

## **Entwurfsbeschreibung**

### **1.) Allgemeines**

Das Wikiportlet ist ein webbasiertes Informationssystem, welches hauptsächlich an einer Universität eingesetzt werden kann. Es bietet eine Vielzahl an Möglichkeiten, um Vorlesungen oder Seminare begleitend zu unterstützen. Die hauptsächlichen Funktionen für den Anwender dienen der Information über Übungsgruppen, Übungsaufgaben, Prüfungen sowie sonstige für das Studium nützliche Informationen. Das Wiki soll in die eine bereits bestehende Portletoberfläche integriert werden, was damit eine Browser- und Betriebssystemunabhängigkeit garantiert. Der bereits bestehende Server selbst baut auf dem Prinzip der Java – Servlet – Technologie auf, die für eine solche Applikation sehr wichtig und hilfreich ist, da erst durch sie die Client Server Architektur umgesetzt werden kann und die wechselseitige Kommunikation realisierbar wird.

### **2.) Produktübersicht**

Das Wiki soll eine Vielzahl von interaktiven Möglichkeiten für den Nutzer bieten. Folgende grundlegende Funktionsübersicht stellt die wichtigsten Funktionen dar:

- Erstellen und anzeigen von Wiki-Seiten / -Artikeln
- Bearbeiten bereits erstellter Artikel
- Anzeigen und Wiederherstellen bereits vergangener Versionen von Artikeln
- Rollensystem für User (Unterteilung in Admin, Dozent, Student, Gast z.B.)
- Suchen nach Stichwörtern

Für die Benutzerrollen ergeben sich folgende Rechte, wobei die höhere Gruppe alle Aktionen der tiefergelegenen Gruppen durchführen kann.

Admin:

- Dateiverwaltung
- Artikelversionsverwaltung
- Konfiguration

Benutzer:

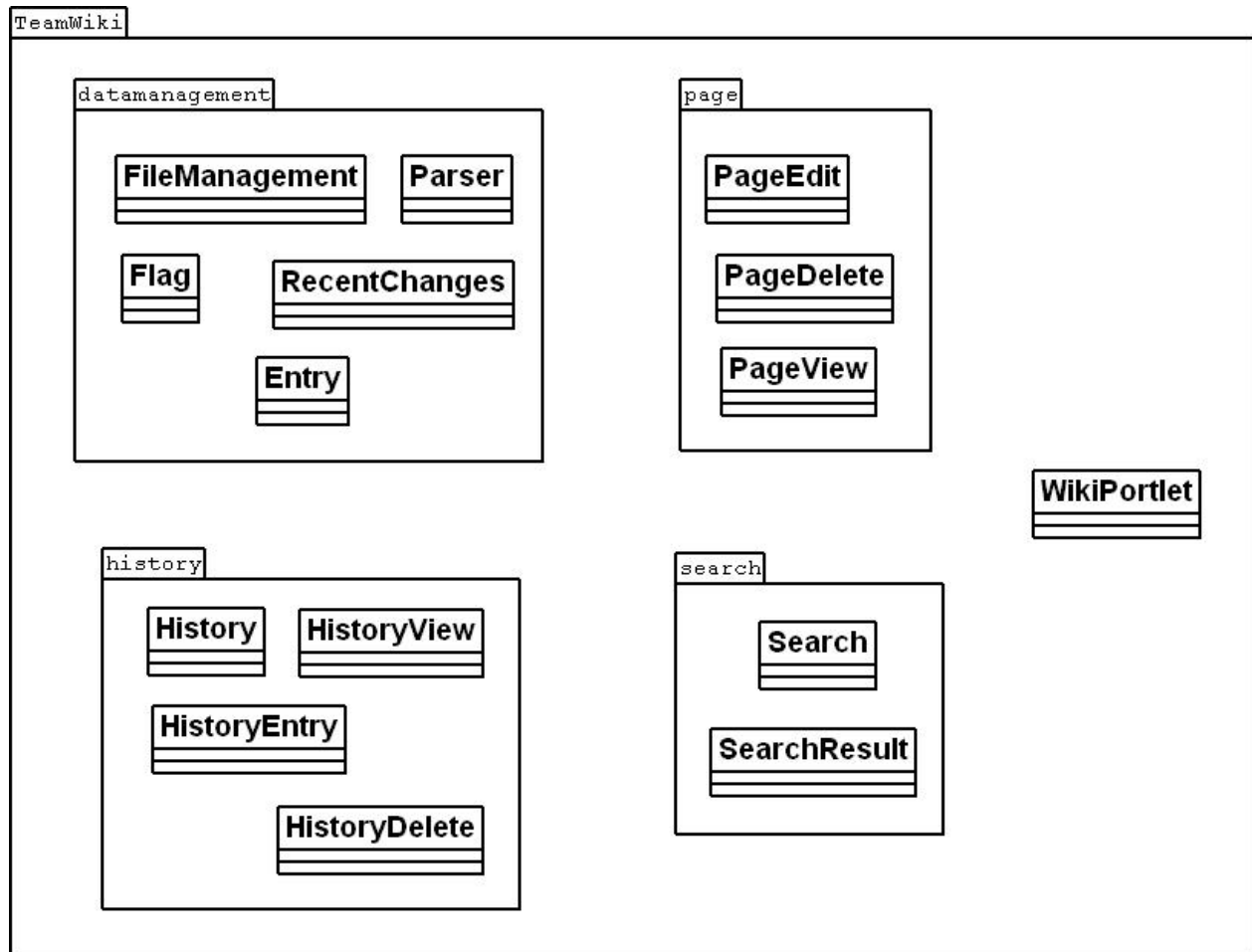
- Artikel erstellen und bearbeiten
- Alte Version eines Artikels wiederherstellen

Gast:

- Lesender Zugriff auf alle Artikel
- Suche durchführen

### 3.) Grundsätzliche Struktur- und Entwurfsprinzipien

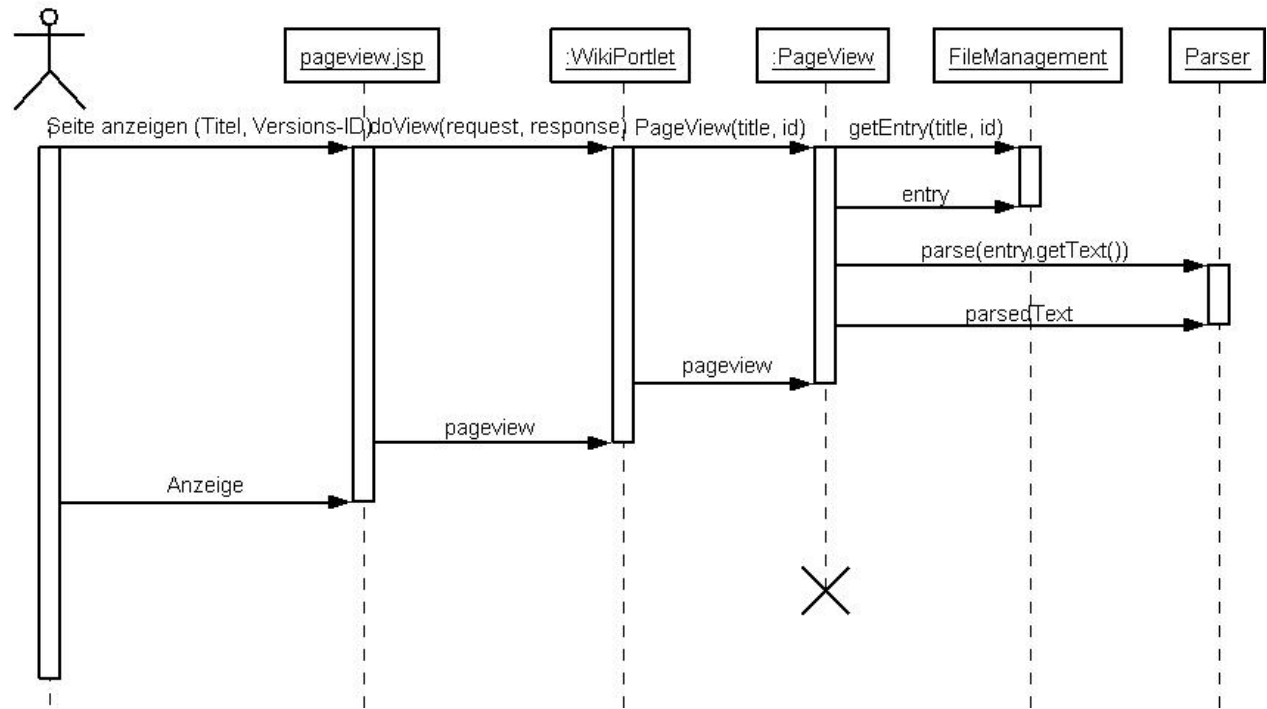
Das Projekt wird so geschrieben, dass es auf einen Portletcontainer läuft, welcher die Java Portlet Specification 1.0 (JSR 168) unterstützt und ist für Jetspeed 2 ausgelegt. Es gliedert sich in 4 Pakete, die in nachfolgendem Paketdiagramm dargestellt sind.



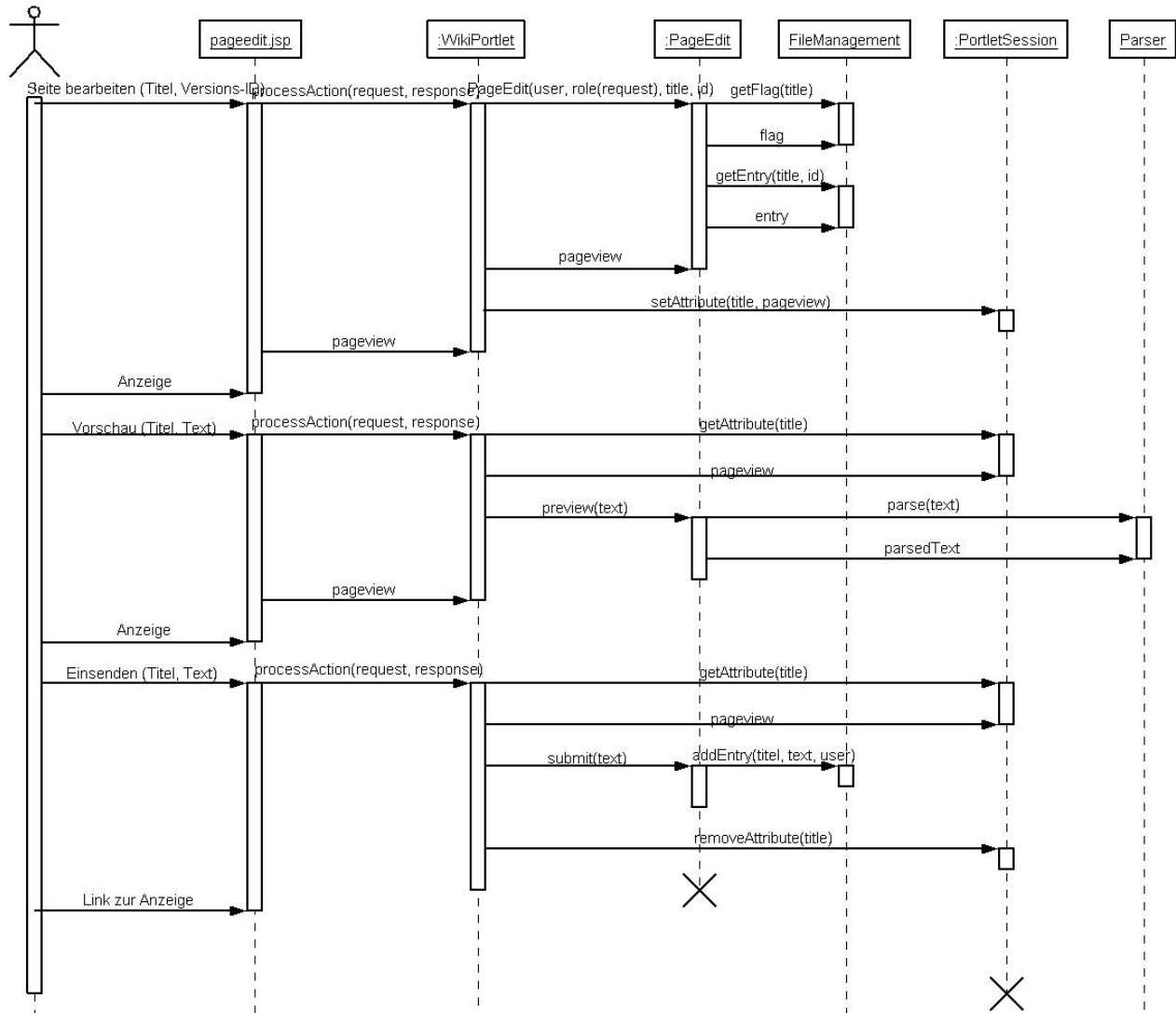
Dabei lassen sich die Pakete dem MVC-Konzept zuordnen. Die Klasse **WikiPortlet** bildet die Controllerkomponente. Die JSP-Dateien realisieren die View-Komponente. Die Pakete **page** und **datamanagement** kann man dem Model zuordnen. Bei **search** und **history** handelt es sich um eigenständigere Ergänzungsmodule, die ebenfalls zum Model gehören. Für den Ablauf ist **WikiPortlet** entscheidend. Dieses fängt Benutzereingaben, z.B. Buttonklick über die URL ab und erzeugt daraufhin Objekte des **Page**-Pakets, die dabei mit Hilfe der **datamanagement**-Klassen die Ausgabe erzeugen. Dann werden diese Objekte den JSP-Objekten übergeben, um die Ausgabe zu realisieren. Die **datamanagement**-Klassen sind hierbei für das Speichern und Lesen der Artikel, als auch für das Parsen des Textes oder andere organisatorische Sachen zuständig.

Für eine genauere Vorstellung folgen 2 Sequenzdiagramme, die das Anzeigen bzw. das Bearbeiten eines Artikels veranschaulichen.

**Anzeigen eines Artikels**



**Bearbeiten eines Artikels**



Mitglieder: Richard Beyer, Cong Bi, Sascha Gübner, Jörg Heinicke, Cornelius Müller  
Jing Wang, Christian Wieland

Für die Datenspeicherung wird XML eingesetzt, wobei jede XML-Datei ein Artikel darstellt. Diese Datei gliedert sich dann wie folgt:

Als Grundinformation stehen die System-Flags zur Verfügung, die die Rechte speichern:

- "Admin-darf-löschen"-Flag -> im allgemeinen true, außer bei systemwichtigen Artikeln, wie z. B. Mitarbeiter, Veranstaltungen oder RecentChanges
- "Benutzer-darf-bearbeiten"-Flag -> für Benutzer-Artikel true, für Systemartikel oder wichtige Artikel, die nicht verändert werden sollen false
- "in-Bearbeitung"-Datum -> gibt an, seit wann ein Artikel für die Bearbeitung gesperrt wurde
- "in-Bearbeitung"-Benutzer -> welcher Benutzer den Artikel gerade für die Bearbeitung gesperrt hält

Danach folgen Unterknoten, die die History darstellen. Jeder Knoten hat dabei die folgenden Informationen:

- wann wurde der Eintrag gemacht
- wer hat ihn gemacht
- Text des Eintrages
- ID

Diese Flags werden dabei auch in der Flag-Klasse genutzt und von ihr ausgewertet.

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

(das Klassendiagramm befindet sich aus Platzgründen auf Seite 8)

### **Paket datamanagement:**

In diesem Packet sind alle Klassen gesammelt, die mit der Datenverarbeitung zu tun haben. Die Hauptklasse ist hierbei FileManagement, über die jeder Zugriff auf einen Artikel läuft. Deshalb wird diese Klasse auch von anderen Paketen aus aufgerufen. Weiterhin finden sich hier die Recent Changes-Verwaltung und der Parser.

### FileManagement:

Diese Klasse ist eine zentrale Stelle der Datenverarbeitung. Sie regelt das Lesen und schreiben von Artikeln und läßt auch spezielle Informationen wie Flags und History aus. Dafür benutzt sie den XML-Helper um den Zugriff auf die Dateien zu vereinfachen. Weiterhin findet man in dieser Klasse Funktionen zum löschen der History oder zum löschen eines gesamten Artikels.

Mitglieder: Richard Beyer, Cong Bi, Sascha Gübner, Jörg Heinicke, Cornelius Müller  
Jing Wang, Christian Wieland

**Entry:**

Speichert alle Informationen zu einem Eintrag, der in einem Artikel steht. Dabei enthält sie die Information, wann, wer, welchen Text einschickte, sowie eine ID jedes Eintrages zur eindeutigen Identifizierung. Diese Klasse wird hauptsächlich zum Datenaustausch zwischen anderen Klassen benutzt.

**XMLHelper:**

Eine Schnittstelle zwischen FileManagement und der XML-Datei um Zugriffe zu kapseln und zu erleichtern.

**Flag:**

Zentrale Klasse des Rechtemanagements. Gibt an, welche Rechte ein Benutzer in einer bestimmten Wiki-Rolle hat. Viele Klassen, die die Benutzereingaben verarbeiten greifen hierauf zu, um zu prüfen, ob der User, in der entsprechenden Wiki-Rolle die nötigen Rechte für eine Aktion hat. Falls nicht kann auch eine Fehlernummer abgefragt werden.

**RecentChanges:**

Diese Klasse registriert Änderungen an Wikiseiten, und fügt einen Link auf kürzlich geänderte Seiten der speziellen Seite "RecentChanges" hinzu. Diese ist nur ein "normaler" Wikieintrag auf den im Menü verlinkt wird und der auch keine History besitzt.

**Parser:**

Dient zum Formatieren des Wiki-Codes nach HTML. Die wichtigste Funktion ist hierbei Parse(), die einen Wiki-Code einliest und ihn dann als in HTML formatierten Text ausgibt. Dabei wird auch das automatische Verlinken vorgenommen.

**Paket page:**

In diesem Paket sind die Klassen, die die eigentliche Darstellung der Funktionen übernehmen. Sie greifen auf Elemente des datamanagements zu und leiten ihre erzeugte Ausgabe mittels der get-Funktionen an die JSP-Dateien weiter.

**PageView:**

Diese Klasse hat die Aufgabe, eine Wikiseite anzuzeigen. Bereits bei der Erzeugung des Objekts holt sich der Konstruktor von der Klasse FileManagement alle nötigen Daten, und speichert sie. Der Text in Wikisyntax wird von der Klasse Parser in HTML-Code umgewandelt. Nun kann die JSP-Datei, welche sich um die Ausgabe kümmert, alle Daten, die sie zur Anzeige braucht, über die get-Methoden bekommen.

Diese Klasse ist allgemeingültig genug, dass sie auch zur Anzeige einer alten Version benutzt werden kann (indem im Konstruktor deren ID übergeben wird), oder zur Druckansicht (dies bedeutet keine Änderung im Model, nur im View).

**PageEdit:**

Diese Klasse kümmert sich um das Editieren von Wikiseiten. Ähnlich wie bei PageView werden alle zu Anfang benötigten Daten im Konstruktor besorgt und der Ausgabe zur Verfügung gestellt. Dabei wird mithilfe der Klasse Flag überprüft, ob der Benutzer die nötigen Rechte besitzt, und der Zugriff auf diesen Artikel für andere Bearbeiter gesperrt. Die Methode preview aktualisiert die Vorschau, submit fügt die Änderungen endgültig hinzu. Außerdem wird bei submit auch die Klasse RecentChanges aufgerufen, welche registriert, dass sich die Seite geändert hat. Gleichzeitig kann auch das Flag geändert werden,

Mitglieder: Richard Beyer, Cong Bi, Sascha Gübner, Jörg Heinicke, Cornelius Müller  
Jing Wang, Christian Wieland

welches bestimmt, ob auch Benutzer die Seite editieren dürfen, und nicht nur Administratoren. Da im Konstruktor die ID des Wikieintrags angegeben werden kann, können mit dieser Klasse auch alte Versionen wiederhergestellt werden.

**PageDelete:**

Mithilfe diese Klasse können Wikiseiten vollständig gelöscht werden. Dabei wird mithilfe der Klasse Flag überprüft, ob der Benutzer die nötigen Rechte besitzt.

### **Paket history:**

In diesem Paket sind die Klassen gesammelt, die das Wiki um die Funktionalität des Verlaufs erweitern.

**History:**

Diese Klasse wird von der Funktion FileManagement.getHistory() zurückgegeben. In dieser Klasse sind die Kurzinformationen, wie Änderungsdatum und Benutzer, der änderte gespeichert. Diese Daten betreffen den Verlauf einer Datei.

**HistoryEntry:**

Die Klasse speichert die Daten eines Eintrages der History-Liste (siehe oben).

**HistoryView:**

Stellt die Komponente dar, die für das Anzeigen der Informationen über den Verlauf zuständig ist.

**HistoryDelete:**

Mithilfe diese Klasse kann die gesamte History einer Wikiseiten gelöscht werden. Dabei wird mithilfe der Klasse Flag überprüft, ob der Benutzer die nötigen Rechte besitzt.

### **Paket search:**

Umfasst alle Funktionen und Anzeigen, die die Suche betreffen.

**Search:**

Führt eine Suche durch. Dabei werden alle Artikel (also alle xml-Dateien) im Volltext durchsucht und anhand der Häufigkeit der gesuchten Wörter und anhand der Aktualität des Artikels ein Ranking erstellt. Über getResults() werden die Ergebnisse dann angezeigt.

**SearchResult:**

Speichert die Informationen über einen Treffer in der Suche. Diese kurzen Informationen werden dann benötigt um die Suchergebnisse auszugeben.

### **WikiPortlet:**

Diese Klasse erweitert GenericPortlet und stellt damit die Schnittstelle zum Portletcontainer dar (in unseren Fall elatePortal bzw. Jetspeed). Die zentrale Methode ist doView, welche die Eingaben vom Browser der Clients entgegennimmt und ein entsprechendes Objekt erzeugt (PageView, PageDelete, HistoryView, HistoryDelete oder Search). Dieses Objekt wird dann JSP übergeben, welches sich um die Anzeige kümmert.

Mitglieder: Richard Beyer, Cong Bi, Sascha Gübner, Jörg Heinicke, Cornelius Müller  
 Jing Wang, Christian Wieland

Eine Ausnahme stellt PageEdit dar: Dieses Objekt wird in der Methode processAction erzeugt, und eine Referenz auf das Objekt in PortletSession gespeichert. Wird dann später z. B. Vorschau gedrückt, so wird das bereits vorhandene Objekt aus PortletSession ausgelesen und weitergenutzt.

Um portletspezifische Dinge vom eigentlichen Model zu trennen, kümmert sich die Methode role um das Mappen von Benutzerrollen des Portletcontainers auf die Rollen im Wiki.

