolg** -

**Nachbedingung Fehlschlag** -

**Akteur** -

**Auslösendes Ereignis** -

**Beschreibung** Erstellen einer Beispielerweiterung, der die in /F 010/ bis /F 022/ definierte Schnittstelle nutzt. Diese Erweiterung implementiert ein "Schwarzes Brett". Es soll möglich sein die Einträge aufzulisten sowie (unbeschränkt) Einträge einzufügen.

**Erweiterungen** -

**Alternativen** -

**/F 031/ - Vorhandene Einträge anzeigen**

**Geschäftsprozess** Vorhandene Einträge anzeigen

**Ziel** Anzeigen der in einer "Schwarzes Brett"-Instanz vorhandenen Einträge

**Kategorie** primär

**Vorbedingung** -

**Nachbedingung Erfolg** -

**Nachbedingung Fehlschlag** -

**Akteur** -

**Auslösendes Ereignis** -

**Beschreibung** Die hinterlegten Einträge, die von Kursteilnehmer eingefügt wurden, werden formatiert ausgegeben.

**Erweiterungen** -

**Alternativen** -

**/F 032/ - Neuen Eintrag einfügen**

**Geschäftsprozess** Neuen Eintrag einfügen

**Ziel** Neuen Eintrag in eine "Schwarzes Brett"-Instanz einfügen

**Kategorie** primär

**Vorbedingung** -

**Nachbedingung Erfolg** -

**Nachbedingung Fehlschlag** -

**Akteur** -

**Auslösendes Ereignis** -

**Beschreibung** Kursteilnehmer können (unbeschränkt) neue Einträge einfügen.

**Erweiterungen** -

**Alternativen** -

**/F 040/ - Existierende Lernressourcen auf Erweiterungsmodell anpassen**

**Geschäftsprozess** Existierende Lernressourcen auf Erweiterungsmodell anpassen

**Ziel** Existierende Lernressourcen auf neues Modell transformieren

**Kategorie** -

**Vorbedingung** -

**Nachbedingung Erfolg** -

**Nachbedingung Fehlschlag** -

**Akteur** -

**Auslösendes Ereignis** -

**Beschreibung** Möglichst viele der schon vorhandenen Funktionalitäten innerhalb des Tabs Lernressourcen, die nicht als statische Funktionalität bestehen bleibt, wird in einzelne Erweiterungen modular ausgegliedert.

**Erweiterungen** -

**Alternativen** -

**3.2 Kann-Funktionen**

**/F 100/ - Löschen**

**Geschäftsprozess** Löschen

**Ziel** Abfragen einer Löschen-Funktion der Extension

**Kategorie** -

**Vorbedingung** -

**Nachbedingung Erfolg** -

**Nachbedingung Fehlschlag** -

**Akteur** RepositoryMainController

**Auslösendes Ereignis** -

**Beschreibung** Prüfen der Extension mithilfe der RepositoryExtension auf eine Löschfunktion. Hierzu werden Description, Actiontext und Controller für ein mögliches neues Element im Toolbaum-Untermenü Löschen abgefragt. Beim Anklicken des Links (Actiontext wird als Text angezeigt, Description als Mouse-Over-Text) wird der Controller als ClosableModalController geöffnet.

**Erweiterungen** -

**Alternativen** -



### 3.3 USE-Case

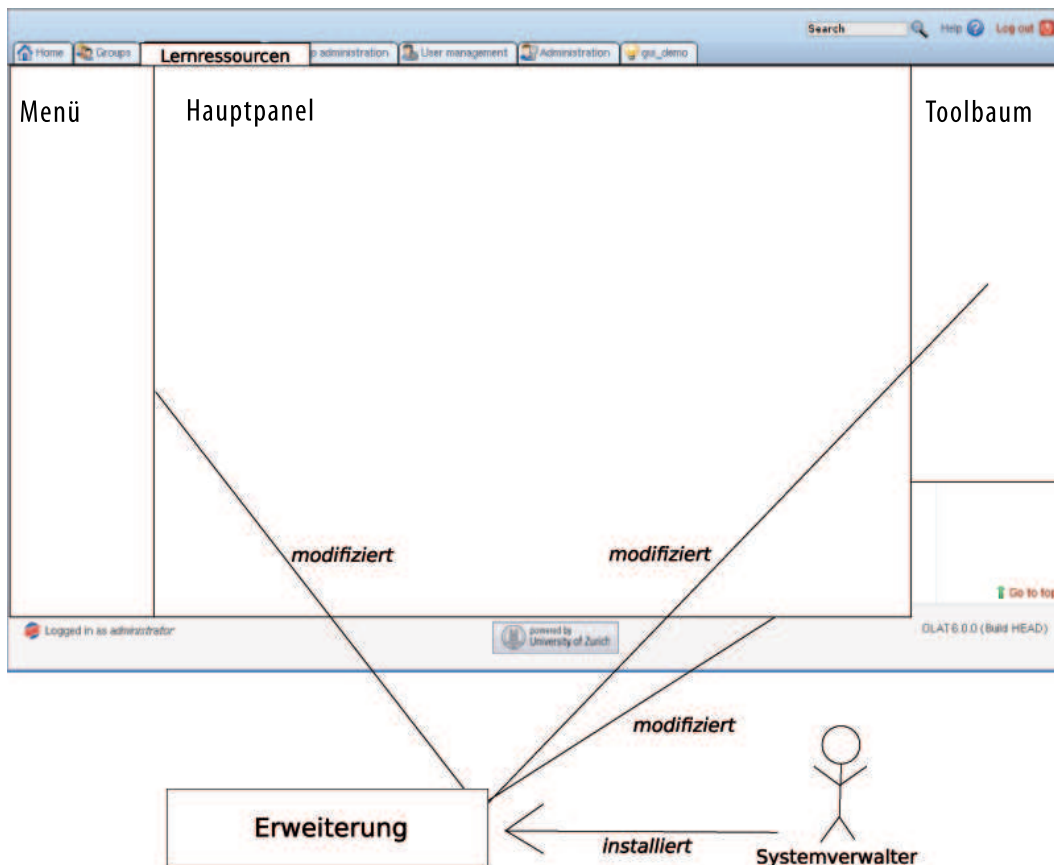


Abbildung 2: Use-Case Diagramm

## 4 Produktdaten

Im jetzigen Modell bestehen keine von uns vorgegebenen Datenstrukturen. Restriktionen werden lediglich durch die OLAT-Umgebung vorgegeben.

## 5 Produktleistungen

/L10/ Es sollte keine Beschränkung für die Anzahl der Erweiterungen geben.

/L20/ Die bestehende Funktionalität darf nicht ungewollt durch Erweiterungen eingeschränkt werden.

## 6 Qualitätsanforderungen

Produktqualität	sehr gut	gut	normal	nicht relevant
<b>Funktionalität</b>				
Angemessenheit	x			
Richtigkeit	x			
Interoperabilität		x		
Ordnungsmäßigkeit	x			
Sicherheit	x			
<b>Zuverlässigkeit</b>				
Reife	x			
Fehlertoleranz		x		
Wiederherstellbarkeit		x		
<b>Benutzbarkeit</b>				
Verständlichkeit			x	
Erlernbarkeit			x	
Bedienbarkeit			x	
<b>Effizienz</b>				
Zeitverhalten			x	
Verbrauchsverhalten			x	
<b>Änderbarkeit</b>				
Analysierbarkeit			x	
Modifizierbarkeit			x	
Stabilität			x	
Prüfbarkeit			x	
<b>Übertragbarkeit</b>				
Anpassbarkeit				x
Installierbarkeit				x
Konformität				x
Austauschbarkeit				x

## 7 Benutzeroberfläche

**/B10/** Die Oberfläche wird über einen Webbrowser dargestellt. Die Benutzeroberfläche wird der von OLAT entsprechen.

**/B20/** Die Benutzeroberfläche ist auf Mausbedienung auszulegen.

**/B30/** Fensterlayout, Dialogstruktur und Mausbedienung entsprechen dem Windows-Gestaltungs-Regelwert (style guide).

## 8 Nichtfunktionale Anforderungen

- Plattformunabhängigkeit

## 9 Technische Produktumgebung

### 9.1 Software

Javafähiges Betriebssystem

### 9.2 Hardware

Internetfähiger PC mit Monitor, Maus, Tastatur

### 9.3 Orgware

Internetzugang

### 9.4 Produkt-Schnittstellen

- Java RE/SDK
- Apache TomCat Server
- Hibernate kompatible Datenbank

## 10 Spezielle Anforderungen an die Entwicklungsumgebung

### 10.1 Software

- Eclipse
- SVN

### 10.2 Hardware

- Ein Server, welcher genug Ressourcen zur Verfügung stellt um eine OLAT-Instanz ausführen zu können.

### 10.3 Orgware

-

## 11 Gliederung in Teilprodukte

-

## **12 Ergänzung**

Um die Möglichkeit, dass die entstehende Funktionalität in das OLAT-Projekt eingeht, zu gewährleisten, muss sich der Quellcode an den diesbezüglichen Richtlinien und Vorgaben von OLAT orientieren. Hier geht es insbesondere um Konventionen der Variablen-, Klassen- und Methodenbenennung sowie die Kommentierung und Dokumentation.