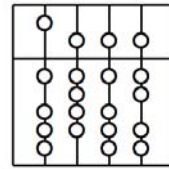




Technische Universität München
Fakultät für Informatik



Diplomarbeit in Informatik

ParLe **(Participating Lecture)** eine eLearningplattform für partizipatorische Vorlesungen

Hans Breidler
Gabelsbergerstr. 85
80333 München

Abgabedatum:
15.11.2006

Aufgabensteller:
Prof. Bernd Brügge, Ph. D.
Lehrstuhl für angewandte Softwaretechnik

Betreuer:
Dipl. Inf. Michael Nagel

Ich versichere, dass ich diese Diplomarbeit selbständig verfasst und nur die angegebenen Quellen und Hilfsmittel verwendet habe.

15.11.2006

Inhaltsverzeichnis:	Seite
1 Anforderungsanalyse.....	4
1.1 Einführung.....	4
1.1.1 Zweck des Systems.....	4
1.1.2 Umfang des Systems.....	4
1.1.3 Ziele und Erfolgskriterien.....	5
1.2 Vorgeschlagenes System.....	5
1.2.1 Übersicht.....	5
1.2.2 Funktionale Anforderungen.....	5
1.2.3 Nichtfunktionale Anforderungen.....	6
1.3 Systemmodell.....	8
1.3.1 Szenarien.....	8
1.3.2 Anwendungsfallmodell für den Akteur Student.....	10
1.3.3 Anwendungsfallmodell für den Akteur Administrator.....	15
1.4 Objektmodell.....	17
1.4.1 Gesamtmodell mit allen Objekten.....	17
1.4.2 Objektmodell der Entitätsobjekte.....	23
1.4.3 Klassendiagramme.....	23
1.4.4 Dynamisches Modell.....	28
1.4.5 Benutzerschnittstelle.....	33
2 Systementwurf.....	36
2.1 Entwurfsziele.....	36
2.2 Vorgeschlagene Softwarearchitektur.....	37
2.2.1 Überblick.....	37
2.2.2 Systemzerlegung in Schichten.....	38
2.2.3 Systemzerlegung in Subsysteme.....	40
2.2.4 Abbildung Hardware- / Software Komponenten.....	45
2.2.5 Management persistenter Daten.....	46
2.2.6 Zugriffskontrolle.....	50
2.2.7 Kontrollfluss.....	50
2.2.8 Randbedingungen.....	51
3 Objektentwurf.....	53
3.1 Einleitung.....	53
3.2 Pakete.....	54
3.3 Schnittstellenspezifikation.....	55
3.3.1 Die Klassen des ParLe Servers.....	55
3.3.2 Die Klassen des ParLe Client.....	56
3.3.3 Erweiterung der Klassenschnittstellen um ein Administrationstool.....	67
4 Testen.....	70
4.1 Komponententest.....	70
4.2 Funktionstest.....	71
5 Benutzerhandbuch.....	75
5.1 Benutzung des ParLe Systems durch den Studenten.....	75
5.2 Datenpflege durch den Administrator.....	78
5.2.1 Installation des Systems.....	78
5.2.2 Einstellen der Daten in die Datenbank.....	79
5.2.3 Verwaltung des laufenden Systems.....	80
6 Literaturverzeichnis.....	82
7 Anhang.....	83

1 Anforderungsanalyse

1.1 Einführung

Die neuen Informations- und Kommunikationstechnologien ermöglichen auch im Bereich „Lernen“ Veränderungen. Die Psychologie reagiert darauf, indem die Forschung sich intensiv damit befasst, wie mit den neuen Möglichkeiten kommuniziert und gelernt wird. Die pädagogische Psychologie richtet ihre Aufmerksamkeit zunehmend auf die Analyse selbstgesteuerter Lernprozesse. Selbstgesteuertes Lernen ist eine Voraussetzung für erfolgreiches Lernen. Bei den verschiedenen Phasen selbstregulierten Lernens kann der interaktive Austausch mit Kommilitonen die Motivationsbildung beeinflussen.

Motivation wird zu einem relevanten Faktor, wenn Studenten oder Lernende selbstgesteuert in medialen Lernangeboten navigieren.

Das Interesse, das ein Lernender gegenüber den Lerninhalten entwickelt, beeinflusst positiv Verarbeitungstiefe und Behaltensleistung. Mediale Lernangebote aktualisieren dann Interesse, wenn sie Interaktivität ermöglichen und leistungsorientiertes individuelles Feedback anbieten. ParLe ist so ein interaktives System, bei dem Studenten mit gleichem Interesse und ähnlichem Kenntnisstand in einer Art strukturiertem Chat miteinander kommunizieren. ParLe bietet ein Bewertungssystem und einen Filtermechanismus an. Durch das Bewertungsschema wird zusätzlich Motivation erzeugt, und durch das Filtersystem wird versucht, die Überfülle von Informationen und mangelnder Bezogenheit der Beiträge zu verringern.

1.1.1 Zweck des Systems

Mit dem zu entwickelnden System ParLe (Participating Lectures) soll in einem interaktiven Prozess erreicht werden, dass von Dozenten angebotene Lehrinhalte durch Kommentierungen von Studenten erweitert und inhaltlich aufbereitet und diskutiert werden. Die Lehrinhalte von Vorlesungen sind in Lehreinheiten unterteilt, zu diesen Lehreinheiten gehört in der Regel ein Satz von Medien in Form von Folien.

Es soll ein elektronisches Diskussionsforum erstellt werden, das es Studenten erlaubt, die zu den Lehreinheiten angebotenen Medien zu kommentieren. Das kann geschehen in Form von Fragen, Ergänzungen und Antworten.

1.1.2 Umfang des Systems

Der Student soll sich auf seinem Arbeitsplatz über ein Fenster Lehreinheiten anzeigen lassen können. Lehreinheiten und Medien sollen ausgewählt werden können. Die zu den Lehreinheiten gehörenden Medien sollen ausgeklappt werden können. In einem zweiten Fenster sollen zur ausgewählten Lehreinheit oder zum ausgewählten Medium bereits vorliegende Kommentare in Form ihrer Titel angezeigt werden. Der Student soll nun gewünschte Titel ausklappen können, um sie zu lesen. Er kann nun selbst Kommentare zu Lehreinheiten, Medien oder bereits eingestellten Kommentaren verfassen und ins System einstellen.

Alle persistenten Daten des ParLe Systems sollen auf einem Datenverwaltungssystem gespeichert werden.

Um die Relevanz von Kommentaren festzustellen, sollen Studenten Bewertungen zu Kommentaren abgeben können. Da das System nicht durch Mehrfachbewertungen

ausgetrickst werden soll, darf ein Student einen Kommentar nur einmal bewerten, und eigene Kommentare dürfen nicht bewertet werden.

Um gut bewertete Kommentare zuerst angezeigt zu erhalten, soll der Student eine Sortierung nach Relevanz vornehmen können. Will der Student nicht mit Informationen überschüttet werden, kann er einen Filter wählen, der nur Kommentare zulässt, die mindestens eine bestimmte Bewertung haben.

1.1.3 Ziele und Erfolgskriterien

Das Ziel ist es, ein lauffähiges Programm zu erstellen, das den Zweck des Systems erfüllt und aus Kostengründen mit Open Source Software (z.B. Datenbank) implementiert ist und aus Kompatibilitätsgründen in Java erstellt wird. Es soll ein Client – Server System erstellt werden; zum Aufruf der Serverobjekte soll Remote Method Invocation (RMI) verwendet werden. Die Daten sollen in einem geeigneten Datenverwaltungssystem gespeichert werden. Zum Umfang der Diplomarbeit gehört es nicht, ein Administrationssystem zum Verwalten der eingestellten Daten (Lehreinheiten, Medien und Nutzer) zu programmieren. Im Benutzerhandbuch wird jedoch erläutert, wie die Daten vom Administrator eingepflegt werden können.

1.2 Vorgeschlagenes System

In diesem Kapitel werden zunächst funktionale und nichtfunktionale Anforderungen behandelt. Danach wird das funktionale Modell mit Anwendungsfällen und Szenarien beschrieben, ein Objektmodell erzeugt, Klassendiagramme zu Anwendungsfällen gebildet, und ein dynamisches Modell mit Sequenzdiagrammen gezeigt.

1.2.1 Übersicht

Es soll ein hierarchisches Diskussionsforum bereitgestellt werden, das Folgendes ermöglicht:

- Darstellung von Lehreinheiten und/oder zugehöriger Medien
- Darstellung von Kommentaren zu ausgewählten Lehreinheiten oder Medienelementen
- Erstellen von Kommentaren zu einer bestimmten Lehreinheit und/oder zu einem zugehörigen Medienelement dieser Lehreinheit
- Bereitstellen eines Bewertungssystems, um die Relevanz von Kommentaren zu bewerten
- Bereitstellen eines Filters, der Kommentare unterdrückt, die schlechter bewertet sind als die ausgewählte Qualitätsstufe
- Implementierung eines Relevanzsortierungsmechanismus, der Kommentare nach der Bewertung sortiert darstellt
- Ändern des Passworts eines registrierten Nutzers

1.2.2 Funktionale Anforderungen

„Eine funktionale Anforderung ist die Spezifikation einer Funktion, die vom System unterstützt werden muss.“ (Brügge, 2004, S.38)₁ Funktionale Anforderungen beschreiben „die Interaktion des Systems mit seiner Umgebung unabhängig von seiner Implementierung.“ (Brügge, 2004, S. 149)₁

Anwender des ParLe Systems sind Studenten, die Vorlesungen (Lectures) hören, zu denen ein ParLe System angeboten wird. Mit dem System interagiert der Student auf folgende Weise:

- Dem Studenten sollen beim ParLe Start alle Lehreinheiten in einem Fenster angezeigt werden. Der Student soll Kommentare zu einer Lehreinheit selbst aufrufen können.

Aus dem Fenster mit allen Lehreinheitstiteln soll eine Lehreinheit auswählbar sein, und die dazugehörigen Kommentare sollen in einem anderen Fenster dargestellt werden.

- Der Student soll alle Medien zu einer gewählten Lehreinheit in Form einer kleinen Vorschau darstellung ausgeklappt erhalten können.
- Der Student soll einzelne Medienelemente auswählen und vergrößern können.
- Der Student soll Kommentare zu einem gewählten Medium auswählen können und sich die Kommentartitel dazu in einem Kommentarfenster ansehen können.
Die Kommentare sollen hierarchisch gegliedert sein. Kommentare der obersten Hierarchiestufe sollen sich direkt auf die gewählte Lehreinheit bzw. das gewählte Medienelement beziehen. Kommentare einer niedrigeren Stufe beziehen sich auf den Kommentar der nächsthöheren Stufe.
- Der Student soll Kommentare lesen können. Dazu soll er die Kommentare selbst ausklappen können. Neben dem Kommentartext soll er auch Name des Autors, Datum der Erstellung und gerundete durchschnittliche Bewertung durch andere Studenten dargestellt erhalten.
- Der Student soll Kommentare verfassen können. Er soll dabei die Möglichkeit haben, Kommentare zu Lehreinheiten, Medienelementen oder bereits verfassten Kommentaren (als Unterkommentar) erstellen zu können.
Bei Kommentaren zu Lehreinheiten oder Medienelementen soll die Einheit markiert sein. Bei einem Unterkommentar soll der Oberkommentar markiert sein. Die Kommentare und deren Titel sollen in ein Formular eingegeben werden können. Der Student soll sich nicht um Zeilenumbruch kümmern müssen. Die Daten über Autor und Erstellungsdatum sollen vom System automatisch vergeben werden.
- Der Student soll Kommentare von anderen nur einmal bewerten können. Dazu soll ihm ein Formular angeboten werden, mit dem er die Bewertung 1 (irrelevant) bis 5 Sterne (höchst zutreffend) vornehmen kann.
- Der Student soll einen Filter setzen können. Dazu soll ihm ein Formular angeboten werden, bei dem er sowohl die Sortierreihenfolge als auch die Filterbedingung wählen kann. Bei der Sortierreihenfolge soll es die Alternativen:
 - o Sortierung nach Erstellungszeit
 - o Sortierung nach Relevanzgeben.

Die Lehreinheiten und deren Medien und die Nutzernamen müssen von einem Administrator direkt in das Datenverwaltungssystem eingegeben werden. Das in der Diplomarbeit erstellte ParLe System stellt dazu keine Anwendungslogik zur Verfügung. Im Benutzerhandbuch wird beschrieben, wie diese Daten eingepflegt werden können. In den Modellen ist die Erweiterung von Klassen beschrieben, um ein Administrationstool dafür zu erstellen.

Der Administrator vergibt bei der Registrierung von Studenten ein Passwort, das der Student später selbst ändern kann.

Der Administrator kann Kommentare und Nutzer sperren. Kommentare sollen dann gesperrt werden, wenn sie politisch inkorrekt sind. Nutzer sollen dann gesperrt werden, wenn sie nicht mehr am ParLe System teilnehmen sollen.

1.2.3 Nichtfunktionale Anforderungen

Hier werden Anforderungen beschrieben, die nicht unmittelbar Bezug zur Funktionalität haben, aber wichtig für die Nutzung sind.

Bedienbarkeit:

Die Benutzung des ParLe Systems soll einfach und weitgehend selbsterklärend sein. Unter einfach wird verstanden, dass eine Transaktion mit maximal drei Mausklicks erledigt werden kann. Unter weitgehend selbsterklärend wird verstanden, dass der Student nur ein kurzes, mit Bildern angereichertes Benutzerhandbuch lesen muss.

Die Anwendung soll ein Hauptfenster haben, das ein Fenster für Lehreinheiten und Medien zur Verfügung stellt und ein davon getrenntes Fenster zur Darstellung der Kommentare.

Durch farbliche Hinterlegung soll eindeutig dargestellt werden, welche Kommentare zu welchen Lehreinheiten oder Medien gehören.

Um zwischen Hierarchiestufen der Kommentare eindeutig unterscheiden zu können, sollen Unterkommentare eingerückt werden.

Die Fenster für Lehreinheiten/Medien und Kommentare sollen sich dynamisch anpassen können.

Um die Bedienung einfach zu halten, sollen Formulare durch die Betätigung von Schaltflächen aufgerufen werden; es sollen nur wenige Schaltflächen angeboten werden für:

- Kommentar eingeben
- Filtern und Sortieren
- Bewerten

Die Schaltfläche „Kommentar“ soll zu einem markierten Kommentar ein Formular bereitstellen, in das ein Kommentar eingegeben werden kann.

Zuverlässigkeit:

Das ParLe System soll mit einem Systemausfall zurechtkommen. Die Ausfallzeit zentraler Komponenten sollte unter zwei Stunden liegen. Persistente Daten auf dem Datenverwaltungssystem sollen nach einem Wiederanlauf konsistent verfügbar sein. Noch nicht abgespeicherte Daten laufender Transaktionen können bei einem Systemausfall verloren gehen.

Zugangsschutz:

Mit dem ParLe System sollen nur Studenten arbeiten können, die registriert sind. Die Registrierung soll über eine zentrale Stelle (Administrator) erfolgen. Die Freigabe des ParLe Systems soll durch ein personenbezogenes Passwort geschützt werden; das Passwort ist vom Studenten änderbar.

Leistungsanforderungen:

Das Programm soll die gleichzeitige Benutzung von mehreren Hundert Nutzern gewährleisten, dabei muss jeder Student, der bei ParLe teilnimmt, ein ParLe Teilsystem auf seinem Rechner installiert haben.

Die Antwortzeit und der Durchsatz werden durch die Leistungsfähigkeit des entfernten lehrstuhleigenen Teilsystems bestimmt.

Implementierungsanforderungen:

ParLe soll plattformunabhängig implementiert sein und auf Hardwareplattformen und Betriebssystemen laufen, die im universitären und studentischen Umfeld weit verbreitet sind.

Bei der Implementierung sollte auch auf wirtschaftliche Aspekte geachtet werden. Es sollten möglichst Produkte verwendet werden, für die keine Lizenzgebühren anfallen.

1.3 Systemmodell

Das Systemmodell beschreibt die funktionalen, strukturellen und dynamischen Modelle des Systems.

1.3.1 Szenarien

„Ein Szenario ist eine Instanz eines Anwendungsfalls.“ (Brügge, 2004, S.160)₁ Für das ParLe System wurden die beiden nachfolgend aufgeführten Szenarien vorgegeben. Diese Szenarien sind der Ausgangspunkt, um die Anwendungsfälle des ParLe Systems zu bestimmen und das Anwendungsfalldiagramm zu erstellen.

Szenario 1: KommentarErstellung:

Szenarioname:	KommentarErstellung
Akteurinstanzen:	Anna Student, Stefan Student, Georg Student, Bernd Dozent
Ereignisfluss:	<ol style="list-style-type: none"> 1.) Prof. Bernd zeigt Folien zur Lehrinheit Strategy Pattern 2.) Studentin Anna hat eine Frage zu Strategy Pattern, die nicht in der Präsentation beantwortet wird. Deswegen wählt sie in ihrem ParLe Client die Lehrinheit Strategy Pattern aus. 3.) Anna erstellt in Form eines Kommentars ihre Frage. 4.) Student Georg liest Annas Frage im ParLe Client als neuen Eintrag. 5.) Student Georg antwortet durch eine Gegenfrage (zusätzliche Frage). 6.) Student Stefan sieht Annas Kommentar und Georgs Gegenfrage im ParLe Client. 7.) Er weiß die Antworten und beantwortet in seinem ParLe Client Annas Frage und Georgs Gegenfrage in einem eigenen Eintrag.

Tabelle 1: Szenario für die **Erstellung von Kommentaren**

Szenario 2: KommentarBewertung:

Szenarioname:	KommentarBewertung
Akteurinstanzen:	Thorsten Student, Georg Student, Anna Studentin, Stefan Student, Doris Studentin, Kathrin Studentin, Studenten
Ereignisfluss:	<ol style="list-style-type: none"> 1.) Studentin Doris liest in ihrem ParLe Client Stefans Antwort auf Annas Frage. 2.) Sie findet die Antwort interessant und bewertet sie positiv (3-5 Sterne) über ihren ParLe Client. 3.) Sie findet Georgs Frage unsinnig und bewertet sie negativ(1-2 Sterne). 4.) Ebenso kommentieren 15 weitere Studenten. 5.) Die Studentin Kathrin hat in ihrem ParLe Client einen Filter eingestellt, der alle Kommentare mit weniger als 3 Sternen ausschließt. Sie sieht deswegen Georgs Gegenfrage und deren Antwort und die weiteren anhängenden Kommentare nicht.

	6.) Student Thorsten hat keinen Filter eingestellt, aber eine Relevanzsortierung nach Bewertungen (die besten stehen dabei oben). Er sieht Stefans Antwort auf Annas Frage vor Georgs Gegenfrage.
--	---

Tabelle 2: Szenario für die **Bewertung, Filterung und Sortierung von Kommentaren**

Für die Administration des Systems werden zwei zusätzliche Szenarien notwendig, die sich mit der Verwaltung des Systems befassen. Es geht hier um die Aufnahme von Lehreinheiten und Medien in das ParLe System und die Registrierung von Nutzern.

Szenario 3: LehreinheitUndMedienVerwaltung:

Szenarioname:	LehreinheitUndMedienVerwaltung
Akteurinstanzen:	Administrator Hans
Ereignisfluss:	<ol style="list-style-type: none"> 1.) Hans gibt eine Lehreinheit in das ParLe System ein. 2.) Die Lehreinheit erscheint im Medienfenster. 3.) Hans wiederholt diesen Vorgang, bis alle Lehreinheiten eingegeben sind. 4.) Hans möchte nun zwischen zwei Lehreinheiten eine neue einschieben; dazu markiert er die Lehreinheit, vor der die neue Lehreinheit erscheinen soll. 5.) Hans möchte nun Medien zu einer Lehreinheit einfügen. Er markiert hierzu die Lehreinheit und gibt die Medien ein. 6.) Er wiederholt dies, bis alle Medien eingegeben sind. 7.) Hans möchte ein neues Medium zwischen zwei vorhandene Medien einschieben; dazu markiert er das Medium, vor dem das neue Medium stehen soll. Er gibt das neue Medium ein. 8.) Hans möchte eine Lehreinheit oder ein Medium ändern. Er markiert die Lehreinheit bzw. das Medium und wählt die Änderungsfunktion. 9.) Hans möchte eine Lehreinheit oder ein Medium löschen. Er markiert die Lehreinheit oder das Medium und löscht die Daten.

Tabelle 3: Szenario für die **Verwaltung von Lehreinheiten und Medien** durch den Administrator

Szenario 4: NutzerUndKommentarVerwaltung:

Szenarioname:	NutzerUndKommentarVerwaltung
Akteurinstanzen:	Administrator Hans
Ereignisfluss:	<ol style="list-style-type: none"> 1.) Hans möchte einen Nutzer anlegen und benutzt dazu das Nutzereingabeformular. 2.) Er wiederholt diesen Vorgang bis alle Nutzer angelegt worden sind. 3.) Hans will einen Nutzer sperren und veranlasst dies über das Nutzereingabeformular. 4.) Hans will einen Kommentar sperren, markiert ihn dazu und lässt ihn sperren.

Tabelle 4: Szenario für **Nutzerverwaltung und Kommentarsperrung**

1.3.2 Anwendungsfallmodell für den Akteur Student

Aus den Szenarien lassen sich für den Akteur Student zunächst zwei übergeordnete Anwendungsfälle mit vier beziehungsweise drei Anwendungsfällen identifizieren:

- LeseUndErstelleKommentare enthält:
 - o AnmeldeNutzer
 - o WähleMedium
 - o LeseKommentare
 - o ErstelleKommentare
- FiltereUndSortiereKommentare enthält:
 - o BewerteKommentar
 - o FiltereKommentare
 - o SortiereKommentare

Ein Verbindungsabbruch zu einem entfernten System erweitert die Anwendungsfälle um den Anwendungsfall ClientServerVerbindungUnterbrochen.

Anwendungsfalldiagramm für den Akteur Student:

Das Anwendungsfalldiagramm zeigt die Funktionalität des Systems aus Sicht des Akteurs Student:

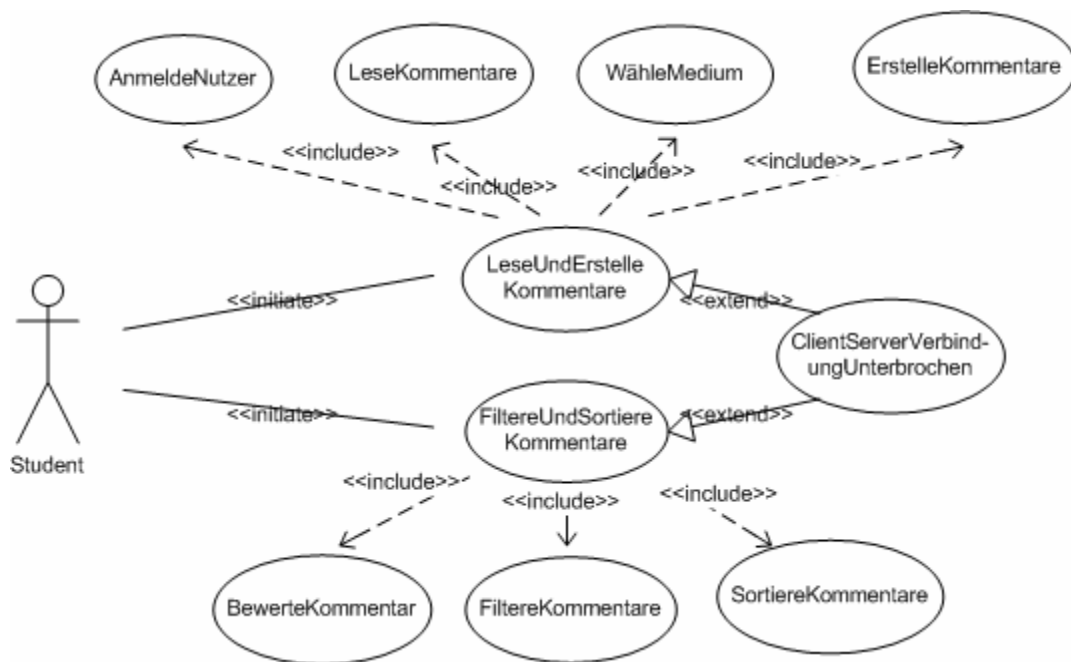


Abbildung 1: Anwendungsfalldiagramm für das ParLe System aus Sicht des **Akteurs Student**. Die Abbildung zeigt die einzelnen Anwendungsfälle, ihre Initialisierung durch den Akteur Student, ihre Enthält- und Erweitertbeziehungen.

Anwendungsfälle:

Nachfolgend werden diese Anwendungsfälle beschrieben.

Anwendungsfall 1: LeseUndErstelleKommentare

Anwendungsfallname:	LeseUndErstelleKommentare
Akteure:	Student

Ereignisfluss:	<ol style="list-style-type: none"> 1.) Der Student meldet sich über ein Anmeldeformular beim System an (beinhaltet den Anwendungsfall AnmeldeNutzer). 2.) Der Student kann eine Lehreinheit oder ein Medium im Medienfenster auswählen und sich dazu die Kommentartitel zeigen lassen (beinhaltet Anwendungsfall WähleMedium). 3.) Der Student kann nun Kommentare im Kommentarfenster lesen (beinhaltet den Anwendungsfall LeseKommentare). 4.) Der Student kann eigene Kommentare (Fragen, Antworten und Ergänzungen) über das Kommentareingabeformular in ParLe einstellen (beinhaltet den Anwendungsfall ErstelleKommentare).
Anfangsbedingung:	Der Student ist bei ParLe registriert.
Abschlussbedingung:	-
Qualitätsmerkmale:	Bei Verbindungsabbruch zum entfernten System tritt Anwendungsfall ClientServerVerbindungUnterbrochen auf.

Tabelle 5: Anwendungsfall Lesen und Erstellen von Kommentaren

Anwendungsfall 2: AnmeldeNutzer

Anwendungsfallname:	AnmeldeNutzer
Akteure:	Student
Ereignisfluss:	<ol style="list-style-type: none"> 1.) Der Student startet das ParLe Programm. 2.) Beim Start des Systems stellt ParLe ein Anmeldeformular bereit. 3.) Der Student gibt Name und Passwort ein und schickt das Formular ab. 4.) Das ParLe System vergleicht die eingegebenen Daten mit den gespeicherten Nutzerdaten im Datenverwaltungssystem und gibt bei Übereinstimmung das ParLe System für die Nutzung frei. Der Student ist als Nutzer identifiziert.
Anfangsbedingung:	Der Student ist bei ParLe registriert
Abschlussbedingung:	Freigabe des Systems bei Übereinstimmung der Nutzerdaten, ansonsten wird ParLe geschlossen.
Qualitätsmerkmale:	Bei Verbindungsabbruch zum entfernten System tritt Anwendungsfall ClientServerVerbindungUnterbrochen auf.

Tabelle 6: Anwendungsfall Anmelden eines Nutzers

Anwendungsfall 3: WähleMedium

Anwendungsfallname:	WähleMedium
Akteure:	Student
Ereignisfluss:	<ol style="list-style-type: none"> 1.) Der Student wählt im Medienfenster eine angebotene Lehreinheit aus. 2.) Das ParLe System zeigt die Lehreinheit im Medienfenster markiert an und die dazugehörigen Kommentare als Kommentartitel im Kommentarfenster. Die Kommentare sind wie eingestellt gefiltert, sortiert und hierarchisch angeordnet. (Die oberste Ebene beinhaltet Kommentare, die direkt zur gewählten Lehreinheit, oder dem gewählten Medium gehören. Die eingerückten Kommentare gehören

	<p>zum jeweiligen Oberkommentar.)</p> <p>3.) Der Student kann nun die Medien zu einer Lehreinheit ausklappen.</p> <p>4.) Das ParLe System zeigt alle zu den ausgeklappten Lehreinheiten gehörenden Medien im Medienfenster an.</p> <p>5.) Der Student kann nun ein Medium auswählen.</p> <p>6.) Das ParLe System zeigt dieses als markiert und die zugehörigen Kommentare (als Titel dargestellt).</p>
Anfangsbedingung:	Der Student ist bei ParLe angemeldet. Es sind schon Medien oder Lehreinheiten vorhanden.
Abschlussbedingung:	Ein Medium oder eine Lehreinheit ist markiert.
Qualitätsmerkmale:	Bei Verbindungsabbruch zum entfernten System tritt Anwendungsfall ClientServerVerbindungUnterbrochen auf.

Tabelle 7: Anwendungsfall Auswählen eines Mediums

Anwendungsfall 4: LeseKommentare

Anwendungsfallname:	LeseKommentare
Akteure:	Student
Ereignisfluss:	<p>1.) Der Student kann einen Kommentar aus den angezeigten Kommentartiteln im Kommentarfenster auswählen.</p> <p>2.) Das ParLe System übernimmt diesen als markiert und zeigt ihn an.</p> <p>3.) Der Student kann, um einen Kommentar zu lesen, das Ausklappen des Kommentars anfordern.</p> <p>4.) Das ParLe System klappt den Kommentartext und die Metadaten Autor, Erstellungsdatum und Bewertung aus und zeigt diese an. Dieser Kommentar gilt auch als markiert.</p>
Anfangsbedingung:	Der Student ist bei ParLe angemeldet. Es sind schon Kommentare vorhanden.
Abschlussbedingung:	Kommentare können ausgeklappt und/oder markiert sein.
Qualitätsmerkmale	Bei Verbindungsabbruch zum entfernten System tritt Anwendungsfall ClientServerVerbindungUnterbrochen auf.

Tabelle 8: Anwendungsfall Lesen von Kommentaren

Anwendungsfall 5: ErstelleKommentare

Anwendungsfallname:	ErstelleKommentare
Akteure:	Student
Ereignisfluss:	<p>1.) Der Student kann nun zu einer gewählten Lehreinheit, einem gewählten Medium oder zu einem gewählten Kommentar einen eigenen Kommentar erstellen; hierzu wählt er das Kommentareingabeformular.</p> <p>2.) Das ParLe System stellt ein Kommentareingabeformular bereit.</p> <p>3.) Der Student füllt dieses Formular aus und bestätigt die Eingabe.</p> <p>4.) Das ParLe System gibt den Kommentar an das Datenverwaltungssystem weiter und zeigt den neuen Kommentar mit an, sofern er nicht vom Filter ausgefiltert wird.</p>

Anfangsbedingung:	Der Student ist bei ParLe angemeldet.
Abschlussbedingung:	Der Kommentar ist persistent im Datenverwaltungssystem gespeichert.
Qualitätsmerkmale	Bei Verbindungsabbruch zum entfernten System tritt Anwendungsfall ClientServerVerbindungUnterbrochen auf.

Tabelle 9: Anwendungsfall Erstellen von Kommentaren

Anwendungsfall 6: FiltereUndSortiereKommentare:

Anwendungsfallname:	FiltereUndSortiereKommentare
Akteure:	Student
Ereignisfluss:	<ol style="list-style-type: none"> 1.) Das ParLe System stellt dem Studenten zum Filtern und Sortieren 3 Funktionen zur Verfügung. 2.) Der Student wählt zur Bewertung von Kommentaren die Funktion Kommentar bewerten (beinhaltet Anwendungsfall BewerteKommentar). 3.) Der Student kann die Sortierung festlegen (beinhaltet Anwendungsfall SortiereKommentare). 4.) Der Student kann so bewertete Kommentare nach Relevanz ausfiltern (beinhaltet Anwendungsfall FiltereKommentare).
Anfangsbedingung:	Der Student ist im ParLe System registriert und hat sich angemeldet. Es sind schon Kommentare zur Lehreinheit/ zum Medium vorhanden. Es ist ein Kommentar ausgewählt.
Abschlussbedingung:	-
Qualitätsmerkmale	Bei Verbindungsabbruch zum entfernten System tritt Anwendungsfall ClientServerVerbindungUnterbrochen auf.

Tabelle 10: Anwendungsfall Bewerten, Filtern und Sortieren von Kommentaren

Anwendungsfall 7: BewerteKommentar:

Anwendungsfallname:	BewerteKommentar
Akteure:	Student
Ereignisfluss:	<ol style="list-style-type: none"> 1.) Der Student wählt auf dem ParLe System das Bewertungsformular aus. 2.) Das ParLe System zeigt nun das Bewertungsformular an. 3.) Der Student kann nun zu einem markierten Kommentar im Bewertungsformular eine Bewertung mit 1 bis 5 Sternen abgeben. 4.) Das ParLe System übernimmt die Bewertung und speichert sie im Datenverwaltungssystem als dem Kommentar und Nutzer zugehörig ab. Die gerundete Durchschnittsbewertung des Kommentars wird neu errechnet.
Anfangsbedingung:	Der Student ist bei ParLe angemeldet. Es sind schon Kommentare zur Lehreinheit bzw. zum Medium vorhanden. Es ist ein Kommentar ausgewählt.
Abschlussbedingung:	Die Bewertung zu einem Kommentar von einem Nutzer ist persistent im Datenverwaltungssystem gespeichert.
Qualitätsmerkmale:	Eigene Kommentare dürfen nicht bewertet werden.

	Jeder Kommentar darf nur einmal von einem Studenten bewertet werden. Beim Verbindungsabbruch zum entfernten System tritt Anwendungsfall ClientServerVerbindungUnterbrochen auf.
--	--

Tabelle 11: Anwendungsfall Bewerten von Kommentaren

Anwendungsfall 8: FiltereKommentare:

Anwendungsfallname:	FiltereKommentare
Akteure:	Student
Ereignisfluss:	<ol style="list-style-type: none"> 1.) Der Student kann eine Filterfunktion auswählen. 2.) Das ParLe System stellt das Filterformular zur Verfügung, welches an einen Nutzer gebunden ist. 3.) Der Student kann über den Filterdialog auswählen, welche Kommentare er sich im Kommentarfenster anzeigen lässt. Er kann Kommentare wählen, die eine Bewertung haben, die besser oder gleich seiner Filterbedingung sind. 4.) Das ParLe System zeigt nur noch Kommentare, die besser oder gleich dem eingestellten Filter sind. Unterkommentare zu den zu schlecht bewerteten Kommentaren werden nicht angezeigt.
Anfangsbedingung:	Der Student ist bei ParLe angemeldet.
Abschlussbedingung:	Zu einem Nutzer ist ein Filter persistent gespeichert.
Qualitätsmerkmale:	Bei Verbindungsabbruch zum entfernten System tritt Anwendungsfall ClientServerVerbindungUnterbrochen auf.

Tabelle 12: Anwendungsfall Filtern von Kommentaren

Anwendungsfall 9: SortiereKommentare:

Anwendungsfallname:	SortiereKommentare
Akteure:	Student
Ereignisfluss:	<ol style="list-style-type: none"> 1.) Der Student kann ein Sortierformular wählen. 2.) Das ParLe System stellt das Sortierformular zur Verfügung. 3.) Der Student kann über das Sortierformular festlegen, welche Sortierung der Kommentare er sich im Kommentarfenster anzeigen lässt. Er kann sich die Kommentare nach Erstellungszeit anzeigen lassen oder die Relevanzsortierung wählen, so dass die Kommentare, die der Filter zulässt, nach ihrer Bewertung angezeigt werden. 4.) Das ParLe System wählt nur die Kommentare aus, die besser oder gleich dem eingestellten Filter sind. Diese sind nach gewählter Sortierung angezeigt.
Anfangsbedingung:	Der Student ist bei ParLe angemeldet.
Abschlussbedingung:	Zu einem Nutzer ist eine Sortierung persistent gespeichert.
Qualitätsmerkmale:	Bei Verbindungsabbruch zum entfernten System tritt Anwendungsfall ClientServerVerbindungUnterbrochen auf.

Tabelle 13: Anwendungsfall Sortieren von Kommentaren

Anwendungsfall 10: ClientServerVerbindungUnterbrochen:

Anwendungsfallname:	ClientServerVerbindungUnterbrochen
---------------------	------------------------------------

Akteure:	Student, Administrator
Ereignisfluss:	<ol style="list-style-type: none"> 1.) Der Akteur will mit dem ParLe System arbeiten und betätigt eine Aktion, die Daten von einem entfernten System anfordert. 2.) Das entfernte System ist nicht erreichbar, da entweder das entfernte System nicht funktioniert oder die Leitung unterbrochen ist. 3.) Der Akteur erhält vom ParLe System keine Antwort auf seine Anforderung.
Anfangsbedingung:	Der Akteur ist im ParLe System registriert.
Abschlussbedingung:	Die Daten im Datenverwaltungssystem bleiben konsistent. Die während der Unterbrechung veränderten Daten werden nicht erfasst.
Qualitätsmerkmale:	-

Tabelle 14: Anwendungsfall Client - Serververbindung unterbrochen

1.3.3 Anwendungsfallmodell für den Akteur Administrator

Aus den Szenarien für den Akteur Administrator lassen sich folgende Anwendungsfälle ableiten:

- VerwalteLehreinheitenUndMedien enthält:
 - o VerwalteLehreinheit
 - o VerwalteMedien
- VerwalteNutzer

Ein Verbindungsabbruch zu einem entfernten System erweitert die Anwendungsfälle um den Anwendungsfall ClientServerVerbindungUnterbrochen.

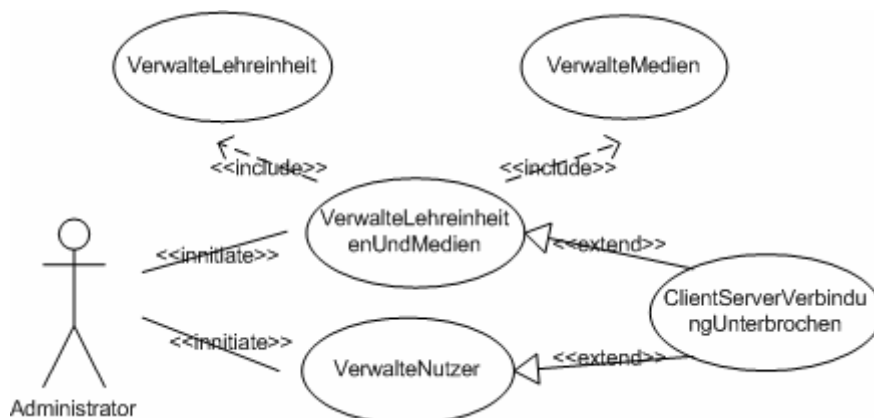


Abbildung 2: Anwendungsfalldiagramm für das ParLe System aus Sicht des Administrators. Die Abbildung zeigt die einzelnen Anwendungsfälle, ihre Initialisierung durch den Akteur Administrator, ihre Enthält- und Erweitertbeziehungen.

Anwendungsfälle:

Nachfolgend werden diese Anwendungsfälle beschrieben.

Anwendungsfall 1: VerwalteLehreinheitenUndMedien:

Anwendungsfallname:	VerwalteLehreinheitenUndMedien
Akteure:	Administrator

Ereignisfluss:	<ol style="list-style-type: none"> 1.) Der Administrator startet das ParLe Administrationssystem. 2.) Das System stellt das Administrationstool bereit. 3.) Der Administrator möchte Lehreinheiten eingeben, ändern oder löschen (beinhaltet Anwendungsfall VerwalteLehreinheit). 4.) Der Administrator möchte Medien eingeben, ändern oder löschen (beinhaltet Anwendungsfall VerwalteMedien).
Anfangsbedingung:	Der Administrator ist im Datenverwaltungssystem registriert.
Abschlussbedingung:	Lehreinheiten und Medien werden wie gewünscht gespeichert.
Qualitätsmerkmale:	Bei Verbindungsabbruch zum entfernten System tritt Anwendungsfall ClientServerVerbindungUnterbrochen auf.

Tabelle 15: Anwendungsfall **Verwalte Lehreinheiten und Medien**

Anwendungsfall 2: VerwalteLehreinheit:

Anwendungsfallname:	VerwalteLehreinheit
Akteure:	Administrator
Ereignisfluss:	<ol style="list-style-type: none"> 1.) Der Administrator gibt Lehreinheiten in das LEeingabeformular ein. Hat er keine Lehreinheit markiert, wird sie am Ende eingefügt. Hat er eine Lehreinheit markiert, erscheint die neue Lehreinheit vor der markierten. 2.) ParLe zeigt die Lehreinheit wie gewünscht im Medienfenster an. 3.) Der Administrator kann den Namen einer Lehreinheit ändern. Dazu markiert er die Lehreinheit und benutzt die Änderungsfunktion des LEeingabeformulars. 4.) ParLe zeigt den geänderten Namen einer Lehreinheit an. 5.) Der Administrator kann Lehreinheiten löschen, er markiert sie dazu und nutzt die Löschfunktion im LEeingabeformular. 6.) Die Lehreinheit wird gelöscht.
Anfangsbedingung:	Der Administrator ist angemeldet.
Abschlussbedingung:	Lehreinheiten werden im Datenverwaltungssystem gespeichert.
Qualitätsmerkmale:	Bei Verbindungsabbruch zum entfernten System tritt Anwendungsfall ClientServerVerbindungUnterbrochen auf.

Tabelle 16: Anwendungsfall **Verwalten von Lehreinheiten**

Anwendungsfall 3: VerwalteMedien:

Anwendungsfallname:	VerwalteMedien
Akteure:	Administrator
Ereignisfluss:	<ol style="list-style-type: none"> 1.) Der Administrator gibt Medien zu einer Lehreinheit in das Medieneingabeformular ein. Hat er eine Lehreinheit markiert, wird das Medium hinten in der gewählten Lehreinheit eingefügt. Hat er ein Medium markiert, wird das neue Medium vor dem markierten eingefügt. 2.) ParLe zeigt die Medien wie gewünscht im Medienfenster an. 3.) Der Administrator möchte Medien ändern. Er markiert sie und nutzt dazu die Änderungsfunktion im Medieneingabeformular.

	4.) ParLe zeigt das geänderte Medium. 5.) Der Administrator kann Medien löschen, er markiert sie dazu und nutzt die Löschfunktion im Medieneingabeformular. 6.) Das Medium wird gelöscht.
Anfangsbedingung:	Der Administrator ist angemeldet.
Abschlussbedingung:	Medien werden im Datenverwaltungssystem gespeichert.
Qualitätsmerkmale:	Bei Verbindungsabbruch zum entfernten System tritt Anwendungsfall ClientServerVerbindungUnterbrochen auf.

Tabelle 17: Anwendungsfall **Verwalten von Medien**

Anwendungsfall 4: VerwalteNutzer:

Anwendungsfallname:	VerwalteNutzer
Akteure:	Administrator
Ereignisfluss:	1.) Der Administrator kann ein Nutzereingabeformular anfordern. 2.) ParLe zeigt das Nutzereingabeformular an. 3.) Der Administrator gibt den Namen und das Passwort ein. 4.) ParLe registriert den Nutzer. 5.) Der Administrator kann einen Nutzer über eine Sperrfunktion im Nutzereingabeformular sperren. 6.) Der Nutzer hat keinen Zugriff mehr auf das ParLe System. 7.) Der Administrator kann einen Kommentar sperren. Hierzu markiert er den Kommentar und wählt die Sperrfunktion. 8.) Der Kommentar kann nicht mehr gelesen werden.
Anfangsbedingung:	Der Administrator ist angemeldet.
Abschlussbedingung:	Nutzerdaten werden im Datenverwaltungssystem gespeichert.
Qualitätsmerkmale:	Bei Verbindungsabbruch zum entfernten System tritt Anwendungsfall ClientServerVerbindungUnterbrochen auf.

Tabelle 18: Anwendungsfall **Verwalten von Nutzern**

1.4 Objektmodell

Das Objektmodell wird dargestellt durch Klassendiagramme. Aus den Anwendungsfällen lassen sich Entitäts-, Grenz- und Steuerungsobjekte ableiten, die die Grundlage des Objektmodells bilden. Im Folgenden wird dargestellt:

- Gesamtmodell mit allen Objekten
- Objektmodell der Entitätsobjekte
- Klassendiagramme für alle Anwendungsfälle

1.4.1 Gesamtmodell mit allen Objekten

Das Modell wurde aufgeteilt in ein Objektmodell für die Anwendungsfälle des Akteurs Student und ein eigenes Modell für die Anwendungsfälle für den Akteur Administrator. Dies geschah, um die Übersichtlichkeit zu wahren.

Objektmodell für die Anwendungsfälle des Akteurs Student

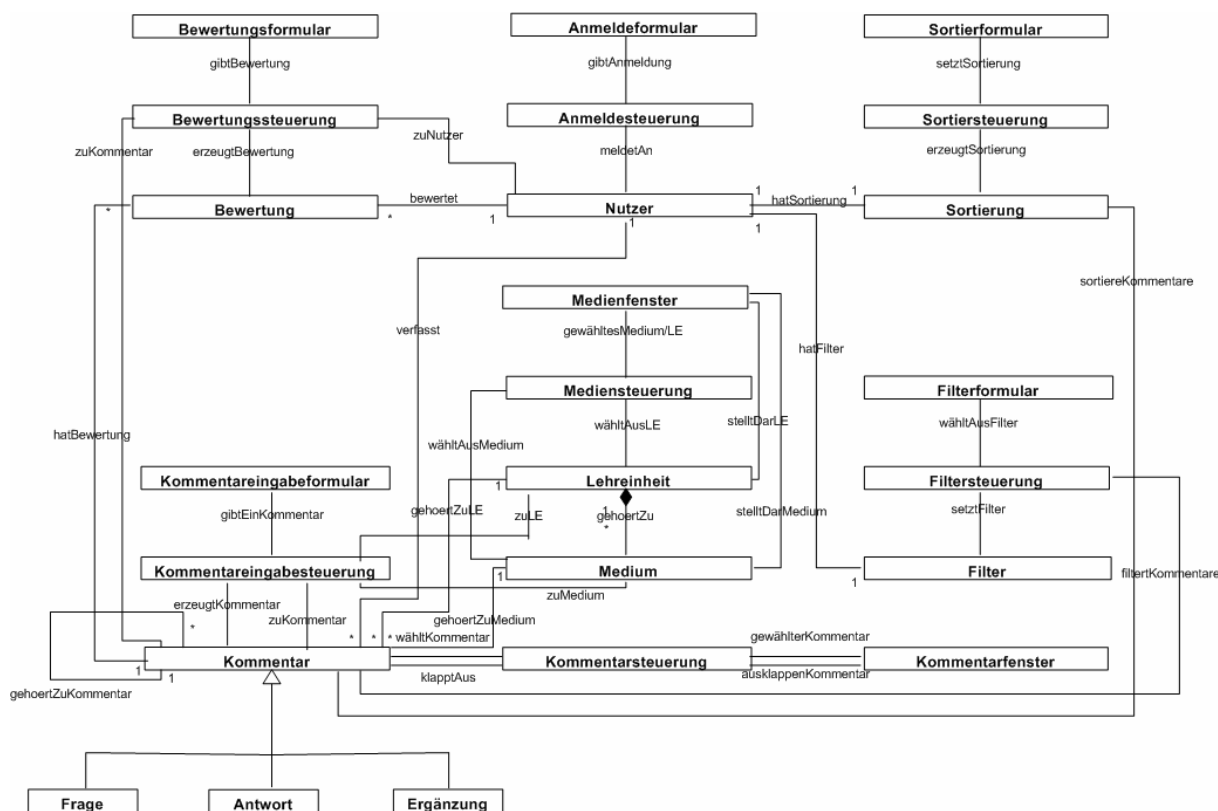


Abbildung 3: Objektmodell des ParLe Systems mit Entitäts-, Grenz- und Steuerungsobjekten für die Anwendungsfälle des Akteurs **Student**. Das Objektmodell ist im Anhang in größerem Format beigefügt.

Darstellung und Erläuterung des Objektmodells für die Anwendungsfälle des Akteurs Student:

Entitätsobjekte:

Entitätsobjekte sind die Repräsentanten der vom System verwalteten persistenten Informationen. Zur Identifikation wurden immer wieder auftretende Substantive in den Anwendungsfällen und Aktivitäten, die das System verfolgen muss, herangezogen. Die Entitätsobjekte sind:

- Nutzer:

Das Entitätsobjekt Nutzer repräsentiert einen Akteur (Student), der Kommentare zu Lehrinheiten und Medien verfasst und bewertet. Ein Nutzer kann über die Relation verfasst zu mehreren Kommentaren gehören, ein Kommentar kann nur zu einem Nutzer gehören. Ein Nutzer kann über bewertet zu einer Bewertung zu einem Kommentar gehören; dies ist nicht möglich, wenn der Kommentar zum selben Nutzerobjekt gehört. Ein Nutzer hat eine Sortierung, entweder nach Relevanz oder nach Erstellungszeit, und hat die Relation hatFilter zu einem Filter, der nicht genügend bewertete Kommentare ausfiltert. Der Nutzer wird durch die Anmeldesteuerung angemeldet. Nutzer können gesperrt sein.

Wenn nachfolgend die Bezeichnung Nutzer verwendet wird, ist damit das Entitätsobjekt gemeint, das einen Akteur systemintern repräsentiert. Wenn von Akteur gesprochen wird, ist der Student oder der Administrator gemeint, der eine Aktion auslösen kann.

- Lehreinheit:

Der Stoff der Vorlesung ist in Lehreinheiten untergliedert. Lehreinheiten werden ins ParLe System eingestellt. Jede Lehreinheit kann mehrere Medien (hier Folien) enthalten. Die Lehreinheit kann kommentiert werden. Zu einer Lehreinheit können mehrere Kommentare gehören. Alle Lehreinheiten werden im Medienfenster dargestellt, und über die Mediensteuerung kann eine Lehreinheit ausgewählt werden.

- Medium:

Jedes Medium gehört zu einer Lehreinheit. Medien können ausgewählt und im Medienfenster angezeigt werden. Das Medium kann kommentiert werden. Medien können über die Mediensteuerung ausgewählt werden. Medien werden ins ParLe System eingestellt.

- Frage:

Eine Frage ist die Form eines Kommentars, bei dem der Akteur Student eine Frage zu einem Medium, einer Lehreinheit oder einem Kommentar hat.

- Antwort:

Eine Antwort ist ein Kommentar, bei dem ein Akteur Student eine Frage eines anderen Studenten beantwortet.

- Ergänzung:

Eine Ergänzung ist die Form eines Kommentars, bei dem ein Student eine Lehreinheit oder ein Medium oder einen Kommentar eines anderen Studenten ergänzend kommentiert.

- Kommentar:

Der Kommentar ist das zentrale Objekt im ParLe System. Die Kommentare gehören zu Lehreinheiten oder Medien oder zu anderen Kommentaren (Oberkommentare). Sie haben die Relation verfasst zu Nutzerobjekten über die Kommentareingabesteuerung, sie sind auswählbar und können ausgeklappt werden. Der Kommentar besteht aus einem Kommentarnamen, einem Erstellungsdatum, einem Autor, einer Durchschnittsbewertung, dem Kommentar selbst und hat eine Zugehörigkeit zu einer Lehreinheit, einem Medium oder einem Oberkommentar. Kommentare können gesperrt sein.

- Bewertung:

Zu jedem fremden Kommentar kann pro Nutzerobjekt über die Bewertungssteuerung genau eine Bewertung von 1 bis 5 Sternen erzeugt werden. Dabei bedeuten 1 Stern irrelevant und 5 Sterne sehr zutreffend. Noch nicht bewertete Kommentare haben 0 Sterne. Jeder Kommentar wird mit dem Durchschnitt seiner Bewertungen angezeigt.

- Filter:

Jeder Nutzer hat einen Filter, der Kommentare über die Filtersteuerung ausschließt, die schlechter bewertet sind als der von ihm eingestellte Filter. Dieser reicht von 0 bis 5 Sternen und wird über die Filtersteuerung gesetzt; hierbei werden gerundete Bewertungen verwendet.

- Sortierung:

Es gibt zwei Arten von Sortierung, die Relevanzsortierung oder die Sortierung nach Erstellungszeit. Ein Nutzer hat eine der beiden Sortiermöglichkeiten. Bei der Relevanzsortierung, die über die Sortiersteuerung auswählbar ist, werden die Kommentare über die Sortiersteuerung nach Relevanz von 5 Sternen bis 0 Sternen (solange der Filter dies zulässt) in dem Kommentarfenster sortiert dargestellt.

Grenzobjekte:

Die Interaktion zwischen Anwender und System wird von Grenzobjekten bewerkstelligt.

Zur Identifikation von Grenzobjekten wurden Benutzerschnittstellenelemente herangezogen, die der Akteur Student braucht, um den Anwendungsfall zu initiieren, Eingabefenster oder Formulare, die der Student braucht, um Daten in das System einzugeben, und Nachrichten, die das System benötigt, um dem Akteur Student zu antworten.

Aus den Anwendungsfällen lassen sich die nachfolgenden Grenzobjekte herleiten:

- Anmeldeformular:

Das beim Start des ParLe Systems dargestellte Formular erfordert die Eingabe des Nutzernamens und des Passworts. Diese Daten werden der Anmeldesteuerung übergeben, die den Nutzer beim ParLe System anmeldet. Bei falscher Eingabe schließt sich das ParLe System.

- Medienfenster:

Im Medienfenster werden die Lehreinheiten und Medien dargestellt. Im Medienfenster kann das gewünschte Medium oder die gewünschte Lehreinheit gewählt werden und dann durch die Mediensteuerung im Medienfenster dargestellt werden. Durch spezielle Auswahl der gewünschten Lehreinheit werden die dazugehörigen Medien aus- oder eingeklappt. Die Medien (Folien) können groß dargestellt werden und sind dabei navigierbar.

- Kommentarfenster:

Im Kommentarfenster werden die Kommentare zur gewählten Lehreinheit oder zum gewählten Medium dargestellt. Diese sind hierarchisch nach ihrer Zugehörigkeit sortiert. Kommentare können über die Kommentarsteuerung gewählt werden. Durch spezielle Auswahl des Kommentarnamens (der zuerst angezeigt wird) werden das Datum, der Autor, die Durchschnittsbewertung und der eigentliche Kommentar über die Kommentarsteuerung ausgeklappt. Diese bleiben bis zum Einklappen oder Wechsel der Lehreinheit oder des Mediums ausgeklappt sichtbar. Die Filtersteuerung und die Sortiersteuerung lässt dabei nur Kommentare und deren Unterkommentare auswählen, die besser oder gleich der Filterbedingung sind und zwar in der gesetzten Sortierreihenfolge. Je nachdem ob nach Relevanz oder Erstellungszeit sortiert wird, ist die Reihenfolge der Kommentare und Unterkommentare bestimmt. Gesperrte Kommentare erscheinen mit hellgrauem Namen und können nicht ausgeklappt werden.

- Kommentareingabeformular:

Über das Kommentareingabeformular kann ein Kommentar erstellt werden, der entweder zu einer Lehreinheit, einem Medium oder einem Oberkommentar gehört. Dieser wird an die Kommentareingabesteuerung übergeben, die daraus den Kommentar erzeugt.

- Filterformular:

Im Filterformular können die Werte von 0 bis 5 Sternen ausgewählt werden, die an die Filtersteuerung übergeben werden; diese setzt dann den Filter für den aktuellen Nutzer und schließt die Kommentare aus, die schlechter bewertet sind als die Filterbedingung. Unterkommentare, die zu ausgeschlossenen Kommentaren gehören, werden ebenfalls ausgefiltert.

- Sortierformular:

Im Sortierformular lässt sich die Sortierart festlegen; dies ist entweder Sortierung nach Relevanz oder stattdessen nach der Erstellungszeit. Die Sortierart wird über die Sortiersteuerung in die Sortierung für den aktuellen Nutzer überführt und die Darstellung im Kommentarfenster so gesetzt. Hierbei wird nach dem Durchschnitt der Bewertungen in Gleitkommazahlen mit doppelter Genauigkeit sortiert, nicht nach gerundeten Werten.

- Bewertungsformular:

Das Bewertungsformular übermittelt eine Bewertung über die Bewertungssteuerung, die eine Bewertung für einen Nutzer und einen Kommentar erzeugt. Dabei kann ein Kommentar eines anderen Nutzers mit 1 bis 5 Sternen bewertet werden. Ein Kommentar darf für einen Nutzer nur einmal bewertet werden.

Steuerungsobjekte:

Steuerungsobjekte verwirklichen die eigentlichen Anwendungsfälle. Zur Identifikation von Steuerungsobjekten wurde die Regel herangezogen, dass jeder Anwendungsfall ein Steuerungsobjekt pro Akteur hat und dass jeder Akteur im Anwendungsfall ein zugehöriges Steuerungsobjekt hat.

Die aus den Anwendungsfällen abgeleiteten Steuerungselemente sind:

- Anmeldesteuerung:

Die Anmeldesteuerung prüft die im Anmeldeformular eingegebenen Daten, d.h. Nutzernamen und Passwörter werden mit den Einstellungen im Datenverwaltungssystem überprüft, und sie meldet den Akteur bei Übereinstimmung der Daten als Nutzer an. Die Anmeldesteuerung gibt die Benutzeroberfläche bei korrekten Daten frei. Bei falschen Eingaben beendet sich das ParLe System.

- Mediensteuerung:

Die Mediensteuerung dient der Auswahl der Medien der Lehreinheiten aus dem Medienfenster und klappt die Medien bei spezieller Auswahl der entsprechenden Lehreinheit im Medienfenster aus oder ein. Durch spezielle Wahl des Mediums (Folie) erscheinen diese im Großformat und können vor und zurück durchgeblättert werden. Zur gewählten Lehreinheit bzw. zum gewählten Medium werden die entsprechenden Kommentare (als Kommentartitel) im Kommentarfenster angezeigt.

- Kommentarsteuerung:

Die Kommentarsteuerung ist für das Aus- und Einklappen der gewählten Kommentare und deren Metadaten (Autor, Datum, Bewertung) zuständig. Die aus- und einzuklappenden Kommentare werden im Kommentarfenster gewählt, das auch die Kommentare anzeigt.

- Kommentareingabesteuerung:

Diese Steuerung regelt das Einstellen der Kommentare in das Datenverwaltungssystem. Dazu werden aus den ins Kommentareingabeformular eingegebenen Daten und der Zugehörigkeit zu einer Lehreinheit, einem Medium, oder einem Oberkommentar die eigentlichen Kommentare erzeugt.

- Filtersteuerung:

Die Filtersteuerung übernimmt das Ausfiltern der Kommentare in der Darstellung im Kommentarfenster, die schlechter als die Filterbedingung bewertet sind. Sie setzt den Filter, der zu einem Nutzer gehört, aus den vom Filterformular übernommenen Daten.

- Bewertungssteuerung:

Die Bewertungssteuerung übernimmt die Daten aus dem Bewertungsformular und erzeugt die Bewertung zwischen dem Kommentar und dem Nutzer.

- Sortiersteuerung:

Die Sortiersteuerung übernimmt die im Sortierformular gesetzte Sortierart und erzeugt die Sortierung. Sie ist für die Sortierung nach Relevanz oder nach Erstellungszeit im Kommentarfenster verantwortlich. Unterkommentare werden dabei ebenfalls nach den oben genannten Bedingungen sortiert. Die gewählte Sortierung gehört zu einem Nutzerobjekt.

Objektmodell für die Anwendungsfälle des Akteurs Administrator

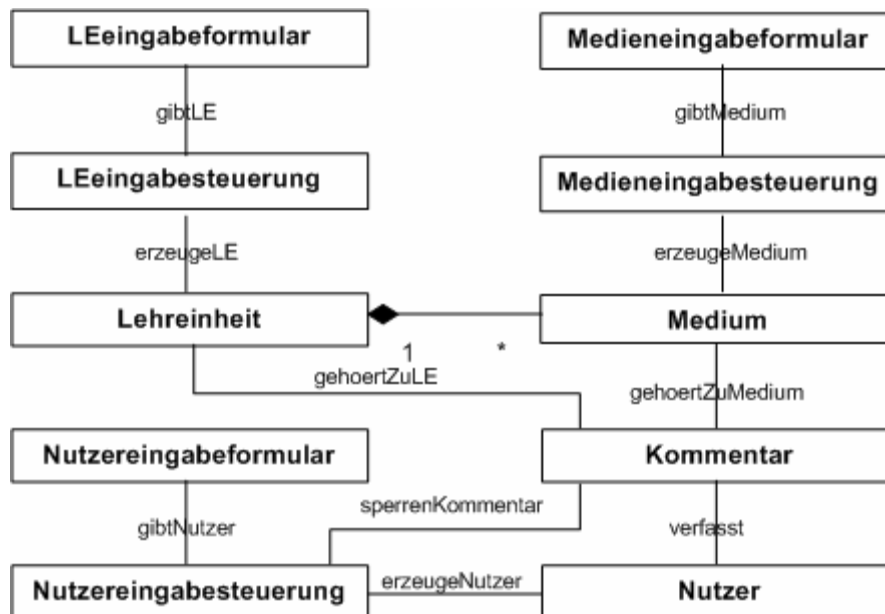


Abbildung 4: Objektmodell des ParLe Systems mit Entitäts-, Grenz- und Steuerungsobjekten für die Anwendungsfälle des Akteurs **Administrator**

Darstellung und Erläuterung des Objektmodells für die Anwendungsfälle des Akteurs Administrator

Näher erläutert werden nur Objekte, die nicht beim Objektmodell für den Akteur Student erwähnt sind. Sind bei Objekten zusätzliche Attribute notwendig, sind nur diese erwähnt.

Entitätsobjekte:

- Nutzer:

Nutzer werden vom Administrator angelegt. Ein Nutzer kann ein Administrator sein.

Grenzobjekte:

- LEingabeformular:

Der Administrator gibt über das LEingabeformular Lehreinheiten in das Parle System ein. Lehreinheiten können über das Formular auch geändert und gelöscht werden.

- Medieneingabeformular:

Der Administrator gibt über das Medieneingabeformular Medien in das Parle System ein. Medien können über das Formular auch geändert und gelöscht werden.

- Nutzereingabeformular:

Der Administrator gibt über das Nutzereingabeformular Name und Passwort von zu registrierenden Nutzern ein. Über das Formular können Nutzer und Kommentare gesperrt werden.

Steuerungsobjekte:

- LEingabesteuerung:

Die LEEingabesteuerung übernimmt die zu erstellenden Lehreinheiten aus dem LEEingabeformular und speichert sie im Datenverwaltungssystem. Sie bearbeitet auch die Änderungen von Lehreinheitstiteln und das Löschen von Lehreinheiten.

- Medieneingabesteuerung:

Die Medieneingabesteuerung übernimmt die zu erstellenden Medien aus dem Medieneingabeformular und speichert sie im Datenverwaltungssystem. Sie bearbeitet auch die Änderungen von Medien und das Löschen von Medien.

- Nutzereingabesteuerung:

Die Nutzereingabesteuerung übernimmt die Eingaben aus dem Nutzereingabeformular und speichert die Daten im Datenverwaltungssystem. Sie bearbeitet auch das Sperren von Nutzern und von Kommentaren.

1.4.2 Objektmodell der Entitätsobjekte:

Entitätsobjekte stehen stellvertretend für Informationen, die das System verwalten soll und die persistent sein sollen. Das Objektmodell der Entitätsobjekte ist die Grundlage für die Strukturierung des Datenverwaltungssystems.

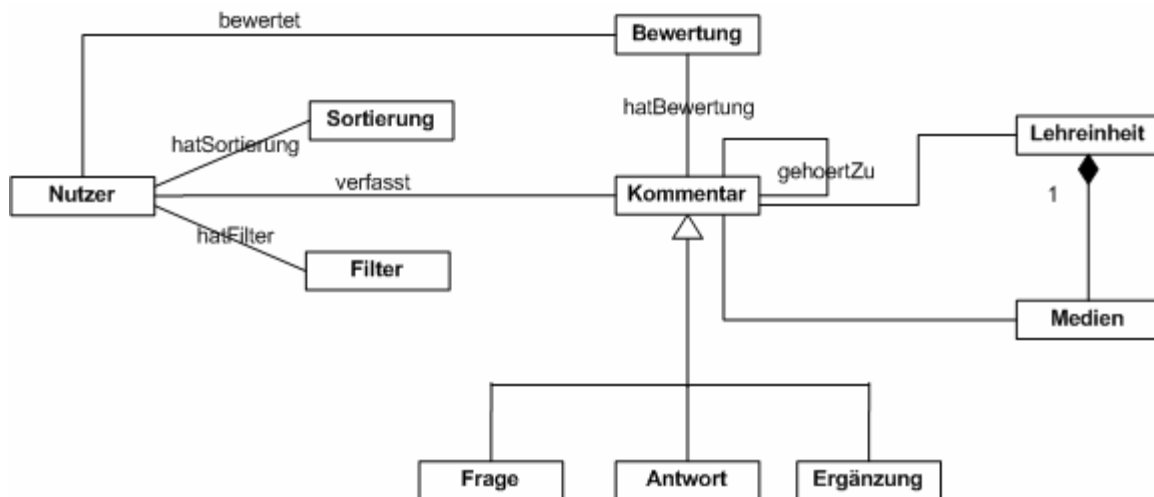


Abbildung 5: Das Objektmodell der Entitätsobjekte für ParLe

Die Abbildung zeigt das Zusammenwirken der einzelnen Entitätsobjekte und ist ein Ausschnitt des Objektmodells von Abbildung 3. Der eigentliche Kern ist das Kommentarobjekt. Dieses besteht aus Frage, Antwort oder Ergänzung. Das Kommentarobjekt gehört zu einem Medium, einer Lehreinheit oder einem Kommentar und kann pro Nutzer eine Bewertung haben. Der Nutzer kann Kommentare verfassen, und er kann einen Kommentar mit einer Bewertung bewerten. Er hat eine nutzerspezifische Sortierung und einen Filter.

1.4.3 Klassendiagramme

Die statische Struktur eines Anwendungsfalles wird im Klassendiagramm beschrieben mit den Objekten, deren Attribute und Operationen. Das Klassendiagramm zeigt auch das Zusammenspiel zwischen den Objekten. In den Boxen der Objekte sind im oberen Feld der Klassenname, im mittleren Feld die Attribute und im unteren Feld die Operationen dargestellt. Für die im Abschnitt 1.3.2 und 1.3.3 identifizierten Anwendungsfälle werden nachfolgend ihre Klassendiagramme dargestellt.

Klassendiagramm für den Anwendungsfall AnmeldeNutzer:

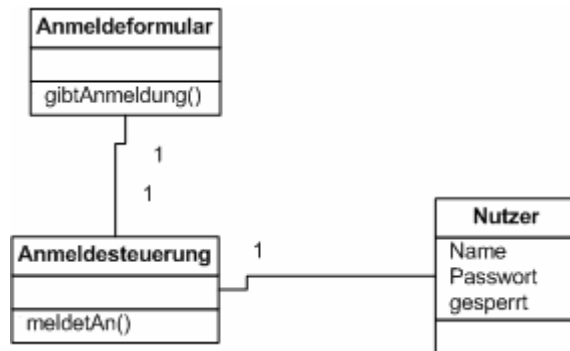


Abbildung 6: Klassendiagramm für den Anwendungsfall **AnmeldeNutzer**

Das Anmeldeformular gibt über `gibtAnmeldung()` Name und Passwort an die Anmeldesteuerung weiter. Diese meldet den Nutzer Student über `meldetAn()` beim ParLe System an. Der Nutzer hat einen Namen, ein Passwort und kann gesperrt sein.

Klassendiagramm für den Anwendungsfall WähleMedium:

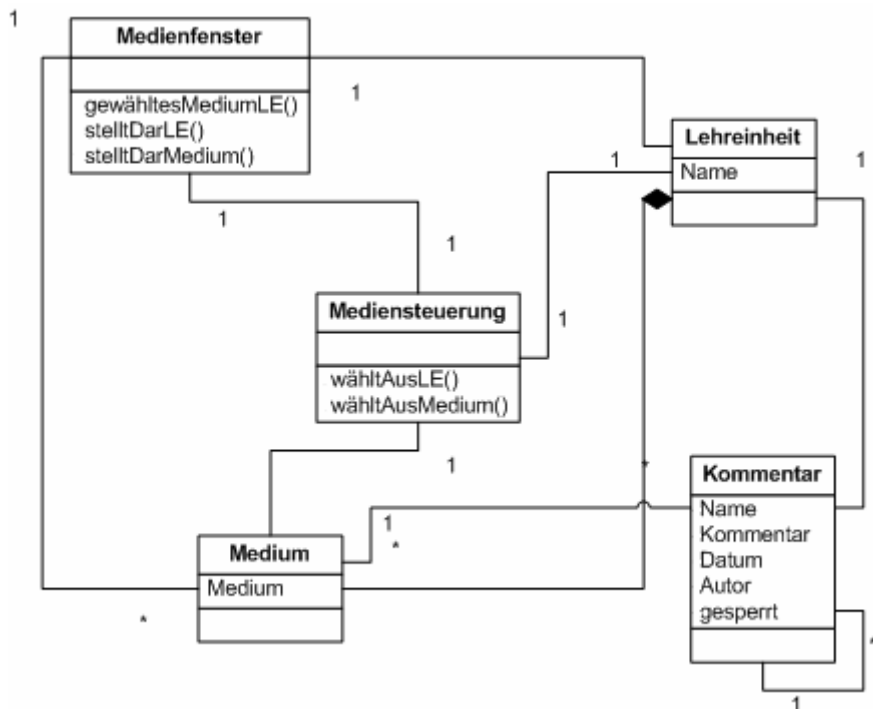


Abbildung 7: Klassendiagramm für den Anwendungsfall **WähleMedium**

Das Medienfenster gibt über `gewähltesMediumLE()` die ausgewählte Lehreinheit oder das ausgewählte Medium an die Mediensteuerung weiter. Diese wählt eine Lehreinheit über `wähltAusLE()` oder ein Medium über `wähltAusMedium()` aus. Im Medienfenster werden die Lehreinheiten über `stelltDarLE()` und die Medien über `stelltDarMedium()` dargestellt. Zu einer Lehreinheit, die aus dem Namen besteht und mehrere Medien beinhalten kann, können Kommentare gehören, die einen Namen, ein Datum, einen Autor und eine Bewertung haben. Zu einer Lehreinheit gehören Medien, die aus Medium bestehen (hier Folien) und die ebenfalls Kommentare haben können.

Klassendiagramm für den Anwendungsfall LeseKommentare:

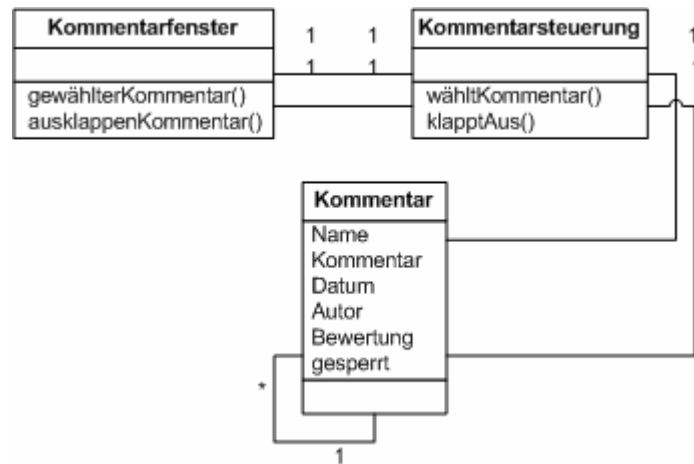


Abbildung 8: Klassendiagramm für den Anwendungsfall **LeseKommentare**

Das Kommentarfenster übergibt über *gewählterKommentar()* einen ausgewählten Kommentar an die Kommentarsteuerung, der von der Kommentarsteuerung über *wähltKommentar()* gewählt oder über *klapptAus()* aus- und eingeklappt werden kann. Das Kommentarfenster stellt über *ausklappenKommentar()* die Kommentare abhängig von der Sortierung dar, wie sie von der Steuerung über *klapptAus()* übernommen werden. Es werden nur Kommentare über die Kommentarsteuerung gewählt, die von der Filtersteuerung zur entsprechenden Lehreinheit oder zum entsprechenden Medium ausgefiltert wurden.

Klassendiagramm für den Anwendungsfall ErstelleKommentare:

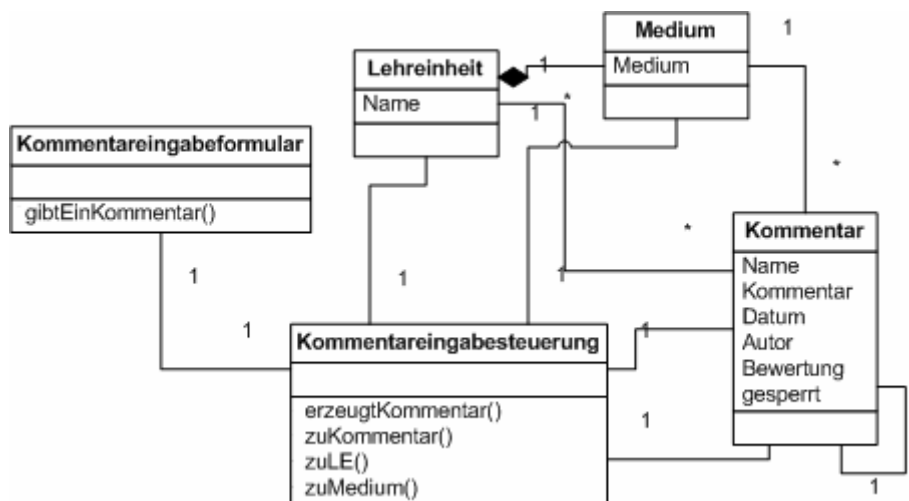


Abbildung 9: Klassendiagramm für den Anwendungsfall **ErstelleKommentare**

Das Kommentareingabeformular gibt die Daten über *gibtEinKommentar()* an die Kommentareingabesteuerung weiter; diese stellt abhängig von der Zuordnung zu einer Lehreinheit, einem Medium oder einem Oberkommentar, den Kommentar über *erzeugtKommentar()* ein; dieser gehört zu einer Lehreinheit *zuLE()*, einem Medium *zuMedium()* oder einem Kommentar *zuKommentar()*. Zu einem Kommentar gehören ein Name, ein Datum, ein Autor und eine Bewertung.

Klassendiagramm für den Anwendungsfall BewerteKommentar:

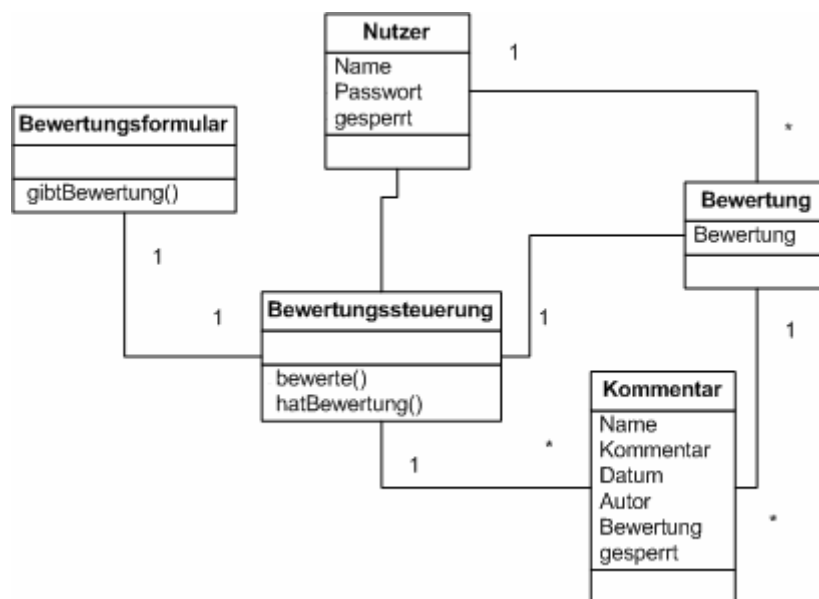


Abbildung 10: Klassendiagramm für den Anwendungsfall **BewerteKommentar**

Das Bewertungsformular gibt über *gibtBewertung()* eine Bewertung an die Bewertungssteuerung weiter, die zu einem Nutzer *bewerte()* und einem Kommentar *hatBewertung()* zugehörig ist. Die Bewertung wird als Bewertung eingestellt.

Klassendiagramm für den Anwendungsfall FiltereKommentare:

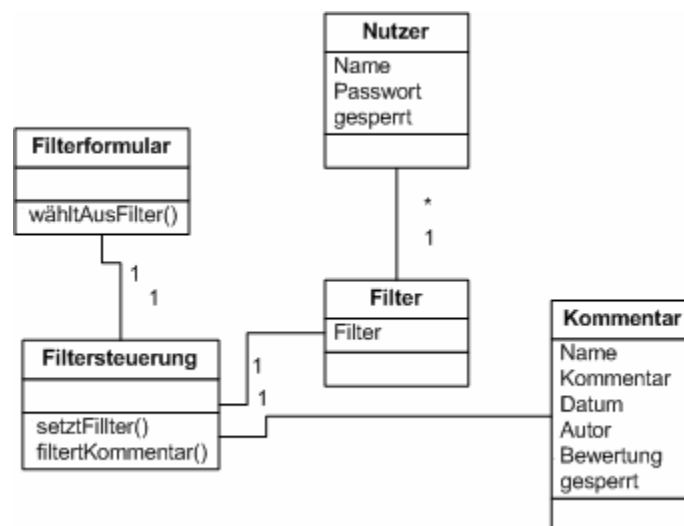


Abbildung 11: Klassendiagramm für den Anwendungsfall **FiltereKommentare**

Über das Filterformular wird über *wähltAusFilter()* ein Filter an die Filtersteuerung weitergegeben, die mit *filtertKommentar()*, die Kommentare ausfiltert, die schlechter als die Filterbedingung sind. Mit *setztFilter()* wird dieser Filter gesetzt, der zu einem Nutzer gehört.

Klassendiagramm für den Anwendungsfall SortiereKommentare:

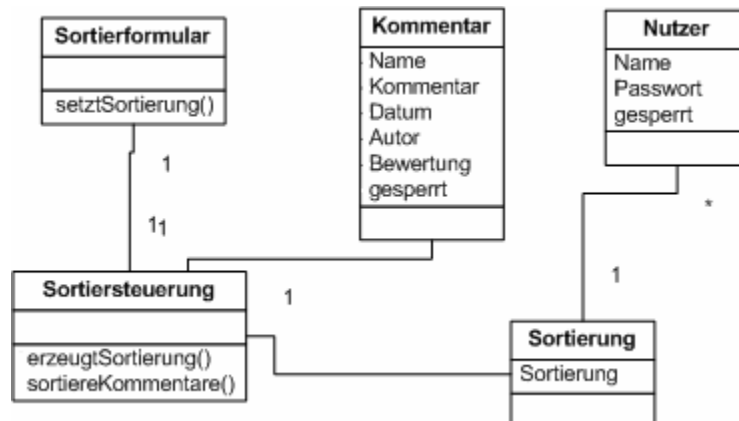


Abbildung 12: Klassendiagramm für den Anwendungsfall **SortiereKommentare**

Die gewählte Sortierart wird über `setztSortierung()` vom Sortierformular an die Sortiersteuerung übergeben, die mit `sortiereKommentare()` die Kommentare sortiert. Die Sortierung, die zu einem Nutzer gehört, wird über `erzeugtSortierung()` als Objekt gespeichert.

Klassendiagramm für den Anwendungsfall VerwalteLehreinheit:

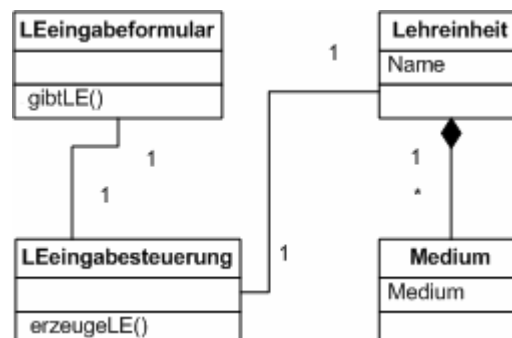


Abbildung 13: Klassendiagramm für den Anwendungsfall **VerwalteLehreinheit**

Das LEingabeformular gibt über `gibtLE()` die Daten der zu erzeugenden, zu ändernden oder zu löschenden Lehreinheit an die LEingabesteuerung weiter. Die LEingabesteuerung erzeugt, ändert oder löscht nun die Lehreinheit über `erzeugeLE()`. Die Lehreinheiten können mehrere Medien beinhalten.

Klassendiagramm für den Anwendungsfall VerwalteMedien:

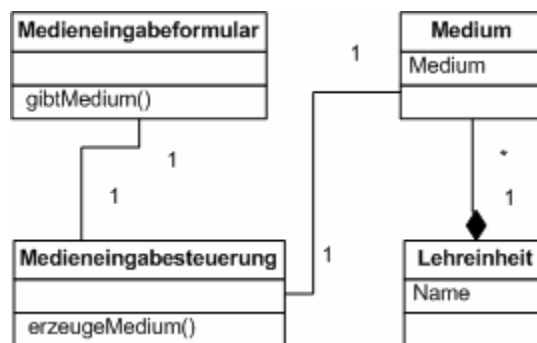


Abbildung 14: Klassendiagramm für den Anwendungsfall **VerwalteMedien**

Das Medieneingabeformular gibt über *gibtMedium()* die Daten des zu erzeugenden, zu ändernden oder zu löschenden Mediums an die Medieneingabesteuerung weiter. Die Medieneingabesteuerung erzeugt, ändert oder löscht nun das Medium über *erzeugeMedium()*.

Klassendiagramm für den Anwendungsfall VerwalteNutzer:

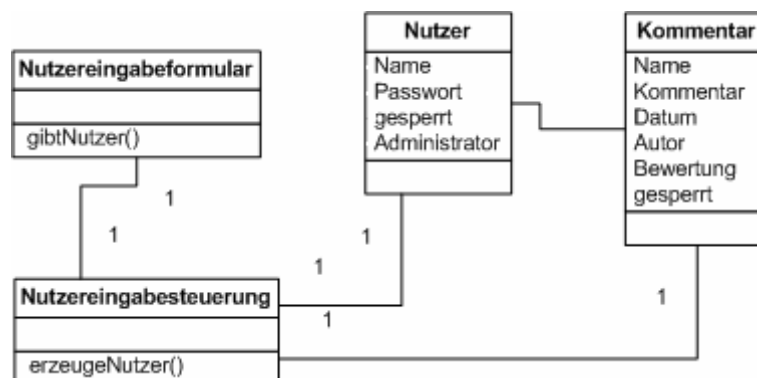


Abbildung 15: Klassendiagramm für den Anwendungsfall **VerwalteNutzer**

Das Nutzereingabeformular gibt über *gibtNutzer()* die Daten zur Erzeugung eines neuen Nutzers an die Nutzereingabesteuerung weiter. Bei Sperrung des Nutzers oder eines Kommentars werden die Daten an die Nutzereingabesteuerung weitergegeben. Die Nutzereingabesteuerung erzeugt über *erzeugeNutzer()* einen Nutzer oder sperrt einen Nutzer oder einen Kommentar.

1.4.4 Dynamisches Modell

Das dynamische Modell beschreibt das interne Verhalten des Systems und wird durch Sequenzdiagramme beschrieben.

Ein Sequenzdiagramm zeigt die Verbindung von Anwendungsfällen mit Objekten. Es beschreibt das Zusammenwirken mehrerer Objekte durch eine Abfolge von Nachrichten, die zwischen den Objekten ausgetauscht werden.

Die Beschreibung der Sequenzdiagramme erfolgt als UML-Notation (Unified Modelling Language), die in der Praxis der Industrie als Standardnotation anerkannt ist.

Für die im Abschnitt 1.3.2 und 1.3.3 identifizierten Anwendungsfälle werden nachfolgend die Sequenzdiagramme dargestellt.

Sequenzdiagramm für den Anwendungsfall AnmeldeNutzer:

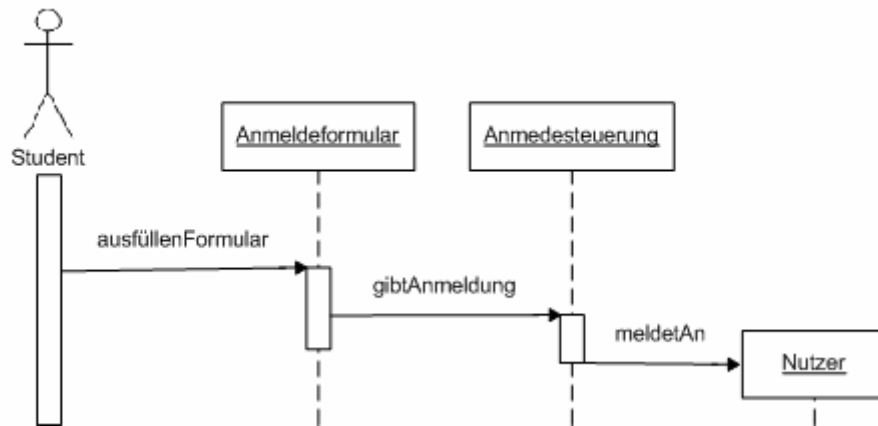


Abbildung 16: Sequenzdiagramm für den Anwendungsfall **AnmeldeNutzer**

Der Student startet das System, und das Anmeldeformular erscheint. Er füllt über *ausfüllenFormular* das Formular aus. Das Formular gibt die Anmeldung über *gibtAnmeldung()* an die Anmeldesteuerung weiter. Diese erzeugt das Nutzerobjekt über *meldetAn()*.

Sequenzdiagramm für den Anwendungsfall WähleMedium:

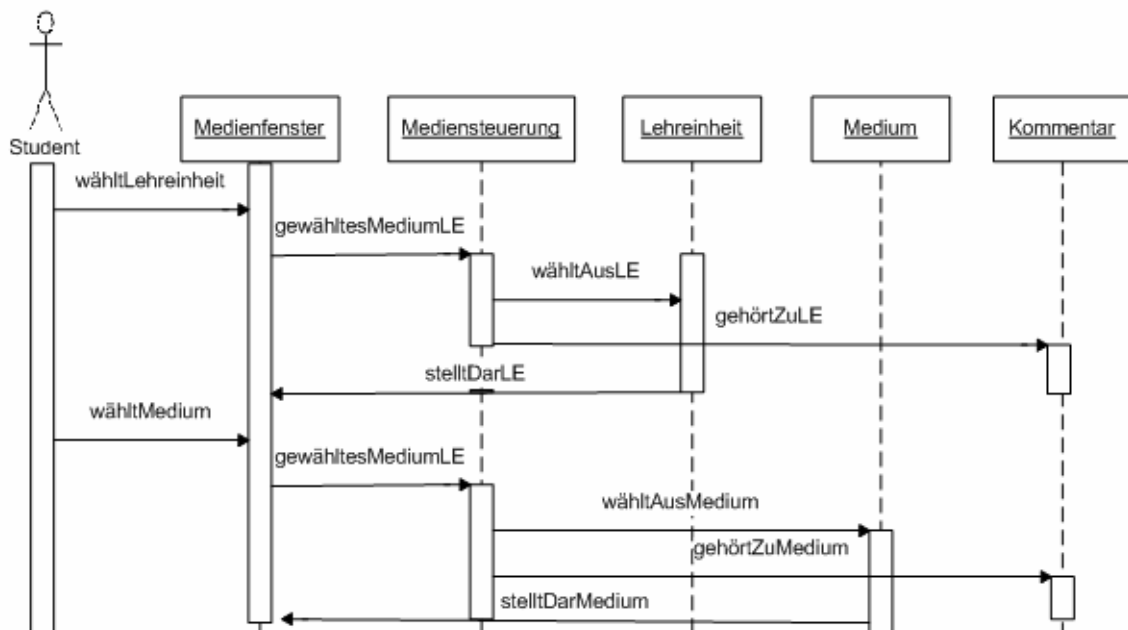


Abbildung 17: Sequenzdiagramm für den Anwendungsfall **WähleMedium**

Der Student hat als Grenzobjekt das Medienfenster; er kann in diesem eine Lehreinheit mit Klick auswählen und mit Doppelklick zum Ausklappen bringen. Dabei werden Medien ausgeklappt und wieder eingeklappt. Die Auswahl wird über *gewähltesMediumLE()* an die Mediensteuerung übergeben. Die Mediensteuerung wählt über die Nachricht *wähltAusLE()* die Lehreinheit aus und über *gehörtZuLE()* die zugehörigen Kommentare. Nun stellt das

Medienfenster die gewählte Lehreinheit als markiert dar und klappt beim Doppelklick die zugehörigen Medien aus. Wird auf eine ausgeklappte Lehreinheit doppelgeklickt, werden die Medien wieder eingeklappt. Der Student kann ein ausgeklapptes Medium über *wähltMedium* auswählen; das Medienfenster übergibt die Nachricht *gewähltesMediumLE()* an die Mediensteuerung. Diese wählt über *wähltAusMedium()* das entsprechende Medium und wählt über *gehörtZuMedium()* die Kommentare aus. Das Medium wird als markiert im Medienfenster über *stelltDarMedium()* dargestellt.

Sequenzdiagramm für den Anwendungsfall LeseKommentare:

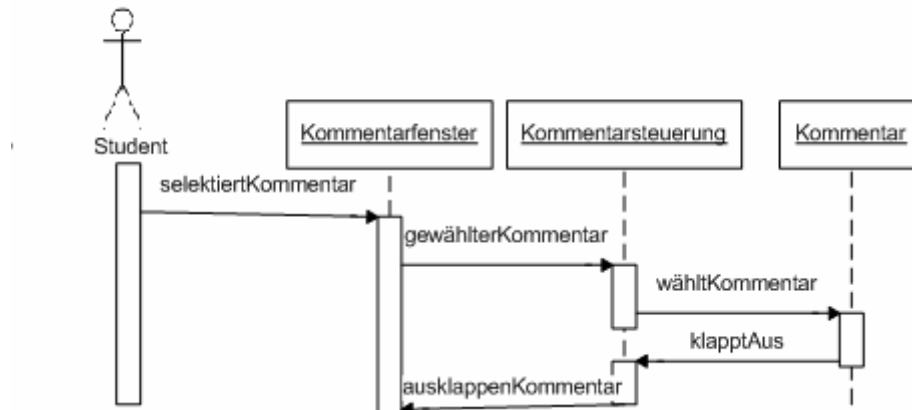


Abbildung 18: Sequenzdiagramm für den Anwendungsfall **LeseKommentare**

Der Student selektiert im Kommentarfenster einen Kommentar, den er markieren oder aus- oder einklappen will. Das Kommentarfenster gibt über *gewählterKommentar()* diese Nachricht an die Kommentarsteuerung. Diese wählt den entsprechenden Kommentar oder klappt ihn aus- oder ein über *wähltKommentar()*. Dieser wird nun wie gewählt über *klapptAus()* von der Kommentarsteuerung übernommen und über *ausklappenKommentar()* im Kommentarfenster angezeigt.

Sequenzdiagramm für den Anwendungsfall ErstelleKommentare:

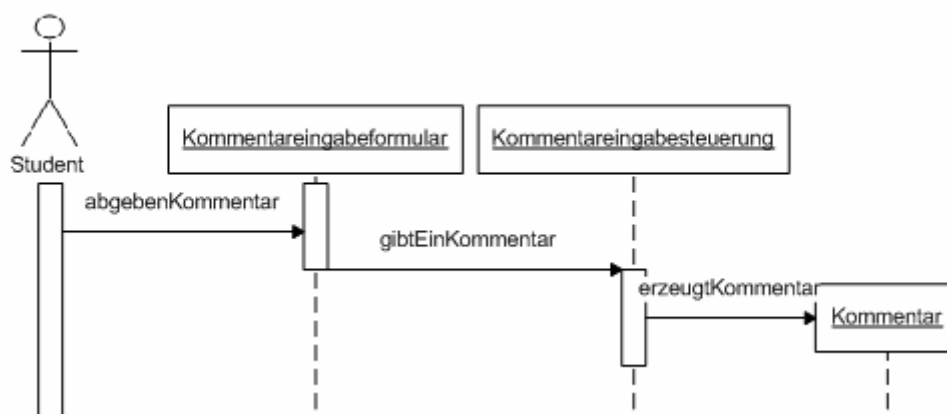


Abbildung 19: Sequenzdiagramm für den Anwendungsfall **ErstelleKommentare**

Der Student gibt einen Kommentar im Kommentareingabeformular ab über *abgebenKommentar*. Das Kommentareingabeformular benachrichtigt über *gibtEinKommentar()* die Kommentareingabesteuerung, die den Kommentar zur jeweiligen Lehreinheit, zum jeweiligen Medium oder Oberkommentar über *erzeugtKommentar()* erstellt.

Sequenzdiagramm für den Anwendungsfall BewerteKommentar:

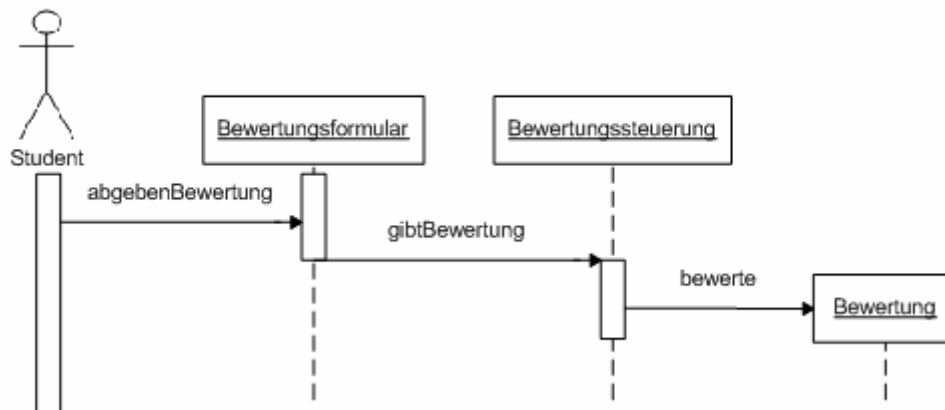


Abbildung 20: Sequenzdiagramm für den Anwendungsfall **BewerteKommentar**

Der Student gibt eine Bewertung im Bewertungsformular über *abgebenBewertung* ab. Dieses benachrichtigt die Bewertungssteuerung über *gibtBewertung()*. Die Bewertungssteuerung erzeugt über *bewerte()* die eigentliche Bewertung, die einem Nutzer und einem Kommentar zugeordnet ist. Die Bewertungssteuerung ordnet dem gewählten Kommentar die Bewertung über *hatBewertung()* zu.

Sequenzdiagramm für den Anwendungsfall FiltereKommentare:

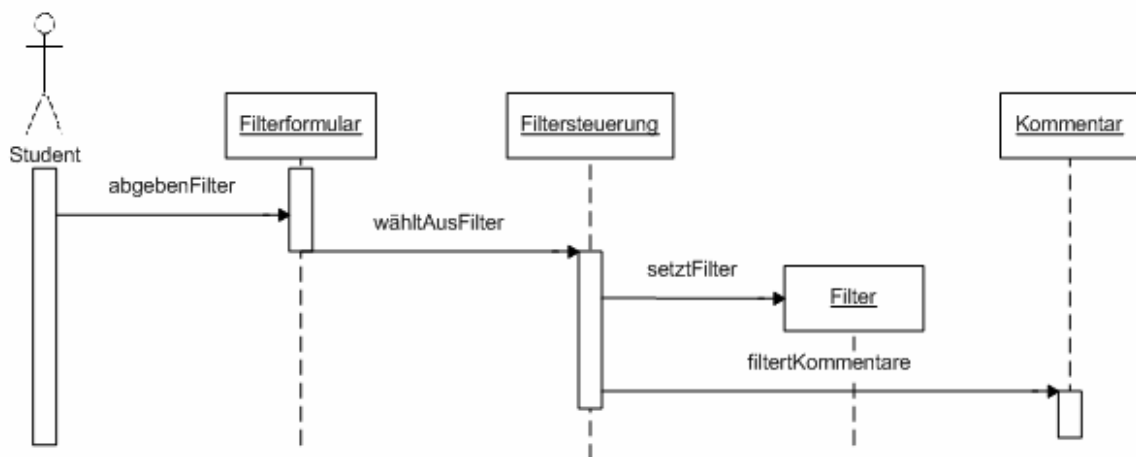


Abbildung 21: Sequenzdiagramm für den Anwendungsfall **FiltereKommentare**

Der Student füllt ein Filterformular über *abgebenFilter* aus. Das Filterformular benachrichtigt die Filtersteuerung über *wähltAusFilter()*. Die Filtersteuerung selbst übernimmt nun zwei Aufgaben, nämlich *setztFilter()*, das den Filter zu einem Nutzer setzt, und *filtertKommentare*, das die Kommentare ausfiltert, die zu schlecht bewertet sind.

Sequenzdiagramm für den Anwendungsfall SortiereKommentare:

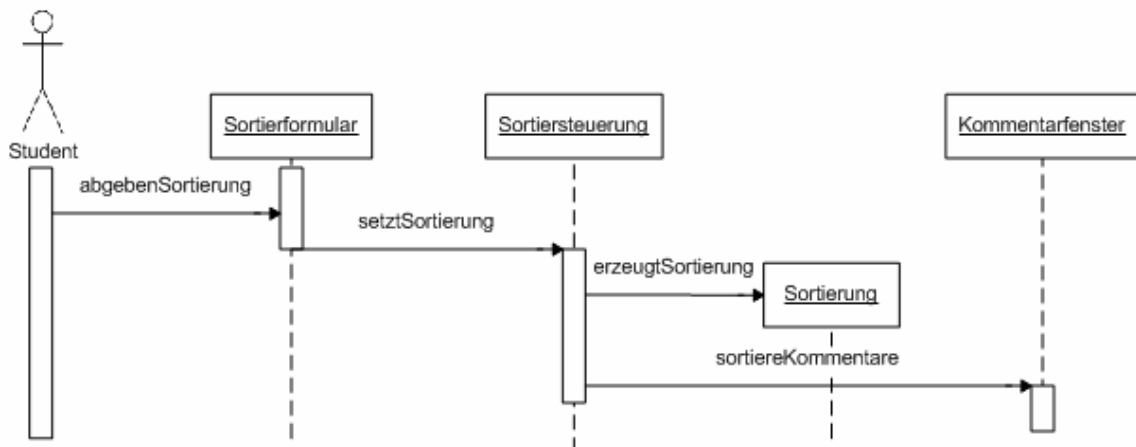


Abbildung 22: Sequenzdiagramm für den Anwendungsfall **SortiereKommentare**

Der Student füllt das Sortierformular über *abgebenSortierung* aus. Das Sortierformular übergibt diesen Wert über die Nachricht *setztSortierung()* an die Sortiersteuerung weiter. Die Sortiersteuerung erzeugt die dem Nutzer zugeordnete Sortierung über *erzeugtSortierung()* und sortiert die Kommentare im Kommentarfenster über *sortiereKommentare()*.

Sequenzdiagramm für den Anwendungsfall VerwalteLehreinheit:

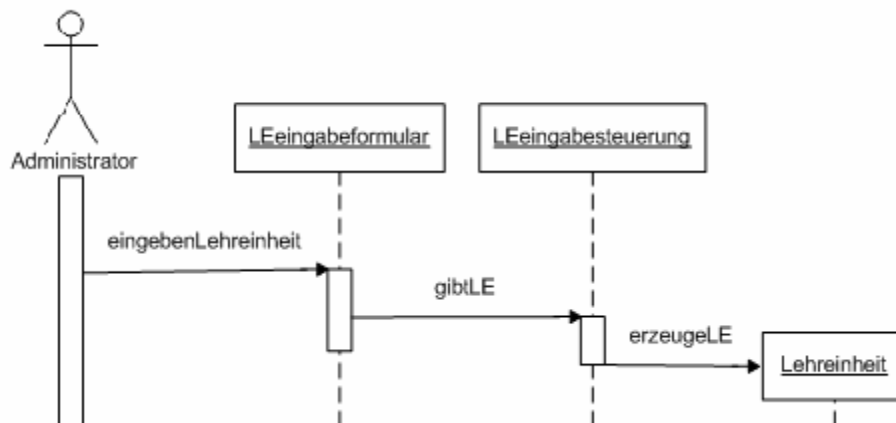


Abbildung 23: Sequenzdiagramm für den Anwendungsfall **VerwalteLehreinheit**

Der Administrator gibt die Daten der zu erzeugenden, zu ändernden oder zu löschenden Lehreinheit über *eingebenLehreinheit* in das LEingabeformular ein. Das LEingabeformular übergibt diesen Wert über die Nachricht *gibtLE()* an die LEingabesteuerung weiter. Diese erzeugt, ändert oder löscht über *erzeugeLE()* die Lehreinheit.

Sequenzdiagramm für den Anwendungsfall VerwalteMedien:

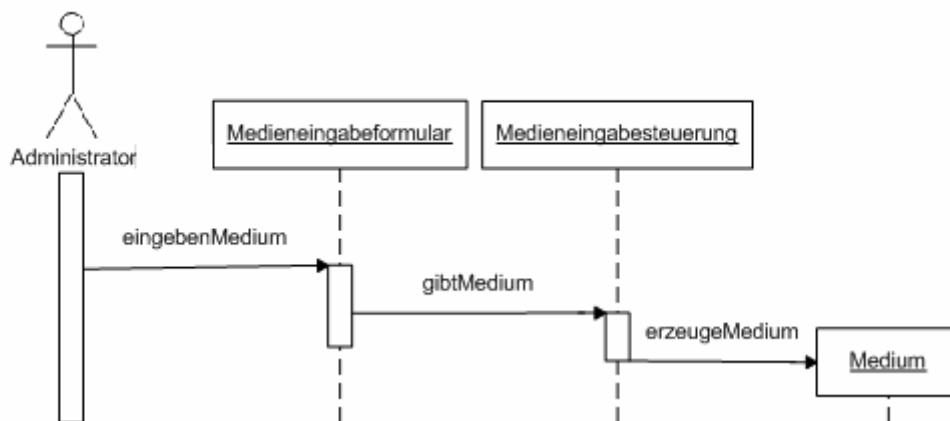


Abbildung 24: Sequenzdiagramm für den Anwendungsfall **VerwalteMedien**

Der Administrator gibt die Daten des zu erzeugenden, zu ändernden oder zu löschenden Mediums über *eingebenMedium* in das Medieneingabeformular ein. Das Medieneingabeformular übergibt diesen Wert über die Nachricht *gibtMedium()* an die Medieneingabesteuerung weiter. Diese erzeugt, ändert oder löscht über *erzeugeMedium()* das Medium.

Sequenzdiagramm für den Anwendungsfall VerwalteNutzer:

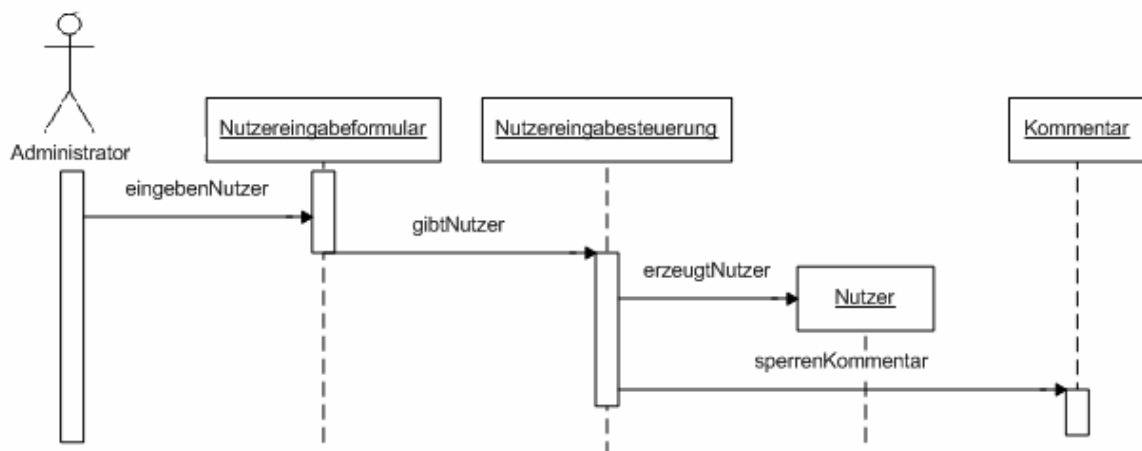


Abbildung 25: Sequenzdiagramm für den Anwendungsfall **VerwalteNutzer**

Der Administrator kann über *eingebenNutzer* einen Nutzer anlegen oder sperren. Er füllt dazu das Nutzereingabeformular aus. Dieses übergibt über *gibtNutzer()* die Nachricht an die Nutzereingabesteuerung weiter. Die Nutzereingabesteuerung erzeugt über *erzeugeNutzer()* einen Nutzer oder sperrt einen Nutzer oder einen Kommentar über *sperrenKommentar()*.

1.4.5 Benutzerschnittstelle

Die Benutzerschnittstelle ist der Teil eines Systems, der den Datenaustausch mit dem Benutzer durchführt. Bei ParLe wird ein Graphical User Interface (GUI) verwendet, das mit der Maus zu bedienen ist. Bei der Gestaltung für den Akteur Student wurde Wert darauf gelegt, wenige markante Elemente zu verwenden. Das Hauptfenster enthält:

- ein MainMenu
- zwei scrollbare Fenster:

- das Medienfenster für die Darstellung von Lehreinheiten und Medien
- das Kommentarfenster für die Darstellung der Kommentare
- drei Schaltflächen für die Auswahl der Funktionen
 - Kommentar einstellen
 - Sortieren und Filtern
 - Bewertung

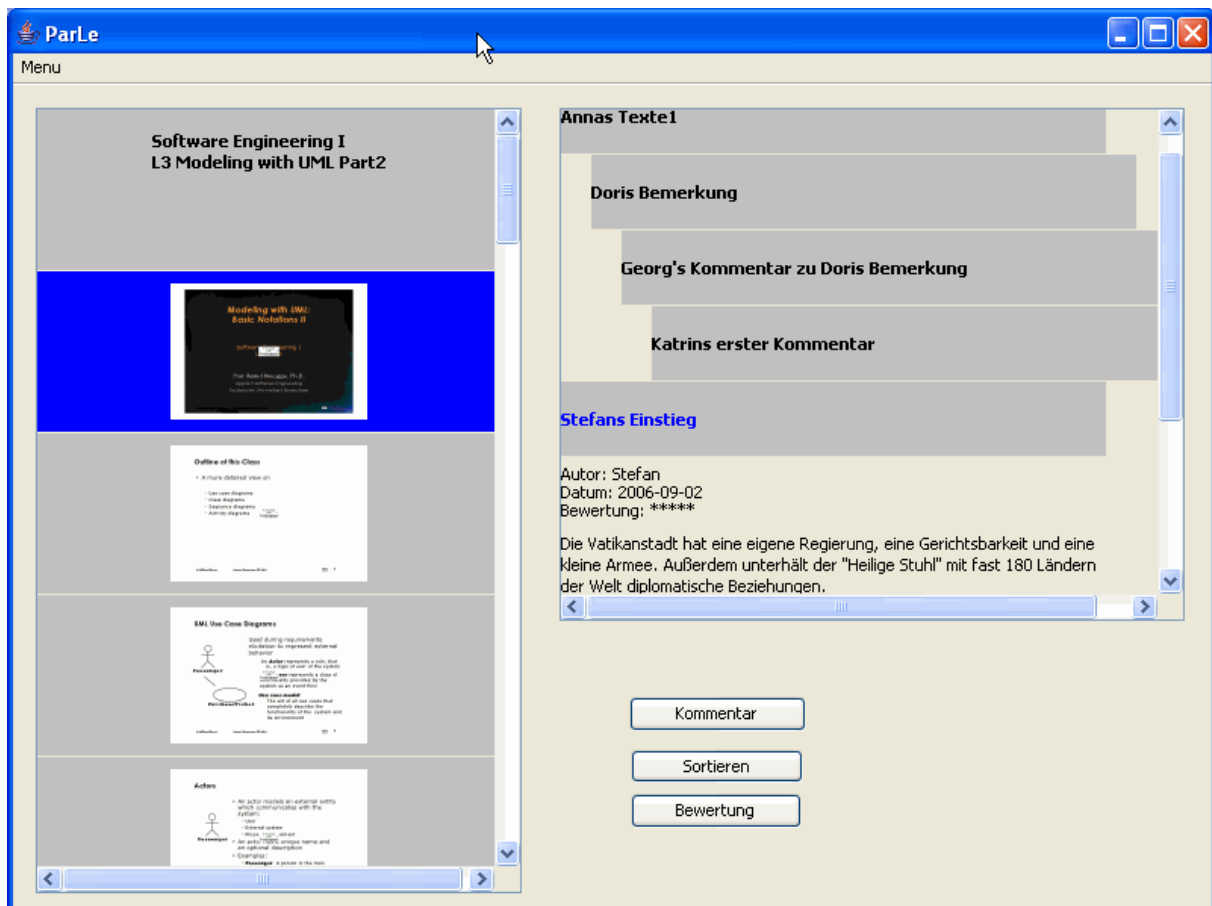


Abbildung 26: Das Hauptfenster von ParLe für den Akteur Student

Das linke Paneel zeigt die Lehreinheiten und lässt dazu Medien (Folien) beim Doppelklick auf die entsprechende Lehreinheit ausklappen. Die Medien sind als Vorschaubjekte klein dargestellt. Diese Lehreinheiten und Medien sind durch Mausklick markierbar. Wenn ein Objekt markiert ist, werden im rechten Paneel die dazugehörigen Kommentartitel angezeigt. Ein Medienelement (Folie) kann durch Doppelklick in einem Fenster vergrößert dargestellt werden, um das Medienelement gut lesen zu können. Diese Ansicht ist vor- und rückwärts navigierbar.

Im rechten Paneel werden die Kommentare dargestellt. Bei Auswahl einer Lehreinheit oder eines Mediums werden nur die Titel der Kommentare hierarchisch gegliedert dargestellt. Durch Doppelklick auf einen Kommentartitel können die Kommentare aus- und eingeklappt werden. Ist ein Kommentar ausgeklappt, erscheint neben dem eigentlichen Kommentar der Autor, das Erstellungsdatum und eine Bewertung.

Die 3 Schaltflächen haben folgende Funktionen:

Die obere Schaltfläche „Kommentar“ stellt das Kommentareingabeformular zur Verfügung, in das ein Kommentar eingegeben werden kann. Dieser Kommentar bezieht sich auf einen markierten Kommentar (Oberkommentar), auf eine Lehreinheit oder ein Medienelement.

Die mittlere Schaltfläche „Sortieren“ öffnet das Sortierformular und das Filterformular.

Beim Sortierformular kann zwischen Relevanz- und Erstellungszeitsortierung gewählt werden. Im Filterformular kann der Filter von 0 bis 5 Sternen gewählt werden.

Die untere Schaltfläche „Bewertung“ öffnet das Bewertungsformular, über das ein markierter Kommentar bewertet werden kann.

Das MainMenu beinhaltet die Funktionen „Passwort ändern“ und „Beenden“. Mit „Passwort ändern“ kann ein Student ein anderes Passwort eingeben. Mit „Beenden“ wird der ParLe Client beendet.

2 Systementwurf

2.1 Entwurfsziele

Die wesentlichen Entwurfsziele des ParLe Systems, abgeleitet aus den nichtfunktionalen Anforderungen, sind Plattformunabhängigkeit, Wirtschaftlichkeit, Skalierbarkeit, Benutzerfreundlichkeit und Verfügbarkeit. Diese Entwurfsziele haben starken Einfluss auf das Konzept zur Erstellung des Systems, die Wahl der Hardware-/Softwareplattform, auf der das System laufen soll, und das Konzept für die persistente Datenhaltung.

Plattformunabhängigkeit:

Dieses Ziel ist abgeleitet aus der Implementierungsanforderung. Das ParLe System soll auf wichtigen im universitären Bereich verbreiteten Betriebssystemen ablaufen können. Wie im nächsten Abschnitt 2.2 (Vorgeschlagene Softwarearchitektur) ausgeführt, ist ParLe ein System nach dem Client – Server Architekturstil.

Das ParLe Serversystem soll deshalb auf Betriebssystemen wie Windows Server, Mac OS X, Linux und Solaris laufen können.

Der ParLe Client soll auf Systemen laufen können, die normalerweise von Studenten genutzt werden; als Beispiele seien Windows XP, Mac OS X und Linux genannt.

Aus diesem Grund wird Java 1.5 als Entwicklungsumgebung genutzt, da für alle oben genannten Systeme Java-Ablaufumgebungen verfügbar sind. Um keine entwicklungs-systemspezifischen Bibliotheken laden zu müssen, sollen möglichst nur Java-Standardbibliotheken verwendet werden.

Wirtschaftlichkeit:

Dieses Entwurfsziel leitet sich auch aus der Implementierungsanforderung ab, möglichst Produkte ohne Lizenzgebühren zu verwenden. Aus Wirtschaftlichkeitsgründen sollen OpenSource Produkte verwendet werden. Beim Datenverwaltungssystem ist deshalb nach einem relationalen DBMS (DataBaseManagementSystem) im OpenSource Bereich zu suchen, das auf allen zuvor genannten Plattformen installierbar ist und einen breiten Funktionsumfang hat.

Skalierbarkeit:

Dieses Entwurfsziel leitet sich aus den Leistungsanforderungen ab. Das System soll bis zu 300 Studenten unterstützen. Um hohe Skalierbarkeit zu erreichen, wird eine Client - Server Architektur notwendig. Die Zahl der gleichzeitig unterstützten Studenten und die Antwortzeit sind dabei letztlich bestimmt durch die Leistungsfähigkeit des Servers.

Benutzerfreundlichkeit:

Das ParLe System soll für den Akteur Student einfach auf seinem Arbeitsplatz zu installieren sein. Das Programm soll über das Netz heruntergeladen werden können, Änderungen sollen nachladbar sein.

Das System soll für den Akteur Student einfach zu bedienen sein; dies ist abgeleitet aus den Bedienbarkeitsanforderungen. Zur Steuerung des Arbeitsplatzes soll ein GUI bereitgestellt werden, das vom Erscheinungsbild her marktverbreiteten Oberflächen ähnlich ist. Das GUI ist eine Benutzerschnittstelle, die die Mensch-Maschine Kommunikation durch graphische Elemente verwirklicht; sie erlaubt zum Beispiel Mausoperationen auf Objekten. Der Akteur

Student soll sich bei der Kommentareingabe nicht um Zeilenumbrüche kümmern brauchen. Bei fehlerhaften Eingaben soll durch einen Hinweis auf den Fehler aufmerksam gemacht werden, und fehlerhafte Eingaben sollen vom System nicht angenommen werden. Medien können groß dargestellt werden, um die Lesbarkeit zu verbessern.

Verfügbarkeit:

Verfügbarkeitsziele lassen sich aus den Zuverlässigkeitsanforderungen ableiten. Zur Wiederinbetriebnahme nach einem Ausfall ist ein BackUp für die persistenten Daten notwendig, was nahe legt, die Daten auf einem Server zu speichern. Ein Ausfall kann erfolgen durch:

- Ausfall des ParLe Servers oder
- Störung des Netzes

Die Anbindung der studenteneigenen Rechner (Clients) an den ParLe Server soll über ein öffentliches Netz (Internet) erfolgen können.

Für dieses Netz kann keine QoS (Quality of Service) gefordert werden. Wenn das Netz nach einem Ausfall wieder verfügbar ist, soll auf die persistenten Daten wieder zugegriffen werden können. Die noch nicht abgespeicherten Daten von laufenden Transaktionen der Studenten können verloren gehen.

2.2 Vorgeschlagene Softwarearchitektur

2.2.1 Überblick

Das ParLe System ist ein System nach dem Client - Server Architekturstil. Der ParLe Server stellt Dienste für den ParLe Client bereit. Er übernimmt die Interaktion mit dem Datenverwaltungssystem.

Der ParLe Client ist für die Interaktion mit dem Akteur zuständig und übernimmt die vom Akteur vorgenommenen Eingaben, führt die eigentliche Verarbeitung aus, initiiert Datenbanktransaktionen und übernimmt die Ausgabe von Informationen an den Akteur.

Die Kommunikation zwischen dem ParLe Client und dem ParLe Server erfordert einen entfernten Methodenaufruf RMI (Remote Method Invocation). Für den entfernten Methodenaufruf sind zusätzliche Klassen notwendig, die im ParLe Server zusammengefasst sind. Diese Klassen stellen Dienste bereit, um mit dem Datenverwaltungssystem zu kommunizieren. Diese Klassen sind:

- das RMIInterface, das die Schnittstelle zur Verfügung stellt
- die RMIImpl, die das Interface implementiert
- der RMIImpl_Stub, der sich automatisch aus der RMIImpl Klasse generiert und
- der RMIServer, der den ParLe Server betreibt

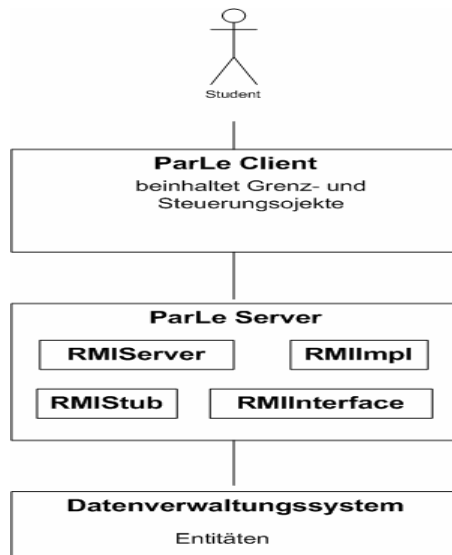


Abbildung 27: Der entfernte Methodenaufruf RMI im ParLe System

2.2.2 Systemzerlegung in Schichten

Das ParLe System lässt sich im 3-Schichtenarchitekturstil darstellen, mit den Schichten:

- Schnittstelle
- Anwendungslogik und
- Datenverwaltungssystem

Schnittstelle

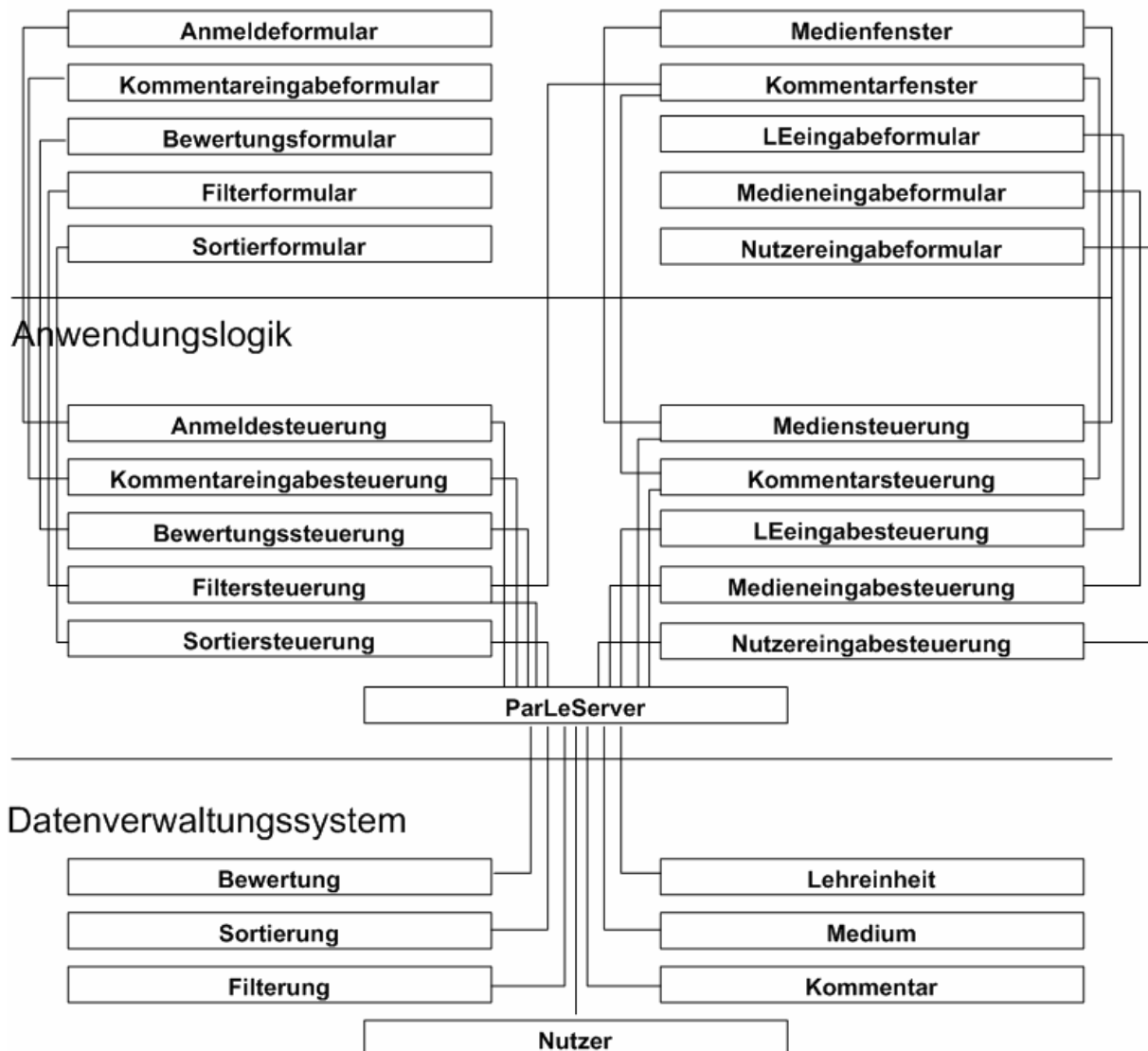


Abbildung 28: ParLe Zerlegung in ein 3 Schichtenmodell

Die Schicht Schnittstelle beinhaltet Grenzobjekte, mit denen der Akteur arbeitet, im ParLe Fall alle Formulare sowie das Medien- und Kommentarfenster. Diese Grenzobjekte wurden in der Anforderungsanalyse identifiziert und hier übernommen.

Die Schicht Anwendungslogik enthält die Steuerungsobjekte, die die Anwendung für ihre eigentliche Arbeit braucht. Zur Schicht Anwendungslogik gehören alle Steuerungen für Formulare sowie Mediensteuerung und Kommentarsteuerung. Diese Steuerungen sind alle aus dem Analysemodell übernommen. Für den entfernten Methodenaufruf gehört in diese Schicht auch der ParLe Server, der die bereits vorher beschriebenen Klassen RMIServer, RMIInterface, RMIImpl und RMIImpl_Stub umfasst.

Die Schicht Datenverwaltungssystem ist verantwortlich für Speichern und Bereitstellen von persistenten Datenobjekten und besteht aus allen Entitätsobjekten, die im Analysemodell ermittelt wurden. Die Verbindung der Entitätsobjekte untereinander ist im Objektmodell der Entitätsobjekte (Abbildung 5) beschrieben. Die Grenzobjekte und die Steuerungsobjekte sind im ParLe Client beinhaltet.

Um die Übersichtlichkeit in der Darstellung zu wahren, wurden nur wesentliche Verbindungen dargestellt.

2.2.3 Systemzerlegung in Subsysteme

Das ParLe System kann auch in funktionale Subsysteme zerlegt werden. Die Subsysteme bestehen aus einer Reihe von Lösungsdomänenklassen. Ein Subsystem ist durch seine Dienste spezifiziert, die es anderen Subsystemen zur Verfügung stellt. Das ParLe System lässt sich in folgende Subsysteme zerlegen:

- Mediensubsystem
- Nutzersubsystem
- Serversubsystem
- Sortiersubsystem
- Kommentarsubsystem
- Bewertungssystem
- Filtersubsystem

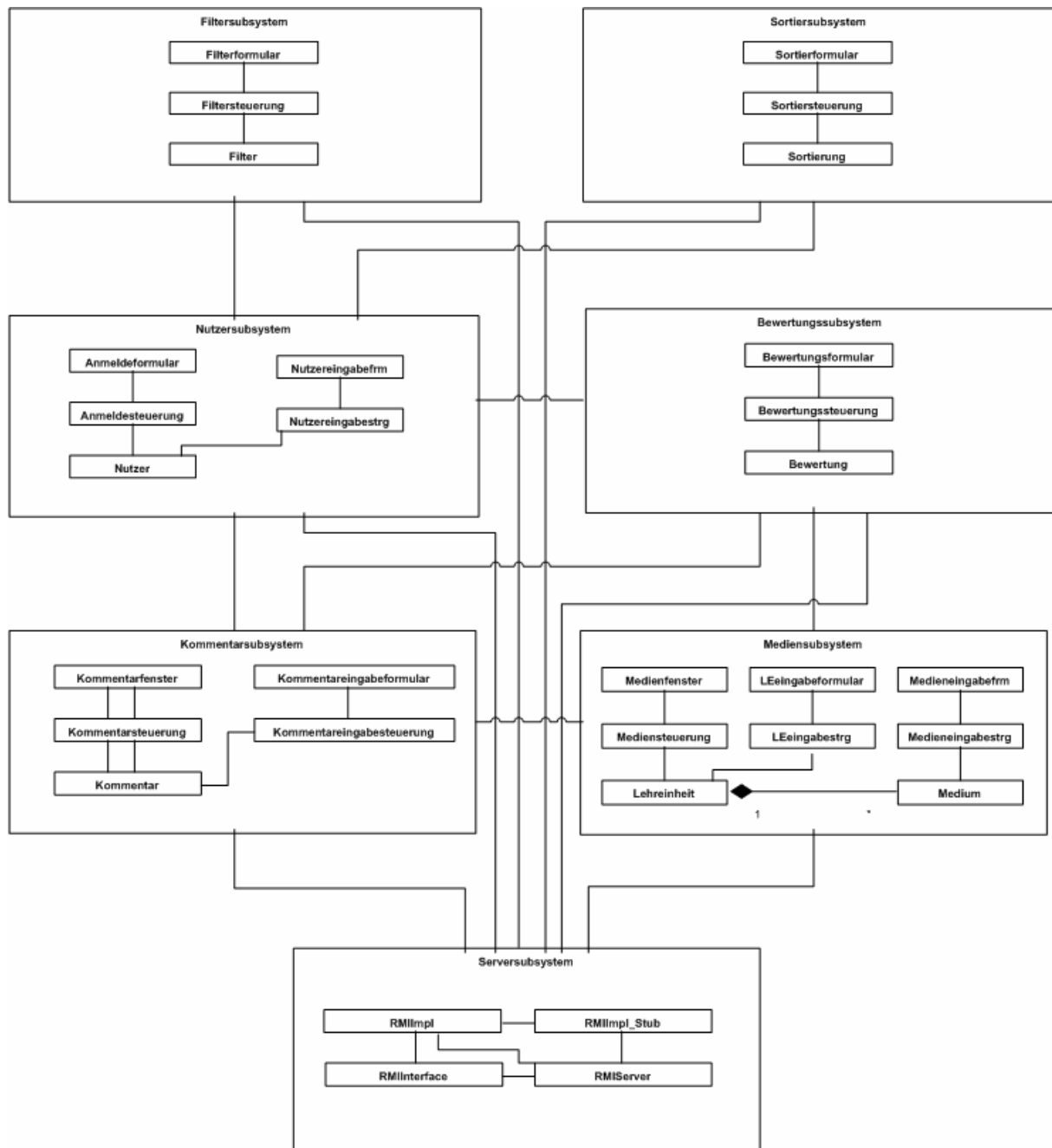


Abbildung 29: Systemzerlegung des ParLe Systems in Subsysteme.

Die Subsysteme sind als UML Klassendigramme dargestellt. Die Linien zeigen Abhängigkeiten zwischen den Subsystemen. Nicht im Detail dargestellt sind die Schnittstellen zwischen den Subsystemen; sie sind bei den einzelnen Subsystemen behandelt und dort im Detail beschrieben.

Die Subsystemzerlegung ist im Anhang in größerem Format beigelegt. Die Begriffe Formular und Steuerung wurden teilweise abgekürzt in frm für Formular und strg für Steuerung.

Mediensubsystem:

Das Mediensubsystem dient der Eingabe von Lehreinheiten und Medien durch den Akteur Administrator. Es stellt für den Akteur Student Lehreinheiten und Medien dar und lässt eine Lehreinheit oder ein Medium auswählen.

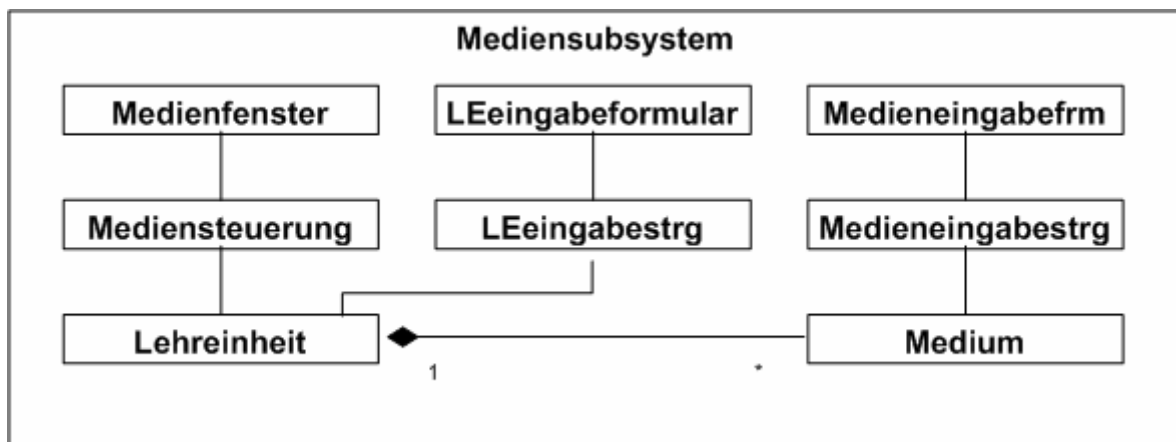


Abbildung 30: Mediensubsystem

Das Mediensubsystem bietet die Möglichkeit für den Akteur Administrator Lehreinheiten oder Medien in das ParLe System einzugeben. Hierzu dienen die Klassen LEingabeformular, LEingabesteuerung, Medieneingabeformular, Medieneingabesteuerung.

Das Mediensubsystem stellt dem Akteur Student die Möglichkeit bereit, Lehreinheiten oder Medien darzustellen, Medien zu Lehreinheiten aus- und einzuklappen und eine Lehreinheit oder ein Medium auszuwählen. Dazu dienen die Klassen Medienfenster, Mediensteuerung, Medium und Lehreinheit. Das Mediensubsystem hat eine Schnittstelle zum Kommentarsubsystem, über die zur gewählten Lehreinheit oder zum gewählten Medium zugehörige Kommentare ausgewählt werden. Es hat auch eine Schnittstelle zum Serversubsystem; diese übergibt Lehreinheiten und Medien.

Kommentarsubsystem:

Das Kommentarsubsystem stellt die zur Lehreinheit oder zum Medium gehörenden Kommentare in gewünschter Sortierreihenfolge und entsprechend der Filterbedingung dar.

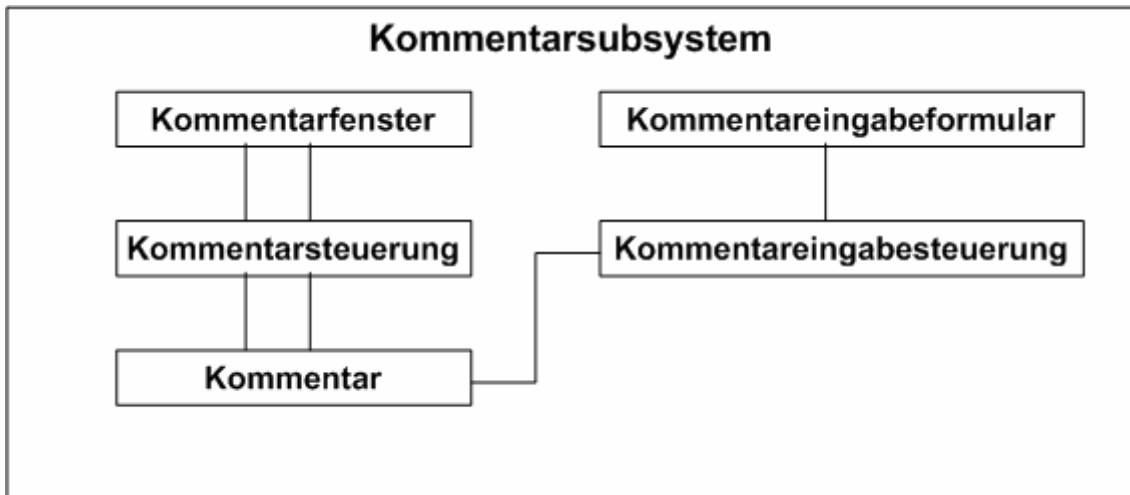


Abbildung 31: Kommentarsubsystem

Das Kommentarsubsystem besteht aus den im Analysemodell gefundenen Klassen Kommentarfenster, Kommentarsteuerung, Kommentareingabeformular, Kommentareingabesteuerung und Kommentar. Es stellt zur ausgewählten Lehreinheit oder zum ausgewählten Medium gehörende Kommentare, die besser oder gleich der Filterbedingung sind, in gewünschter Sortierreihenfolge im Kommentarfenster dar. Im Kommentarfenster kann ein Kommentar ausgewählt oder durch Doppelklick aus- oder eingeklappt werden. Es können Kommentare eingestellt werden, die zu einer Lehreinheit, einem Medium oder einem gewählten Kommentar gehören. Das Kommentarsubsystem hat zu meisten Subsystemen Schnittstellen. Die Schnittstelle zum Mediensubsystem übergibt Informationen, zu welcher Lehreinheit oder zu welchem Medium ein Kommentar gehört, der zu erstellen oder darzustellen ist. Die Schnittstelle zum Bewertungssystem bietet Informationen über die Bewertung zu einem Kommentar. Die Schnittstelle zum Nutzersubsystem übergibt Informationen, zu welchem Nutzer ein Kommentar gehört. Der Filter und die Sortierung gehören zu einem Nutzerobjekt. Die Schnittstelle zum Serversubsystem übergibt die Kommentare.

Bewertungssystem:

Das Bewertungssystem bewertet nutzerbezogene Kommentare.

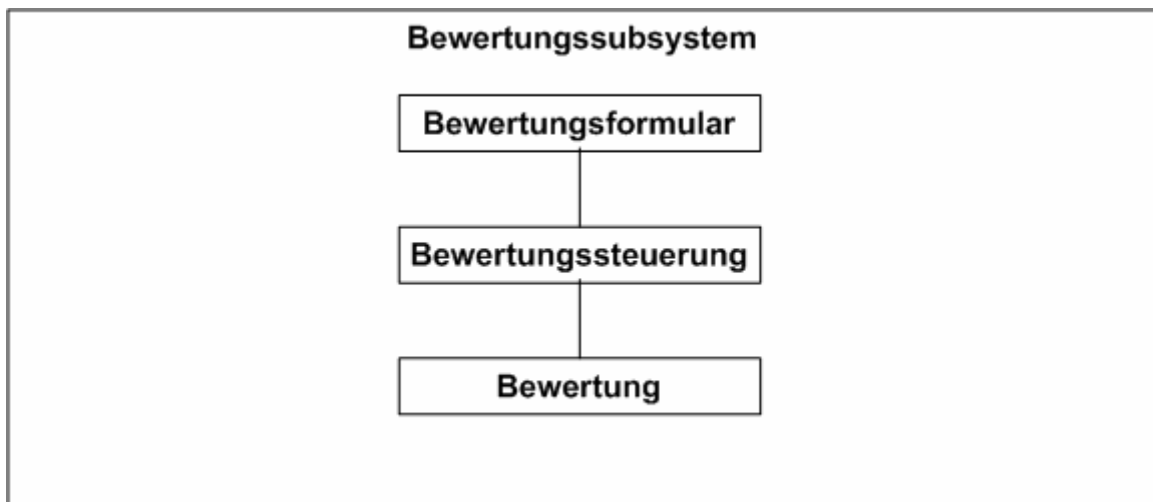


Abbildung 32: Bewertungssystem

Auch im Bewertungssystem kommen alle Klassen aus dem Analysemodell. Das Bewertungssystem stellt eine Bewertung her zwischen einem Nutzer und einem Kommentar, sofern dies zulässig ist. Eigene Kommentare dürfen nicht bewertet werden, ein Kommentar eines anderen darf nur einmal pro Nutzer bewertet werden. Das Bewertungssystem hat Schnittstellen zum Kommentarsystem, zum Nutzersystem und zum Serversystem. Die Schnittstellen zum Kommentarsystem und Nutzersystem stellen eine Beziehung her zwischen einer Bewertung, einem Nutzer und einem Kommentar. Über das Serversystem wird die Bewertung übergeben.

Nutzersystem:

Das Nutzersystem übernimmt die Anmeldung des Nutzerobjekts und die Eingabe des Nutzerobjekts durch den Akteur Administrator.

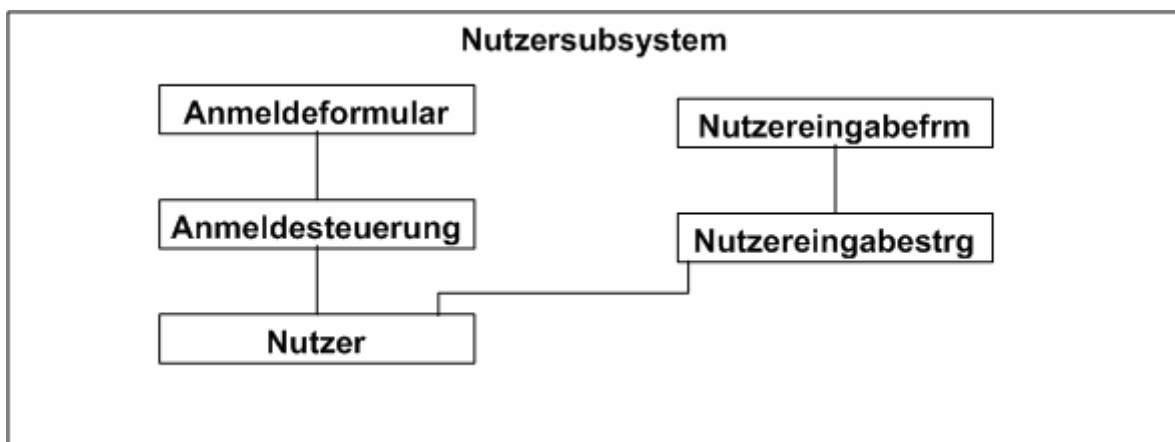


Abbildung 33: Nutzersystem

Die Klassen des Nutzersystems stammen aus dem Analysemodell. In diesem Subsystem sind die Funktionen Anmeldung des Nutzers durch den Akteur Student und Eingabe des Nutzers durch den Akteur Administrator zusammengefasst. Das Subsystem hat Schnittstellen zum Kommentarsystem, zum Bewertungssystem, zum Filtersystem, zum Sortiersystem und zum Serversystem.

Die Schnittstelle zum Kommentarsystem stellt eine Relation her zwischen einem Kommentar und einem zugehörigen Nutzer.

Die Schnittstelle zum Bewertungssystem stellt zu einem Nutzer eine kommentarbezogene Bewertung bereit. Die Schnittstelle zum Filtersystem übergibt den zum Nutzer gehörenden Filter. Die Schnittstelle zum Sortiersystem übergibt die Information, wie die nutzerbezogene Sortierung geschehen soll. Über das Serversystem werden Nutzer ausgetauscht.

Filtersubsystem:

Das Filtersubsystem übernimmt die Filterung der Kommentare.

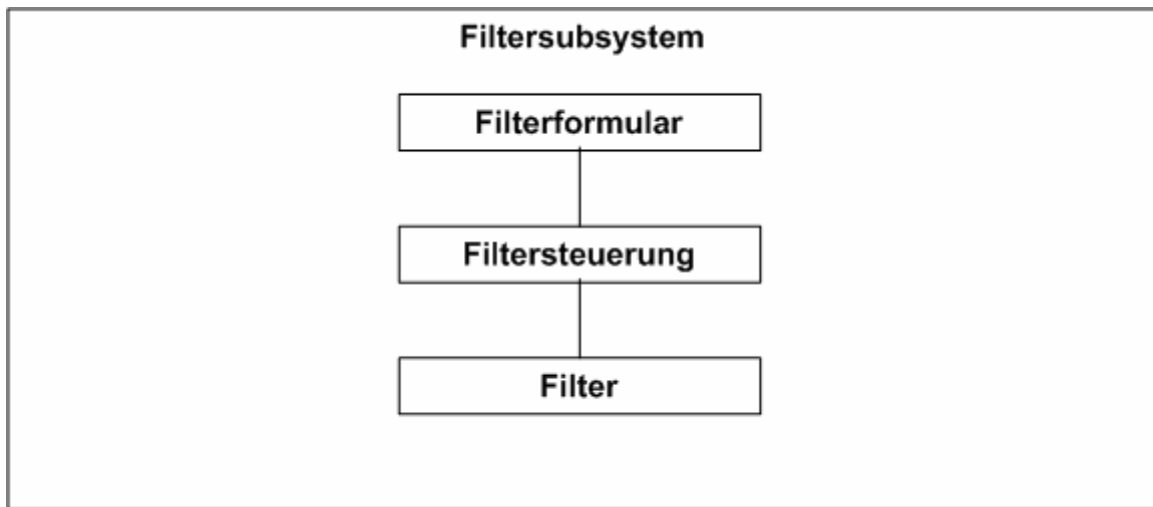


Abbildung 34: Filtersubsystem

Die Klassen des Filtersubsystems stammen aus dem Analysemodell. Das Filtersubsystem stellt einen vom Akteur Student gewählten Filter bereit. Es hat Schnittstellen zum Nutzersubsystem und zum Serversubsystem. Die Schnittstelle zum Nutzersubsystem stellt eine Relation zwischen Nutzer und Filter dar. Die Schnittstelle zum Serversubsystem stellt für den Filter den Zugriff zum Datenverwaltungssystem zur Verfügung. Die Verbindung zum Kommentarsubsystem erfolgt über das Nutzersubsystem, da ein Filter immer zu einem Nutzer gehört.

Sortiersubsystem:

Das Sortiersubsystem übernimmt die Sortierung der Kommentare.

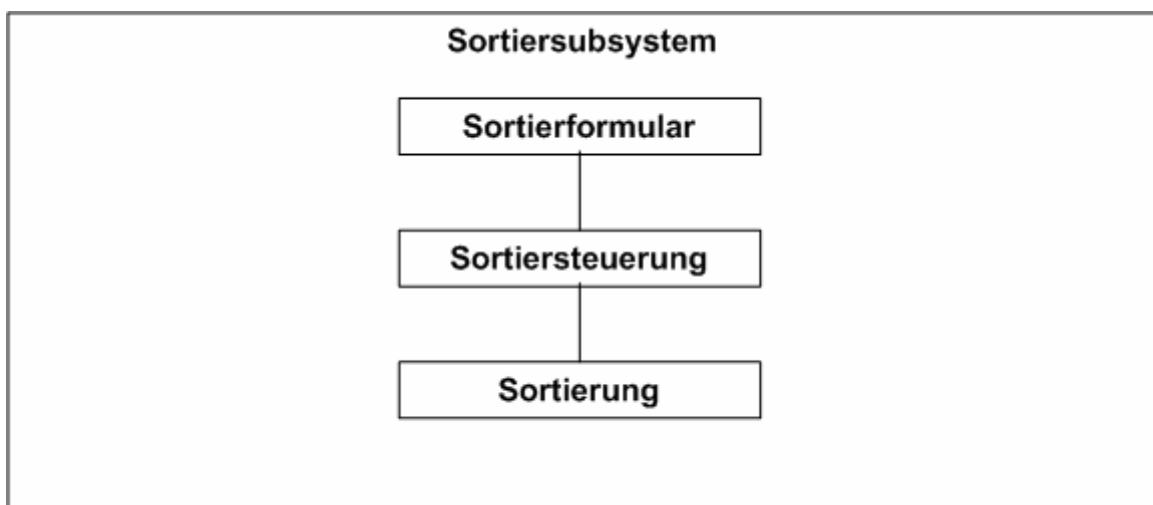


Abbildung 35: Sortiersubsystem

Die Klassen des Sortiersubsystems stammen aus dem Analysemodell. Das Sortiersubsystem stellt eine vom Akteur Student gewählte Sortierung der Kommentare bereit. Es hat

Schnittstellen zum Nutzersubsystem und zum Serversubsystem. Die Schnittstelle zum Nutzersubsystem stellt eine Relation zwischen Nutzer und Sortierung her. Die Schnittstelle zum Serversubsystem stellt für die Sortierung den Zugriff zum Datenverwaltungssystem zur Verfügung. Die Verbindung zum Kommentarsubsystem erfolgt über das Nutzersubsystem, da die Sortierung immer zu einem Nutzer gehört.

Serversubsystem:

Das Serversubsystem implementiert den Zugriff auf das Datenverwaltungssystem über RMI (Remote Method Invocation).

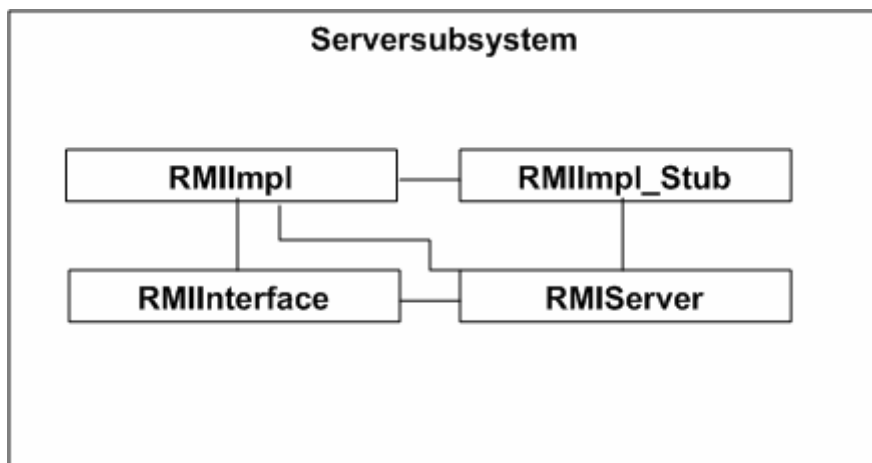


Abbildung 36: Serversubsystem

Die Klassen des Serversubsystems sind erst im Systementwurf gefunden worden, um RMI zu implementieren. Das Serversubsystem ist mit allen anderen Subsystemen verbunden und dient dem Einstellen und Auslesen der Entitätsobjekte aus dem Datenverwaltungssystem für die Überführung zu den systeminternen lokalen Objekten. Diese lokalen Objekte auf dem ParLe Client sind notwendig für die Darstellung der Lehreinheiten, Medien und Kommentare auf dem GUI des ParLe Clients.

2.2.4 Abbildung Hardware-/Software Komponenten

ParLe ist ein verteiltes System, da es von unterschiedlichen Akteuren an unterschiedlichen Rechnern genutzt wird. Es ist deshalb nötig, das System zu trennen in einen Knoten Benutzermaschine (Client) und Servermaschine (Server).

Hardware/Software-Abbildung als UML-Verteilungsdiagramm:

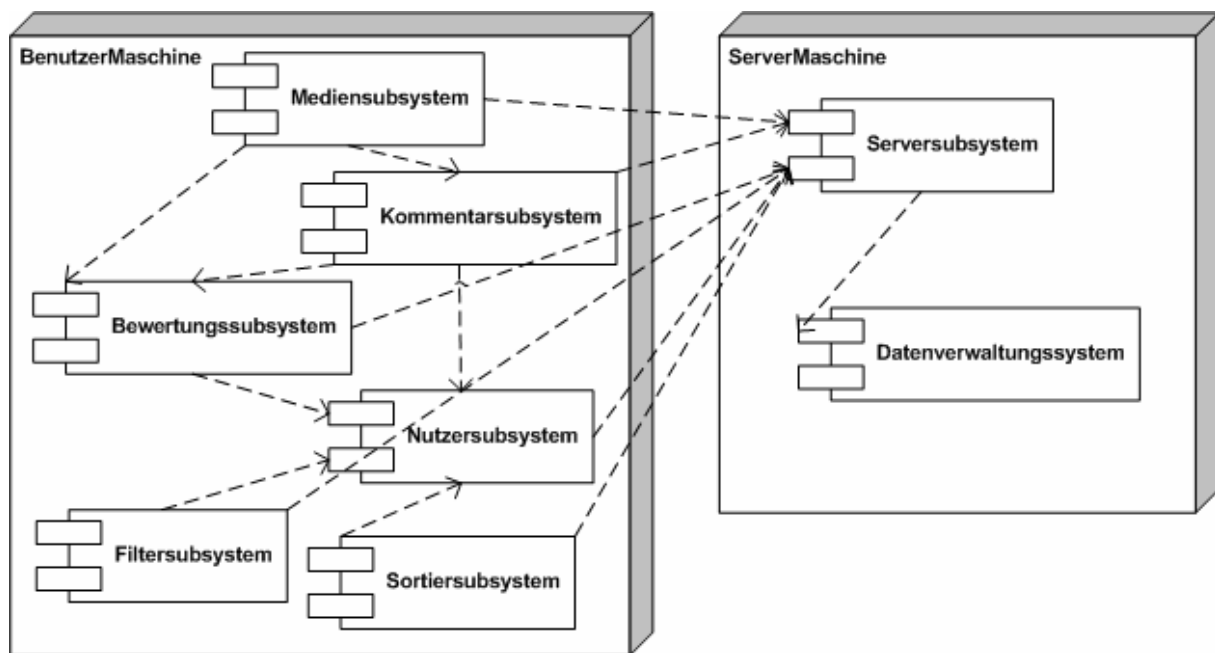


Abbildung 37: Hard-/Softwareabbildung des ParLe Systems als UML Verteilungsdiagramm

Auf der Benutzermaschine befinden sich das Mediensubsystem, das Kommentarsubsystem, das Nutzersubsystem, das Bewertungssystem, das Filtersubsystem und das Sortiersubsystem. Diese Subsysteme bieten Formulare und Fenster für die Interaktion mit den Akteuren Student und Administrator, und die clientseitige Anwendungslogik repräsentiert durch die entsprechenden Steuerungen.

Die Servermaschine stellt die Dienste zum Arbeiten auf dem Datenverwaltungssystem bereit. Das Serversubsystem und das Datenverwaltungssystem sollten auf einem Rechner liegen. Die Abbildung 37 zeigt, dass von jedem Subsystem auf der Benutzermaschine eine Kommunikationsverbindung zum Serversubsystem besteht. Es besteht daher eine starke Kopplung. Diese Kopplung kann verringert werden durch die Einführung eines RMIClient auf der Benutzermaschinenseite, der die Kommunikation mit dem Serversubsystem bündelt. Die Klasse RMIClient wird mit dem Serversubsystem zu einem neuen Kommunikationssystem zusammengefasst.

2.2.5 Management persistenter Daten

Persistente Entitätsobjekte sind Datenobjekte, die dauerhaft, über die einmalige Anwendung hinaus, verfügbar sein sollen.

Beim ParLe System sind die Entitätsobjekte aus dem Objektmodell abgeleitet:

- Nutzer
- Lehreinheit
- Medium
- Kommentar
- Bewertung
- Sortierung
- Filter

Als Datenverwaltungssystem wurde ein relationales Datenbankmanagementsystem gewählt. Die Gründe dafür sind:

- Da ParLe einen Mehrbenutzerbetrieb für mehrere Hundert Nutzer ermöglichen soll, ist Nebenläufigkeitskontrolle und Zugriffskontrolle wichtig.
- Da ParLe eine hohe Verfügbarkeit bieten soll, ist eine Wiederherstellung der Ausprägung der Datenbank nach Systemausfall wichtig. Die Datenbank soll BackUp fähig sein.
- Relationale Datenbanken bieten eine ausgereifte Technologie und skalieren gut bei großen Datenbeständen.
- Eine Datenbank ist ein gutes Instrument für den Einsatz in heterogenen Umgebungen. Dies ist in der Aufgabenstellung gefordert.
- Im Betrieb von ParLe können große Datenmengen anfallen, für die eine Speicherung in Dateien nicht sinnvoll ist.

Konkret wurde das Datenbank Management System PostgreSQL 8.1 gewählt, da es ein OpenSource Produkt ist, auf vielen Plattformen installierbar ist, einen breiten Funktionsumfang hat und dem SQL92 Standard weitgehend entspricht.

Im konzeptuellen Entwurf einer Datenbank steht das Entity-Relationship Modell (ER-Modell) im Vordergrund. Beim ER-Modell gibt es Entitäten (Gegenstände) und Relationen (Beziehungen) zwischen den Entitäten. Darüber hinaus gibt es Attribute und Rollen. „Man abstrahiert ähnliche Gegenstände zu Gegenstandstypen ... , die man graphisch als Rechtecke darstellt.... Beziehungen werden auf analoge Weise zu Beziehungstypen ... abstrahiert. Die Beziehungstypen werden als Rauten mit entsprechender Beschriftung repräsentiert.“ (Kemper/Eickler, 2001, S.35)₂ Attribute charakterisieren Gegenstände und deren Beziehungen. Schlüssel (bei der Darstellung unterstrichen dargestellt) sind die minimale Menge von Attributen, deren Werte eindeutig identifiziert sind. Da PostgreSQL in Attributen keine Umlaute unterstützt, werden diese umschrieben (z.B.: ö zu oe).

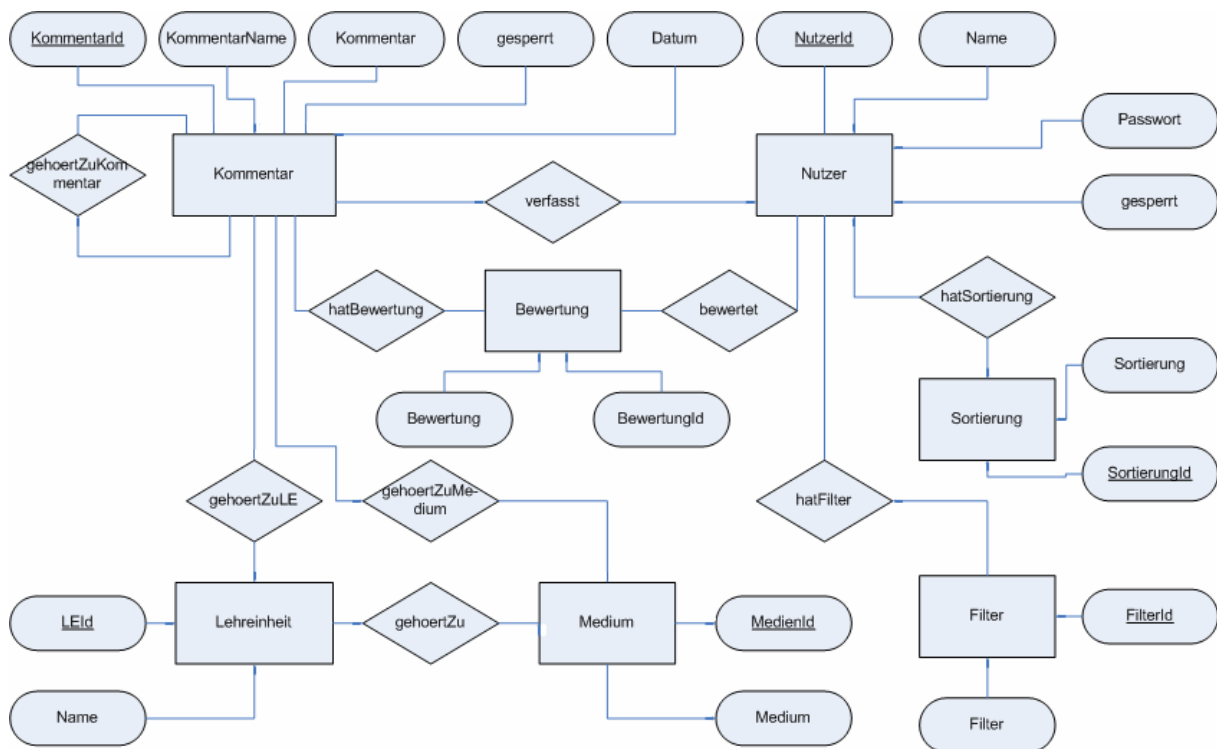


Abbildung 38: ER-Diagramm der Entitäten des ParLe Systems

Bei ParLe gibt es 7 Entitäten:

- Nutzer mit den Attributen NutzerId (Schlüssel), Name, Passwort, gesperrt
- Lehreinheit mit den Attributen LEId (Schlüssel), Name
- Medium mit den Attributen MedienId (Schlüssel), Medium
- Kommentar mit den Attributen KommentarId (Schlüssel), KommentarName, Kommentar, gesperrt, Datum
- Bewertung mit den Attributen BewertungId, Bewertung
- Filter mit den Attributen FilterId, Filter
- Sortierung mit den Attributen SortierungId, Sortierung

ParLe hat 9 Relationen:

- verfasst zwischen Kommentar und Nutzer mit der Beziehung N:1 (von einem Nutzer können mehrere Kommentare sein, aber jeder Kommentar hat nur einen Verfasser).
- bewertet zwischen Nutzer und Bewertung mit der Beziehung 1:N (eine Bewertung kann von einem Nutzer abgegeben werden, und jeder Nutzer kann mehrere Bewertungen abgeben).
- hatBewertung zwischen Kommentar und Bewertung mit der Beziehung 1:N (Eine Bewertung gehört zu einem Kommentar, und jeder Kommentar kann mehrere Bewertungen haben).
- gehoertZu zwischen Lehreinheit und Medium mit der Beziehung 1:N (Ein Medium gehört zu einer Lehreinheit, und jede Lehreinheit kann mehrere Medien beinhalten).
- gehoertZuKommentar zwischen Kommentar und Kommentar mit der Beziehung N:1 (ein Kommentar gehört zu einem Oberkommentar, und ein Oberkommentar kann mehrere Unterkommentare haben).
- gehoertZuLE zwischen Kommentar und Lehreinheit mit der Beziehung N:1.
- gehoertZuMedium zwischen Kommentar und Medium mit der Beziehung N:1.
- hatSortierung zwischen Nutzer und Sortierung mit der Beziehung 1:1 (Ein Nutzer hat eine gewählte Sortierung, und eine Sortierung hat einen Nutzer).
- hatFilter zwischen Nutzer und Filter mit der Beziehung 1:1 (Ein Nutzer hat einen Filter gewählt, und ein Filter hat einen Nutzer).

Nach der Darstellung im ER-Modell folgt die Umsetzung in ein relationales Modell. Ziel dabei ist ein Modell in 3.Normalform, um nicht denselben Fakt mehrfach zu speichern, was zu Anomalien führen könnte (bei Update- oder Deleteanweisungen). Die Bedingung dabei ist, dass kein Nichtschlüsselattribut einen Fakt einer Attributmenge darstellt. Ein Schlüsselattribut ist ein in einem Schlüssel enthaltener Wert. Das bei ParLe verwendete Relationsschema in 3. Normalform lautet wie folgt:

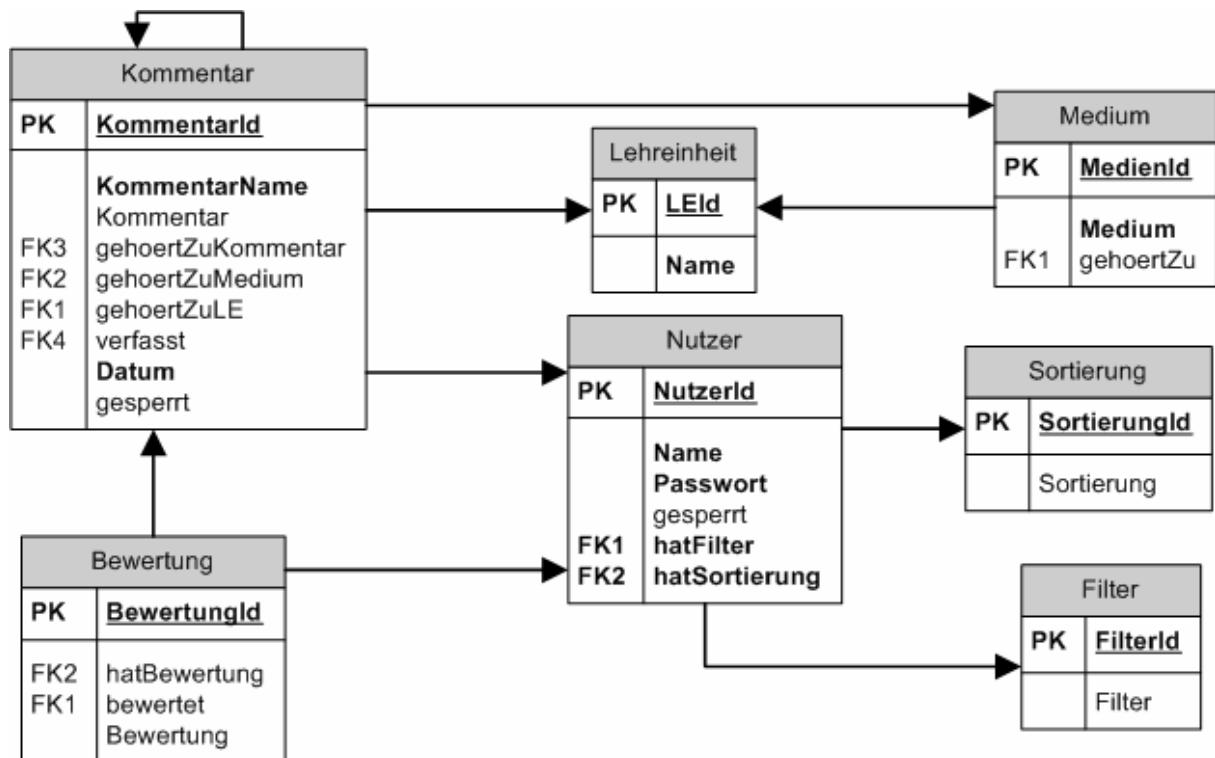


Abbildung 39: Das **Datenbankschema** von ParLe in 3. Normalform

Bei der Transformation des ER-Diagramms in ein Datenbankschema in 3. Normalform wird zuerst eine Tabelle zu jeder Entität mit den zugehörigen Attributen gebildet. Danach werden die Relationen dargestellt zwischen den Entitäten mit dem Schlüssel der zu ihnen gehörenden Entitäten und ihren Attributen. Die neun Relationen *verfasst*, *bewertet*, *hatBewertung*, *gehörtZU*, *gehörtZuKommentar*, *gehörtZuMedium*, *gehörtZuLE*, *hatSortierung*, *hatFilter* lassen sich mit den anderen Relationen zusammenfassen. Relationen mit gleichen Schlüsseln werden zusammengefasst. Die Entität *Lehreinheit* bleibt unverändert, da sie auf der 1 - Seite aller mit ihr verbundenen Beziehungen steht. Die Relation *gehörtZu* zwischen *Lehreinheit* und *Medium* hat den Schlüssel *MedienId* und wird somit mit *Medium* zusammengefasst; dabei erhält *Medium* den Fremdschlüssel, der sich auf die *LEId* der *Lehreinheit* bezieht. Da *Kommentar* auf der N - Seite der Relationen *gehörtZuKommentar*, *gehörtZuLE*, *gehörtZuMedien* und *verfasst* liegt, werden bei der Entität *Kommentar* folgende Fremdschlüssel eingefügt:

- *gehörtZuKommentar* mit *KommentarId* des Oberkommentars
- *verfasst* mit der *NutzerId* eines Nutzers
- *gehörtZuLE* mit *LEId* einer *Lehreinheit*
- *gehörtZuMedium* mit *MedienId* eines Mediums

Da *Bewertung* auf der N - Seite der Relationen *hatBewertung* und *bewertet* liegt, werden folgende Fremdschlüssel eingefügt:

- *hatBewertung*, die *KommentarId* des dazugehörigen Kommentars
- *bewertet*, die *NutzerId* des dazugehörigen Nutzerobjekts

Der *Nutzer* hat als Fremdschlüssel *hatFilter* und *hatSortierung*, die auf die *FilterId* des Filters und die *SortierId* der *Sortierung* verweisen.

Da die einzelnen Relationen nun der 3. Normalform entsprechen, ist auch das Gesamtschema in 3. Normalform.

Die Umsetzung der SQL-Statements im PostgreSQL 8.1 Dialekt ist im Benutzerhandbuch aufgeführt.

Im Datenbankschema ist für den Namen der *Lehreinheit* ein Feld von 200 Zeichen vorgesehen. Beim Löschen einer *Lehreinheit* werden die dazugehörigen *Medien* kaskadierend

mitgelöscht. Beim Ändern der LEId werden die dazugehörigen Medien auf den neuen Wert eingestellt.

Medien sind als Typ OID dargestellt, das ByteArrays verwendet und deshalb für die Darstellung von Folien geeignet ist. Beim Löschen eines Mediums werden alle dazugehörigen Kommentare und Unterkommentare kaskadierend gelöscht. Beim Ändern der MedienId werden die dazugehörigen Kommentare angepasst.

Im Datenbankschema ist für den Kommentarname ein Feld von 200 Zeichen vorgesehen, der Kommentar selbst ist als bytea umgesetzt. Wird ein Kommentar gelöscht, werden kaskadierend alle Unterkommentare gelöscht. Wird die KommentarId verändert, passen sich die Unterkommentare entsprechend an.

Der Name des Nutzers und das Passwort sind auf 80 Zeichen limitiert. Wird ein Nutzer gelöscht, wird in der Tabelle Bewertung die NutzerId auf null gesetzt; die Bewertung geht also nicht verloren. Wird ein Kommentar gelöscht oder geändert, löscht/ändert sich auch die Referenz in der Tabelle Bewertung.

Da der JDBC-Treiber für PostgreSQL deutsche Umlaute und einige Sonderzeichen nicht unterstützt, müssen diese explizit umkodiert werden. Dies geschieht in Anlehnung an die HTML Kodierung für Umlaute. Bei den Spalten des Datenbankschemas wird wie oben gezeigt auf Umlaute verzichtet. Der Akteur Student kann Umlaute der deutschen Sprache und alle gängigen Sonderzeichen (z.B.: €) verwenden und sich auch anzeigen lassen; diese werden explizit in der nun zusätzlich einzufügenden Klasse Umlaut umkodiert.

2.2.6 Zugriffskontrolle

Der Akteur Student hat in einem Mehrbenutzersystem unterschiedliche Zugriffsmöglichkeiten auf Funktionen und Daten. Bei ParLe muss sich der Student zu Beginn authentifizieren.

Ein Anmeldedialog erfordert die Eingabe des Nutzernamens und seines Passworts. Ist der Student nicht registriert, oder stimmt die Kombination aus Passwort und Nutzername nicht mit den gespeicherten Daten überein, beendet sich der ParLe Client. Ist der Student authentifiziert, wird der ParLe Client zur Bedienung freigeschaltet.

Jeder authentifizierte Akteur Student hat lesenden Zugriff zu allen Informationen. Kommentare können gesperrt werden, um z.B. politisch inkorrekte Informationen nicht mehr zugänglich zu machen.

Einzelne Nutzer können gesperrt werden. Die dazugehörigen Akteure haben dann keinen Zugang mehr zum ParLe System. Ihre Kommentare bleiben aber weiterhin im System enthalten.

Jeder authentifizierte Student kann Kommentare erstellen und unter folgenden Bedingungen bewerten:

- Ein Kommentar kann von einem Nutzer nur einmal bewertet werden.
- Ein selbst erstellter Kommentar kann nicht bewertet werden.

Das ParLe System vergleicht dabei den Nutzer mit dem Urheber des Kommentars über eine Zugriffskontrollliste.

Eine Verschlüsselung wird nicht eingesetzt.

Der Akteur Administrator hat nach Anmeldung die Möglichkeit Lehreinheiten und Medien zu verwalten. Er kann Nutzer registrieren, sperren und Kommentare sperren.

2.2.7 Kontrollfluss

Das ParLe System ist ereignisgetrieben gesteuert. Ein externes von einem Akteur angestoßenes Ereignis, wird mit Ereignisbehandlungsroutinen abgefangen. Für die

ereignisgetriebene Steuerung werden zwei neue Klassen eingeführt und zwar ParLeApp und ParLeMainFrame.

Die Klasse ParLeApp startet das UserInterface des ParLe Client.

Die Klasse ParLeMainFrame beinhaltet die GUI des ParLe Systems. Sie ist für die Darstellung der in den Grenzobjekten enthaltenen Daten zuständig und gibt bei Ereignissen die benötigten Werte an die Grenzobjekte ab, die ihrerseits die Steuerungsobjekte aufrufen. Die Einführung der ParLeMainFrame Klasse hat den Vorteil, dass bei Erweiterung des Systems (z.B. Richtung browserbasierter Anwendung) nur darzustellende Objekte zu überarbeiten sind, alle anderen Klassen sind wieder verwendbar.

2.2.8 Randbedingungen

Zu den Randbedingungen gehören Aussagen, wie das System gestartet, initialisiert und heruntergefahren werden kann, sowie Aussagen zum Vorgehen beim Ausfall von Teilsystemen.

Um das ParLe System zu betreiben ist ein Administrator notwendig. Dieser muss den ParLe Server starten und verwalten.

Start des ParLE Server Systems:

Auf einem Server muss PostgreSQL 8.1 installiert werden, und es muss ein Verzeichnis zur Verfügung gestellt werden, in das Medien temporär zwischengespeichert werden können, um sie dann über RMI an den ParLe Client zu übertragen; dies ist eine Besonderheit von PostgreSQL, das BLOBs (Binary Large Objects) nur in ein Verzeichnis auslesen kann. „Es ist ... nicht möglich, den Inhalt einer in der Datenbank gespeicherten Graphik einfach als Ergebnis einer Abfrage auf dem Bildschirm auszugeben“ (Hartwig, 2001, S. 112)₃.

Der Pfad der Datenbank und das Passwort der Datenbank müssen eingegeben werden. Die Java Ablaufumgebung muss eingerichtet werden mit Klassenpfaden. Die RMI Registrierung muss gestartet werden. Näheres siehe hierzu im Abschnitt Benutzerhandbuch.

Start des ParLe Client Systems:

Das ParLe Client System soll über das Netz heruntergeladen werden können und direkt gestartet werden können. Näheres hierzu im Abschnitt Benutzerhandbuch.

Serververwaltung:

Zur Initialisierung des ParLe Servers müssen Daten der Lehreinheiten und Medien in die Datenbank eingepflegt werden und Nutzer angelegt werden.

Im laufenden Betrieb können Lehreinheiten und/oder Medien geändert, hinzugefügt oder gelöscht werden. Dies geschieht kaskadierend, so dass sich die zugehörigen Fremdschlüssel bei Änderung anpassen und beim Löschen die dazugehörigen Informationen ebenfalls gelöscht werden.

Der Administrator kann Nutzer und Kommentare sperren.

Die Programmierung dieser Administrationsfunktionen ist nicht Aufgabe dieser Diplomarbeit. Im Benutzerhandbuch wird aber beschrieben, wie ein Administrator dabei vorgehen kann.

Beendigung des Systems:

Der ParLe Client wird durch die Menufunktion „Beenden“ beendet. Der ParLe Server wird beendet, indem der Serverprozess vom Administrator explizit „gekillt“ wird.

Ausfälle:

Beim Clientausfall muss dieser neu gestartet werden.

Beim Serverausfall muss dieser neu gestartet werden. Während des Ausfalls besteht für den ParLe Client keine Verbindung zu den Serverdiensten und zur Datenbank.

Beim Datenbankausfall muss die Datenbank neu gestartet werden; z.B. aus dem BackUp. Noch nicht gesicherte Daten können verloren gehen.

Bei Ausfall der Netzverbindung bleiben die gespeicherten Daten konsistent, aber die laufenden Transaktionen werden nicht zu Ende geführt, und die Daten gehen verloren.

3 Objektentwurf

3.1 Einleitung

Im Objektentwurf werden Pakete definiert, Standardkomponenten gesucht, eventuell noch fehlende Teile identifiziert und die Klassenschnittstellen der Objekte im Detail beschrieben. Als verfügbare Standardkomponente wird in ParLe das relationale Datenbanksystem PostgreSQL 8.1 eingesetzt; andere Standardkomponenten konnten nicht identifiziert werden. Für die Kommunikation zwischen dem ParLe Client und dem ParLe Server wird das Programmgerüst Java RMI (Remote Method Invocation) angewandt. Der Vorteil eines solchen Programmgerüsts ist es, die Wiederverwendbarkeit zu erhöhen und die Erweiterbarkeit zu vereinfachen.

„RMI ... ist der Mechanismus in Java, um entfernte Objekte und deren Angebote zu nutzen. ... Mit RMI ist es also möglich, auf hohem Abstraktionsniveau zu arbeiten. Stellvertreter nehmen die Daten entgegen und übertragen sie zum Server. Nach der Antwort präsentiert der Stellvertreter das Ergebnis.“ (Ullenboom, 2006, S. 1216)⁴

RMI hat 4 Schichten:

- Application-Layer
- Proxy-Layer
- Remote-Reference-Layer
- Transport-Layer

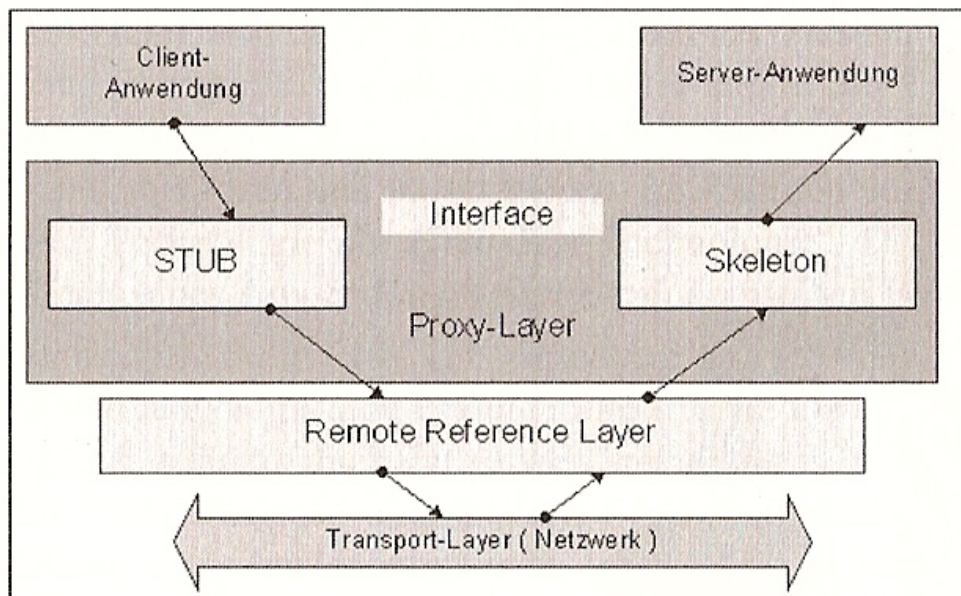


Abbildung 40: RMI-Architektur (Tabatt & Wolf, 2004, S.460)⁵

Zum Application-Layer gehören die Client- und die Serveranwendung; die Clientanwendung ruft externe Methoden auf, während die Serveranwendung die aufgerufenen Methoden implementiert.

Der Proxy-Layer übersetzt den Aufruf der Methode aus dem aufrufenden Objekt in einen Aufruf einer externen Methode. Zum Proxy-Layer gehören Stub (clientseitig) und Skeleton (serverseitig).

Der Remote-Reference-Layer übernimmt die Abbildung zwischen Proxy-Layer und dem Transport-Layer.

Der Transport-Layer ist für den eigentlichen Transport der Daten zwischen Client und Server verantwortlich.

Für einen entfernten Methodenaufruf über RMI spricht:

- RMI ist der Standard für entfernten Methodenaufruf in Java.
- RMI bildet entfernte Methoden wie lokale Methoden ab und erlaubt damit dem ParLe Client und dem ParLe Server sowohl auf getrennten Rechnern als auch auf einem Rechner betrieben zu werden. In der in dieser Diplomarbeit realisierten Anwendung werden der ParLe Client und der ParLe Server auf einem Rechner betrieben.
- RMI stellt Methoden bereit, mit der alle Steuerungsobjekte mit allen Entitätsobjekten kommunizieren können.

3.2 Pakete

Nachfolgend wird die Verteilung der Klassen von ParLe auf Pakete vorgenommen. Das ParLe System wird auf zwei Pakete verteilt:

- Paket ParLe Server
- Paket ParLe Client

Das Paket ParLe Server realisiert die Verbindung zur Datenbank und implementiert mit den nachfolgend bezeichneten Klassen die serverseitige RMI Funktionalität:

- Die Klasse RMIServer ist die Hauptklasse der Serveranwendung.
- Das Interface RMIInterface listet die auf dem Server integrierten Methoden auf; dies ist notwendig für RMI.
- Die Klasse RMIImpl realisiert die Programmierung dieses Interfaces.
- Die Klasse RMIImpl_Stub ist für RMI Aufrufe notwendig und wird automatisch generiert.

Das Paket ParLe Client ist zuständig für die Interaktion mit dem Akteur. Es übernimmt die Ein- / Ausgabe der Daten, greift über RMI auf den ParLe Server zu und übernimmt alle Bearbeitungsaufgaben, wie Anzeigen aufbereiten, Sortieren, Filtern usw..

Nachfolgend werden die Klassen des ParLe Client aufgeführt:

- Die Hauptklasse der Anwendung ist ParLeApp. Sie ist für den Start und Betrieb des ParLe Clients zuständig.
- Die Klasse RMI Client ist der clientseitige Stellvertreter der RMIImpl Klasse und stellt Methoden bereit, um zwischen der ParLe Clientanwendung und dem RMI Server zu kommunizieren.
- Die ParLeMainFrame Klasse ist für Ereignisbehandlung zuständig und bietet den Rahmen für die Darstellung der Grenzobjekte.

Zusätzlich sind im ParLe Client noch alle Klassen aus den Subsystemen enthalten:

- Mediensubsystem
- Kommentarsubsystem
- Bewertungssystem
- Nutzersubsystem
- Sortiersubsystem
- Filtersubsystem
- Die Klasse Umlaut kodiert Sonderzeichen um, um JDBC Konformität zu erreichen.

Bei der Realisierung zeigt sich, dass noch zwei zusätzliche Klassen eingeführt werden müssen:

- Die Klasse Sortierer erweitert das Interface Comparator und gibt an, wie zwei Kommentarobjekte zu vergleichen sind.

- Die Klasse Zuschneider ist dafür zuständig, die Zeilenzahl eines Kommentars zu ermitteln und den Kommentar entsprechend wortweise umzubereiten.

3.3 Schnittstellenspezifikation

Nachfolgend werden alle Klassen und ihre Schnittstellen beschrieben, Operationen und Parameter inbegriffen. In den folgenden Darstellungen sind Methoden *kursiv* und Attribute normal geschrieben.

3.3.1 Die Klassen des ParLe Servers

RMIServer
+ main(argv:string[])

Tabelle 19: Die Klasse **RMIServer**

Die RMIServer Klasse hat nur eine main Methode zum Ablauf des Servers.

RMIImpl
+ getLEName():ArrayList + getCountMedien(LEId:int):int + getMedien(LEId:int):ArrayList + getMedienID(MedienId:int):ArrayList + getKommentarFromLE(LEId:int,Filter:Filter):ArrayList + getbewertet(KommentarId:int):double + getKommentarFromMedium(MedienId:int,Filter:Filter):ArrayList + getKommentarFromKommentar(KommentarId:int,Filter:Filter):ArrayList + setKommentarLE(Name:string,Kommentar:string,gehörtZuLE:int,verfasst:Nutzer) + setKommentarMedium(Name:string, Kommentar:string, gehörtZuMedium:int, verfasst:Nutzer) + setKommentarKommentar(Name:string, Kommentar:string, gehörtZuKommentar:int, verfasst:Nutzer) + isBewertet(NutzerId:int,KommentarId:int):boolean + setBewertung(NutzerId:int,KommentarId:int,Bewertung:int) + idRegistriert(Name:string,Passwort:string):int + setPassword(Name:string,Passwort:string) + getSortierung(NutzerId):Sortierung + getFilter(NutzerId):Filter + setSortierung(NutzerId:int,Sortierung:Sortierung) + setFilter(NutzerId:int,Filter:Filter)

Tabelle 20: Das Interface **RMIInterface** und die Klasse **RMIImpl**

Die RMIImpl Klasse implementiert die im RMIInterface definierten Methoden. Das RMIInterface und dessen Implementierung RMIImpl haben die oben dargestellten Methoden zum Zugriff auf die Datenbank. Zu beachten ist, dass über RMI nur serialisierbare Objekte übertragen werden können, deshalb wird im Folgenden mit ArrayLists serialisierbarer Objekte gearbeitet.

Die Datenbankzugriffe sind aus Performancegründen so gestaltet, dass keine unnötigen Objekte aus der Datenbank ausgelesen werden.

Die Methode *getLEName()* liefert alle Lehreinheiten in einer ArrayList einer ArrayList aus Name und LEId der Lehreinheiten. Die *getCountMedien()* Methode gibt zur LEId der gewählten Lehreinheit die Zahl der dazugehörigen Medien aus. *getMedien()* gibt bei Eingabe der LEId der Lehreinheit eine ArrayList zurück, die aus einer ArrayList aus MedienId, gehörtZu und Medium besteht. Dabei müssen die Medien, um sie serialisierbar zu machen, in ein ByteArray überführt werden.

Die Methode *getMedienID()* gibt bei Eingabe einer MedienId eines Mediums die ArrayList der Attribute des Mediums zurück. *getKommentarFromLE()* gibt zu einer LEId einer gegebenen Lehreinheit und dem Filter die Liste (ArrayList) der dazugehörigen Kommentare als ArrayList ihrer Attribute zurück, wobei auf die Filterbedingung und Sortierung geachtet wird. Die Methode *getbewertet()* gibt, bei Eingabe einer KommentarId die aktuelle Durchschnittsbewertung zurück. Die Methode *getKommentarFromMedium()* gibt zu einem gegebenen Medium (MedienId) und dem übergebenen Filter die dazugehörigen Kommentare zurück, falls sie dem Filter entsprechen. Ebenso gibt *getKommentarFromKommentar()* zur KommentarId eines gegebenen Kommentars und des Filters die direkt dazugehörigen Unterkommentare aus.

Die *setKommentarLE()* Methode speichert Kommentare, die direkt zu einer Lehreinheit gehören, mit den Attributen KommentarName, Kommentar, Datum, gesperrt, verfasst und gehörtZuLE in der Datenbank ab. Ebenso speichert *setKommentarMedium()* einen direkt zu einem Medium gehörenden Kommentar mit den oben genannten Attributen ab, wobei statt gehörtZuLE gehörtZuMedium überführt wird. Kommentare, die zu Oberkommentaren gehören, werden mit *setKommentarKommentar()* mit den Attributen KommentarName, Kommentar, Datum, gesperrt, verfasst und gehörtZuKommentar in die Datenbank überführt.

Die Methode *isBewertet()* testet bei Eingabe einer NutzerId eines Nutzers und einer KommentarId eines Kommentars, ob schon eine Bewertung gespeichert ist, und gibt true aus, wenn ein Eintrag in der Datenbank vorhanden ist. *setBewertung()* speichert eine Bewertung in die Datenbank, unter Übergabe der NutzerId eines Nutzers, der KommentarId des zu bewertenden Kommentars und der eigentlichen Bewertung.

Die Methode *idRegistriert()* gibt nach Übergabe eines Passworts und eines Namens aus, mit welcher NutzerId dieser Nutzer registriert ist, falls es einen entsprechenden Eintrag in der Datenbank zum eingegebenen Tupel gibt. Die Methode *getSortierung()* ruft zu einer gegebenen NutzerId die dazugehörige Sortierung auf. Die Methode *setSortierung()* stellt die gewünschte Sortierung in die Datenbank ein. Die Methode *getFilter()* holt den Filter zu einer gegebenen NutzerId aus der Datenbank. Der zu einem Nutzer gehörende Filter wird über *setFilter()* in die Datenbank eingestellt.

Auf die Darstellung der Klasse *RMIImpl_Stub* wird verzichtet, da diese sich selbst aus der Klasse *RMIImpl* generiert.

3.3.2 Die Klassen des ParLe Client

ParLeApp
+ packFrame: boolean
+ main(args:string)

Tabelle 21: Die Klasse **ParLeApp**

Die Klasse *ParLeApp* ist die Hauptklasse des Pakets *ParLe Client*. Sie hat eine *main()* Methode, die mit *run()* das *UserInterface* startet.

RMIClient
<pre> + getServer() + getLEName():ArrayList + getCountMedien(LEId:int):int + getMedien(LEId:int):ArrayList + getMedienID(MedienId:int):ArrayList + getKommentarFromLE(LEId:int,Filter:Filter):ArrayList + getbewertet(KommentarId:int):double + getKommentarFromMedium(MedienId:int,Filter:Filter):ArrayList + getKommentarFromKommentar(KommentarId:int,Filter:Filter):ArrayList + setKommentarLE(Name:string,Kommentar:string,gehörtZuLE:int,verfasst:Nutzer) + setKommentarMedium(Name:string, Kommentar:string, gehörtZuMedium:int, verfasst:Nutzer) + setKommentarKommentar(Name:string, Kommentar:string, gehörtZuKommentar:int, verfasst:Nutzer) + isBewertet(NutzerId:int,KommentarId:int):boolean + setBewertung(NutzerId:int,KommentarId:int,Bewertung:int) + idRegistriert(Name:string,Passwort:string):int + getSortierung(NutzerId):Sortierung + getFilter(NutzerId):Filter + setPassword(Name:string,Passwort:string) + setSortierung(NutzerId:int,Sortierung:Sortierung) + setFilter(NutzerId:int,Filter:Filter) </pre>

Tabelle 22: Die Klasse **RMIClient**

Die Klasse `RMIClient` ist die clientseitige Schnittstelle, die mit dem `ParLe Server` kommuniziert. Die Methode `setServer()` setzt den Servernamen auf den Namen des gewünschten Servers. Die anderen Methoden rufen über RMI die Methoden der `RMIIImpl` Klasse auf. Sie heißen wie ihre entfernten Methoden.

ParLeMainFrame
<pre> + Nutzer:Nutzer + Filter:Filter + Sortierung:Sortierung + allE:ArrayList + prleKomm:ArrayList ... </pre>
<pre> + jbInit() + jPanelMedien_mouseClicked(MouseEvent e) + vorButton_mouseClicked(MouseEvent e) + zurueckButton_mouseClicked(MouseEvent e) + jPanelKommentar_mouseClicked(MouseEvent e) + jButtonInsert_mouseClicked(ActionEvent e) + jButtonInsertOK_mouseClicked(MouseEvent e) + jButtonSort_mouseClicked(MouseEvent e) + jButtonSortOK_mouseClicked(MouseEvent e) + jButtonBewertung_mouseClicked(MouseEvent e) + jButtonBewertungOK_mouseClicked(MouseEvent e) + jButtonPasswortOK_mouseClicked(MouseEvent e) + dlg_focusLost(FocusEvent e) </pre>

+ JMenuItemEnde_actionPerformed(ActionEvent e) + JMenuItemPasswort_actionPerformed(ActionEvent e) + JButtonPasswortÄndern_mouseClicked(MouseEvent e)
--

Tabelle 23: Die Klasse **ParLeMainFrame**

Die ParLeMainFrame Klasse beinhaltet die GUI des ParLe Systems. Sie ist für die Darstellung der in den Grenzobjekten enthaltenen Daten zuständig und gibt bei Ereignissen die benötigten Werte an die Grenzobjekte ab. Zum Aufruf einiger Methoden anderer Klassen sind Variablen eingeführt, die die Darstellung und den aktuellen Zustand der ausgegebenen Fenster betreffen und für deren Darstellung relevant sind. Es sind neben den Attributen der enthaltenen GUI-Objekte im Wesentlichen die beiden ArrayLists *alLE* und *prleKomm*, die im Falle von *alLE* die Liste der darzustellenden Lehreinheiten und Medien beinhaltet und im Falle von *prleKomm* die Liste der dazu darzustellenden Kommentarobjekte. Zu beachten ist, dass die internen Objekte zur Darstellung noch weitere Attribute benötigen, als die persistent in der Datenbank gespeicherten. Dies ist in den entsprechenden Objektklassen definiert.

Für jede Aktion auf dem GUI gibt es eine Unterklasse, die die Steuerung der Aktion einleitet. Die Unterklassen sind hier nicht erwähnt.

Die Methode *jbInit()* lädt beim Starten des Servers den ParLeMainFrame mit den am Anfang über das Medienfenster darzustellenden Lehreinheiten (über *stelleDarLE()* der Klasse Medienfenster) und stellt den Anmeldedialog für die Eingabe in das Anmeldeformular bereit. *jPanelMedien_mouseClicked()* fängt Mausklicks auf dem Paneel auf, die dem Medienfenster übergeben werden. Ein einfacher Klick wählt ein Medium oder eine Lehreinheit aus, während der Doppelklick auf eine Lehreinheit die zugehörigen Medien aus- und einklappt. Ein Doppelklick auf ein Medium vergrößert dieses in einem neuen Dialog. In diesem Dialog ermöglicht der *vorButton_mouseClicked()* und der *zurückButton_mouseClicked()* die Navigation durch die Medien (Folien) einer Lehreinheit.

jPanelKommentar_mouseClicked() fängt Mausklicks auf dem Kommentarpaneel auf und übergibt sie an das Kommentarfenster. Dabei ist für einen einfachen Klick eine Markierung abzugeben oder aufzuheben. Ein Doppelklick klappt den eigentlichen Kommentar mit seinen Metadaten auf. Die Ereignisbehandlungsroutine *jButtonInsert())_mouseClicked* öffnet den Dialog *dlgInsert*, in dem die Kommentare eingestellt werden können. Der OK-Button zum Einstellen des Kommentars löst *jButtonInsertOK_mouseClicked()* aus, das die Daten dem Kommentareingabeformular übergibt.

Ähnlich ist es beim Sortieren und Filtern. Hier wird der Dialog *dlgSort*, der die Eingabe einer Sortierreihenfolge und eines Filters ermöglicht, durch die Aktion *jButtonSort_mouseClicked()* aufgerufen und durch den OK-Button über *jButtonSortOK_mouseClicked()* an das Sortierformular und das Filterformular geschickt.

Eine Bewertung kann über den Dialog *dlgBewertung* abgegeben werden; dieser wird durch die Routine *jButtonBewertung_mouseClicked()* beim Klicken auf den Button Bewertung aufgerufen. Beim Klicken auf den OK-Button des Dialogs wird die Ereignisbehandlungsroutine *jButtonBewertungOK_mouseClicked()* ausgelöst, welche die Klasse Bewertungsformular informiert. *jButtonPasswortOk_mouseClicked()* behandelt den am Anfang angezeigten Anmeldedialog und übergibt die Werte an das Anmeldeformular. *dlg_lostFocus()* schließt die vergrößerte Darstellung der Medien beim Verlassen des Dialogs.

menuItemEnde_actionPerformed() beendet den ParLe Client. Die Routine *menuItemPasswort_actionPerformed()* gibt bei Wahl des Menüpunkts einen Dialog aus, in

dem das Passwort geändert werden kann. Diese Eingabe wird über Klick auf den OK –Button über *jButtonPasswortÄndern_mouseClicked()* dem Anmeldeformular übergeben.

Medienfenster
+ <i>gewähltesMediumLE(e:MouseEvent,alLE:ArrayList, Filter:Filter, Sortierung:Sortierung):ArrayList</i>
+ <i>stelltDarLE(alLE:ArrayList,Filter:Filter,Sortierung:Sortierung):ArrayList</i>
+ <i>stelltDarMedium(med:Medium):Medium</i>

Tabelle 24: Die Klasse **Medienfenster**

Über *gewähltesMediumLE()* aus der Eingabe von *jPanelMedien_mouseClicked()* wird an die Mediensteuerung das *MouseEvent* *e* und die *ArrayList* *alLE* übergeben. Zur Ausgabe gelangt eine *ArrayList*, die aus den nun anzuzeigenden Lehreinheiten und Medien besteht.

Die Methode *gewähltesMediumLE()* ruft die Methode *wähltAusLE()*, mit den Attributen *e* vom Typ *MouseEvent*, der *ArrayList* *alLE*, dem *Filter* und der *Sortierung* auf; zur Rückgabe gelangt die *ArrayList*, die die Grundlage für die darzustellende *ArrayList* *alLE* bildet. Die Klasse *ParLeMainFrame* zeigt die in der Liste enthaltenen Lehreinheiten und Medien der Reihenfolge nach von oben nach unten im *jPanelMedien* an.

stelltDarLE() übernimmt die neue *ArrayList* *alLE*, den *Filter* *Filter* und die *Sortierung* *Sortierung*. Zur Ausgabe kommt eine *ArrayList*, die die Grundlage für *prleKomm*, die Liste der darzustellenden Kommentare, bildet. Diese wird vom *ParLeMainFrame* übernommen und im *jPanelKommentar* von oben nach unten dargestellt. Dabei wird auch die globale Variable *prleKomm* auf die Ausgabe von *stelltDarLE()* gesetzt. Über *stelltDarMedium()* wird ein einzelnes *Medium* zur Darstellung im Dialog *dlg*, der das vergrößerte *Medium* beim Doppelklick darstellt, übergeben.

Mediensteuerung
+ <i>wähltAusLE(e:MouseEvent,alLE:ArrayList,Filter:Filter,Sortierung:Sortierung):Object[]</i>
+ <i>wähltAusMedium(e:MouseEvent,alLE:ArrayList):Medium</i>

Tabelle 25: Die Klasse **Mediensteuerung**

Die Methoden *stelltDarLE()* und *gewähltesMediumLE()* des Medienfensters rufen *wähltAusLE()* auf. Diese Methode holt sich von *gewähltesMediumLE()* oder *stelltDarLE()* die Eingaben *e*, *alLE*, *Filter* und *Sortierung*, die aus dem *MouseEvent*, der Liste der angezeigten Medien und Lehreinheiten, dem *Filter* und der *Sortierung* bestehen. Zur Ausgabe kommt ein *Object[]*, das aus der *ArrayList* der anzuzeigenden Lehreinheiten und Medien und der *ArrayList* der darzustellenden Kommentare besteht.

Bei einfachem Klick setzt die Mediensteuerung nur das Attribut *markiert* des gewählten Objekts auf *true*, die aller anderen anzuzeigenden Objekte auf *false* und erzeugt daraus die neue im Medienfenster darzustellende *ArrayList* der Lehreinheiten und Medien und die neue darzustellende *ArrayList* der Kommentare, welche im Kommentarfenster angezeigt werden.

Die *ArrayList* der im *jPanelKommentar* darzustellenden Kommentare wird über die Methoden *getKommentarFromLE()*, *getKommentarFromMedium()* und *getKommentarFromKommentar()*, unter Beachtung der *Sortierung* und dem *Filter*, aus der Datenbank erstellt. Beim Doppelklick auf eine Lehreinheit setzt die Mediensteuerung die Attribute *markiert* auf *true* und *ausgeklappt* auf den jeweils anderen booleschen Wert. Auch

hier wird das Attribut markiert der anderen Objekte der ArrayList *aLE* der im Medienfenster darzustellenden Objekte auf *false* gesetzt.

Beim Übergang vom ein- zum ausgeklappten Zustand einer Lehreinheit werden unter dieser die dazugehörigen Medien angezeigt (d.h. eingefügt), und die dahinter in der *aLE* ArrayList stehenden Objekte werden entsprechend der Zahl der auszuklappenden Medien nach hinten verschoben.

Beim Übergang vom aus- zum eingeklappten Zustand werden die zur gewählten Lehreinheit gehörenden Medien aus der ArrayList *aLE* entfernt, so dass die nächste Lehreinheit direkt hinter der gewählten liegt.

Die Methode *wähltAusMedium()*, die von *gewähltesMediumLE()* aufgerufen wird, holt sich über das Medienfenster von der ArrayList *aLE* und dem MouseEvent *e* die Eingabedaten. In dem Fall, dass es sich beim gewählten Element um ein Medium handelt, und es sich um einen mehrfachen Klick handelt, gibt die Methode ein Medium für das Medienfenster zurück, welches über *stelltDarMedium()* im Dialog *dlg* angezeigt wird.

Lehreinheit
<ul style="list-style-type: none"> - LEId: int - Name: string - AnzahlMedien: int - markiert: boolean - ausgeklappt: boolean - altMedien: int
<ul style="list-style-type: none"> + get...():Object + set...(ob:Object)

Tabelle 26: Die Klasse **Lehreinheit**

Die Klasse *Lehreinheit* beinhaltet das bereits bei der Identifizierung der Entitätsobjekte erwähnte Attribut *Name*, welches auch in der Datenbank gespeichert ist. Zur Darstellung der *Lehreinheit* in einem Paneel des *ParLeMainFrame* sind neben dem in der Datenbank gespeicherten Attribut *LEId* auch die Anzahl der Medien (*AnzahlMedien*), *markiert*, *ausgeklappt* und *altMedien* einzuführen. Die *Lehreinheiten* werden über *getLEName()* aus der Datenbank ausgelesen. Die Anzahl der Medien wird von der Mediensteuerung aus der Datenbank über *getCountMedien()* ermittelt; sie dient der Bereitstellung des Platzes für die auszuklappenden Medien in der *aLE* ArrayList, die im Medienfenster über *gewähltesMediumLE()* ausgegeben wird.

Das Attribut *markiert* wird gesetzt, um eine gewählte *Lehreinheit* bei der Darstellung zu erkennen. Das *ausgeklappt* Attribut kennzeichnet eine *Lehreinheit*, deren Medien ausgeklappt sind. Um bei Änderungen in der Datenbank nur die bei der Ausklappung dargestellten Medien aus der ArrayList darzustellender *Lehreinheiten* und Medien beim Einklappen wieder zu entfernen, wird das Attribut *altMedien* beim Ausklappen gesetzt. Zu allen hier privat gehaltenen Methoden werden Getter- und Settermethoden angeboten.

Medium
<ul style="list-style-type: none"> - MedienId: int - Medium: Image - gehoertZu: int - markiert: boolean
<ul style="list-style-type: none"> + get...():Object + set...(ob:Object)

Tabelle 27: Die Klasse **Medium**

Neben den persistenten Attributen `MedienId`, `Medium` und `gehörtZu`, die auch in der Datenbank gespeichert sind und über `getMedien()` ausgelesen werden, wird für die Darstellung im `jPanelMedien` auch noch das `markiert` Attribut benötigt; dieses identifiziert ein ausgewähltes Medium. Auch hierzu werden die Getter- und Settermethoden dieser privat gehaltenen Attribute verwendet.

Kommentarfenster
+ gewählterKommentar(prleKomm:ArrayList,e:MouseEvent):Kommentar + ausklappenKommentar(prleKomm:ArrayList,e:MouseEvent):ArrayList

Tabelle 28: Die Klasse **Kommentarfenster**

Bei der Ereignisbehandlung von `jPanelKommentar_mouseClicked()`, der Routine die beim Klick auf das `jPanelKommentar` ausgelöst wird, wird die Kommentarsteuerung über `ausklappenKommentar()` aufgerufen. Diese Methode übernimmt eine `ArrayList` der darzustellenden Kommentare und gibt sie an die Ereignisbehandlungsroutine `jPanelKommentar_mouseClicked()` weiter, die die Kommentare in Reihenfolge ihres Auftretens in der übergebenen `ArrayList` darstellt. Methode `gewählterKommentar()` ruft die Kommentarsteuerung auf, um an die Daten des gewählten Kommentars zu kommen. Die Häufigkeit des Klicks wird über das `MouseEvent e` ermittelt. Ausgegeben wird der gewählte Kommentar.

Kommentarsteuerung
+ wähltKommentar(prleKomm:ArrayList,e:MouseEvent):Kommentar + klapptAus(prleKomm:ArrayList,e:MouseEvent):ArrayList

Tabelle 29: Die Klasse **Kommentarsteuerung**

Die Methode `klapptAus()` wird von dem `Kommentarfenster` über `ausklappenKommentar()` aufgerufen. Dabei wird über `klapptAus()` die `ArrayList` ausgegeben, die die Grundlage für die Darstellung im `jPanelKommentar` bildet. Die Methode `wähltKommentar()`, bei der der aktuell angeklickte Kommentar für `gewählterKommentar()` ermittelt wird, dient der Auswahl eines gewählten Kommentars.

Bei einem einfachen Klick wird das Attribut `markiert` des gewählten Kommentars entweder auf `true` gesetzt oder, wenn er bereits markiert war, auf `false`. Bei allen anderen Kommentaren wird das Attribut `markiert` auf `false` gesetzt. Bei einem mehrfachen Klick wird der Kommentar aus- oder eingeklappt. Es werden die Attribute `markiert` und `ausgeklappt` auf `true` gesetzt. Das Attribut `markiert` der anderen Kommentare wird ebenso wieder auf `false` gesetzt. Bei der Wahl der darzustellenden Kommentare wird über die globale `ArrayList prleKomm` nach dem markierten Element, das einen Kommentar darstellt, gesucht.

Kommentar
- KommentarId: int - KommentarName: string - Kommentar: string - gehörtZuKommentar:int - gehörtZuMedium:int - gehörtZuLE:int - verfasst: int - Datum: Date - gesperrt: boolean

- eingerückt: int - beginnt: int - AnzahlZeilen: int - Bewertung: double - ausgeklappt: boolean - markiert: boolean
+ get...():Object + set...(ob:Object)

Tabelle 30: Die Klasse **Kommentar**

Die Klasse Kommentar benötigt für die Darstellung im JPanelKommentar neben den persistent in der Datenbank gespeicherten Attributen KommentarId, KommentarName, Kommentar, gehoertZuKommentar, gehoertZuMedium, gehoertZuLE, verfasst, Datum und gesperrt zur Darstellung noch die Attribute beginnt, AnzahlZeilen, eingerückt, Bewertung, ausgeklappt und markiert.

Das Attribut beginnt gibt die y-Koordinaten an, bei der der Kommentar beginnt. Das Attribut AnzahlZeilen wird über den Zuschneider über die Methode *getLineCount()* bestimmt, der der darzustellende Kommentarstring zu übergeben ist. Das Attribut eingerückt bestimmt die x-Koordinate, um den ein Kommentar nach rechts zu verschieben ist.

Die Bewertung ist der Durchschnitt der zum Kommentar gehörenden Bewertungen. Dieser wird über *getbewertet()* bei Eingabe der KommentarId des Kommentars ausgegeben. Der Parameter ausgeklappt gibt an, ob der Kommentartext und die Metadaten ausgeklappt sind. Das Attribut markiert gibt an, ob der Kommentar markiert ist. Es kann höchstens ein Kommentar markiert sein. Für alle privat gehaltenen Attribute werden Getter- und Settermethoden angeboten.

Kommentareingabeformular
+ gibtEinKommentar(KommentarName:string, Kommentar:string, alle:ArrayList, prleKomm:ArrayList, NutzerId:int):ArrayList

Tabelle 31: Die Klasse **Kommentareingabeformular**

Die Klasse Kommentareingabeformular wird von *jButtonInsertOK_mouseClicked()* aufgerufen und gibt den KommentarName und den Kommentar an die Kommentareingabesteuerung über den Aufruf *erzeugtKommentar()* weiter. Ebenso werden die ArrayLists alle und prleKomm und die NutzerId übergeben. Zur Ausgabe gelangt eine ArrayList, die die darzustellenden Kommentare enthält. Die globale ArrayList prleKomm der Klasse ParLeMainFrame wird auf den Wert dieser Ausgabe gesetzt.

Kommentareingabesteuerung
+ erzeugtKommentar(KommentarName:string, Kommentar:string, alle:ArrayList, prleKomm:ArrayList, NutzerId:int):ArrayList + zuLE(alle:ArrayList):Lehreinheit + zuMedium(alle:ArrayList):Medium + zuKommentar(prleKomm:ArrayList):Kommentar

Tabelle 32: Die Klasse **Kommentareingabesteuerung**

Die Klasse Kommentareingabesteuerung wird vom Kommentareingabeformular aufgerufen. Beim Aufruf der Methode *erzeugtKommentar()* der Eingabesteuerung wird der zu erzeugende Kommentar in Form seiner zwei Daten KommentarName und Kommentar übergeben. Die

Ermittlung der Zugehörigkeit zu einer Lehrinheit, einem Medium oder einem Oberkommentar wird über die Methoden *zuLE()*, *zuMedium()* und *zuKommentar()* erreicht. Diese werden von der Methode *erzeugtKommentar()* aufgerufen und übernehmen die Daten aus den ArrayLists *alle* und *prleKomm*. Die Methode *zuLE()* prüft, ob im *alle* eine Lehrinheit markiert ist, *zuMedium()* prüft, ob in *alle* ein Medium gewählt ist, und über *zuKommentar()* wird ermittelt, ob ein Kommentar ausgewählt ist. *erzeugtKommentar()* gibt über die Methoden *setKommentarLE()*, *setKommentarMedium()* oder *setKommentarKommentar()* des RMIClient diese Daten nun je nach Zugehörigkeit in die Datenbank ein.

Bewertungsformular
+ gibtBewertung(Bewertung:int, NutzerId:int, KommentarId:int,prleKomm:ArrayList): ArrayList

Tabelle 33: Die Klasse **Bewertungsformular**

Die Klasse *Bewertungsformular*, aufgerufen von *jButtonBewertungOK_mouseClicked()*, gibt die eingegebene Bewertung an die Bewertungssteuerung weiter und gibt als Ausgabe die ArrayList, die die Kommentare beschreibt, zurück. Dabei werden der aktuelle Nutzer und der gewählte Kommentar in Form ihrer Ids eingegeben. Diese Daten werden der Mediensteuerung über den Aufruf *erzeugtBewertung()* übergeben.

Bewertungssteuerung
+ erzeugtBewertung(NutzerId:int,KommentarId:int,Bewertung:int,prleKomm:ArrayList): ArrayList
+ zuNutzer(KommentarId:int,NutzerId:int):int
+ zuKommentar(KommentarId:NutzerId:int):int

Tabelle 34: Die Klasse **Bewertungssteuerung**

Die Klasse *Bewertungssteuerung* übernimmt von der Klasse *Bewertungsformular* die gewählte Bewertung durch den Methodenaufruf *erzeugtBewertung()*. Sie untersucht über *zuNutzer()*, ob der Kommentar von demselben Nutzer stammt wie die Bewertung. Das geschieht, indem der Autor (d.h. seine Id) mit dem aktuellen Nutzer verglichen wird. Wenn das der Fall ist, wird keine Bewertung abgegeben.

Über *zuKommentar()* wird überprüft, ob der aktuelle Nutzer zu diesem Kommentar bereits eine Bewertung abgegeben hat. Dies wird über *isBewertet()* realisiert. Sind die beiden Überprüfungen negativ, wird über *erzeugtBewertung()* die Bewertung vorgenommen. Die Bewertungssteuerung stellt die Bewertung über *setBewertung()* in die Datenbank ein.

Bewertung
- NutzerId:int
- KommentarId:int
- Bewertung:int
+ get...()
+ set...(...)

Tabelle 35: Die Klasse **Bewertung**

Die Klasse *Bewertung* besteht auch intern nur aus den persistent gespeicherten Attributen *NutzerId*, *KommentarId* und *Bewertung*. Die Bewertung wird über *erzeugtBewertung()* der Bewertungssteuerung in die Datenbank überführt. Ausgelesen wird die Bewertung von der

Kommentarsteuerung, um die Durchschnittsbewertung eines Kommentars zu ermitteln, über die Methode *getbewertet()*.

Anmeldeformular
+ gibtAnmeldung(nutzerName:string,passwort:string):Nutzer

Tabelle 36: Die Klasse **Anmeldeformular**

Das Anmeldeformular wird über *jButtonPasswortOK_mouseClicked()* aufgerufen. Die Anmeldesteuerung wird über *meldetAn()* von der Methode *gibtAnmeldung()* aufgerufen. Dabei werden die Strings Name und Passwort übergeben. Das Anmeldeformular gibt den Nutzer an die ParLeMainFrame Klasse weiter, und diese setzt die Variable Nutzer auf das Nutzerobjekt.

Anmeldesteuerung
+ meldetAn(Name:string,Passwort:string):Nutzer

Tabelle 37: Die Klasse **Anmeldesteuerung**

Die Methode *meldetAn()*, die über *gibtAnmeldung()* aufgerufen wird, erhält als Eingabe die Daten über den Nutzernamen und das Passwort, vergleicht über *idRegistriert()* der RMIClient Klasse die Variablen Name und Passwort. Bei Übereinstimmung gibt sie den ParLe Client frei und setzt über das Anmeldeformular die globale Variable Nutzer der ParLeMainFrame Klasse auf das angemeldete Nutzerobjekt. Bei falschen Daten schließt sich der ParLe Client. Die Anmeldesteuerung erzeugt ein Nutzerobjekt, welches eine Beziehung zu einem Filter und einer Sortierung hat. Das Nutzerobjekt kann zu jedem Kommentar eine Bewertung haben. Zu jedem Kommentar gehört als Autor ein Nutzer.

Nutzer
- NutzerId:int - Name:string - gesperrt:boolean - hatSortierung:int - hatFilter:int
+ get...(...):... + set...(...)

Tabelle 38: Die Klasse **Nutzer**

Die Klasse Nutzer hat die persistenten Attribute NutzerId, Name, gesperrt, hatSortierung und hatFilter. Ein Nutzer verfasst mehrere Kommentare, wobei ein Kommentar nur zu einem Nutzer gehören kann. Ebenso kann zu einem Kommentar eine Bewertung abgegeben werden. Der Nutzer hat eine Sortierung und einen Filter.

Sortierformular
+ setztSortierung(Nutzer:Nutzer,Sortierung:Sortierung,prleKomm:ArrayList,Filter:Filter):Object[]

Tabelle 39: Die Klasse **Sortierformular**

Die Methode *setztSortierung()* wird von der *ParLeMainFrame* Klasse über *jButtonSortierungOK_mouseClicked()* aufgerufen. Sie erhält als Eingabe das aktuelle Nutzerobjekt aus dem globalen Objekt *Nutzer*, die darzustellende Kommentarliste *prleKomm*, die Filterung durch *Filter* und die gewählte Sortierung. Sie ruft *erzeugtSortierung()* und *sortiereKommentare()* der *Sortiersteuerung* auf. Zur Rückgabe gelangt das *Object[]*, das aus der Sortierung und der *ArrayList* der auszugebenden Kommentare besteht. *jButtonSortierungOK_mouseClicked()* setzt die Variable *Sortierung* auf den ersten Rückgabewert des *Object[]*. Die Sortierung kann entweder nach Erstellungszeit oder nach Relevanz erfolgen.

Sortiersteuerung
+ erzeugtSortierung(Nutzer:Nutzer,Sortierung:Sortierung):Sortierung
+ sortiereKommentare(prleKomm:ArrayList,Filter:Filter,Sortierung:Sortierung):ArrayList

Tabelle 40: Die Klasse **Sortiersteuerung**

Die Klasse *Sortiersteuerung* gibt die gewünschte Sortierung zu den im *Sortierformular* eingegebenen Attributwerten *Nutzer*, *Sortierung* über *erzeugtSortierung()* an die Datenbank weiter. *erzeugtSortierung()* gibt die Sortierung für die Ausgabe im *Object[]* von *setztSortierung()* aus. Die Methode *sortiereKommentare()* sortiert die Kommentare über *getKommentarLE()*, *getKommentarMedium()* und *getKommentarKommentar()* aus der Datenbank und gibt die auszugebende *ArrayList* der darzustellenden Kommentare zurück.

Sortierung
- SortierungId:int
- Sortierung:int
+ get...(...):...
+ set...(...)

Tabelle 41: Die Klasse **Sortierung**

Die Klasse *Sortierung* repräsentiert programmintern eine Sortierung entweder als *Sortierung* nach Erstellungszeit oder nach Relevanz. Sie hat die beiden Attribute *SortierungId* und *Sortierung*. Die *Sortierung* wird bei *jButtonPasswortOK_mouseClicked()* und *jButtonSortierungOK_mouseClicked()* der *ParLeMainFrame* Klasse aus der Datenbank über *getSortierung()* des *RMIClient* ausgelesen und die Variable *Sortierung* auf diesen Wert gesetzt. Eine *Sortierung* gehört zu einem *Nutzer*.

Filterformular
+ wähltAusFilter(Nutzer:Nutzer, Sortierung:Sortierung, prleKomm:ArrayList, Filter:Filter):Object[]

Tabelle 42: Die Klasse **Filterformular**

Die Methode *wähltAusFilter()* wird von der *ParLeMainFrame* Klasse über *jButtonFilterOK_mouseClicked()* aufgerufen. Sie erhält als Eingabe das aktuelle Nutzerobjekt aus der globalen Variablen *Nutzer* vom Typ *Nutzer*, die darzustellende Kommentarliste *prleKomm*, die Filterung durch *Filter* und die gewählte Sortierung. Sie ruft *setztFilter()* und *filtertKommentare()* der *Filtersteuerung* auf. Zur Rückgabe gelangt das *Object[]*, das aus dem *Filter* und der *ArrayList* der auszugebenden Kommentare besteht. *jButtonFilterOK_mouseClicked()* setzt die Variable *Filter* auf den ersten Rückgabewert des *Object[]*. Der *Filter* kann die Werte 0 bis 5 haben.

Filtersteuerung
+ <i>setztFilter</i> (Nutzer:Nutzer,Filter:Filter)
+ <i>filtertKommentare</i> (prleKomm:ArrayList,Filter:Filter,Sortierung:Sortierung):ArrayList

Tabelle 43: Die Klasse **Filtersteuerung**

Die Klasse Filtersteuerung gibt den gewünschten Filter zu den im Filterformular eingegebenen Attributwerten Nutzer, Filter über *setztFilter()* an die Datenbank weiter. Die Methode *setztFilter()* gibt den Filter für die Ausgabe im Object[] von *wähltAusfilter()* aus. Die Methode *filtertKommentare()* holt die Kommentare, die der Filterbedingung genügen, über *getKommentarLE()*, *getKommentarMedium()* und *getKommentarKommentar()* aus der Datenbank und gibt die auszugebende ArrayList der darzustellenden Kommentare zurück.

Filter
- FilterId:int
- Filter:int
+ <i>get...(...):...</i>
+ <i>set...(...)</i>

Tabelle 44: Die Klasse **Filter**

Die Klasse Filter repräsentiert programmintern eine Filterung nach Kommentaren mit 0 bis 5 Sternen, je nach eingestelltem Filter. Sie hat die beiden Attribute FilterId und Filter. Der Filter wird bei *jButtonPasswortOK_mouseClicked()* und *jButtonFilterOK_mouseClicked()* der ParLeMainFrame Klasse aus der Datenbank über *getFilter()* des RMIClient ausgelesen und die Variable Filter auf diesen Wert gesetzt.

Umlaut
+ <i>getUmlaut</i> (in:string):string
+ <i>getFromUmlaut</i> (in:string):string

Tabelle 45: Die Klasse **Umlaut**

Die Klasse Umlaut dient der Umkodierung der deutschen Umlaute und einiger Sonderzeichen für den Datenbankzugriff über JDBC. Der JDBC Treiber unterstützt keine deutschen Sonderzeichen.

Die Methode *getUmlaut()* kodiert die entsprechenden Zeichen in Form zugelassener Zeichenfolgen. Diese wird zum Einstellen in die Datenbank aufgerufen; ihr ist dabei der umzukodierende String zu übergeben. Für die Rückkodierung ist die Methode *getFromUmlaut()* zuständig. Ein ihr übergebener String umkodierter Sonderzeichen wird in einen String mit diesen Sonderzeichen zurückkodiert.

Sortierer
+ <i>getRelevanz</i> (Eingabe:ArrayList):ArrayList

Tabelle 46: Die Klasse **Sortierer**

Vergleicher
+ <i>compare</i> (ob1:Object,ob2:Object):int

Tabelle 47: Die Unterklasse **Vergleicher**

Die Klasse *Sortierer* dient der Durchführung der Relevanzsortierung. Sie sortiert die Kommentare einer Hierarchiestufe nach ihrer Relevanz über *getRelevanz()*. Dazu hat sie eine Unterklasse *Vergleicher*, die das Interface *Comparator* für Kommentare implementiert. Diese vergleicht in der *compare()* Methode das Attribut *Bewertung* und sortiert dies absteigend.

Zuschneider
+ <i>getLineCount</i> (in:string):int + <i>getKommentarCut</i> (in:string):string

Tabelle 48: Die Klasse **Zuschneider**

Die Klasse *Zuschneider* hat zwei Aufgaben. Sie ermittelt über *getLineCount()* für die Kommentarsteuerung die Zeilenzahl wortweise umgebrochener Texte in TrueType Schriftarten. Hierzu muss der darzustellende String eingegeben werden. Über die Methode *getKommentarCut()* wird der eingegebene String wortweise umgebrochen, indem Zeilenumbruchzeichen eingefügt werden. Dieser Text kann nun wortweise umgebrochen dargestellt werden.

3.3.3 Erweiterung der Klassenschnittstellen um ein Administrationstool

Um Lehreinheiten, Medien und Nutzer in das ParLe System einbringen und verwalten zu können, ist ein Administrationstool notwendig, das zusätzlich zu den im Analysemodell ermittelten Objekten Erweiterungen in den Klassen *RMImpl*, *RMIClient* und *Nutzer* erfordert. Bei den erweiterten Klassen sind nachfolgend nur die zusätzlichen Attribute und Methoden aufgeführt und beschrieben, um die Übersichtlichkeit zu wahren.

Die Programmierung des Administrationstools gehört nicht zum Umfang der Diplomarbeit.

Die Klasse *RMImpl* muss um nachfolgende Methoden erweitert werden:

RMImpl
... + <i>setLE</i> (alLE:ArrayList,Name:string) + <i>getHöchsteLEId</i> ():int + <i>setMedium</i> (alLE:ArrayList,MedienAdresse:string) + <i>getHöchsteMedienId</i> ():int + <i>setNutzer</i> (Name:string,Passwort:string)

Tabelle 49: Erweiterung der Klasse **RMImpl**

Die Methode *setLE()* gibt eine neue Lehreinheit vor einer markierten Lehreinheit in die Datenbank ein; die nach der eingefügten Lehreinheit liegenden Lehreinheiten werden nach hinten verschoben. Dabei müssen alle mit größerer LEId zuerst hinter die höchste LEId verschoben werden und nach der Einfügung wieder um eine Position weniger zurück. Dieser Wert wird über *getHöchsteLEId()* ermittelt.

Ist keine Lehreinheit markiert, wird die einzufügende Lehreinheit an die letzte Stelle gesetzt. Über *setMedium()* wird ein Medium in die Datenbank eingefügt. Es erscheint vor dem markierten Medium in der zugehörigen Lehreinheit. Dazu muss wieder die höchste MedienId über *getHöchsteMedienId()* ermittelt werden. Die dahinter liegenden Medien werden an den Platz nach der höchsten MedienId gesetzt und das neue Objekt eingefügt. Ist kein Medium markiert, wird es an die letzte Stelle der markierten Lehreinheit eingefügt. Bei *setMedium()* muss die Adresse des Bildes gewählt werden; dieses wird in ein temporäres Verzeichnis des

Servers gelegt, um von dort in die Datenbank eingefügt zu werden. Dies liegt daran, dass PostgreSQL keine direkten BLOBs unterstützt.
Über *setNutzer()* kann ein neuer Nutzer registriert werden.

In der Klasse RMIClient müssen die gleichen Methoden eingeführt werden, die die RMIImpl Methoden aufrufen.

Nutzer
...
Administrator:boolean
+ getAdministrator():boolean
+ setAdministrator(Administor:boolean)

Tabelle 50: Erweiterung der Klasse **Nutzer**

Die Klasse Nutzer (und die Datenbank) muss um das boolesche Attribut Administrator erweitert werden.

LEeingabeformular
+ gibtLE(alLE:ArrayList,Name:string)

Tabelle 51: Die Klasse **LEeingabeformular**

Die Methode *gibtLE()* ruft die Methode *erzeugeLE()* der Klasse LEeingabesteuerung mit den Attributen alLE, der ArrayList der Lehreinheiten und Medien, und dem Namen der Lehreinheit auf.

LEeingabesteuerung
+ erzeugeLE(alLE:ArrayList,Name:string)

Tabelle 52: Die Klasse **LEeingabesteuerung**

Die LEeingabesteuerung setzt nun mithilfe von *getHöchsteLEId()* die LEIds der Lehreinheiten, die hinter der einzufügenden Stelle liegen, auf einen Wert höher als die höchste LEId, erzeugt mit *setLE()* des RMIClient die neue Lehreinheit und setzt die LEIds der dahinter liegenden Lehreinheiten wieder hinter die neu eingefügte Lehreinheit zurück.

Medieneingabeformular
+ gibtMedium(alLE:ArrayList,MedienAdresse:string)

Tabelle 53: Die Klasse **Medieneingabeformular**

Die Methode *gibtMedium()* ruft die Methode *erzeugeMedium()* der Klasse Medieneingabesteuerung mit den Attributen alLE, der ArrayList der Lehreinheiten und Medien und der Adresse des Mediums (Folie) auf.

Medieneingabesteuerung
+ erzeugeMedien(alLE:ArrayList,Name:string)

Tabelle 54: Die Klasse **Medieneingabesteuerung**

Die Medieneingabesteuerung setzt nun mithilfe von *getHöchsteMedienId()* die MedienIds der Medien, die hinter der einzufügenden Stelle liegen, auf einen Wert höher als die höchste MedienId, erzeugt mit *setMedium()* des RMIClient das neue Medium und setzt die MedienIds der nachfolgenden Medien wieder hinter das neu eingefügte Medienelement zurück.

Nutzereingabformular
+ gibtNutzer(Name:string,Passwort:string,gesperrt:boolean)

Tabelle 55: Die Klasse **Nutzereingabformular**

Die Methode *gibtNutzer()* ruft die Methode *erzeugeNutzer()* der Nutzereingabesteuerung auf. Sie hat die Attribute Name, Passwort und gesperrt.

Nutzereingabesteuerung
+ erzeugeNutzer(Name:string,Passwort:string,gesperrt:booeelan)

Tabelle 56: Die Klasse **Nutzereingabesteuerung**

Die Klasse Nutzereingabesteuerung ruft mit der Methode *erzeugeNutzer()* die Methode *setNutzer()* der RMIClient Klasse auf. Diese speichert über einen entfernten Methodenaufruf die neuen Nutzerdaten in der Datenbank ab.

4 Testen

Beim Testen wird untersucht, ob es zwischen dem erwarteten, in Modellen spezifizierten Verhalten, Unterschiede zum implementierten System gibt. Beim ParLe System wird eine vereinfachte Teststrategie verfolgt, da es sich um ein überschaubares System handelt, und die Implementierung im Rahmen dieser Diplomarbeit durch eine Person erfolgt.

Die Teststrategie sieht vor:

- Komponententests und
- Funktionstests auf Basis eines funktionsfähigen Systems

Der Komponententest soll Unterschiede finden und beheben zwischen der Spezifizierung einer Klasse und ihrer Implementierung. Komponententests sind vorgesehen für die Klassen Mediensteuerung, Kommentarsteuerung und Kommentareingabesteuerung, da es sich hierbei um komplexe Klassen handelt.

Bei Funktionstests sollen Unterschiede gefunden und behoben werden zwischen den funktionalen Anforderungen und der tatsächlichen Implementierung. Die Testfälle werden aus den Anwendungsfällen abgeleitet.

Testfälle sind für folgende Anwendungsfälle vorgesehen:

- AnmeldeNutzer
- WähleMedium
- LeseKommentare
- ErstelleKommentare
- BewerteKommentar
- FiltereKommentare
- SortiereKommentare

Die Funktionstests werden für alle Anwendungsfälle geplant, so dass ein Systemtest für funktionale Anforderungen damit indirekt erfolgt.

Systemleistungstests wurden nicht geplant, da das zu implementierende Programm den Localhost als Server verwendet.

4.1 Komponententest

Für den Komponententest wird die Äquivalenzmethode gewählt. Dabei werden Tests in Äquivalenzklassen aufgeteilt, die gleiches Verhalten zeigen. Wie oben erwähnt wird ein Testvorgehen nur für die Klassen Mediensteuerung, Kommentarsteuerung und Kommentareingabesteuerung beschrieben.

Äquivalenzklassen für die Mediensteuerung:

Für den Test der Mediensteuerung wird eine ArrayList der Lehreinheiten und Medien benutzt, die aus mehreren Lehreinheiten besteht, die ihrerseits wieder mehrere Medien besitzen. Die einzelnen Lehreinheiten und Medien haben bereits mehrere Kommentare, die unterschiedlich bewertet sind. Bei der Sortierung wird unterschieden nach Erstellungszeitsortierung und Relevanzsortierung.

Für die Mediensteuerung ergeben sich hieraus 27 Äquivalenzklassen:

- 12 Äquivalenzklassen für einfachen Mausklick auf eine Lehreinheit mit allen Filter- und Sortiermöglichkeiten. Die Lehreinheit wird markiert, die Kommentartitel werden angezeigt, dazu gibt es 6 Filter- und 2 Sortiermöglichkeiten für die Kommentare.
- 2 Äquivalenzklassen für doppelten Mausklick auf eine Lehreinheit. Medien der Lehreinheit werden ausgeklappt oder eingeklappt.
- 12 Äquivalenzklassen für einfachen Mausklick auf ein Medium mit allen Filter- und Sortiermöglichkeiten für die Kommentare.

- Eine Äquivalenzklasse für doppelten Mausklick auf ein Medium. Die Großdarstellung des gewählten Mediums wird aufgerufen.

Für den Test ist ein Testtreiber notwendig, der die verschiedenen Äquivalenzklassen austestet und ein Teststumpf, der den Zugriff auf den RMIClient simuliert.

Äquivalenzklassen für die Kommentarsteuerung:

Für die Kommentarsteuerung gibt es 4 Äquivalenzklassen:

- 2 Äquivalenzklassen für einfachen Mausklick auf einen Kommentar. Der Kommentar wird markiert oder die Markierung zurückgenommen.
- 2 Äquivalenzklassen für doppelten Mausklick auf ausgeklapptes und eingeklapptes Kommentarfeld.

Der Testtreiber muss die 4 Äquivalenzklassen zu einer von ihm übergebenen ArrayList der Kommentare abdecken. Es ist hier kein Teststumpf notwendig.

Äquivalenzklassen für die Kommentareingabesteuerung:

Die Kommentareingabesteuerung hat 4 Äquivalenzklassen:

- 2 Äquivalenzklassen für eine ausgewählte Lehreinheit oder ein Medium
- 2 Äquivalenzklassen für markierten oder nicht markierten Kommentar.

Der Testtreiber muss unterschiedliche ArrayLists der Lehreinheiten und Medien und unterschiedliche ArrayLists der Kommentare bereitstellen.

4.2 Funktionstest

Nachfolgend werden Testfälle für die oben beschriebenen Anwendungsfälle aus der Anforderungsanalyse erstellt. Damit werden die funktionalen Anforderungen überprüft. Die Tests selbst erfolgen an einem implementierten System mit allen Funktionen.

Testfall: AnmeldeNutzer:

Testfallname:	AnmeldeNutzer_AllgemeinerFall
Anfangsbedingungen:	Student startet ParLe Client.
Ereignisfluss:	<ol style="list-style-type: none"> 1.) Der Student ist nicht registriert und bestätigt mit OK. 2.) Der ParLe Client wird geschlossen. 3.) Der Student lässt sich registrieren 4.) Der Administrator registriert den Studenten 5.) Der Student startet den ParLe Client und gibt Name und/oder Passwort falsch ein und bestätigt mit OK. 6.) Der ParLe Client schließt sich. 7.) Der Student startet den ParLe Client wieder und gibt Name und Passwort richtig ein. 8.) Der ParLeMainFrame wird freigeschaltet.
Abschlussbedingungen:	Das ParLe System ist zur Nutzung bereit.

Tabelle 57: Testfall AnmeldeNutzer_AllgemeinerFall

Testfall: WähleMedium:

Testfallname:	WähleMedium_AllgemeinerFall
Anfangsbedingungen:	Der Student ist bei ParLe angemeldet.
Ereignisfluss:	1.) Der Student klickt eine angebotene Lehreinheit im

	<p>Medienfenster an oder doppelklickt eine Lehreinheit im Medienfenster an zum Ausklappen der dazugehörigen Medien.</p> <p>2.) Die gewählte Lehreinheit ist markiert und die zugehörigen Kommentartitel werden im Kommentarfenster angezeigt.</p> <p>3.) Der Student klickt ein ausgeklapptes Medium im Medienfenster an.</p> <p>4.) Das gewählte Medium ist markiert, und die zugehörigen Kommentartitel werden im Kommentarfenster angezeigt.</p> <p>5.) Der Student klappt weitere Medien aus.</p> <p>6.) Diese werden ausgeklappt.</p> <p>7.) Der Student will die Medien einer Lehreinheit wieder einklappen und doppelklickt auf die Lehreinheit.</p> <p>8.) Der ParLe Client klappt die Lehreinheit wieder ein.</p> <p>9.) Der Student kann durch Doppelklick auf ein Medium das Bild groß darstellen lassen.</p> <p>10.) Der ParLe Client stellt das Bild in einem Dialog groß dar.</p>
Abschlussbedingungen:	Die gewählten Kommentare werden angezeigt.

Tabelle 58: Testfall WähleMedien_AllgemeinerFall

Testfall: LeseKommentare:

Testfallname:	LeseKommentare_AllgemeinerFall
Anfangsbedingungen:	Der Student ist bei ParLe angemeldet.
Ereignisfluss:	<p>1.) Der Student klickt den Kommentartitel an.</p> <p>2.) Der gewählte Kommentar ist markiert.</p> <p>3.) Der Student doppelklickt auf das Paneel, das den Kommentartitel beinhaltet.</p> <p>4.) Der gewählte Kommentar ist nun markiert. Der Kommentartext und die Metadaten Autor, Bewertung, Datum werden ausgeklappt.</p> <p>5.) Der Student klappt einen weiteren Kommentar aus.</p> <p>6.) Der ParLe Client zeigt das wie gewünscht an.</p> <p>7.) Der Student will den ersten Kommentar wieder einklappen und doppelklickt auf diesen.</p> <p>8.) Der ParLe Client klappt den Kommentartext und die Metadaten wieder ein. Der gewählte Kommentar ist nun markiert.</p> <p>9.) Der Student will keinen Kommentar ausgewählt haben und klickt deswegen auf den markierten Kommentar.</p> <p>10.) Kein Kommentar ist im ParLe Client markiert.</p>
Abschlussbedingungen:	Die gewählten Kommentare werden wie gewünscht angezeigt.

Tabelle 59: Testfall LeseKommentare_AllgemeinerFall

Testfall: ErstelleKommentare:

Testfallname:	ErstelleKommentare_AllgemeinerFall
Anfangsbedingungen:	Der Student ist bei ParLe angemeldet.
Ereignisfluss:	<p>1.) Der Student ruft den Kommentareingabedialog über eine Schaltfläche auf.</p> <p>2.) Der ParLe Client bietet den Eingabedialog an.</p> <p>3.) Der Student hat noch keine Lehreinheit und kein Medium gewählt. Er betätigt den OK-Button.</p> <p>4.) Der ParLe Client bemerkt den Fehler und gibt die Meldung aus,</p>

	<p>dass keine Lehreinheit oder kein Medium gewählt ist.</p> <p>5.) Der Student wählt eine Lehreinheit oder ein Medium aus.</p> <p>6.) Der ParLe Client zeigt die Lehreinheit oder das Medium als markiert an.</p> <p>7.) Der Student ruft den Kommentareingabedialog über eine Schaltfläche auf und gibt entweder keinen Kommentarnamen und/oder keinen Kommentartext ein. Er betätigt den OK-Button.</p> <p>8.) Der ParLe Client gibt eine Meldung aus, dass kein Kommentarnamen oder Kommentartext eingegeben wurde.</p> <p>9.) Der Student ruft den Kommentareingabedialog über eine Schaltfläche auf. Er gibt den Kommentarnamen und den Kommentar richtig ein. Er betätigt den OK-Button.</p> <p>10.) ParLe stellt den Kommentar in die Datenbank ein und zeigt ihn je nach Filtereinstellung an.</p> <p>11.) Der Student will einen Unterkommentar zu einem Kommentar erstellen und wählt deshalb den Oberkommentar aus. Er ruft den Kommentareingabedialog auf und gibt Kommentarnamen und Kommentartext ein.</p> <p>12.) ParLe stellt den Kommentar zu einem Kommentar gehörend in die Datenbank ein.</p>
Abschlussbedingungen:	Es wird ein Kommentar in die Datenbank eingestellt.

Tabelle 60: Testfall ErstelleKommentar_AllgemeinerFall

Testfall: BewerteKommentar:

Testfallname:	BewerteKommentar_AllgemeinerFall
Anfangsbedingungen:	Der Student ist bei ParLe angemeldet. Es sind schon Kommentare vorhanden.
Ereignisfluss:	<p>1.) Der Student ruft den Bewertungsdialog über eine Schaltfläche auf.</p> <p>2.) Der ParLe Client bietet den Bewertungsdialog an.</p> <p>3.) Der Student wählt eine Bewertung von 1 bis 5 Sternen aus. Er betätigt den OK-Button. Kein Kommentar ist ausgewählt.</p> <p>4.) Der ParLe Client gibt eine Meldung aus, dass kein Kommentar gewählt wurde.</p> <p>5.) Der Student wählt einen Kommentar aus, den er schon bewertet hat oder der von ihm selbst stammt und wählt eine Bewertung von 1 bis 5 Sternen aus. Er betätigt den OK-Button.</p> <p>6.) Der Parle Client gibt eine Meldung aus, dass der Kommentar schon vom Nutzer bewertet ist oder von ihm selbst stammt.</p> <p>7.) Der Student wählt einen fremden Kommentar, den er noch nicht bewertet hat. Er wählt eine Bewertung von 1 bis 5 Sternen aus. Er betätigt den OK-Button.</p> <p>8.) ParLe stellt die Bewertung in die Datenbank ein. Der ParLe Client gibt den Kommentar mit seiner neuen Durchschnittsbewertung, je nach Filterbedingung, sortiert aus.</p>
Abschlussbedingungen:	Eine Bewertung zu einem Kommentar wird eingestellt.

Tabelle 61: Testfall BewerteKommentar_AllgemeinerFall

Testfall: FiltereKommentare:

Testfallname:	FiltereKommentare_AllgemeinerFall
Anfangsbedingungen:	Der Student ist bei ParLe angemeldet.
Ereignisfluss:	1.) Der Student ruft den Filterdialog über eine Schaltfläche auf. 2.) Der ParLe Client bietet den Filterdialog an. 3.) Der Student wählt einen Filter von 1 bis 5 Sternen aus. Er betätigt den OK-Button. 4.) ParLe stellt den Filter zu einem Nutzer in die Datenbank ein. Der ParLe Client gibt nun die Kommentare je nach Filterbedingung sortiert aus.
Abschlussbedingungen:	Ein Filter zu einem Nutzer wird in die Datenbank eingestellt.

Tabelle 62: Testfall FiltereKommentare_AllgemeinerFall

Testfall: SortiereKommentare:

Testfallname:	SortiereKommentare_AllgemeinerFall
Anfangsbedingungen:	Der Student ist bei ParLe angemeldet.
Ereignisfluss:	1.) Der Student ruft den Filterdialog über eine Schaltfläche auf. 2.) Der ParLe Client bietet den Filterdialog an. 3.) Der Student wählt eine Sortierung entweder nach Erstellungszeit oder nach Relevanz aus. Er betätigt den OK-Button. 4.) ParLe stellt die Sortierung zu einem Nutzer in die Datenbank ein. Der ParLe Client gibt nun die Kommentare je nach Filterbedingung sortiert aus.
Abschlussbedingungen:	Eine Sortierung zu einem Nutzer wird in die Datenbank eingestellt.

Tabelle 63: Testfall SortiereKommentare_AllgemeinerFall

5 Benutzerhandbuch

5.1 Benutzung des ParLe Systems durch den Studenten

Einrichtung des Systems auf dem Client:

Um das ParLe System auf dem Client einzurichten, muss das Nutzersystem mit Java Runtime Environment (JRE) 1.5 und Java Web Start ausgestattet sein. Java Web Start ist eine benutzerfreundliche Möglichkeit Java Programme über das Netz herunter zu laden und zu starten. Bei Änderungen am Client Programm wird das ParLe Programm automatisch heruntergeladen. Auf dem Bildschirm erscheint der ParLeMainFrame mit dem Medienfenster und dem Kommentarfenster und dem Anmeldedialog.

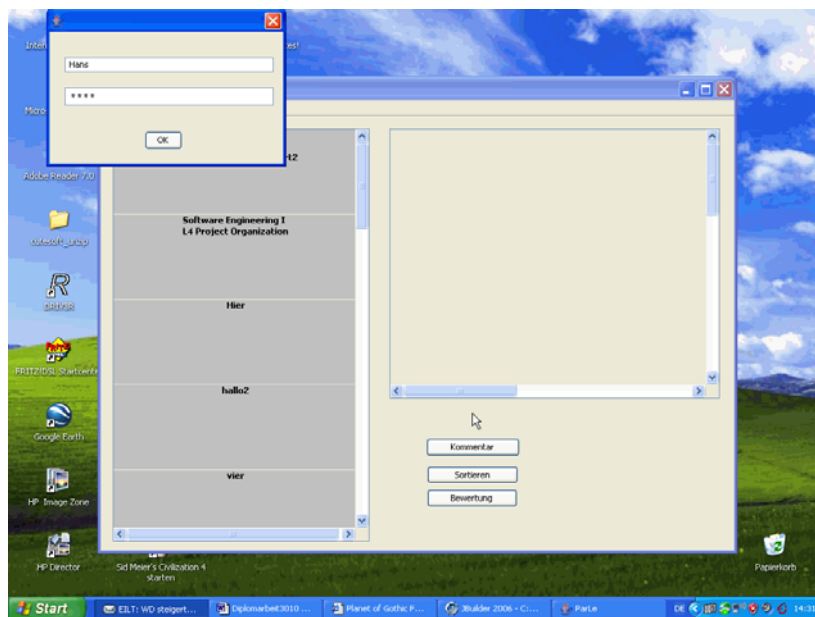


Abbildung 41: ParLeMainFrame mit Anmeldedialog

Anmeldung beim System:

Um das System nutzen zu können, muss sich der Student über den Anmeldedialog anmelden. Dazu muss er im oberen Feld den Nutzernamen und im unteren Feld das Passwort eintragen. Mit dem „OK“-Button wird die Anmeldung abgeschickt. Bei erfolgreicher Anmeldung wird der ParLeMainFrame freigeschaltet, und im Medienfenster sind alle Lehreinheiten aufgelistet. Voraussetzung für die Anmeldung ist die Registrierung des Nutzers durch den Administrator, die im Abschnitt 5.2. behandelt wird. Bei einer fehlgeschlagenen Anmeldung schließt sich der ParLe Client.

Auswahl von Lehreinheiten:

Der Student kann nun eine Lehreinheit auswählen; dies geschieht mit einem Klick mit der Maus. Die gewählte Lehreinheit ist nun blau hinterlegt dargestellt. Im Kommentarfenster auf der rechten Seite erscheinen die direkt zur Lehreinheit gehörenden Kommentare; es werden dabei nur die Kommentartitel angezeigt.

Auswahl von Medien (Folien):

Der Student kann durch Doppelklick auf eine Lehreinheit die dazugehörigen Medien (Folien) ausklappen. Die Lehreinheit wird markiert dargestellt. Die Medien werden als Medienelemente (kleine Vorschau-Bilder) dargestellt und sind auswählbar. Die Auswahl geschieht durch Anklicken eines Medienelements. Der Hintergrund des Medienobjekts wird grau hinterlegt und die zugehörigen Kommentartitel im Kommentarfenster angezeigt.

Vergrößerung der Medienelemente:

Durch Doppelklick auf ein Medienelement in der Vorschau wird dieses gewählt und groß dargestellt, um den Inhalt besser lesen zu können. Im Kommentarfenster werden die zum Medienelement gehörenden Kommentare angezeigt. Ist ein Medienelement groß dargestellt, kann mit der Schaltfläche vorwärts/rückwärts navigiert werden. Die nachfolgenden oder davor liegenden Medienelemente dieser Lehreinheit werden groß dargestellt, und im Kommentarfenster erscheinen die dazugehörigen Kommentare. Im Medienfenster ist das groß dargestellte Objekt markiert.

Lesen von Kommentaren:

Zunächst werden nur Kommentartitel angezeigt. Durch Klick auf das graue Feld mit dem Kommentartitel wird dieser markiert und die Schrift blau dargestellt. Durch Anklicken auf einen markierten Kommentar wird dieser wieder als unmarkiert dargestellt. Um den Kommentartext lesen zu können, muss auf das graue Feld mit dem Kommentartitel doppelgeklickt werden. Der Kommentartitel wird blau markiert, und die Metadaten und der Kommentartext werden nun ausgeklappt. Die Metadaten benennen den Autor, das Erstellungsdatum und die Bewertung. Darunter steht der eigentliche Kommentartext. Durch Doppelklick auf einen ausgeklappten Kommentar kann das Feld wieder eingeklappt werden, und es erscheint nur noch der Kommentartitel.

Die Kommentare sind hierarchisch dargestellt, ein Unterkommentar ist zu seinem Oberkommentar eingerückt dargestellt.

Erstellen von Kommentaren:

Der Akteur Student kann eigene neue Kommentare zu Lehreinheiten oder Medien eingeben. Hierzu muss die entsprechende Lehreinheit bzw. das entsprechende Medium gewählt (markiert) werden. Die bereits vorhandenen Kommentare zu dieser Lehreinheit oder dem Medienelement werden angezeigt.

Mit der Schaltfläche „Kommentar“ öffnet sich ein Dialog, bei dem der Kommentartitel (Kommentarname genannt) und der eigentliche Kommentartext eingegeben werden können. Beide Felder sind Pflichtfelder. Der Kommentartitel ist auf 3 Zeilen oder 200 Zeichen begrenzt. Der Kommentartext hat keine Begrenzung.

Über die Schaltfläche „Einfügen“ wird der Kommentar eingestellt.



Abbildung 42: Dialog zum **Eingeben** von **Kommentaren**

Der Akteur Student muss sich nicht um die Zeilenumbrüche kümmern, da diese vom System übernommen werden. Es wird wortweise umgebrochen.

Der Student kann Kommentare eingeben, die sich direkt auf eine Lehreinheit oder ein Medium beziehen. Dazu darf er keinen Kommentartitel markiert haben. Er kann aber auch Unterkommentare zu einem Kommentar abgeben; hierzu muss der Oberkommentar markiert sein.

Sortieren und Filtern von Kommentaren:

Durch Klick auf die Schaltfläche „Sortieren“ wird ein Dialog geöffnet, der die Relevanzsortierung ermöglicht.

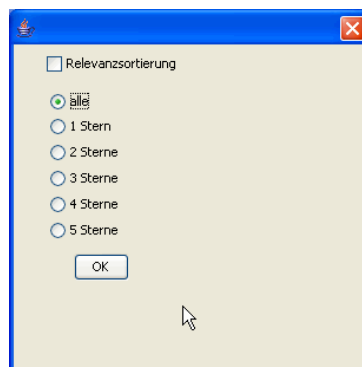


Abbildung 43: Dialog zum **Sortieren** und **Filtern** von **Kommentaren**

Durch Setzen des entsprechenden Häkchens auf die Relevanzsortierung wird eine Sortierung nach Relevanz vorgenommen. Bei Wahl der Relevanzsortierung erscheinen die Kommentare in der Reihenfolge ihrer Bewertung; die besten stehen oben. Zunächst sind die Oberkommentare nach Relevanz sortiert, danach die zu einem Oberkommentar gehörenden Unterkommentare. Ist das Häkchen nicht gesetzt, wird nach Erstellungszeit sortiert.

Zusätzlich kann ein Filter ausgewählt werden, um sich nur noch Kommentare anzeigen zu lassen, die gerundet mindestens der Filterbedingung entsprechen. Wird zum Beispiel die Filterbedingung 3 Sterne gesetzt, werden nur noch Kommentare angezeigt, die eine Bewertung von 3 oder höher haben. Hat ein Oberkommentar eine Bewertung, die außerhalb der Filterbedingung liegt, werden zugehörige Unterkommentare nicht angezeigt, auch wenn ihre Bewertung der Filterbedingung entspricht.

Bewertung:

Kommentare können vom Studenten bewertet werden; hierzu ist der Bewertungsdialog über die Schaltfläche „Bewertung“ auszuwählen. Die Bewertung bezieht sich auf den jeweils markierten Kommentar.

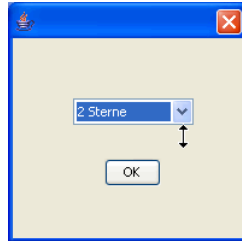


Abbildung 44: Bewertungsdialog

In einer ComboBox kann nun zwischen 1 bis 5 Sternen gewählt werden. 5 Sterne bedeuten, dass dieser Kommentar höchst zutreffend ist. Ein Stern bedeutet, dass der Kommentar irrelevant ist. Über den „OK“-Button wird diese Bewertung abgegeben. Nicht bewertete Kommentare haben die Bewertung 0. Jeder Akteur Student kann einen Kommentar nur einmal bewerten, eigene Kommentare können nicht bewertet werden. Das System überprüft dies und gibt eine Meldung aus, wenn gegen diese Regel verstoßen wird.

Ändern des Passworts:

Über den Menüpunkt „Passwort ändern“ kann der Student ein eigenes Passwort eingeben. Zu Beginn vergibt der Administrator ein Passwort. Zum Ändern gibt der Student das neue Passwort in zwei PasswordFields ein und bestätigt durch Drücken des „OK“-Buttons.

Beenden der Anwendung:

Durch den Menüpunkt „Beenden“ wird die Client-Anwendung beendet.

5.2 Datenpflege durch den Administrator

Für das ParLe Server System ist der Administrator zuständig. Der Administrator sollte gute Kenntnisse über die Verwaltung von Datenbanksystemen haben.

5.2.1 Installation des Systems

Der Administrator muss eine PostgreSQL 8.1 Datenbank auf dem Server einrichten. Dies ist zum Beispiel im Detail beschrieben im Buch PostgreSQL Professionell und Praxisnah von Jens Hartwig erschienen bei Addison-Wesley³.

Die Tabellen Lehreinheit, Medium, Nutzer, Kommentar, Bewertung, Filter und Sortierung sind wie im Anhang 1 beschrieben in den Tablespace einzufügen.

In den Methoden der Klasse RMIImpl muss bei der Connection con in der Methode *DriverManager.getConnection(...)* der Parameter auf den Pfad der Datenbank eingestellt werden. Hierbei sind noch der Username der Datenbank und das Passwort einzugeben.

Der Java Development Kit (JDK 1.5) muss installiert sein. Im Classpath muss das Verzeichnis gesetzt sein, in dem die ParLe Server Klassen liegen. Zusätzlich muss die Bibliothek des org.postgresql.Driver (postgresql8.2dev-503.jdbc.jar) eingebunden werden. Zum RMI-Betrieb ist es nötig, auf dem Server ein temporäres Verzeichnis einzurichten, in das Medien, die in BLOB-Form vorliegen, für ihre Übertragung durch RMI vorübergehend geladen werden können. Dies liegt daran, dass PostgreSQL keine direkten BLOBs unterstützt; diese können nur in ein Verzeichnis ausgegeben werden. Die Klassen müssen nun kompiliert werden. Jetzt kann die RMI Registry gestartet werden.

Die Datei java.policy muss auf folgendes geändert werden:

```
Permission java.net.SocketPermission "(hostname:PortNummer)", "listen,
accept, connect";
```

Bevor die ParLe Client Anwendung für Java Web Start eingerichtet wird, muss der Administrator in der Klasse RMIClient den serverName auf den Namen des verwendeten Servers setzen. Auf dem Server müssen die für Java Web Start benötigten Dateien bereitgestellt werden. Das Vorgehen ist unterschiedlich für die Plattformen Linux/Solaris und Windows und ist beschrieben im Installguide unter der Adresse http://java.sun.com/products/javawebstart/1.2/de/docs/installguide_de.html.

Bei dem in dieser Diplomarbeit implementierten System wurde mit dem Localhost gearbeitet, da der Server und der Client auf demselben Rechner liegen. Dies war eine Vorgabe des Instituts.

5.2.2 Einstellen der Daten in die Datenbank

Der Administrator muss die Lehreinheiten, Medien und Nutzer in die Datenbank einpflegen.

Eingabe der Daten der Lehreinheiten:

Die Lehreinheiten müssen über

```
INSERT INTO Lehreinheit (LEId,Name) values (...,...);
```

einggegeben werden. Die Reihenfolge der Darstellung wird bestimmt durch ihre Id; diese kann automatisch erstellt werden oder individuell vergeben werden (Siehe hierzu Abschnitt Lehreinheiten ändern, hinzufügen und löschen).

Eingabe der Daten der Medien:

Zunächst müssen die Medien (Folien) im jpeg Format auf ein Verzeichnis im Server gebracht werden. Dies ist nur zum Einlesen der Medien notwendig, danach kann der Inhalt des Verzeichnisses wieder gelöscht werden. Dieses Vorgehen ist nötig, da PostgreSQL BLOBs nicht direkt unterstützt. PostgreSQL stellt eine spezielle Form zur Verfügung, die es ermöglicht, eine binäre Datei in die Datenbank zu laden. Das Einstellen funktioniert über lo_import(<Dateiname>). Das zugehörige SQL-Statement lautet:

```
INSERT INTO Medium (MedienId,Medium,gehörtZu) VALUES
(1,lo_import('/5.jpg'),2);
```

Hierbei ist der absolute Pfad einzugeben. Die Reihenfolge der Medienelemente wird bestimmt durch ihre MedienId; diese kann automatisch erstellt werden oder individuell vergeben werden (Siehe hierzu Abschnitt Medienelemente ändern, hinzufügen und löschen).

Registrieren von Nutzern:

Der Administrator kann Nutzer einpflegen. Dazu gibt er den Namen des Nutzers und ein Passwort mit folgendem Statement in die Datenbank ein:

```
INSERT INTO Nutzer (NutzerId,Name,Passwort) VALUES (1,Stefan','Ste');
```

5.2.3 Verwaltung des laufenden Systems

Im laufenden Betrieb können Lehreinheiten und/oder Medien geändert, hinzugefügt oder gelöscht werden. Kommentare können gelöscht oder gesperrt werden, Nutzer können gesperrt werden.

Lehreinheiten ändern, hinzufügen und löschen:

Lehreinheiten können über folgenden Befehl geändert werden:

```
UPDATE Lehreinheit SET ...=... WHERE ...;
```

Dabei kann der Name und/oder die LEId geändert werden. Sollen Lehreinheiten hinzugefügt werden, geschieht das über den Befehl:

```
INSERT INTO Lehreinheit (LEId,Name) VALUES (...);
```

Die Lehreinheit wird hinter der letzten Lehreinheit eingefügt. Soll eine Lehreinheit zwischen zwei vorhandene Lehreinheiten eingeschoben werden, muss Platz geschaffen werden, und die dahinter liegenden Lehreinheiten müssen auf freie Plätze nach hinten verschoben werden.

Dies geschieht durch den Befehl:

```
UPDATE Lehreinheit SET LEId=LEId+... WHERE ...;
```

Die Id der dahinter liegenden Lehreinheiten wird entsprechend geändert. Beim Ändern dieser LEIds werden die dazugehörigen Medien kaskadierend geändert, auch die dazugehörigen Kommentare. Die Fremdschlüssel `gehörtZu` von `Medium` und `gehörtZuLE` des Kommentars werden dabei auf den neuen Wert gesetzt, so dass die Medien und Kommentare immer noch richtig zu ihren Lehreinheiten zugeordnet sind. Nach dem Einfügen können die verschobenen Lehreinheiten auf Plätze hinter der neuen Lehreinheit zurück verschoben werden.

Das Löschen von Lehreinheiten erfolgt über den Befehl:

```
DELETE FROM Lehreinheit WHERE ...;
```

Dies führt kaskadierend zum Löschen aller zugehörigen Medien und Kommentare.

Medien ändern, hinzufügen und löschen:

Bei einem Medium kann die `MedienId` geändert werden:

```
UPDATE Medium SET MedienId=... WHERE ...;
```

Wird die `MedienId` geändert, wird das Medium entsprechend verschoben, die zugehörigen Kommentare kaskadierend angepasst. D.h. ihr Fremdschlüsselattribut `gehörtZuMedium` wird auf den neuen Wert gesetzt.

Soll der Inhalt geändert werden (z.B. Fehlerkorrektur), muss ein neues Medium mit dem neuen Inhalt eingestellt werden und der Wert der Spalte `Medium` des neuen Objekts ermittelt werden. Dieser Wert wird nun in das zu ändernde Medienelement gesetzt, damit wird das Medium des Medienelementes durch das neue Medium ersetzt:

```
UPDATE Medien SET Medium=... WHERE MedienId=...;
```

Dann muss das neu eingeführte Medienelement wieder gelöscht werden.

Wird ein neues Medium in einer Lehreinheit hinzugefügt, erhält es eine Id, die es hinten anfügt. Soll ein Medium zwischen zwei vorhandene geschoben werden, muss Platz geschaffen werden, und die dahinter liegenden Medien müssen auf freie Plätze nach hinten verschoben werden. Dies geschieht durch den Befehl:

```
UPDATE Medium SET MedienId=MedienId+... WHERE ...;
```

Die `MedienId` der dahinter liegenden Medienelemente wird entsprechend geändert. Beim Ändern dieser `MedienIds` werden die dazugehörigen Kommentare kaskadierend geändert. Der Fremdschlüssel `gehörtZuMedium` wird dabei auf den neuen Wert gesetzt, so dass die Kommentare immer noch richtig ihren Medien zugeordnet sind. Nach dem Einfügen können die verschobenen Medien wieder auf Plätze hinter dem eingefügten Medium zurück verschoben werden.

Das Löschen von Medien erfolgt über den Befehl:

```
DELETE FROM Medium WHERE ...;
```

Dies führt kaskadierend zum Löschen aller zugehörigen Kommentare.

Löschen und Sperren von Kommentaren:

Kommentare können über den Befehl

```
DELETE FROM Kommentar WHERE ...;
```

gelöscht werden.

Der Administrator kann Kommentare sperren, indem er gesperrt auf t setzt:

```
UPDATE Kommentar SET gesperrt='t' WHERE ...;
```

Die Kommentare bleiben in der Datenbank erhalten, sie werden aber dem Akteur Student nicht mehr angezeigt.

Sperren von Nutzern:

Um einem Akteur die Berechtigung zum Arbeiten mit dem ParLe System zu entziehen, kann der Nutzer gesperrt werden:

```
UPDATE Nutzer SET gesperrt='t' WHERE ...;
```

Beenden des Systems:

Zum Beenden des Servers muss der Serverprozess explizit „gekillt“ werden.

6 Literaturverzeichnis

1. Bernd Brügge, Allen H. Dutoit: Objektorientierte Softwaretechnik, Pearson Studium, München 2004
2. A. Kemper, A. Eickler: Datenbanksysteme, Oldenbourg, München, 2001, 4. Auflage
3. Jens Hartwig: PostgreSQL, Addison-Wesley, München, 2001
4. Christian Ullenboom: Java ist auch eine Insel, Galileo Press, Bonn 2006, 5. Auflage
5. Peter Tabatt, Henry Wolf: Java Programmieren mit JBuilder, Software & Support Verlag, Frankfurt 2004, 3. Auflage
6. Danny Ayers, Hans Bergsten et. al.: Professional Java Server Programming, Wrox Press Ltd., Birmingham 1999
7. David Flanagan: Java in a Nutshell Deutsche Ausgabe für Java 1.4, O'Reilly Verlag, Köln 2003, 4. Auflage
8. Steven Holzner: Java 2 Blackbook, Coriolis Group, Scottsdale Arizona 2001
9. Bruce Eckel: Thinking in Java, Prentice Hall, Upper Saddle River NJ 2003, 3. Edition
10. Andreas Krapp, Bernd Weidenmann (Hrsg.): Pädagogische Psychologie, Beltz PVU, Weinheim, 2001, 4. Auflage
11. P. Graessle, H. Baumann, P. Baumann: UML 2.0 projektorientiert, Galileo Press, Bonn, 2004, 3. Auflage

7 Anhang

Anhang 1: SQL-Tabellen für ParLe

```
CREATE TABLE Lehreinheit (  
  LEid SERIAL NOT NULL,  
  Name VARCHAR(200),  
  CONSTRAINT Lehreinheit_pk PRIMARY KEY (LEid)  
);
```

```
CREATE TABLE Medium (  
  MedienId SERIAL NOT NULL,  
  Medium OID,  
  gehoertZu INT,  
  CONSTRAINT Medien_pk PRIMARY KEY (MedienId),  
  CONSTRAINT gehoertZu_fk  
  FOREIGN KEY (gehoertZu) REFERENCES Lehreinheit (LEid) ON UPDATE CASCADE ON DELETE  
  CASCADE  
);
```

```
CREATE TABLE Nutzer (  
  NutzerId SERIAL NOT NULL,  
  Name VARCHAR(80) NOT NULL,  
  Passwort VARCHAR(80) NOT NULL,  
  Gesperrt BOOL,  
  CONSTRAINT Nutzer_pk PRIMARY KEY (NutzerId)  
);
```

```
CREATE TABLE Kommentar (  
  KommentarId SERIAL NOT NULL,  
  KommentarName VARCHAR (200) NOT NULL,  
  Datum DATE NOT NULL,  
  Kommentar bytea,  
  verfasst INT,  
  gehoertZuLE INT,  
  gehoertZuMedium INT,  
  gehoertZuKommentar INT,  
  gesperrt BOOL,  
  CONSTRAINT Komm_pk PRIMARY KEY (KommentarId),  
  CONSTRAINT gehoertZuKommentar_fk  
  FOREIGN KEY (gehoertZuKommentar) REFERENCES Kommentar (KommentarId) ON DELETE  
  CASCADE,  
  CONSTRAINT gehoertZuLE_fk  
  FOREIGN KEY (gehoertZuLE) REFERENCES Lehreinheit (LEid) ON UPDATE CASCADE ON  
  DELETE CASCADE,  
  CONSTRAINT gehoertZuMedium_fk  
  FOREIGN KEY (gehoertZuMedium) REFERENCES Medium (MedienId) ON UPDATE CASCADE ON  
  DELETE CASCADE,  
  CONSTRAINT von_fk  
  FOREIGN KEY (verfasst) REFERENCES Nutzer (NutzerId) ON DELETE CASCADE,  
);
```

```
CREATE TABLE Bewertung (  
  BewertungId SERIAL NOT NULL,  
  hatBewertung INT,  
  bewertet INT,  
  Bewertung INT,  
  CONSTRAINT Bewertung_pk PRIMARY KEY (BewertungId),  
  CONSTRAINT Nutzer_fk  
  FOREIGN KEY (bewertet) REFERENCES Nutzer (NutzerId) ON DELETE SET NULL,
```

```
CONSTRAINT Kommentar_fk  
FOREIGN KEY (hatBewertung) REFERENCES Kommentar (KommentarId) ON DELETE CASCADE  
ON UPDATE CASCADE  
);
```

```
CREATE TABLE Filter (  
FilterId SERIAL NOT NULL,  
hatFilter INT,  
Filter INT,  
CONSTRAINT Filter_pk PRIMARY KEY (FilterId),  
CONSTRAINT Filter_fk  
FOREIGN KEY (hatFilter) REFERENCES Nutzer (NutzerId) ON DELETE CASCADE  
ON UPDATE CASCADE  
);
```

```
CREATE TABLE Sortierung (  
SortierungId SERIAL NO NULL,  
hatSortierung INT,  
Sortierung INT,  
CONSTRAINT Sortierung_pk PRIMARY KEY (SortierungId),  
CONSTRAINT Sortierung_fk  
FOREIGN KEY (hatSortierung) REFERENCES Nutzer (NutzerId) ON DELETE CASCADE ON  
UPDATE CASCADE  
);
```

Anhang 2: Glossar

Im Glossar sind nur Begriffe erläutert, die im Zusammenhang mit dem ParLe System verwendet wurden.

Akteur:

Der Akteur, bei ParLe ein Student oder Administrator, ist ein Teilnehmer am ParLe System. Dieser muss als Nutzer registriert und angemeldet sein.

Bewertung:

Die Kommentare können von Studenten bewertet werden. Die Bewertungsskala reicht von einem Stern (irrelevant) bis zu 5 Sternen (höchst zutreffend). Die Gesamtbewertung ermittelt sich aus dem Durchschnitt der abgegebenen Bewertungen. Noch nicht bewertete Kommentare haben die Relevanz von 0 Sternen.

Bewertungssystem:

Das Bewertungssystem bewertet nutzerbezogene Kommentare.

Filter:

Der Akteur Student kann eine Filterbedingung festlegen und damit nur Kommentare abrufen, die besser oder gleich der Filterbedingung sind. Die Filterbedingungen sind von 0 bis 5 Sterne. Unterkommentare von dadurch ausgeschlossenen Oberkommentaren werden nicht angezeigt, auch wenn ihre Bewertung besser ist.

Filtersystem:

Über das Filtersystem kann der Akteur Student einen Filter zur Auswahl von Kommentaren setzen.

Kommentare:

Kommentare sind der Kern des ParLe Systems. Sie führen zu einer Erweiterung, inhaltlichen Aufbereitung und Diskussion der Lehrinhalte von Vorlesungen. Sie zeigen offene Fragen auf und führen durch Antworten und Ergänzungen zu einer interaktiven Anreicherung des Inhalts der bereitgestellten Informationen. Kommentare beziehen sich auf Lehreinheiten oder Medien. Kommentare der obersten Hierarchiestufe gehören zu Lehreinheiten oder Medien, Unterkommentare sind hierarchisch gegliedert und eingerückt und stehen unter den zugehörigen Oberkommentaren. Ein Kommentar hat einen Autor, einen Kommentartitel, ein Erstellungsdatum, den eigentlichen Kommentarinhalt und kann Bewertungen haben. Kommentare werden im Kommentarfenster dargestellt.

Kommentarsystem:

Stellt die zur Lehreinheit oder zum Medium gehörenden Kommentare in gewünschter Sortierreihenfolge und entsprechend der Filterbedingung dar.

Lehreinheiten:

Der Stoff der Vorlesung (Lecture) ist in Lehreinheiten untergliedert. Jede Lehreinheit kann Medien (hier Folien) enthalten. Lehreinheiten werden vom Administrator in das ParLe System eingefügt und können von Studenten kommentiert werden.

Medien:

Zu Lehreinheiten können vom Administrator Medien (im ParLe Fall bestehend aus Folien) bereitgestellt werden, die den Lehrinhalt im Detail darstellen. Die einzelnen Medienelemente (Medium) sind dabei vom Studenten kommentierbar.

Mediensubsystem:

Das Mediensubsystem bietet dem Administrator die Möglichkeit Lehreinheiten und Medien in das ParLe System einzustellen, und gibt dem Studenten die Möglichkeit eine Lehreinheit oder ein Medium auszuwählen.

Nutzer:

Der Nutzer ist ein Entitätsobjekt, das einen Akteur (im ParLe Fall einen Studenten oder einen Administrator) repräsentiert.

Nutzersubsystem:

Der Administrator kann über das Nutzersubsystem neue Nutzer anlegen. Es übernimmt auch die Anmeldung des Nutzerobjekts.

ParLe:

ParLe (Participating Lectures) ist ein Programm für ein elektronisches Diskussionsforum, das es Studenten erlaubt, Lehreinheiten und zugehörige Medien zu kommentieren.

ParLe Client:

Der ParLe Client beinhaltet das UserInterface, das die Lehreinheiten, Medien und Kommentare anzeigt. Er holt dazu die Daten über den ParLe Server vom Datenverwaltungssystem. Er bietet darüber hinaus die Funktionen zum Einstellen neuer Kommentare, zur Bewertung der Kommentare anderer und zur Filterung und Sortierung der Kommentare nach Vorgaben. Diese Daten werden an den ParLe Server zur permanenten Speicherung übermittelt. Über den ParLe Client kann auch das Passwort geändert werden.

ParLe Server:

Der ParLe Server beinhaltet den Teil des Programms, der die Daten aus dem Datenverwaltungssystem ausliest und diese für die Übertragung zum ParLe Client zur Verfügung stellt. Er übernimmt auch die Eingaben der Kommentare und deren Bewertungen vom ParLe Client und stellt diese ins Datenverwaltungssystem ein.

Serversubsystem:

Das Serversubsystem implementiert den Zugriff auf das Datenverwaltungssystem über RMI.

Sortierung:

Kommentare können in ihrer Hierarchiestufe nach der Erstellungszeit oder Relevanz sortiert werden. Mit Hilfe einer Relevanzsortierung sollen die Kommentare in ihrer Hierarchiestufe in der Reihenfolge ihrer Bewertung angezeigt werden. Nicht bewertete Kommentare erscheinen dabei nach den bewerteten Kommentaren, weil sie die Bewertung 0 haben.

Sortiersubsystem:

Das Sortiersubsystem übernimmt die Sortierung entweder nach Erstellungszeit oder nach Relevanz. Diese Sortierung kann im Sortiersubsystem festgelegt werden.

Student:

Der Student steht sowohl für die weibliche als auch für die männliche Form von Studierenden.