

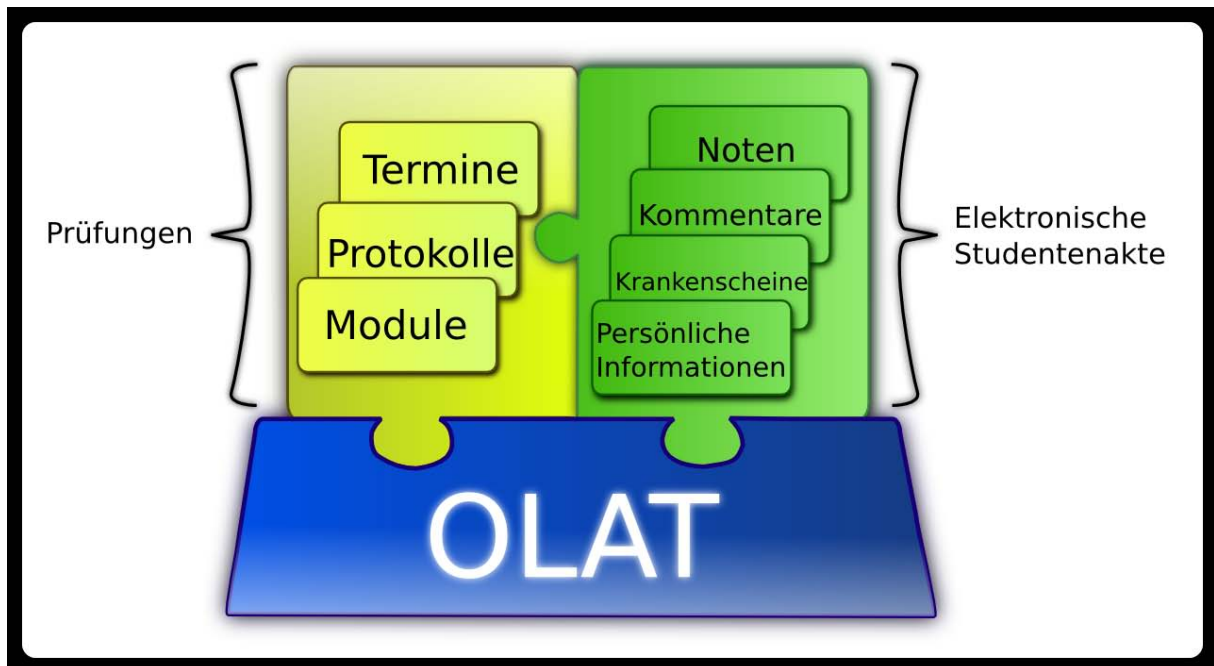
# Entwurfsbeschreibung

## 1. Allgemeines

Das Produkt OLAT-PA ist eine Instanz des OLAT-LMS welche zur Verwaltung von Prüfungen eingesetzt wird. Damit steht Studenten, Prüfungsverantwortlichen und dem Prüfungsamt eine einheitliche, zentrale Stelle zur Verwalten von Prüfungsanmeldungen zu Verfügung. Außerdem beinhaltet das System die ESA (Elektronische Studentenakten) in welcher sämtliche Ergebnisse, Krankmeldungen und Kommentar des zugehörigen Studenten gespeichert werden. Die hier vorgestellte Version ist dabei durch ein Update der bereits laufenden OLAT-PA Version entstanden

## 2. Produktübersicht

OLAT-PA gliedert sich grob in zwei Teilsysteme, das Prüfungsverwaltungssystem und die elektronische Studentenakte.



Gesamtübersicht OLAT-PA ; Grafik aus Bachelorarbeit[Gerber 2009] Seite 10 Abb. 4.2.

Das Prüfungsverwaltungssystem realisiert dabei die zur Durchführung von Prüfungen relevanten Schritte wie die Anmeldung, Terminierung, Protokollierung und das Eintragen und Veröffentlichen

swp11-2

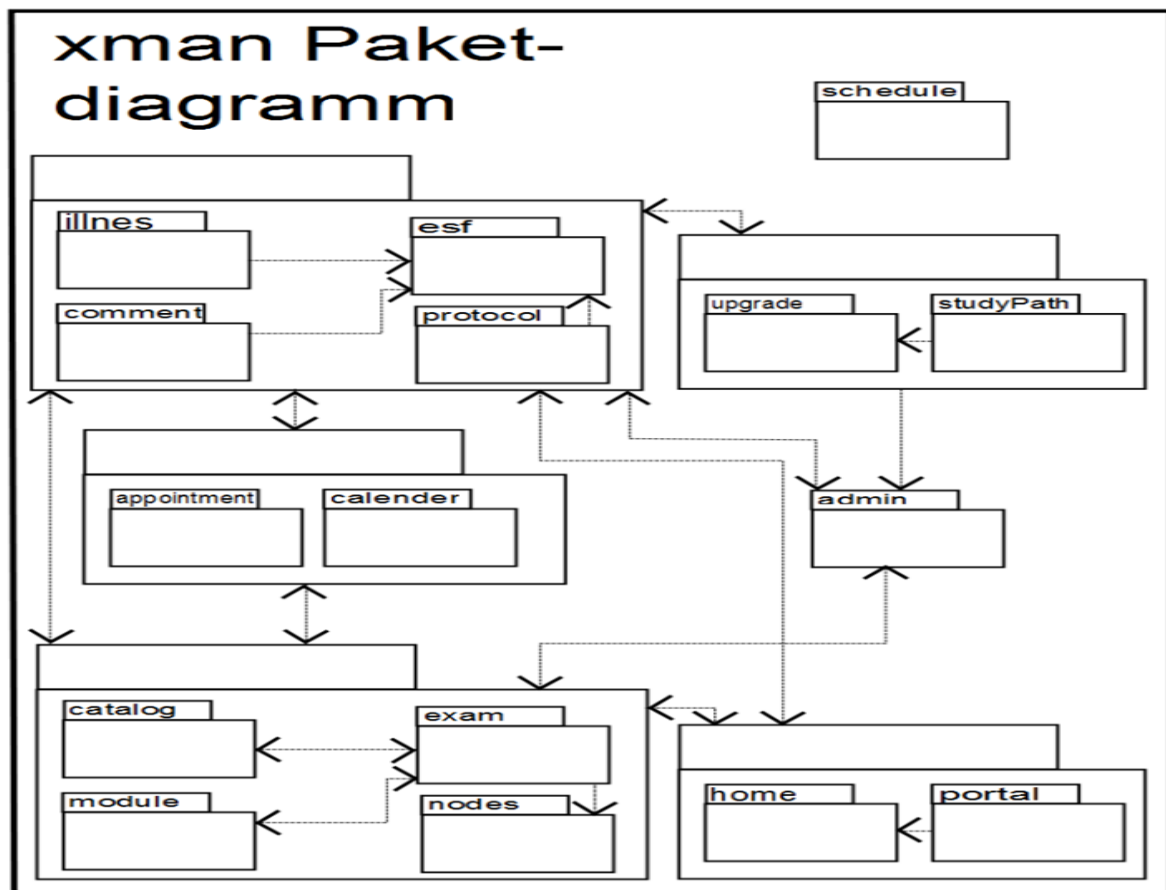
Projektleiter : Matthias Haeßner | Dokument erstellt von Axel Fischer

der Ergebnisse.

In der Studentenakte werden neben den zu den Prüfungen gehörenden Ergebnissen auch Krankmeldungen, Kommentare und weitere persönliche Information wie Name, Vorname, Matrikelnummer oder Studiengang verwaltet und dargestellt.

### 3. Grundsätzliche Struktur- und Entwurfsprinzipien für das Gesamtsystem

Das Paketdiagramm soll einen Überblick über die Pakete des Olat-PA-Moduls und deren Beziehungen zueinander geben. Um eine gute Übersicht zu garantieren sind die Pakete in mehrere Komponenten zusammen gefügt wurden. Eine Komponente besteht aus den Paketen `esf` und den Paketen `illnes`, `comment`, `protocol`, die alle von `esf` genutzt werden. Eine andere große Komponente besteht aus den Paketen `exam`, `nodes`, `module` und `catalog`, welche eng miteinander verknüpft sind. Die Pakete `appointment` und `calender` sind in einer Komponente zusammengefasst, da sie die selben Beziehungen zu den anderen Paketen haben. Weiterhin bilden die Pakete `upgrade`, `studyPath` und `home`, `portal` jeweils eine Komponente. Das Paket `schedule` nutzt weder ein anderes Paket von `xman` noch wird es von ein anderen genutzt. Das Admin-Paket stellt unter anderen Controller für die `esf`, `module`, `studypath` zur Verfügung und hat dementsprechende Beziehungen zu den anderen Paketen.

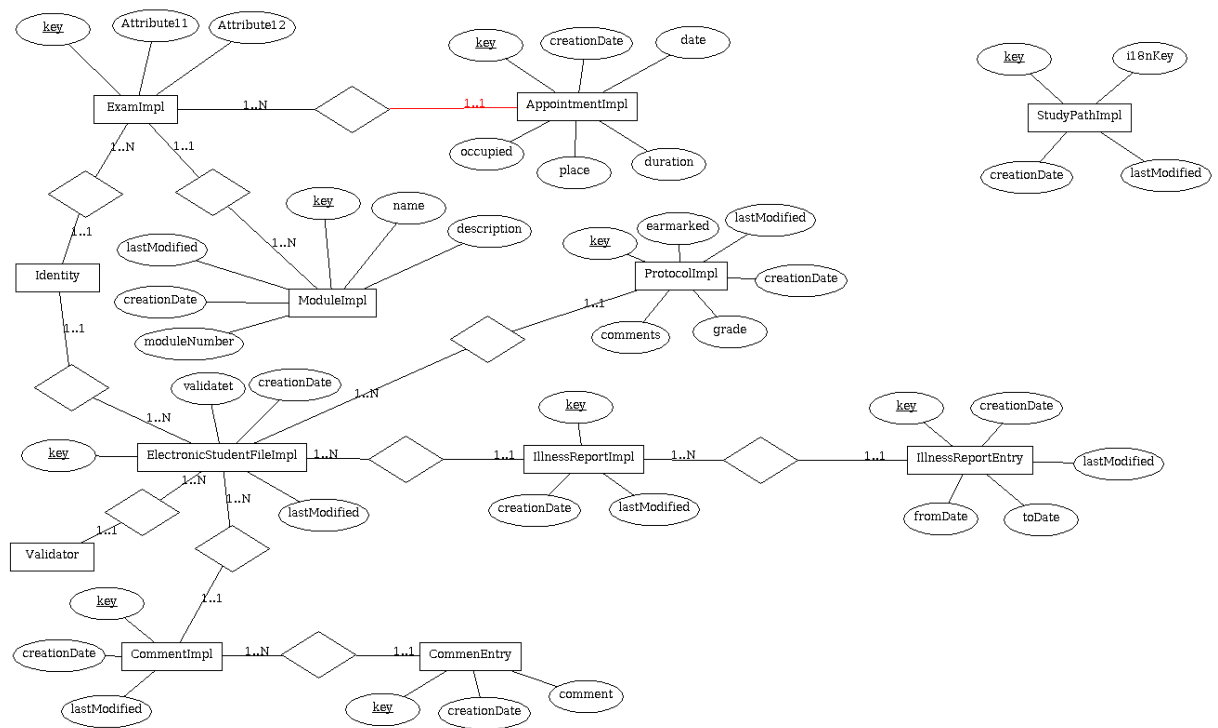


## Datenbankentwurf

Die folgende Abbildung stellt den Datenbankentwurf für das OLAT-PA System in einem Entity-Relationship-Modell dar.

Das Mapping der zugehörigen Objekte auf die Datenbank erfolgt dabei durch Hibernate, wobei die genaue Spezifikation des Mappings in einer Mapping-Datei erfolgt. Besonders zu beachten ist zudem das in den Entity-Mengen keine Fremdschlüssel bezeichnet werden müssen.

Der genaue Entwurf zur Realisierung der Hibernate mappings wurde bereits in der vorangegangenen Analyse des Basisprojekts diskutiert und wird hier nicht erneut dargestellt.

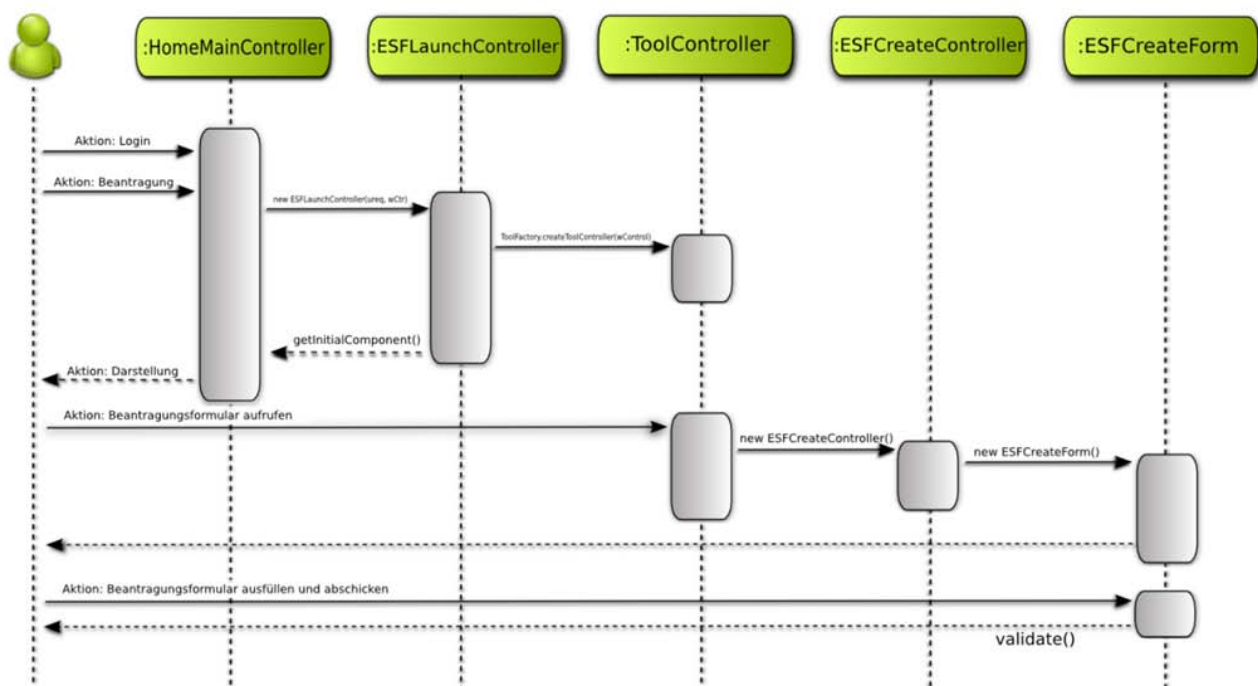


## 4. Grundsätzliche Struktur- und Entwurfsprinzipien der einzelnen Pakete

### ESA

Bevor eine Studentenakte vom Student beantragt werden kann, muss sich dieser im System über den von OLAT zur Verfügung stehenden Registrationsprozess registrieren. Danach kann er seine persönliche Studentenakte beantragen, wobei er seinen Vornamen, Nachnamen, Studserv-Emailadresse, Matrikelnummer und Studiengang angeben muss. Diese Beantragung erfolgt über einen Link auf der Home Seite des Benutzers, nach dessen Betätigung ein ESFLaunchController erzeugt wird, welcher die Erstellung der neuen Akte regelt. Dieser Prozess wird im unten dargestellten Sequenzdiagramm noch einmal genauer dargestellt.

Diese noch nicht validierte Akte wird mit den anderen gespeichert und kann daraufhin von einem Benutzer mit Rolle Prüfungsamt händisch validiert werden.



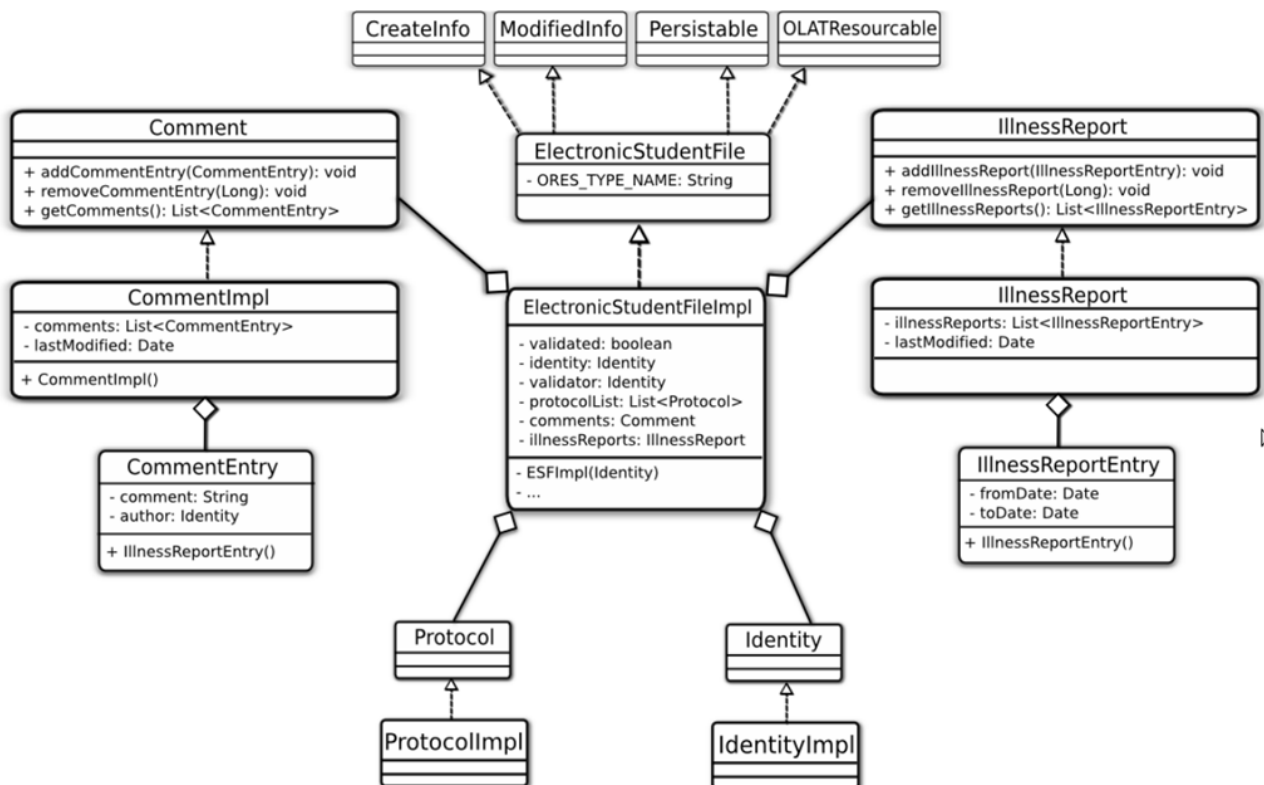
Sequenzdiagramm – Beantragung einer Studentenakte; Grafik aus Bachelorarbeit[Gerber 2009] Seite 25 Abb. 4.10

An diesem Vorgang ändert sich auch mit der Übertragung auf die neue OLAT Version nichts und auch die Klassen bleiben unverändert. Lediglich der Aufruf des ESFLaunchControllers geschieht nach dem

Update nicht mehr im HomeMainController selber, sondern über eine Extension, welche als Funktion im Lastenheft genauer beschrieben ist.

Auch der Aufbau der Studentenakte selbst und deren Anzeige bleiben gleich. Da sich wenig an den OLAT Klassen verändert hat, sollte die Verwendung der Xman Pakete auch mit der neuen Version keine Probleme bereiten. Im nächsten Bild wird nochmal der Aufbau der Studentenakte in Form eines Klassendiagramms dargestellt. Weitere Informationen zum Aufbau und Darstellung entnehmen sie bitte der angegebenen Bachelorarbeit.

Verwaltet werden können Studentenakten von Personen mit der Rolle Prüfungsamtmitarbeiter und/oder Administrator. Hierfür haben diese eine eigene Seite, welche an anderer Stelle genauer erläutert wurde. Auf dieser können sie sich Studentenakten ansehen oder diese mit Hilfe der Controller validieren, erzeugen oder verändern. Auch an diesen Implementierungen ändert sich gegenüber der vorherigen OLAT Version nichts. Lediglich die Rolle des ExamAdmin, wird auf andere Art und Weise implementiert, was als Funktion im Lastenheft genauer dargestellt wurde.

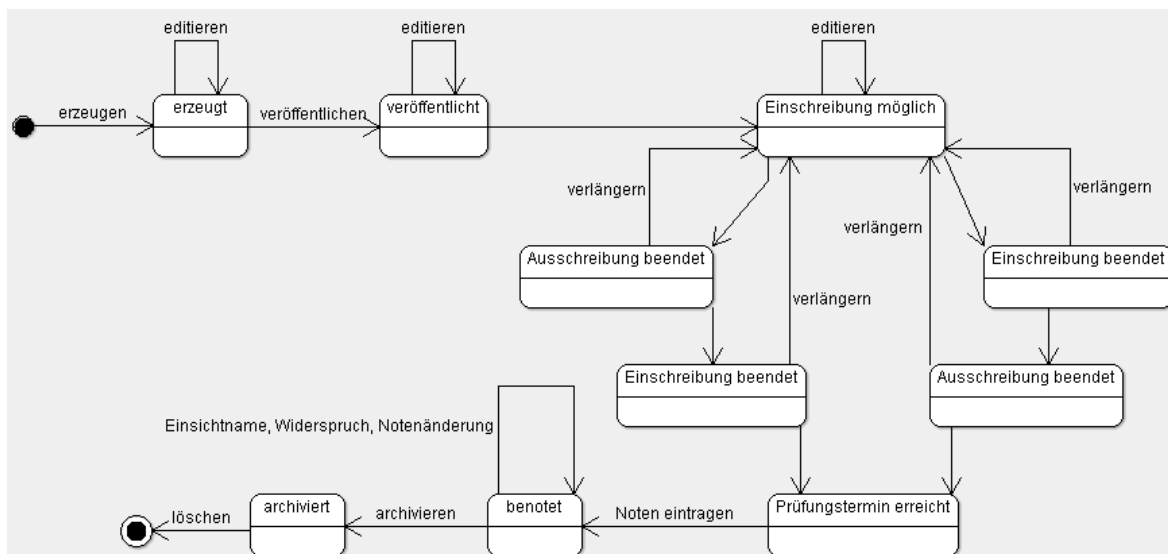


Klassendiagramm – Elektronische Studentenakte; Grafik aus Bachelorarbeit[Gerber 2009] Seite 26 Abb. 4.11

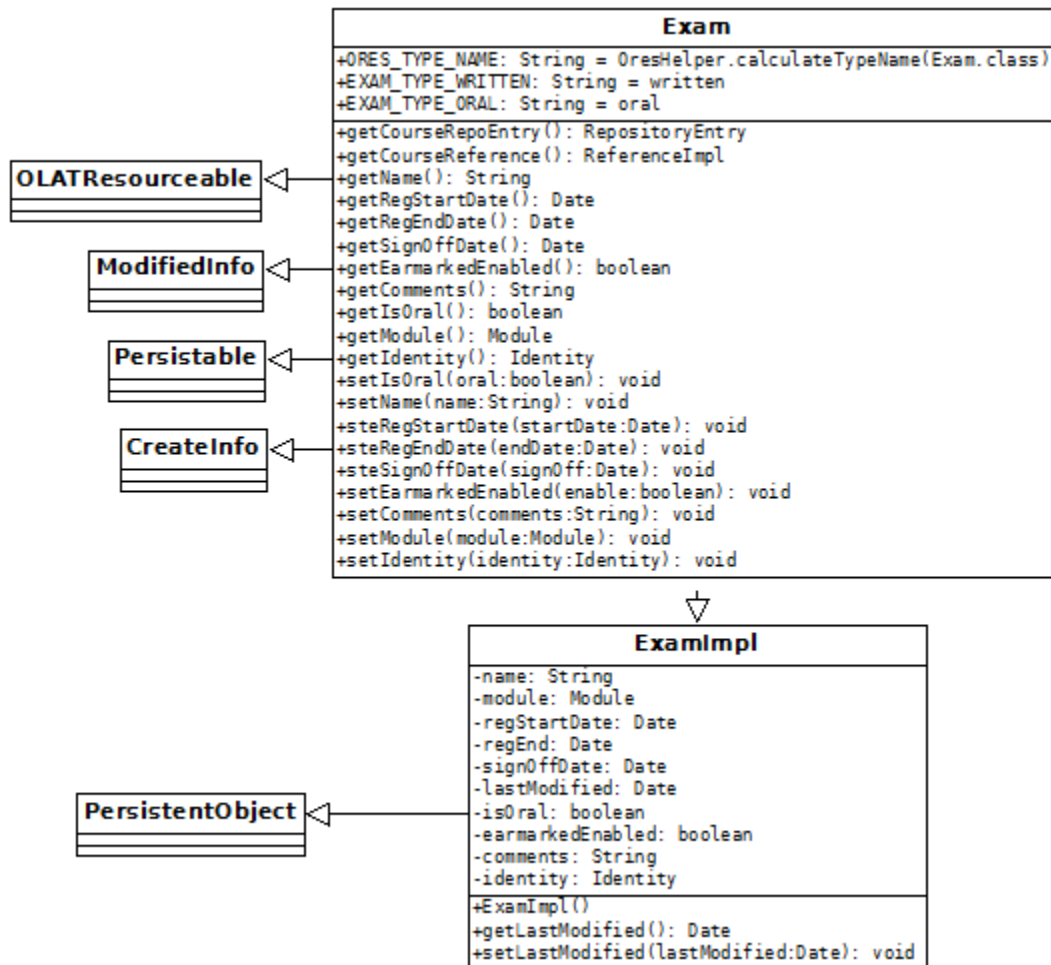
## Prüfung

Der grundsätzliche Lebenszyklus einer Prüfung ändert sich während des Updates nicht und wird im folgendem kurz dargestellt. Zunächst muss eine Prüfung durch einen entsprechenden Prüfungsverantwortlichen erzeugt, ihre relevanten Details spezifiziert, der Termin durch das Prüfungsamt bestätigt und schließlich veröffentlicht werden.

Nun können sich Studenten, vorausgesetzt sie besitzen eine validierte Studentenakten, zu der jeweiligen Prüfung an- und abmelden wobei es möglich ist die Prüfung bis zum Prüfungstermin zu editieren und insbesondere die Fristen für die An- und Abmeldung zu verändern. Der Verantwortliche für die Prüfung hat zudem die Möglichkeit Studenten manuell Ein- oder Auszutragen falls z.B. die jeweilige Frist verstrichen sein sollte. Eine solche Aktion zieht einen Vermerk in den Studentenakten des betroffenen Studenten nach sich. Nach erfolgreicher Durchführung und Korrektur der Prüfung können nun die Noten eintragen und durch die Studenten eingesehen werden. Es folgt die Einsichtnahme der Studenten in die Klausur und die Behebung eventueller Widerspruchsfälle, gegebenenfalls durch Korrektur der betroffenen Noten und Kommentarvermerke in die jeweiligen Prüfungsprotokolle. Ist die Frist zur Einsichtnahme abgelaufen werden die Ergebnisse der Prüfung an das Prüfungsamt übersendet und die Prüfung geschlossen. Nach einer bestimmten Frist (z.B. 1 Jahr) sollte die Prüfung gelöscht oder extern archiviert werden.



Prüfungen werden wiederum auf zwei verschiedene Arten interpretiert, als Lernressourcen sowie als Kursbaustein. Zur Realisierung des ersten Ansatzes werden Änderungen im OLAT-Code nötig. Analog wie in der jetzigen Version wird ein RepositoryHandler für das Entry Prüfung in der zugehörigen Factory registriert und in einer Klasse ExamHandler, die dann die nötigen Controller zur Verfügung stellt, implementiert. Der zweite Ansatz wird wiederum über den Erweiterungspunkt für Kursbausteine durchgeführt.

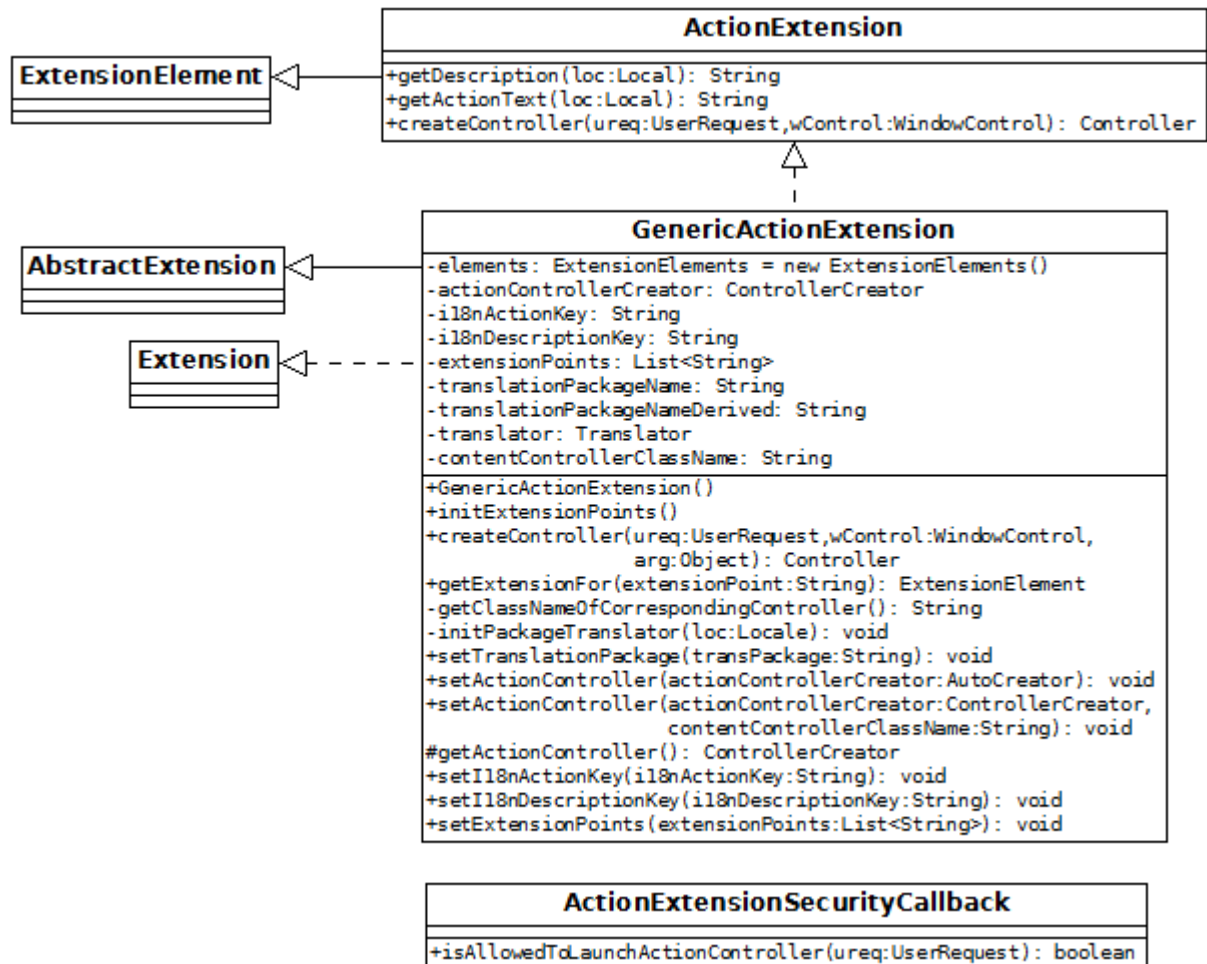


Das Interface Exam dient der Erstellung und Verwaltung von Prüfungen und enthält neben dem Namen der Prüfung, auch Daten über den Registrationszeitraum, Anmeldefrist, Art der Prüfung (mündlich oder nicht-mündlich) und weitere. Das Interface selbst verpflichtet bei der Implementation nur zur Definition von gettern und settern, mit denen diese Werte abgerufen bzw. gesetzt werden können. Es gibt zwei getter `getCourseRepoEntry()` und `getCourseReference()` welche erst genauer definiert werden müssen und nicht mit settern versehen wurden.

`ExamImpl` stellt eine einfache Implementation von `Exam` dar und realisiert die beiden zuletzt erwähnt getter durch einen Datenbankzugriff, um so die Kursreferenz der Prüfung zurückzugeben. Exam findet in folgenden OLAT-PA-Paketen Verwendung: `de.xman`

- appointment
- calendar
- catalog
- nodes
- portal
- protocol

## Actionextension



Das Interface ActionExtension erweitert das funktionslose Interface ExtensionElement und wird von GenericActionExtension implementiert. Actionextension dient hierbei als Extensionpoint in den Erweiterungen eingehangen werden können.

Bisher Findet Actionextension Verwendung in HomeMainControllerExtension, welche einen zusätzlichen Link innerhalb der von GenericActionExtension erstellten Linkliste einbindet. Der so erstellte Link ist Teil einer ActionExtension, die einen ExamListController enthält. Dieser Controller erstellt nach einer erfolgten Nutzerinteraktion mit dem Link eine Liste, welche die abgelegten Prüfungen des Studenten beinhaltet.

Geplant ist die Actionextension gemäß dem Beispiel com.xyz.demoextension um zu setzten um so den Extensionpoitn möglichst effektiv zu nutzen. Hierbei wird der Controller durch den unter org.olat.core.gui.control.creator befindlichen Autocreator durch Spring instantiiert. Durch diese Änderung ist es möglich den Controller durch eine Erweiterung von BasicController dar zu stellen, wodurch die Seperate Erzeugung von Actionextensions innerhalb von OLAT-PA vermieden werden kann.



swp11-2

Projektleiter : Matthias Haeßner | Dokument erstellt von Axel Fischer

---

Einen weiteren Einsatz wird der Extensionpoint der ActionExtension finden, um den ESF-LaunchController, welcher bisher durch Manipulation des OLAT-Codes ergänzt wurde, als Erweiterung einzuhängen. Dies bietet einen großen Vorteil : da der ESF-LaunchController durch diese Art der Implementation keine Änderungen am ursprünglichen System vornimmt werden zukünftige Updates erleichtert.