

# 4. Aufgabenserie

## Entwurfsbeschreibung der Softwarestudie

### I. Allgemeines

Die vorliegende Softwarestudie basiert auf dem Konzept des OLAT-Portals. Die Grundsatzanforderungen an die Softwarestudie liegen in der Umsetzung der vorgegebenen Spezifikation unter Beachtung der vorhandenen Design- und Strukturkonzepte.

### II. Produktübersicht

Die vorliegende Softwarestudie ermöglicht auf Basis des OLAT-Portals ausschließlich die Nutzung von Gruppen. Zudem ist der Gastzugang auf der Startseite entfernt.

Innerhalb des Portals existiert nun ein weiterer Tab „Öffentlicher Bereich“, in dem alle Gruppen des Portals angezeigt werden. Des Weiteren kann man nach bestimmten Nutzern mit dem Ziel suchen, an diese eine Email zu verschicken.

### III. Struktur- und Entwurfsprinzipien im Gesamtsystem

Eine OLAT-Erweiterung besteht aus Klassen für Seitendefinition, Seitendarstellung sowie Controllern zum Steuern des Inhalts (diese heißen gemäß den bisherigen OLAT-Konventionen \*SiteDef, \*Site und \*Controller). Die Seitendefinition ist dafür verantwortlich, Tab-Elemente zu erstellen, und wird in der `olat_extensions.xml` eingebunden. Die Klasse Site erstellt die Site selbst und lädt das Kernstück, den Controller. In dem Controller wird ein Menu-Baum erstellt und in diesen die Navigationsunterpunkte hinzugefügt. Die event-Methode verwaltet die Aktionen, wie etwa Klicks auf das Menu. Außerdem werden die Namen für die Sprachübersetzung definiert sowie Controller für die weitere Funktionalität verlinkt.

Möchte man zudem Änderungen an einer bestehenden OLAT-Installation bezüglich bestimmter bereits vorhandener Funktionalitäten vornehmen, so ist die Datei `olat_config.xml` von zentraler Bedeutung, da in ihr zentrale Systemeinstellungen einer OLAT-Instanz verwaltet werden. Die Einstellungen werden hierbei entsprechend den zugehörigen Modulen wie z. B. LoginModule, UserModule und DBModule unterteilt.

Tabs, englisch Sites, werden in OLAT in der Konfigurationsdatei „`olat_extensions.xml`“ im Ordner „`webapp/WEB_INF`“ verwaltet.

## IV. Struktur- und Entwurfsprinzipien im Paket

### Konkretisierung der Deaktivierung des Gastlogins und des Lernressourcen-Tabs:

Um den Gast-Login zu entfernen muss man im LoginModule der Datei olat\_config.xml eine Zeile ändern. In der Zeile

```
<GuestLoginLinks enabled="true" />
```

muss die Einstellung „true“ auf „false“ gesetzt werden. Wenn man nun den Tomcat-Server stoppt und erneut startet, ist der Gast-Login entfernt.

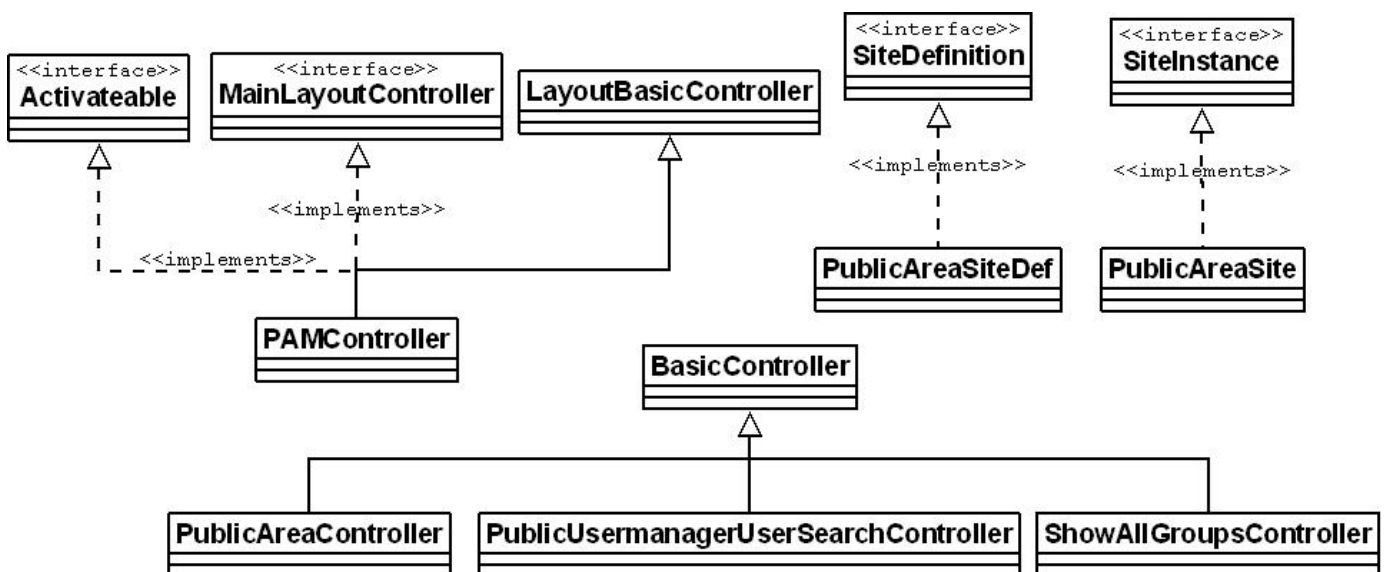
In der Datei olat\_extensions.xml kann man das standardmäßig aktivierte Tab „Lernressourcen“ deaktivieren, indem man die Zeile

```
<bean id="olatsites_repository" class="org.olat.repository.site.RepositorySiteDef" singleton="false" />
```

auskommentiert, d. h. man setzt davor die Zeichen „!--“ und dahinter „-->“. Nach einem Neustart des Tomcat-Servers ist die Site „Lernressourcen“ entfernt.

### Überblick über unser Paket:

Das folgende UML-Schema gibt nun einen Überblick über das von uns erstellte Paket mit den neuen Funktionalitäten. Dabei haben wir die bestehende Strukturordnung im OLAT eingehalten und Analogien zu bereits vorhanden Paketen konsequent genutzt, wie nachfolgend erläutert.



### **Zur Anzeige der Gruppen:**

Der Controller `ShowAllGroupsController` soll alle auf der Plattform entstandenen Gruppen im öffentlichen Bereich anzeigen. Da dies im öffentlichen Bereich aufrufbar sein soll, wird der Controller aus dem `PAConroller` (Controller des öffentlichen Bereichs) heraus initiiert.

Die Oberklasse aller OLAT - Gruppen ist die `BusinessGroup`. Um alle `BusinessGroups` aus dem System verwalten zu können, muss der im OLAT integrierte Gruppen - Manager `BusinessGroupManagerImpl` referenziert werden. Ist dies geschehen, so kann man mit seinen Methoden bestimmte Gruppen aufrufen, finden oder definieren. Die Methode `BusinessGroupManagerImpl.getInstance()` liefert die Instanz des `BusinessGroupManager`. Die Methode `.getAllBusinessGroups()` gibt eine Liste aller erstellten `BusinessGroups` als Rückgabewert.

Der verwendete bereits in OLAT enthaltene `TableController` verwaltet die Ausgabetabelle aller Gruppen. Diesem übergeben wird eine Tabellenkonfiguration. Anschließend werden mit der Methode `.addColumnDescriptor()` die Spalten - Köpfe benannt. Zum Schluss erhält der `TableController` das eigentliche Datenmodell aller Gruppen, `BusinessGroupTableModelWithType` - in diesem ist der Gruppenname und die zugehörige Beschreibung enthalten. Final wird die Methode `.modelChanged()` des Tabellen - Controllers aufgerufen, um der graphischen Ausgabe die Aktualisierung dessen bekannt zu geben, somit auszugeben.

### **Zur Nutzersuche und Emailfunktion:**

In unserer Studie haben wir den Punkt Benutzersuche mit dem Controller `PublicUsermanagerUserSearch` vernetzt. Dieser ist eine Kopie vom `UsermanagerUserSearchController`, welcher die Suchfunktion des OLAT-Administratorbereichs bereitstellt. Er liegt ebenso im Ordner `/org/olat/admin/user`.

Unter anderem wird dabei ein Formular zur Eingabe der Suchdaten erstellt, sowie eine Ergebnislistendarstellung. Dabei wurde von uns eine Kopie des ursprünglichen Controllers erstellt, um Funktionalitäten wie zum Beispiel die Möglichkeit, Nutzer zu editieren, zu entfernen, da diese Funktion nur dem Administrator zur Verfügung stehen sollte und darum in einem öffentlichen Bereich nicht sinnvoll ist.