Designbeschreibung für
„W.U.T.“

1. Allgemeines

Sinn und Zweck unseres Programms soll es sein, die Übungsverwaltung rechnergestützt zu organisieren. Um dies zu erleichtern, soll das Programm für jedermann rund um die Uhr erreichbar sein. Das lässt sich am besten lösen, indem man es webbasiert erstellt. D.h. mit einem Internetzugang als Grundlage und Voraussetzung soll es Ziel sein, dass jeder Benutzer von überall Zugriff auf das System hat.

Die Applikation befindet sich auf einem Server und wird dort betrieben. Diese ist dann per HTML-Client zu erreichen. Alle rechenintensiven Aufgaben werden auf dem Server verarbeitet. Unter der Verwendung von Servlets ist das sehr von Vorteil, da dadurch der Client des Benutzers entlastet wird, denn dieser lädt nur die für ihn notwendigen Daten runter.

Die Anwendung ist für genau eine Lehrveranstaltung einer Fakultät einsetzbar und das Layout der Applikation lässt sich aufgrund der HTML-Umsetzung an das des Lehrstuhls anpassen. Die Oberfläche ist klar gegliedert und dementsprechend aufgebaut. Das erleichtert die Navigation für den Benutzer. Aufwendige Darstellung wird durch einfache und übersichtliche Gestaltung ersetzt. Dadurch wird die Belastung der Internetverbindung verringert, um auch den Benutzern mit Modem eine bequeme und komfortable Bedienung zu gewährleisten.

2. Produktübersicht

Das Programm bietet die Möglichkeit sich über das Internet einzuloggen. Dabei gibt es sechs unterschiedliche Rollen, die jeweils unterschiedliche Rechte besitzen. Des weiteren ist die Rolle des Benutzers von entscheidender Bedeutung, wie das Programm mit den Benutzern umgeht und auf deren Anfragen reagiert.

Wenn man sich in der Student- oder Gastrolle einloggt, erreicht man ein HTML-Menü, das Informationen, Aufgabenstellungen und Uploadmöglichkeiten für Lösungen der aktuellen Serie beinhaltet sowie den persönlichen Punktestand (nicht bei Gast) und Angaben über das eigene Profil.

Für Mitarbeiter und Dozenten gilt Ähnliches. Auch diese erreichen ein Menü, welches speziell für ihren Anwendungsbereich angepasst ist. D.h. Webformulare zur Seminarerstellung (beinhaltet Übungsgruppen, deren SHK und Klausuren), Übungsserienerstellung (beinhaltet zugehörige Termine, Uploadmöglichkeiten der Aufgabenstellung und Punktzahl) und Einsichtmöglichkeiten in die Klarlisten der Punktstände von allen Seminarteilnehmern.

Studentische Hilfskräfte verfügen über die Möglichkeit die Lösungen der Studenten, für die nur sie zuständig sind, runter zu laden und die Korrekturergebnisse mit Hilfe eines Webformulars zu übertragen.

Der Administrator erhält die Einsicht in alle Daten, die in den XML-Dateien gespeichert sind. Er kann diese ändern und löschen. Dies geschieht entweder über Webformular oder direkten Upload von XML-Dateien.

3. Grundsätzliche Design-Entscheidungen

Für unser Opensource Projekt benutzen wir die Servletarchitektur. Die Servlets bieten den Vorteil, dass der Anwender sie nicht vom Server herunter laden muss wie es beispielsweise bei einem Applet der Fall ist. Sie werden also direkt vom Server aus gestartet.
Außerdem erleichtern Servlets die Erstellung von Benutzeroberflächen, da sie auf HTML basieren.

In unserem Programm ist die Datenbankverwaltung die zentrale Schnittstelle, da sämtliche Daten aus der XML-Datei ausgelesen und geschrieben werden. Die Benutzer erhalten eine persönliche ID, um sie später in der XML-Datei eindeutig identifizieren zu können. Zudem bekommen sie Name, Vorname, Rolle, Emailadresse und natürlich ihren Login.
Funktionen wie Login() und Logout() dürfen nicht fehlen, da sich jeder User, sei es der Administrator oder der Student, in das Programm ein- bzw. ausloggen können muss. Diese Funktion sendet die Daten des Benutzers an die Datenbankverwaltung und sucht in der XML-Datei, ob der Login vorhanden ist. Falls ja, dann liefert sie ein Okay zurück.
Jede Rolle wird von der Klasse Benutzer vererbt, somit erhalten sie neben den Login- und Logoutfunktionen auch die anderen benötigten Attribute.

//Student
Der Student kann sich in eine jeweilige Übungsgruppe eintragen. Dabei liest er alle in der XML-Datei existenten Uebungsgruppen über Datenbankverwaltung ein, um eine Übersicht über diese zu bekommen und sich für eine Ausgewählte einzuschreiben. Dies wird dann über Datenbankverwaltung in der XML-Datei abgespeichert. Ähnlich funktioniert die Verbindung zwischen dem Studenten und der Klausur. Jedoch kann sich der Student nur einschreiben, wenn er die erforderlichen Punkte besitzt.
Der Student besitzt die Möglichkeit, seine Lösung der aktuellen Serie hoch zu laden. Dies geschieht, in dem die Dateien in das Verzeichnis des für ihn zuständigen SHK verschoben werden. Dabei wird eine Bestätigung über Datenbankverwaltung in der Datenbank abgelegt.

//Mitarbeiter/Dozent
Der Mitarbeiter bzw. Dozent kann sich aus der XML-Datei Klarlisten sowie auch Aushänge erstellen. Dabei stellt er eine Anfrage an die XML Verwaltung. Diese unterscheidet zwischen Anfrage auf Klarlisten oder Aushang. Die XML Verwaltung parst die erforderlichen Daten in eine Liste und gibt sie den Mitarbeiter bzw. den Dozenten zurück.

//SHK
Der SHK bekommt eine Linkliste der hoch geladenen Lösungsdateien der Studenten ausgegeben. Es wird geprüft, ob eine Bestätigung für das Hochladen in der Datenbank liegt, um nicht existente Dateien herauszufiltern und den Dateinamen zu erhalten. In einer weiteren Funktion werden die Korrekturergebnisse, die durch die Schnittstelle eines Webformulars kommen, den einzelnen Studenten zugeordnet und über Datenbankverwaltung in den jeweiligen in der XML-Datei abgelegten Profilen verzeichnet.

//Gast
Der Gast hat keine spezifischen Funktionen.

//Admin

Der Administrator (Admin) besitzt eine Assoziation zur Datenbankverwaltung, damit er dort Einträge direkt lesen, ändern und löschen kann. Er ist als einziger befähigt, eine Sicherungskopie der Datenbestände zu erzeugen.

//Uebungsgruppe

Diese Klasse kann nur durch Mitarbeiter/Dozent über Datenbankverwaltung erstellt und verwaltet werden.

//Uebungsserie

Diese Klasse kann nur durch Mitarbeiter/Dozent über Datenbankverwaltung erstellt und verwaltet werden. Dabei wird über Datentransfer die aktuelle Aufgabenstellung als ps- oder pdf-Datei hoch geladen.

//Datentransfer

Diese Klasse bekommt die Quell- bzw. Zieldatei und verwaltet den Down- bzw. Upload.

//Aushang

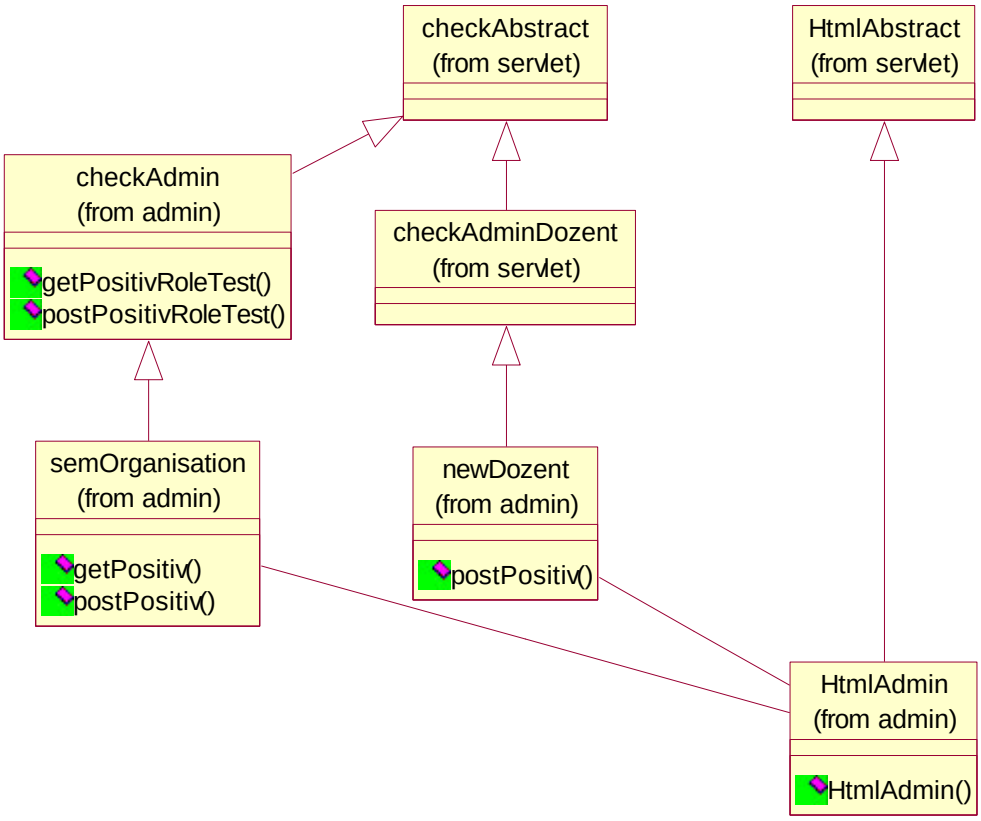
Der Aushang zieht sich alle Informationen über Studenten und ihre Punktestände aus Datenbankverwaltung. Daraus wird dann der HTML-Quellcode zur Ausgabe generiert.

//Klarlisten

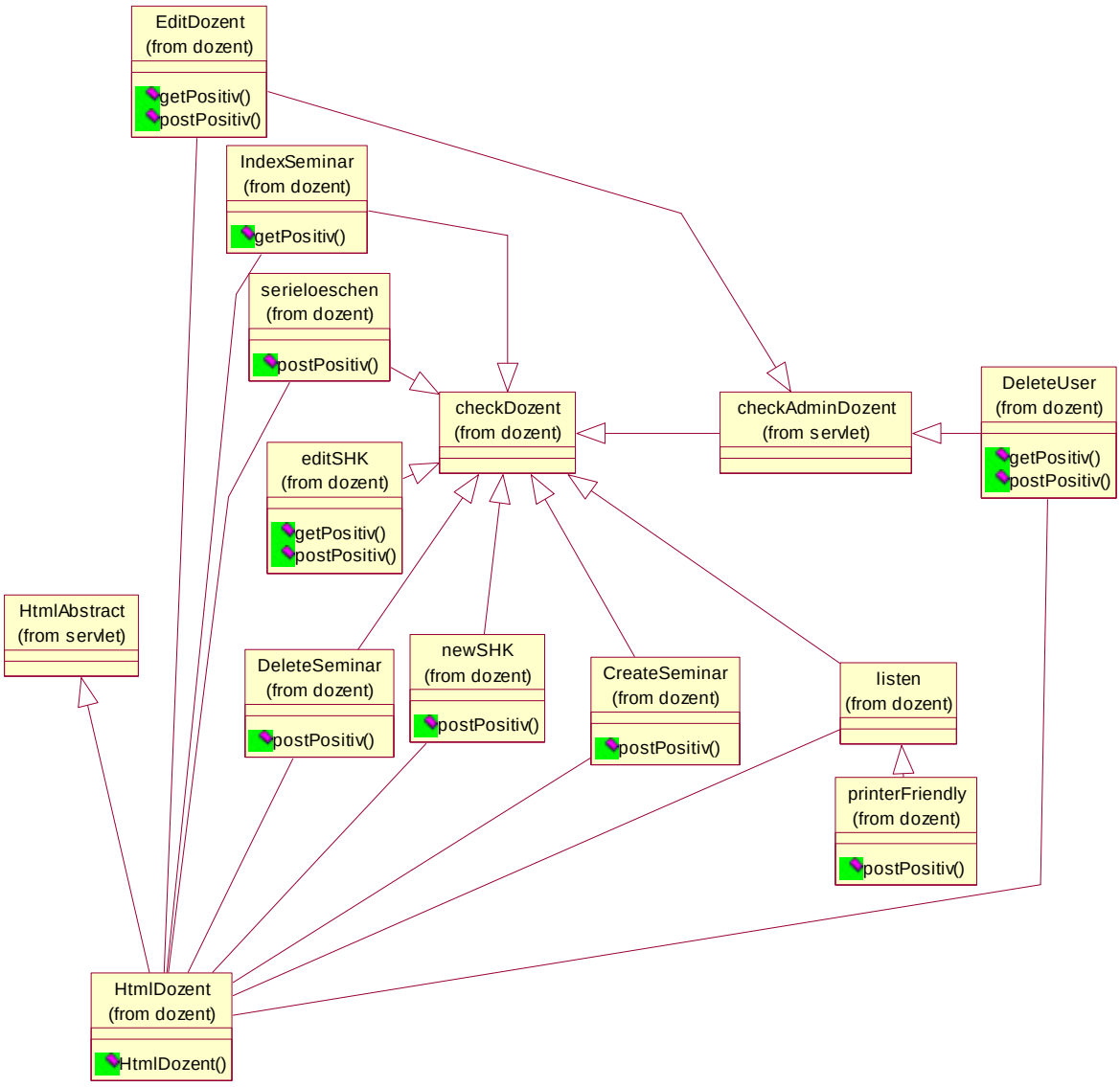
Siehe Aushang. Zusätzlich zeigt Klarlisten Name und Vorname an.

4. Paket- und Klassenstruktur

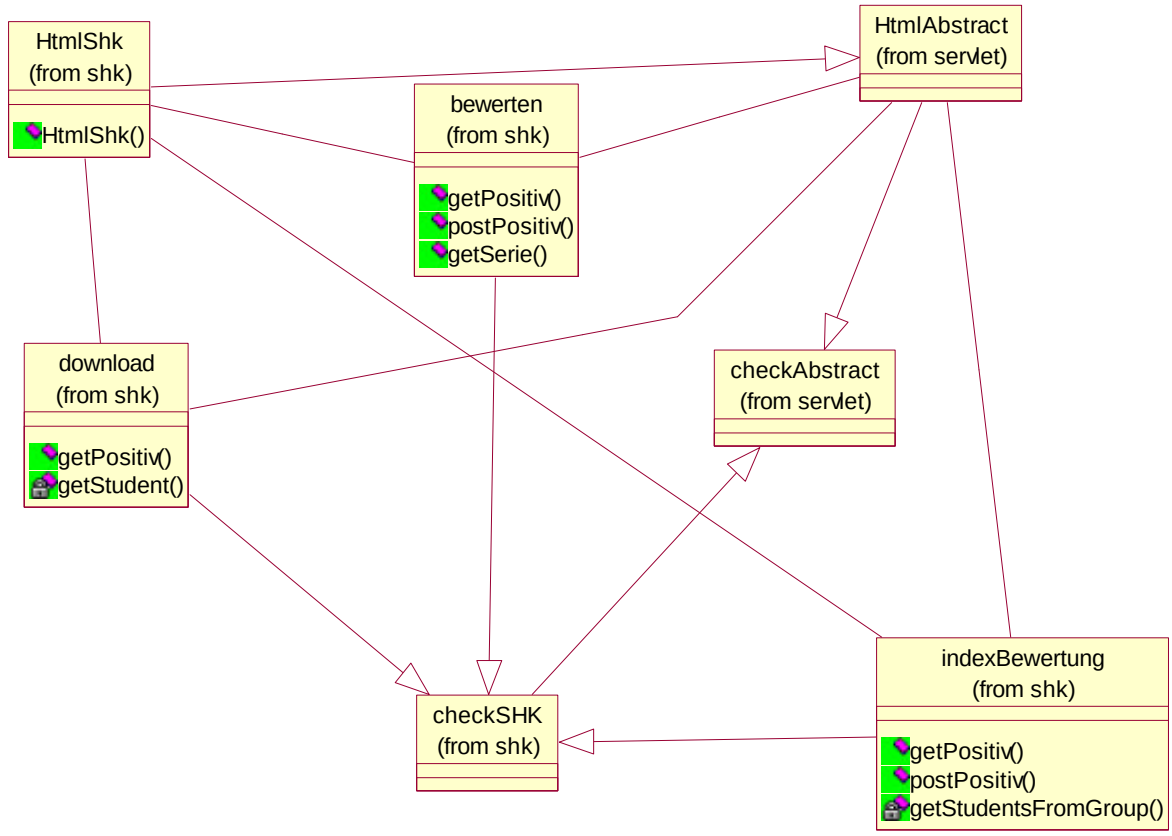
Statisches Modell Admin:



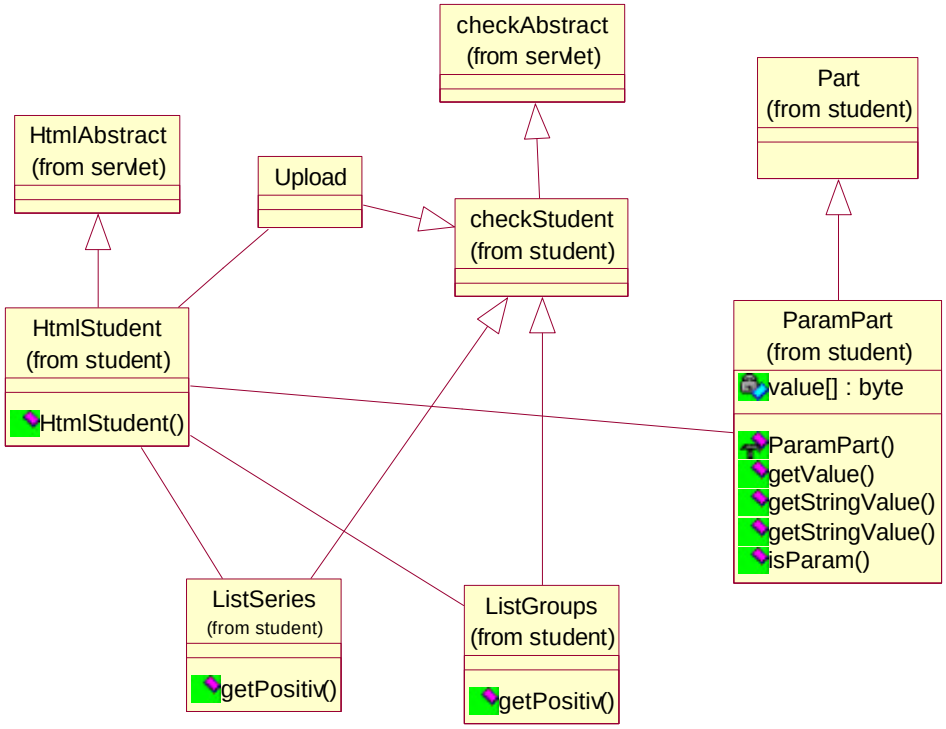
Statisches Modell Dozent:



Statisches Modell SHK:



Statisches Modell Student:



Dynamisches Modell:

/LF010/
Szenario 1: Admin loggt sich das erste mal ein (legt neuen Benutzer an)
Bedingung: - erster Anmeldeversuch am System von einer Person mit Benutzername „admin“ und Passwort „admin“
Ergebnis : - Admin wird aufgefordert sein Passwort zu ändern
- (siehe Szenario 2 und 3)
Szenario 2: Admin loggt sich ein und legt einen neuen Nutzer mit falschen Daten an
Bedingung: - erfolgreiches Einloggen des Admins
- Eingabe von Daten die nicht möglich sind, z.B. E-Mail ohne „@“ oder anlegen eines Nutzers der bereits existiert.
Ergebnis : - Ausgeben einer Warnmeldung und markieren des Fehlers
- Auffordern den Fehler zu berichtigen oder den Dialog abzubrechen
Szenario 3: Admin loggt sich ein und legt einen neuen Nutzer an
Bedingung: - erfolgreiches Einloggen des Admins
- richtige Eingabe des neues Nutzers
Ergebnis : - eintragen der Daten in die XML-Datei
- Ausgeben einer Meldung des erfolgreichen Anmeldens
- Verschicken einer Mail mit den Daten an den eingetragenen Nutzer

/LF020/
Szenario 1: Termine zweier Seminare bei ein- und demselben Übungsgruppenleiter überschneiden sich
Bedingung: - Uhrzeit, Wochentag und Wochenrhythmus der anzulegenden Übungsveranstaltung überschneidet sich mit einer anderen
- der Übungsgruppenleiter beider Veranstaltungen ist ein- und derselbe
Ergebnis : - das Seminar wird nicht angelegt
- Anzeigen einer Warnung, dass das Seminar zu diesem Termin nicht angelegt werden kann
Szenario 2: Der Übungsgruppe wird ein SHK zugeteilt, der gar nicht existiert
Bedingung: - es wird ein SHK angegeben, der in der Benutzerverwaltung gar nicht erfasst ist
Ergebnis : - eine Meldung wird angezeigt, dass die genannte SHK nicht bekannt ist
- das Seminar wird nicht angelegt
Szenario 3: Korrektes Anlegen des Seminars
Bedingung: - die eingegebenen Daten sind korrekt und lösen keine Konflikte aus
Ergebnis : - die Übungsveranstaltung wird angelegt und die Daten werden in die XML-Datei geschrieben

/LF030/
Szenario 1: Anmeldung von bereits bekanntem Nutzer
Bedingung: - Name + Vorname + E-Mail sind bereits in dieser Zusammenstellung bekannt
Ergebnis : - Ablehnen der Anmeldung und Ausgabe einer Fehlermeldung
Szenario 2: Anmelden in einer, in der Zwischenzeit, voll gewordenen Gruppe
Bedingung: - die Gruppe in die sich der Nutzer anmelden will, hat in der Zwischenzeit das Maximum der möglichen Nutzer ereicht
Ergebnis : - Ablehnen der Anmeldung und Ausgabe einer Fehlermeldung,

sowie das Aufzeigen der Fehlerstelle mit Möglichkeit der Berichtigung oder des kompletten Abbruchs
Szenario 3: Anmelden ohne Zwischenfall
Bedingung: - Nutzer ist noch nicht angemeldet und Gruppe noch nicht voll
Ergebnis : - Daten werden in die XML-Datei geschrieben
- Ausgabe einer Nachricht der erfolgreichen Anmeldung und Weiterleitung ins System
- Benutzer bekommt Mail mit seinen Daten

/LF040/
Szenario 1: Benutzer existiert nicht
Bedingung: - der eingegebene Username ist dem System nicht bekannt
Ergebnis : - es wird eine Meldung ausgegeben, das der Benutzer nicht existiert mit dem Hinweis, dass man sich erst registrieren (Neuanmelden) muss
- es erfolgt kein Zugang zum System
Szenario 2: Falsches Passwort
Bedingung: - das eingegebene Passwort stimmt nicht mit dem zugehörigen Passwort in der XML-Datei überein
Ergebnis : - eine Fehlermeldung, die einen fehlgeschlagenen Loginversuch vermeldet, erscheint
- es geschieht keine Anmeldung am System
Szenario 3: erfolgreicher Loginvorgang
Bedingung: - Username und zugehöriges Passwort stimmen mit den in der XML-Datei gespeicherten Daten überein
Ergebnis : - der Benutzer gelangt in den internen, auf seine Rolle zugeschnittenen Bereich des Systems

/LF050/
Szenario 1: Passwort != Kontrollpasswort
Bedingung: - Password-Feld1 ungleich Password-Feld2
Ergebnis : - Warnmeldung das die beiden Passwörter nicht übereinstimmen
- Passwort nicht in der XML-Datei ändern
- Formular erneut aufrufen
Szenario 2: neue E-Mail Adresse entspricht nicht der Norm
Bedingung: - „@“ fehlt
Ergebnis : - Warnmeldung, das die E-Mail nicht korrekt sein kann, und markieren des Feldes im Formular
- E-Mail nicht in der XML-Datei ändern
Szenario 3: beinhaltet Szenario 1 und 2, mit beiden Bedingungen und Ergebnissen
Szenario 4: Eingegebene Daten sind OK
Bedingung: - Gegenteil von 1 und 2
Ergebnis : - Ändern der Daten in der XML-Datei
- Bildschirmmitteilung über die erfolgreiche Änderung mit Ausgabe der neuen Nutzerdaten
Szenario 5: Neue Übungsgruppe ist inzwischen voll
Bedingung: - In der Zeit zwischen Aufrufen des Formulars und Bestätigung ist die Gruppe voll geworden
Ergebnis : - Warnmeldung, dass die ausgewählte Gruppe mittlerweile schon voll ist und die Bitte, eine andere Gruppe zu suchen

/LF060/
Szenario 1: erfolgreiches Ausloggen
Bedingung: - Benutzer klickt auf den Button/Link zum ausloggen
Ergebnis : - User wird vom System abgemeldet

- Hinweis für erfolgreiches Logout wird angezeigt

/LF070/
Szenario 1: Upload einer nicht-PDF-Datei
Bedingung: - Datei hat nicht die Endung „.pdf“
Ergebnis : - verwerfen der Datei
- Meldung Ausgeben, das die Datei nicht das richtige Format hat und der Hinweis, dass nur PDF-Dateien zulässig sind
Szenario 2: Dozent oder Mitarbeiter laden Aufgabenstellung hoch
Bedingung: - gültige Datei nach Szenario 1
- Angabe der Mindestpunktzahl
- Angabe des Datums
Ergebnis : - Speichern der Datei auf dem Server
- eintragen der notwendigen Daten in die XML-Datei
- Ausgabe einer Bildschirmnachricht über das erfolgreiche Anlegen einer neuen Übungsserie und den dazugehörigen Daten
Szenario 3: Szenario 2 + falsches Datum
Bedingung: - Eingegebenes Datum ist vor dem aktuellen
Ergebnis : - Speichern der Datei in einen TEMP-Ordner
- Warnmeldung, dass das Datum nicht richtig sein kann und die Bitte, dieses zu ändern oder den Dialog komplett zu beenden;
beim Beenden Datei aus dem TEMP-Ordner löschen, bei Berichtigung verschieben in den Ordner für die Aufgaben und eintragen der notwendigen Daten in die XML-Datei

/LF080/
Szenario 1: Eintragen von Ergebnissen einer Übungsserie, deren Bearbeitungszeit noch nicht abgelaufen ist
Bedingung: - die SHK versucht Ergebnisse für eine nicht abgeschlossene Übungsserie einzutragen
Ergebnis : - es wird eine Fehlermeldung angezeigt, dass diese Übungsserie noch nicht korrigiert werden kann
- die eingegebenen Daten werden verworfen und nicht in die XML-Datei geschrieben
Szenario 2: Eintragen von Ergebnissen für einen Student, der keine Übungsserie eingereicht hat
Bedingung: - die SHK versucht Ergebnisse für eine abgeschlossene Serie einzutragen
- der Student, dem die Ergebnisse zugeschrieben werden sollen, hat keine Lösung für diese Serie hochgeladen
Ergebnis : - der Vorgang wird abgebrochen, ohne Daten in die XML-Datei zu schreiben
- eine Fehlermeldung mit dem Hinweis, dass der betreffende Student diese Übungsserie gar nicht eingereicht hat, wird ausgegeben
Szenario 3: Fehlerfreier Korrekturvorgang
Bedingung: - die Bearbeitungszeit der Übungsserie ist abgelaufen
- der betroffene Student hat die Übungsserie gelöst
- die SHK hat die Korrekturergebnisse eingetragen
Ergebnis : - die Daten werden in die XML-Datei geschrieben und die Lösung wird als korrigiert vermerkt
- Hinweis über die erfolgreiche Eintragung der Korrekturergebnisse

/LF090/
Szenario 1: Keine Ergebnisse vorhanden
Bedingung: - noch keine relevanten Informationen in der XML-Datei
Ergebnis : - anzeigen einer Seite mit der Nachricht das noch keine Daten vorliegen
Szenario 2: Ergebnisse vorhanden
Bedingung: - relevante Einträge in der XML-Datei
Ergebnis : - Berechnen der gesamten Punktzahl
- Warnmeldung bei weniger als der minimal notwendigen Punktzahl um den Kurs zu bestehen
- Anzeigen der Punkte und Bemerkungen der SHK’s zu den einzelnen Übungsserien

/LF100/
Szenario 1: Student will sich trotz nicht erreichter Punktzahl für die Klausur anmelden
Bedingung: - die vom Dozent/Mitarbeiter festgelegte geforderte Punktzahl ist höher als die Summe der erreichten Punkte der Übungsserien
Ergebnis : - es wird eine Meldung ausgegeben, die den Studenten darauf hinweist, dass die Gesamtsumme der in den Übungsaufgaben erreichten Punkte nicht genügt, um an der Klausur teilzunehmen
- die Anmeldung bleibt erfolglos und der Student wird nicht zur Klausur angemeldet
Szenario 2: Student erfüllt alle Bedingungen, um zur Klausur zugelassen zu werden
Bedingung: - die Summe der erreichten Punkte in den Übungsserien ist mindestens genauso hoch wie die vom Dozent/Mitarbeiter festgelegte geforderte Punktzahl
Ergebnis : - es wird eine Meldung über die erfolgreiche Anmeldung zur Klausur ausgegeben
- es wird ein Vermerk in die XML-Datei geschrieben, dass der Student zur Klausur angemeldet ist

/LF110/
Szenario 1: Keine Ergebnisse vorhanden
Bedingung: - keine Studenten sind zur Übung angemeldet
- oder es liegen noch keine korrigierten Lösungen vor
Ergebnis : - anzeigen einer Seite mit der Nachricht das noch keine Daten vorliegen
Szenario 2: Ergebnisse vorhanden
Bedingung: - mindestens ein Student ist zur Übung angemeldet
- oder mindestens eine korrigierte Lösung liegt vor
Ergebnis : - erzeugen einer Textdatei, die im Browser angezeigt wird und heruntergeladen oder ausgedruckt werden kann
/LF120/
[analog zu /LF110/]

5 :: Testkonzept
Jede PP muss für ihre implementierten Programmteile mehrere Testdatensätze bereitstellen, die möglichst alle Eventualitäten abdecken. Des Weiteren muss genau dokumentiert werden, was mit der XMLDatei
passieren soll. Dabei ist zu jedem Testdatensatz die daraus resultierende XML-Datei in Textform anzugeben, um sie dann mit der

tatsächlichen XML-Datei vergleichen zu können und so Rückschlüsse über die Korrektheit des Programms zu bekommen. Durch das Programm darf auf keinen Fall ein Fehler in die XML-Datei geschrieben werden, da das gesamte Programm darauf aufbaut.

6 :: Dokumentationsanweisung:
Jede PP ist verpflichtet sich an die JavaDoc Konvention zu halten. Im JavaDoc werden alle programmiertechnisch wichtigen Dinge dokumentiert, wie den Autor der Klasse, welche Methode sie enthält usw.. Die Dokumentation der Methoden muss mindestens die Angabe der geforderten Parameter und Rückgabewerte enthalten.
Die Textdokumentation umfasst hingegen weniger technische Details und erläutert eher des Zusammenspiel der einzelnen Klassen und warum man für die geforderte Aufgabe diese Klasse implementiert hat.