Gruppe: ueb 11
Bearbeitet von: Nicky Fritsch, Patrick Wabnitz, Christian Beyerl, Andrea Münsz
zuletzt bearbeitet: 09.06.2003

1. Designbeschreibung Übungsverwaltungsprogramm

1.1.Allgemeines:

Die Software soll im universitären Umfeld eingesetzt werden und die Verwaltung und Organisation des Übungsbetriebs wesentlich erleichtern.
Dazu muss sie für jeden Übungsteilnehmer und die angestellten Korrektoren 24 Stunden am Tag erreichbar sein. Natürlich drängt sich da die Implementierung als Webapplikation auf, da sie genau diese Voraussetzungen erfüllen kann.
Der Zugang zum Internet und ein gängiger Browser sind die einzigen Voraussetzungen für die Studenten um diese Applikation nutzen zu können. Diese Möglichkeiten haben die meisten Studenten sogar von zu Hause oder mindestens in der Universität.
Nun sollte der Clientrechner des Studenten nicht zu sehr belastet und auch kein Breitbandinternetanschluß vorausgesetzt werden.
Die Verwendung von Servlets trägt allen diesen Voraussetzungen Rechung, da die rechenintensiven Aufgaben auf dem Server durchgeführt werden und nicht den Clientrechner belasten. Außerdem ist lädt der Benutzer nur die für ihn nötigen Daten in Form von Ergebnissen oder Übungsaufgaben herunter und keine Applets oder anderen Daten, die für das Funktionieren der Applikation nötig sind. Somit wird auch die Belastung der Internetverbindung minimiert.

Nun soll die Software in den verschiedensten Fächern und Fakultäten der unterschiedlichsten Einrichtungen genutzt werden, und muss damit auch die Möglichkeit bieten die Benutzeroberfläche und andere Informationen an die jeweiligen Umstände anzupassen.
Außerdem soll die Oberfläche natürlich nicht aufwendig oder langsam sein. Um diese Probleme zu lösen wurde die Benutzeroberfläche konsequent durch HTML-Seiten realisiert.
Das heißt die Anzeige kann gut und einfach gestaltet sein und ist nicht besonders aufwendig in der Darstellung. Die Anpassung an die Gegebenheiten der einzelnen Veranstaltungen ist dadurch gewährleistet, dass der Dozent und der Administrator die Seiten für die Veranstaltung selber verändern und an die Umstände anpassen können.

Dank der Realisierung des Projekts als Webapplikation und dessen Installation auf einen der Universität gehörigen Servers ist es ohne weiteres möglich die Applikation auch von einer großen Anzahl von Benutzern gleichzeitig nutzen zu lassen, ohne die Funktionsweise oder die Performance der Applikation zu beeinflussen.
Durch die Verwendung der Stylesheet Technologie ist es möglich ohne den Quellcode der Applikation zu verändern die grafische Oberfläche auf die des verwendenden Lehrstuhls einzustellen.
Auf diese Weise kann das Produkt in die Homepage des Lehrstuhls eingebunden werden ohne das sich das Produkt optisch von dieser abhebt.
Das Produkt benötigt aufgrund der Servlet Technologie einen Servlet fähigen Container zum Beispiel einen Apache Tomcat Server. Außerdem sind Cookies unbedingt erforderlich!
Auf der Grundlage dieser Technologien entwickeln wir eine Software die den Anforderung in diesen Anwendungsgebieten gewachsen ist.

Gruppe: ueb 11
Bearbeitet von: Nicky Fritsch, Patrick Wabnitz, Christian Beyerl, Andrea Müns
zuletzt bearbeitet: 09.06.2003

1.2.Produktübersicht

Beim Starten der Installierten Applikation wird zuerst die Startseite erzeugt.
Diese beinhaltet alle öffentlich zugänglichen Informationen, wie die Einträge des moderierten Diskussionsforums sowie allgemeine Informationen zu der Vorlesung
für die das Produkt verwendet wird.
Um in den aktiven Bereich der Übungsverwaltung zu kommen kann man sich jetzt über den Button <Login> mittels Matrikelnummer und Passwort in die Applikation einloggen. Ist der Benutzer noch nicht registriert, wird über den Button <registrieren> das Registrieren Seite aufgerufen, wo er die Möglichkeit hat seine Matrikelnummer und seine E-Mail Adresse auf dem Universitätsaccount einzutragen und ein vorläufiges Passwort auf diesem generieren zu lassen
(bzw. ein eigenes zu wählen) um damit Zugang zu der Applikation zu erhalten.
Die so entstandenen Logindaten werden in einer user.xml Datei auf dem Universitätsserver gespeichert.
Hat der User den Button <Login> betätigt, wird das Login Seite aufgerufen, wo er Matrikelnummer / Name und Passwort eingeben kann.

Diese werden eingelesen und mit dem Eintrag in der user.xml verglichen.
Bei Falschen Passwort erhält der User keinen Zugriff, bekommt die Nachricht „neu einloggen“ / „Password oder Benutzer falsch“ und kehrt in den Login-Bereich zurück.
Bei mehrmaligen Fehlversuchen wird eine Warnung ausgegeben und auf die Startseite zurückgeworfen um evtl. Passwort hacks grob zu verhindern.
Sind die eingegebenen Daten als korrekt verifiziert werden die dem Login zugeordnete Benutzerspezifische Informationen und Servletinfos überprüft und ausgegeben.
Somit erhält jeder Benutzer nur die für ihn zulässigen Funktionen.
Nun gibt es drei verschiedene Benutzer mit denen man sich in die Applikation einloggen kann:
als Dozent, Korrektor oder als Übungsteilnehmer. Diese sind in der Klasse Benutzer so festgelegt und beinhalten die jeweiligen Benutzerspezifischen Informationen, die an die Klasse Menue übergeben werden. Die Klasse Menue erzeugt nun eine HTML Seite welche die Buttons darstellt, die für den jeweiligen Nutzer zugänglich sein sollen.

Vorläufiger Produktaufbau für die Benutzergruppen:

Änderungen liegen im Bereich des Implementierungsaufwandes
mögliche zusätzliche Funktionen können folgen.

Betrachten wir nun die Applikation unter Verwendung des Benutzers Dozent:
Dieser hat nach Aufbau der für ihn generierten Seite die Auswahl aus 4 verschiedenen Unterpunkten, welche als Buttons auf dem Bildschirm gekennzeichnet sind. Das sind <Übungsgruppenverwaltung>, <Aufgabenveröffentlichung>, <Gruppenübersicht> und <Verwaltung Diskussionsforum>.
Durch Betätigung des Buttons <Übungsgruppenverwaltung> wird das Klasse UEbungsgruppenverwaltung aufgerufen, welches die Möglichkeit bietet, das man die Anzahl , den Raum und den Übungsleiter von Übungsgruppen aus einer gruppen.xml
Datei ausliest und die neu festgelegten Werte speichert.
Durch Betätigung des Buttons <Aufgabenveröffentlichung> wird das Klasse Aufgabenveroeffentlichung aufgerufen, welches die Möglichkeit bietet, Übungsaufgaben und Materialien zur Vorlesung durch Upload in die Datenbank zu veröffentlichen.
Auserdem werden die Abgabetermine aus einer termine.xml Datei ausgelesen und man kann dann neue Termine einfügen oder Alte ändern und diese wieder in der Datei speichern.
Durch Betätigung des Buttons <Gruppenübersicht> wird das Klasse Gruppenübersichtverwaltung aufgerufen. Hier hat der Dozent die Möglichkeit eine geordnete Liste über die erreichten Punkte der Übungsteilnehmer, die aus der Datei List.xml ausgelesen wurde, zu bearbeiten und neu zu speichern. In dieser Datei sind außerdem die Klausureinschreibungen der Übungsteilnehmer eingetragen und können ausgelesen und bei Bedarf geändert werden.

Gruppe: ueb 11
Bearbeitet von: Nicky Fritsch, Patrick Wabnitz, Christian Beyerl, Andrea Müns
zuletzt bearbeitet: 09.06.2003

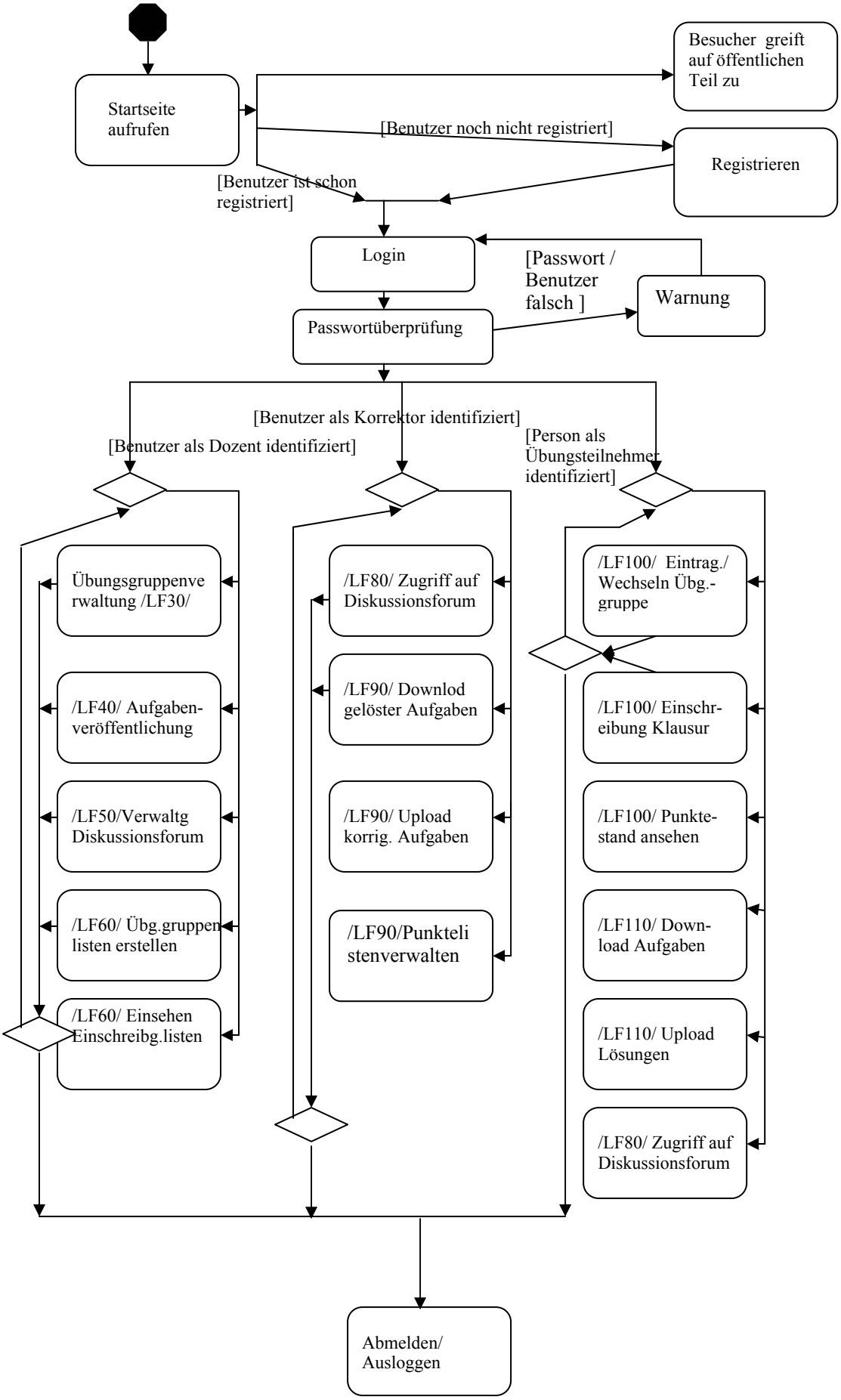
Betrachten wir nun die Applikation unter Verwendung des Benutzers Übungsteilnehmer:
Dieser hat nach Aufbau der für ihn generierten Seite die Auswahl aus 4 zusätzlichen Unterpunkten, <Diskussionsforum> , <Statusverwaltung>, <Accountverwaltung> und <Übungsaufgabenverwaltung> die über Buttons auf dem Bildschirm ausgegeben werden.
Durch Betätigen des Buttons <Statusverwaltung> wird das Klasse Status geöffnet.
Hier hat der Übungsteilnehmer die Möglichkeit sich zu Übungsgruppen einzuschreiben oder in eine andere zu wechseln.
Weiter werden in der Statusverwaltung noch die erreichten Punkte des Übungsteilnehmers aus den korrigierten Übungsaufgaben ausgegeben.
Weiter kann er sich in der Statusverwaltung noch zur Klausur einschreiben, dazu benutzt er den Button <Klausureinschreibung> und betätigt den Button <Bestätigen>, dann ist der Übungsteilnehmer automatisch zu der aktuell anstehenden Klausur eingeschrieben.
In <Übungsaufgabenverwaltung> stehen die Links zu den vom Dozenten auf dem Server abgelegten Übungsaufgaben, die sich der Student damit herunter laden kann. Des Weiteren kann der Übungsteilnehmer durch Betätigen des Buttons <Lösungen Hochladen> seine Lösungen der Übungsaufgaben auf die Server hochladen.

Betrachten wir nun die Applikation unter Verwendung des Benutzers Korrektor:
Dieser hat nach Aufbau der für ihn generierten Seite die Auswahl aus 4 verschiedenen Unterpunkten, <Diskussionsforum>, <Accountverwaltung>, <Punktelisten> und <Verwaltung Korrektur> die als Buttons auf dem Bildschirm gekennzeichnet sind.
Wählt man den Menüpunkt <Verwaltung Korrektur> wird das Klasse KorrekturVerwaltung geöffnet.
Der Korrektor hat nun die Möglichkeit auf die Daten zuzugreifen und aus dem Pool von eingesendeten Lösungen, der von den Übungsteilnehmer bearbeiteten und hoch geladenen Übungsaufgaben, sich welche herunter zu laden und zu korrigieren.
Die mit Kommentaren versehenen korrigierten Lösungen können so ebenfalls wieder in die Datenbank hochgeladen werden. Außerdem kann er mittels dem Button <Punktelisten> die Punkteliste der erreichten Punkte und andere Statistiken ausgeben lassen und neue Punktvergaben in die Liste eintragen und Speichern.

Auf der nächsten Seite folgt ein Diagramm zur Veranschaulichung.

Gruppe: ueb 11
Bearbeitet von: Nicky Fritsch, Patrick Wabnitz, Christian Beyerl, Andrea Münsz
zuletzt bearbeitet: 09.06.2003

Aktivitätsdiagramm:



Gruppe: ueb 11
Bearbeitet von: Nicky Fritsch, Patrick Wabnitz, Christian Beyerl, Andrea Müns
zuletzt bearbeitet: 09.06.2003

1.3.Grundsätzliche Design-Entscheidungen

Äußeres Erscheinungsbild:

Um das Erscheinungsbild des Designs im Webbrowser schnell ändern zukönnen benutzen wir Stylesheets ausgelagert in eine .css Datei welche u. a. Das Hintergrundbild oder div. Schriftformationen beinhaltet.
Da es sich um ein universitäres Verwaltungsprogramm handeln soll verzichten wir auf zu viele Farben und Bildchen und setzen lieber auf Übersichtlichkeit und Funktionalität.
Dazu benutzen wir u. a. dynamische Tabellen und Menüs welche auch bei unterschiedlichen Bildschirmauflösungen benutzt werden können.
Das Bewegen zwischen verschieden Seiten oder Abrufen der Informationen soll über eindeutige Buttons geschehen welche zusammenhängend dargestellt sind z.B. in einer Menüführung links am Bildschirm und nicht über wild verstreute Links, um Übersichtlichkeit und Funktionalität zu wahren.
Einfache Benutzerspezifische Informationen sollen im selben Fenster / Tabelle dargestellt werden.
Andere Sachen wie das Diskussionforum wird extra dargestellt da dieses etwas komplexer ist.
Das Login / Logout Menü ist so gestaltet, dass Fehler minimiert werden und der Benutzer eindeutig weiß was zu machen ist. Ach sollte man nicht einfach auf einer Seite feststecken und evtl. nur über den Browser wechseln können, deswegen wird auf jeder Seite ein entsprechendes Menü oder ein Button zur Hauptseite gesetzt. Somit ist die einwandfreie Seitenführung gewährleistet.

Internes:

Die Klassenstruktur ist so gehalten, dass wichtige Funktionen gleicher Art auch in einer Klasse bleiben aber nicht zu viel oder zu wenig Servlets / Klassen gebraucht werden.
Sicher kann mehrere Methoden in einer einzigen Klasse bauen diese wäre aber sehr unübersichtlich. Baut man aber zu viele gehen die einzelnen zusammenhänge zwischen Attributen und Methoden verschiedener Klassen evtl. verloren.
Die .java Dateien sind gut strukturiert und dokumentiert was spätere Änderungen vereinfacht.
Zur Entscheidung zu Servlettechnologie siehe auch 1.1.

Ausgehend von einem Hauptservlet was u.a. die Startseite generiert haben wir uns entschieden nach dem Login und deren Überprüfung durch GetPass 3 weitere Servlets für jeden Benutzertyp zu erzeugen. Gemeinsame Atribute und Methoden werden von der Oberklasse „Benutzer“ vererbt.
Je ein eigenes Servlet für den Benutzer zu erzeugen hat den Vorteil dass, doGet und doPost was von der generierten Html Seite durch die Buttons und „action= …“ aufgerufen wird nur innerhalb der Benutzergruppe bleibt und nicht etwa grundverschiedene Funktionen der unterschiedlichen Benutzer an 1 Startservlet gesendet werden muss.
Des Weiteren haben wir entschieden eine Oberklasse Menue zu benutzen die für den Aufbau der dynamischen Menüführung zuständig ist.
Sie stellt dementsprechend Benutzerspezifische Buttons/Menüs sowie allgemein für alle zugängliche Funktionen zur Verfügung. Durch die Vererbungsstruktur und Erschaffen von Subklassen für einzelne Menüfunktionen können später leicht zusätzliche Buttons oder aber auch neue Benutzerspezifische Funktionen ergänzt werden sowie das Html Layout ohne Probleme angepasst werden.

Da es sich um eine Software handelt die an verschieden Universitäten und Instituten sowie unter änderbaren Domainen genutzt wird empfiehlt es eine Configklasse zu erstellen die über diverse Subklassen Anpassungen und Einstellungen speichert oder verwaltet.
So z.B. der Mailserver, die URL, Lage des Stylsheets, Ordner für Dateien, Gruppeneinstellungen …

Gruppe: ueb 11
Bearbeitet von: Nicky Fritsch, Patrick Wabnitz, Christian Beyerl, Andrea Müns
zuletzt bearbeitet: 09.06.2003

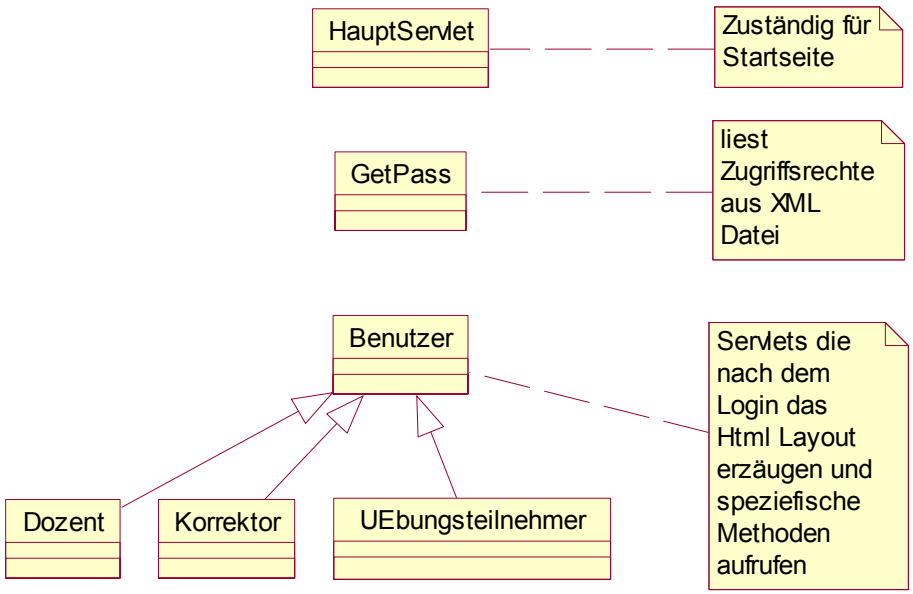
Verwaltungstechnische Funktionen der einzelnen Benutzer werden am Besten unter der Oberklasse „Verwaltung“ zusammengefasst die grundlegende Methoden bereitstellt (z.B. DB Zugriff)
In mehreren Subklassen, welche durchaus weitere Hierarchien fortsetzen, werden dann einzelne Methoden bereitgestellt die von den Benutzern gefordert werden. Also Dateiverwaltung mit Unterklassen sowie Gruppenverwaltung, Accountverwaltung und so weiter.

Dies hat den Vorteil, dass im späteren Implementierungsverlauf oder aber auch bei geänderten Anforderungen an das Programm sowie weitere Wünsche eines Neuen Kunden diese ohne großen Aufwand ergänzt werden können.
Siehe auch Qualitätsbestimmungen im Pflichtenheft.

1.4. Paket und Klassenstruktur

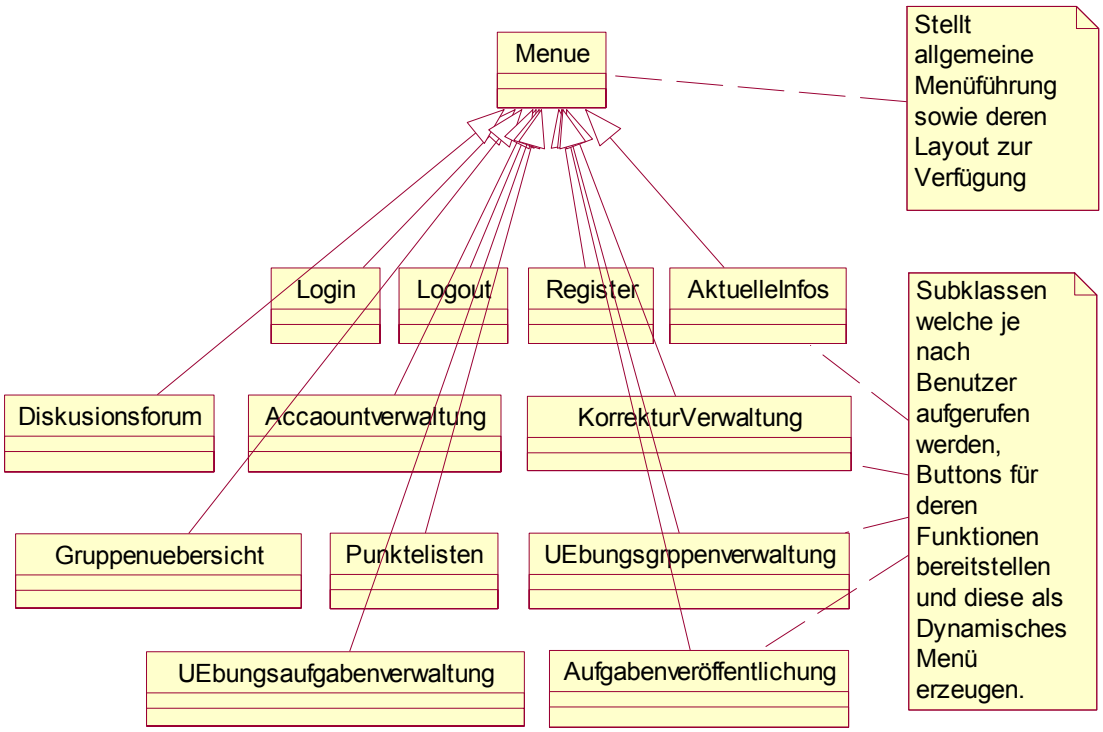
Im folgendem wird in statischen Diagrammen unsere Vorläufige Klassenstruktur veranschaulicht.
Der Übersicht halber haben wir erstmal auf Integrierung von Methoden und Attributen verzichtet.
Es sollte aber klar sein das durch die Namensgebung der Klasse gewisse Eigenschaften gegeben sind.
So ist selbstverständlich, das der Benutzer „UEbungsteilnehmer“ Attribute wie Matrikelnummer, Name, Übungsgruppe und so weiter hat.

Des Weiteren ist das hier natürlich nur vorläufiges Modell, da gegebenenfalls durch Testergebnisse und neuen Ideen im weiterem Implementierungsverlauf dieses angepasst werden muss/kann.



weitere Erklärungen siehe auch 1.3.

Gruppe: ueb 11
Bearbeitet von: Nicky Fritsch, Patrick Wabnitz, Christian Beyerl, Andrea Münsz
zuletzt bearbeitet: 09.06.2003



Durch Unterteilung in einzelne Klassen können jederzeit Änderungen einfach vorgenommen sowie zusätzlich neue Buttons und Funktionen integriert werden.
Diese werden von dem jeweiligen Servlet aufgerufen und die erzeugte Html Seite sendet auch wieder nur an dieses zurück.