

Designbeschreibung

1. Allgemeines

Die Kooperationsplattform ist ein webbasiertes Verwaltungssystem, das die Veranstaltungen und Prüfungen eines Instituts einer Universität verwaltet. Es stellt dabei entsprechende Funktionen zur Verfügung, mit denen man Prüfungen/Veranstaltungen bearbeiten (löschen, ändern, erstellen) kann. Desweiteren können sich an dieser Plattform Studenten anmelden und sich dann in die Prüfungen und Veranstaltungen einschreiben.

Die Bedienung des Programmes findet mittels eines WebBrowsers statt. Dieser Browser kommuniziert mit einem Server, der 24 Stunden erreichbar ist.

Der Server basiert auf dem Jakarta-Server und der Java-Servlet-Technologie. Wir werden den Server mittels Struts implementieren, da dies eine gute Portabilität und Erweiterungsfähigkeit gewährleistet.

Die Daten die in der Kooperationsplattform gehandhabt werden benötigen einen gewissen Sicherheitsstandard, der durch Java relativ einfach zu realisieren ist. Außerdem werden wir diese Daten in XML-Dateien speichern, was von auch wieder von Java sehr gut unterstützt wird.

Ausserdem soll noch die Möglichkeit bestehen, mittels der Kooperationsplattform sich an ein Verwaltungsprogramm für eine Veranstaltung anzumelden.

2. Produktübersicht

Um die meisten Funktionen der Kooperationsplattform zu nutzen muß sich der Student, Dozent und Administrator an der Kooperationsplattform anmelden. Dozenten und Administratoren können von einem Administrator erstellt werden, wobei ein Administrator beim Start der Kooperationsplattform vorhanden ist.

Der Student kann sich über die Plattform registrieren.

Hat sich ein Student, Dozent oder Administrator eingeloggt werden die für ihn vorgesehen Optionen angezeigt.

Der Student hat folgende Möglichkeiten:

- Registrieren
- Profil bearbeiten
- Anmelden an Prüfungen und Veranstaltungen
- Abmelden von Prüfungen und Veranstaltungen

Der Dozent hat diese Möglichkeiten:

- Erstellen von Prüfungen oder Veranstaltungen
- Bearbeiten von seinen Prüfungen oder Veranstaltungen
- Erstellen von Zulassungslisten
- Bearbeiten von Zulassungslisten
- Teilnehmerlisten anzeigen lassen
- Scheinliste für die eigenen Veranstaltungen erstellen

Der Administrator hat alle Möglichkeiten des Dozenten und folgende:

- Löschen von Prüfungen und Veranstaltungen
- Bearbeiten beliebiger Prüfungen und Vorlesungen
- Erstellen neuer Dozenten und Administratoren
- Bearbeitung aller Listen

Scheinlisten dienen zur langfristigen Speicherung der erhaltenen Scheine eines Studenten.

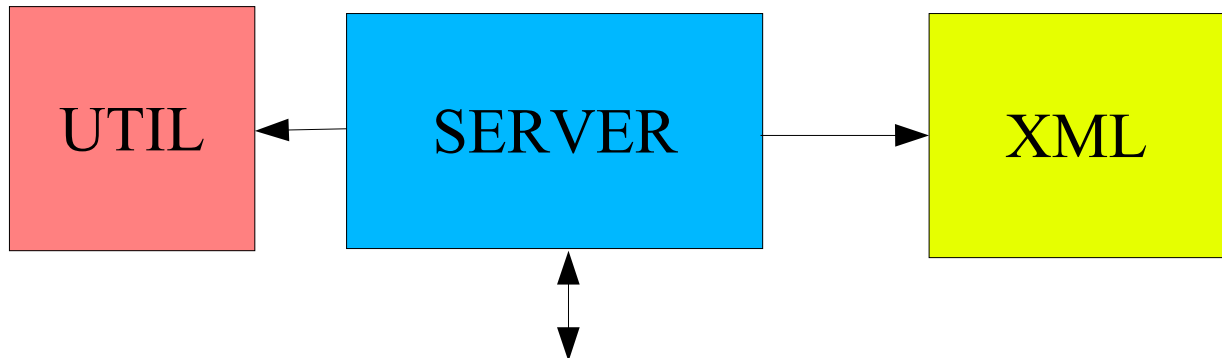
Zulassungslisten können negative oder positive Listen für eine Prüfung, Klausur oder Veranstaltung enthalten. Steht der Student in einer negativen Zulassungsliste, dann kann er sich auf keinen Fall in die Prüfung oder Veranstaltung einschreiben. Ist der Student aber in einer positiven Zulassungsliste enthalten, so kann er sich auf jeden Fall in eine Prüfung oder Veranstaltung einschreiben.

Steht der Student nicht in einer Zulassungsliste, und besteht für die Veranstaltung eine Zulassungsbedingung, so muss diese Zulassungsbedingung überprüft werden. Solche Zulassungsbedingungen sind für Prüfungen, Vorlesungen, Seminare und Praktikas einer oder mehrere bestimmte Scheine. Diese werden dann anhand der Scheinlisten für einen Studenten überprüft. Im Falle

einer Klausur bei einer Vorlesung sind die Zulassungsvoraussetzungen eine bestimmte Punkteanzahl der Übungsserien.

3. Grundsätzliche Struktur- und Entwurfsprinzipien für das Gesamtsystem

Die grundsätzliche Struktur umfasst 4 Komponenten.



Schnittstelle zu anderen Verwaltungsprogrammen

Die Serverkomponente enthält die komplette Kommunikation mit dem User sowie die Oberflächengestaltung. In unserem Fall wird der Server mittels Struts erstellt blablabla ... Konstantin (MVC und auf Struts eingehen)

Die Komponente XML enthält alle Klassen und Funktionalitäten im Bezug zur langfristigen Datenspeicherung und Datenbereitstellung für den Server.

Wir legen zur Speicherung der Daten 5 Xml-Dateien an.

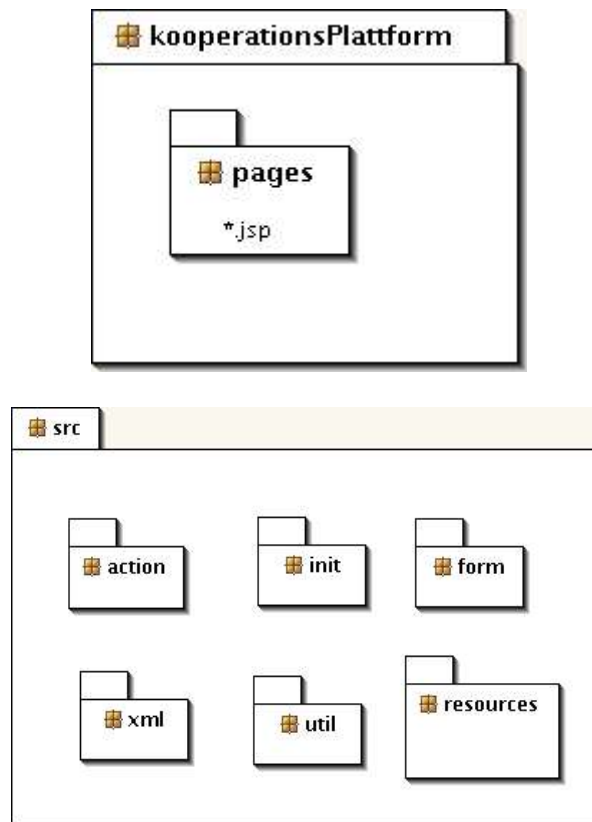
Die einzelnen Xml-Dateien speichern folgendes:

- Userdaten (Login, Password, Name, Vorname, Emailadresse)
- Veranstaltungsdaten (Name der Veranstaltung, Dozent, Veranstaltungsort und Veranstaltungszeitpunkt, Uebungsgruppen, Klausurdaten, teilnehmende Studenten, Zulassungsbedingung für Veranstaltung, Zulassungsdaten für Klausur, Übungspunkte für Studenten)
- Prüfungen (Name der Prüfung, Ort und Zeitpunkt und eventuell Zulassungsbedingungen)
- Scheinlisten (Name des Scheins und Studenten die diesen Schein erhalten haben)
- Zulassungslisten (Name der Prüfung, Veranstaltung oder Klausur für die die Zulassungsliste gilt und die betroffenen Studenten)

Die Komponente UTIL enthält Hilfsklassen, die für spezielle Aufgaben beim Server nötig sind. Ein Beispiel dafür ist die Klasse Passwordgenerator, die ein zufälliges Passwort generiert, mit dem sich der User beim ersten Login anmelden muß.

4. Grundsätzliche Struktur- und Entwurfsprinzipien der einzelnen Pakete

Das komplette Projekt hat folgende Packagestruktur:



Pages enthält die jsp-Dateien für die Oberfläche.

Das Package src enthält alle Javodateien wir für das Projekt benötigen.

Die Packages action und form enthalten die Klassen zur Behandlung der Useraktionen.

Xml enthält alle Klassen die mit den Xml-Dateien zu tun haben.

Das Package Util enthält Hilfsklassen die man für die Kooperationsplattform benötigt. Als erstes benötigt man ein Passwortgenerator bei der Registrierung eines Studenten. In dieses Package kommt auch noch die Schnittstelle zur Anbindung von Verwaltungsprogrammen für Veranstaltungen. Ein Beispiel für solch ein Programm ist der Uebmananger.

Wie schon erwähnt behandeln die Klassen im Package xml, die komplette Datenspeicherung.

Wir haben uns dafür entschieden die Daten in 5 verschiedene Dateien zu speichern.

Mittels der Klasse User werden alle Daten für die User gespeichert und bearbeitet. Das wichtigste Eigenschaft dabei ist, das jeder Login eines Users eindeutiger Schlüssel ist. Generell haben wir 4 verschiedene User.

1. Administratoren
2. Dozenten
3. Studenten
4. Studenten die ihre Anmeldung noch bestätigen müssen

Studenten die ihre Anmeldung noch bestätigen müssen, bekommen eine Email zugesandt, in der der Login und ein zufälliges Passwort enthalten ist. Der Student muss nun sich mit diesem Passwort an der Plattform anmelden. Schlägt dieses Anmelden dreimal fehl, so wird der Student für die Kooperationsplattform gesperrt und er muss sich mit dem Admin in Verbindung setzen.

Bei Prüfungen(diese werden durch die Klasse Exam behandelt) haben wir den den Namen der Prüfung als eindeutigen Schlüssel definiert. Hier können sich Studenten an eine Prüfung anmelden und abmelden. Dabei wird der Login des Studenten für die entsprechende Prüfung gespeichert.

Die Klasse Event behandelt die Vorlesungen, Seminare und Praktikas die an einer Fakultät angeboten

werden. Hierzu haben wir ein allgemeines Event definiert.

Dieses Event kann Vorlesungsveranstaltungen, Übungen und Klausuren enthalten, wenn es eine Vorlesung repräsentiert. Wenn ein Event ein Seminar ist, dann kann es Vorlesungsveranstaltungen und Klausuren enthalten. Bei Praktikas besteht die Möglichkeit Übungen und Klausuren einzurichten. Der Schlüssel für die Events ist das Name des Eventes. Namen für die Übungen, Klausuren und Vorlesungsveranstaltungen müssen global in dem Xml-Dokument nicht eindeutig sein. Sie müssen nur innerhalb eines Events eindeutig sein. Dadurch kann man jedes Event mit seinem Namen, und jede Vorlesungsveranstaltung, Übung, Klausur mit dem Eventnamen und dem Namen für die Vorlesungsveranstaltung, Übung oder Praktikum ansprechen.

Wenn Vorlesungsveranstaltungen regelmässig für ein Event stattfinden, kann man eine Vorlesung (Lecture) einrichten, wo man Wochentag, Anfang und Ende einrichten kann. Falls dies aber nicht der Fall ist, zum Beispiel bei einer Blockvorlesung, so soll man die Termine für diese Blockveranstaltung in der Beschreibung für das Event eintragen.

Übungen sind immer regelmässig, und werden wie Vorlesungen(Lectures) eingerichtet.

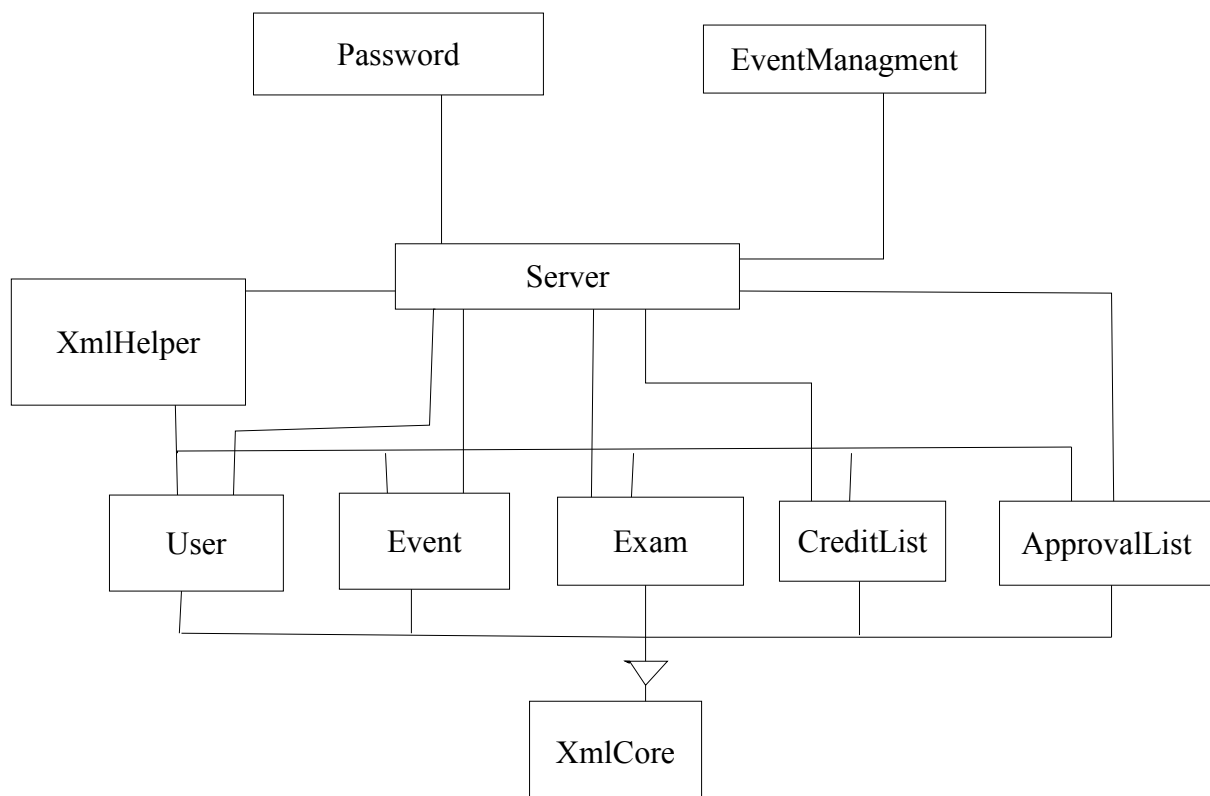
Klausuren für eine Event haben eine Termin sowie einen Anfangszeitpunkt und einen Endzeitpunkt.

Desweiteren besteht die Möglichkeit für einen Studenten der in eine Veranstaltung eingetragen ist die Punkteanzahl und die Anwesenheit zu speichern. Über diese beiden Werte kann dann die Zulassung für eine Klausur zu einer Veranstaltung überprüft werden.

Mittels der Klasse CreditList werden die Scheine eines Studenten langfristig gespeichert. Der eindeutige Schlüssel ist hier der Name des Scheines. Der Student der diesen Schein hat wird mittels seines Logins gespeichert.

Die Klasse ApprovalList speichert die positiven Zulassungslisten für Prüfungen, Veranstaltungen und Klausuren von Veranstaltungen. Hier wird wiederum der Login des Studenten in der entsprechenden Liste gespeichert. Es ist auch wichtig das ein Student nicht gleichzeitig in einer positiven und negativen Liste zur der gleichen Veranstaltung, Prüfung oder Klausur stehen darf, da dies ein Widerspruch wäre. In der XmlHelper Klasse stehen Methoden die auf verschiedene XmlKlassen zugreifen müssen. Ein Beispiel ist dafür die Überprüfung der Zulassungsvoraussetzung für eine Prüfung.

Statisches Modell:



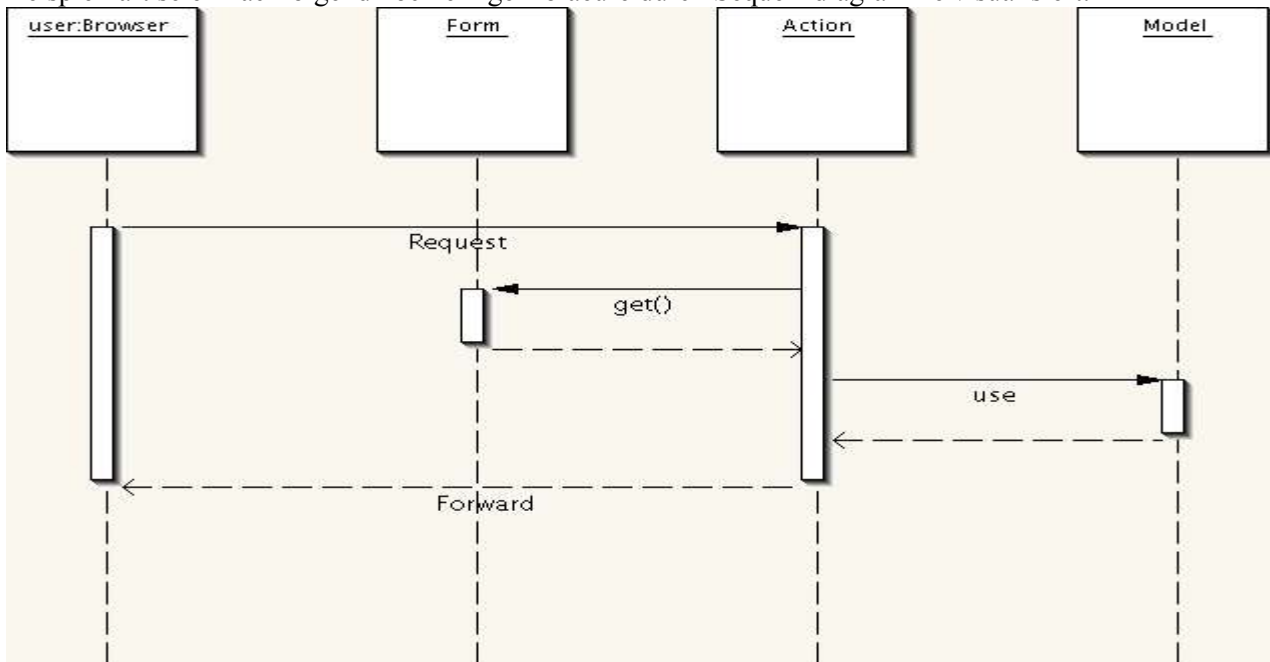
Die Klasse EventManagement dient eigentlich dazu um Verwaltungsprogramme für Veranstaltungen mit der Kooperationsplattform zu verbinden. Wenn dies geschehen ist sollten sich Studenten, mittels dieser Schnittstelle, an den entsprechenden Veranstaltungen, die mit diesem Verwaltungsprogramm gehandhabt werden, anmelden können.

Dynamisches Modell:

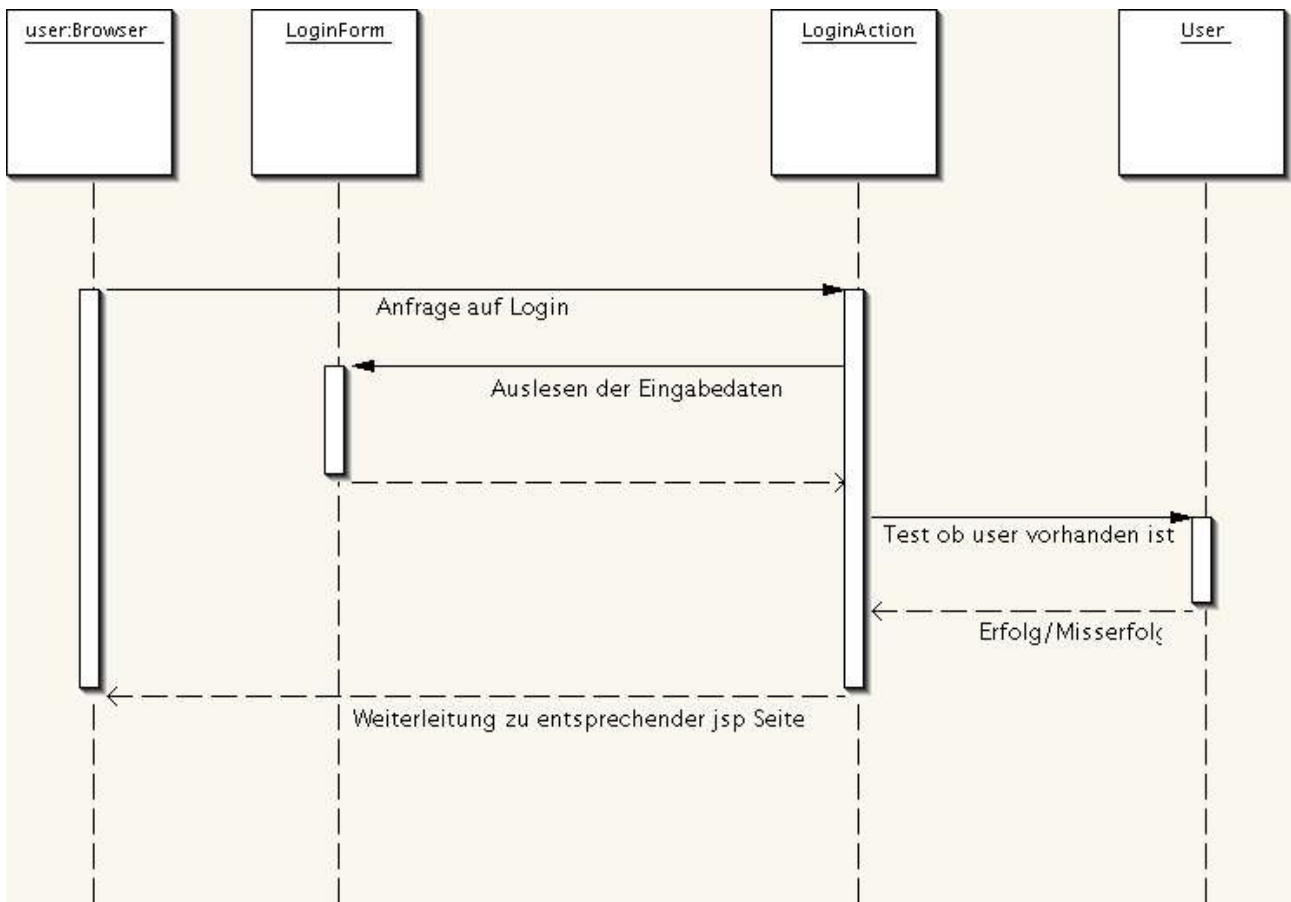
Es folgen einige Sequenzdiagramme die den prinzipiellen Ablauf der Aktionen beschreiben:

Das folgende Sequenzdiagramm zeigt den prinzipiellen Ablauf eines vom User ausgelösten Requests. Der Request wird durch die Actionklasse verarbeitet. Eventuell vorhandene (vom User eingegebene) Daten werden von der Actionklasse aus der Formklasse ausgelesen und mittels des Models verarbeitet. Das Model gibt dann eventuell Daten oder einen Wert zurück, woraufhin die Actionklasse ein Forward (response) an den User zurückschickt.

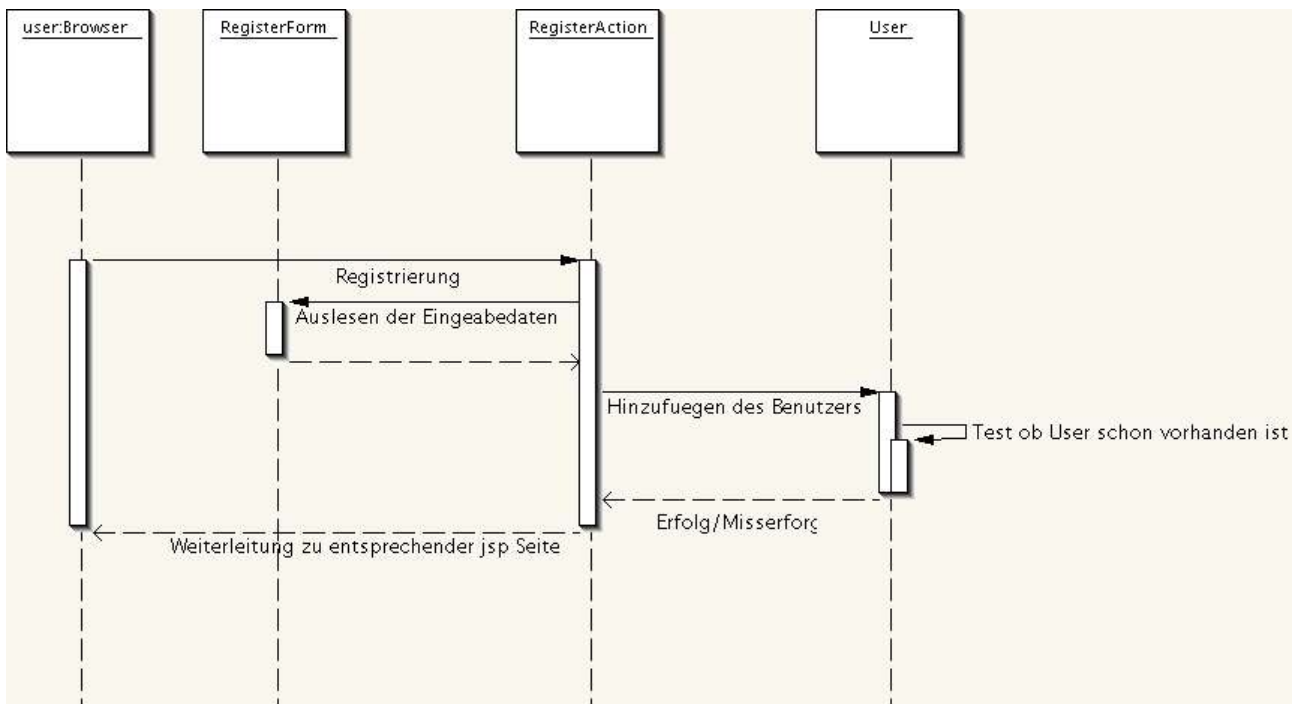
Beispielhaft seien nachfolgend noch einige Abläufe durch Sequenzdiagramme visualisiert:



Login eines Users:



Registrieren eines Studenten:



Einschreiben in eine Veranstaltung:

