Finale Version Gruppe: ueb 11
Bearbeitet von: Raphael Doehring, Patrick Wabnitz 25.05.03

Designbeschreibung WebAssign

Allgemeines

Voraussetzungen:

Die Software soll im universitären Umfeld eingesetzt werden und die Verwaltung und Organisation des Übungsbetriebs wesentlich erleichtern. Dazu muss sie für jeden Studenten und die angestellten Korrekteren 24 Stunden 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 und diese Möglichkeiten haben die meisten Studenten sogar von zu Hause oder mindestens in der Universität.
Nun sollte der Clientrechner des Studenten nicht mit zu sehr belastet werden 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 Kursleiter die Seiten für die Veranstaltung selber verändern und an die Umstände anpassen kann.
Da bei der Nutzung dieser Software unterschiedlichste Szenarien möglich sind, muss dem in der Architektur Rechnung getragen werden. Deshalb hat man sich dafür entschieden, das die Möglichkeit besteht, die das System auf unterschiedliche Rechner zu verteilen. Das heißt es ist möglich unterschiedliche Plattformen für die Ausführung der unterschiedlichen Aufgaben zu benutzen. Das macht sich besonders bei Anwendungsfällen mit sehr vielen Benutzern bezahlt, da nicht eine Maschine nötig ist, die die ganze Arbeit macht, sondern zum Beispiel die Datenbank anfragen von einem separaten System erledigt werden können.
Auf der Grundlage dieser Technologien ist es möglich eine Software zu entwickeln die den Anforderung in diesen Anwendungsgebieten gewachsen ist und sich nicht auf einen bestimmten Anwendungsbereich (z.B. Systeme mit weniger als 100 Nutzern oder einen bestimmten Fachbereich) beschränkt.

Finale Version Gruppe: ueb 11
Bearbeitet von: Raphael Doehring, Patrick Wabnitz 25.05.03

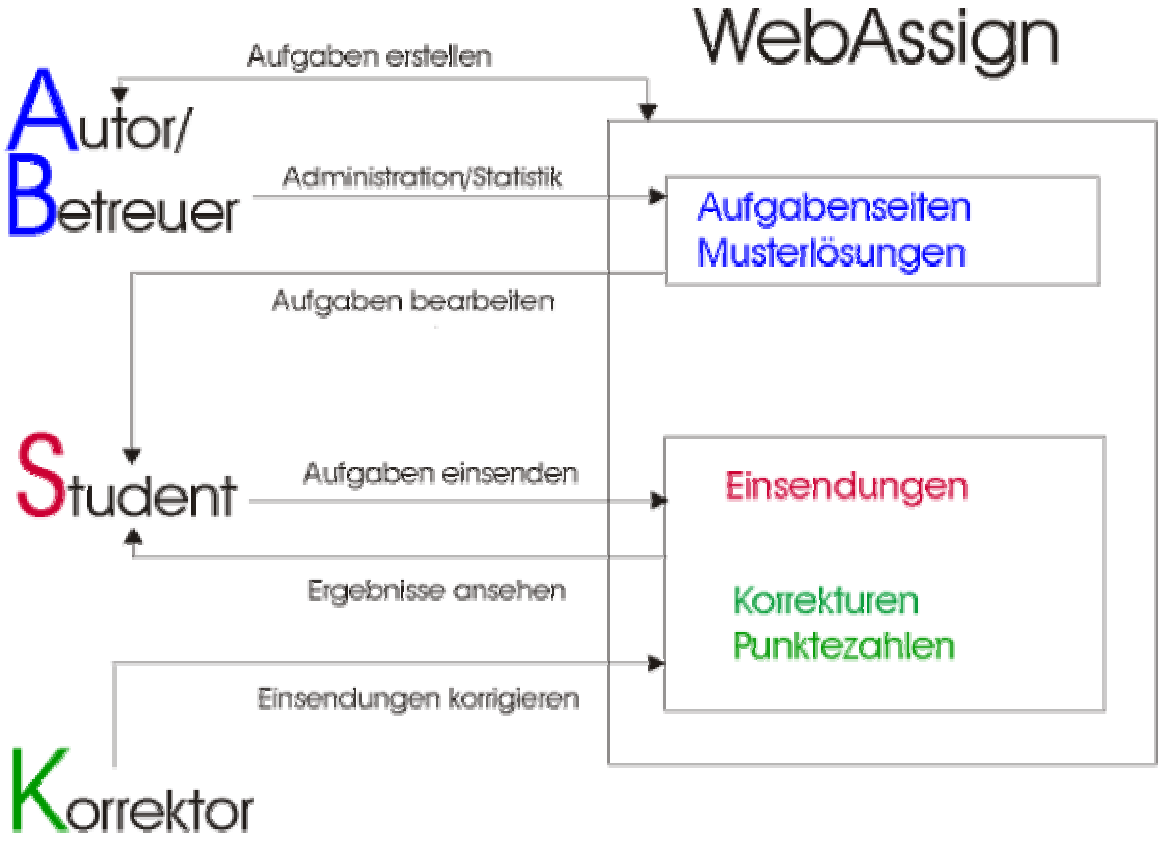
Designbeschreibung WebAssign

Produktübersicht

Grundstruktur:

Um die Grundstruktur des Software Projektes zu verstehen ist das zentrale Servlet von großer Bedeutung. Das Servlet webassign ist für die Entgegennahme der Anfragen des Clients verantwortlich. Es leitet diese Anfrage an den WebApplikationServer weiter. Dieser extrahiert aus der Anforderung des Benutzers, die nötigen Daten zur Authentifizierung bzw. die gewünschte Anfrage an das System. Nun werden die extrahierten Informationen an die zuständigen Stellen weitergeleitet, bzw. bei der Authentifizierung auf die dafür notwendigen Ressourcen zugegriffen. Dies kann unter Umständen auch ein LDAP-Verzeichnisserver sein.
Der WebAssignService bietet die Grundfunktionalitäten GET, POST und PUT, die je nach http-request gewählt werden. Sie beinhalten die Basisfunktionen, wie eröffnen eines Antwortdatenstroms. Dabei kommt dem EinsendungsPostService eine besondere Bedeutung zu, da er einerseits für die Weiterleitung der Anfragen an das Modul WebAssingModel zuständig ist, andererseits aber auch schon eine automatische Korrektur von Aufgaben anstoßen würden. Die dazu implementierte Schnittstelle würde das Korrekturservlet ansprechen.
Das Modul WebAssignModel ist nun für die Kommunikation mit dem Benutzer je nach seinen Rechten verantwortlich. Es entscheidet, wer welche Möglichkeiten der Bearbeitung, Eingabe und Veränderung welcher Informationen hat. Die dafür nötigen Zugriffe auf die verwendete Datenbank werden in extra Klassen (Database) gekapselt und von den benötigenden Klassen angesprochen.

hier eine kleine Übersicht:



Finale Version Gruppe: ueb 11
Bearbeitet von: Raphael Doehring, Patrick Wabnitz 25.05.03

Designbeschreibung WebAssign

Die Benutzerschnittstelle:

Die Benutzerschnittstelle von WebAssign besteht aus einem Pool von HTML-Seiten, die teilweise verändert und angepasst werden können, um die Gestaltung noch mehr an die Gegebenheiten anpassen zu können. Um die Webseiten für Übungen oder Kurse dynamisch gestalten zu können gibt es bei WebAssign eine bestimmte Menge an Variablen, die der Kursleiter, der die Webseiten ändert mit einbauen kann. Diese werden bei der Anzeige der Seite dann durch deren aktuellen Inhalte ersetzt. Die Variablen werden je nach Sichtbarkeit unterschieden. Dabei werden globale, kursspezifische und personenbezogene Variablen unterschieden. Wichtige Variablen enthalten z.B. natürlich den Namen oder die E-Mailadresse des Nutzers. Aber auch auf die Ergebnisse einer eventuellen Vorkorrektur kann man über solche Variablen zugreifen.

Die Dienste:

Die WebAssignServices sind die eigentlichen Dienste, die der User in Anspruch nimmt. Sie bieten die Möglichkeit auf Daten der Übungen zuzugreifen, oder Aufgaben herunter bzw. Lösungen hochzuladen. Auch dieser Teil kann im Bereich der Webseiten dynamisch gestaltet werden und somit können die unterschiedlichen Anwender der Software auch das Aussehen der Basisfunktionen gut an ihre Bedürfnisse anpassen. Dieser Bereich der Software ist der, den der Anwender hauptsächlich sieht in wahrnimmt. Die anderen Module der Software werden für den Benutzer transparent geladen und ausgeführt.

Finale Version Gruppe: ueb 11
Bearbeitet von: Raphael Doehring, Patrick Wabnitz 25.05.03

Designbeschreibung WebAssign

Grundsätzliche Designentscheidungen

Verwendung von Servlets:
Für das OpenSource Softwareprojekt WebAssign wird als Basis die Servletarchitektur verwendet. Servlets bieten anders als Applets einfacher die Möglichkeit Serverressourcen zu nutzen und müssen vom Client nicht heruntergeladen werden. Natürlich kommt genauso die Programmiersprache Java zum Einsatz, mit allen damit verbundenen Vorteilen.
Ein weiterer wichtiger Vorteil von Servlets ist die einfache Erstellung von Benutzeroberflächen, die auf HTML-Basis erstellt werden können und auch in diesem Projekt eine besondere Rolle spielen. Es ist somit möglich einfach eine strukturierte Benutzeroberfläche zu erstellen, die jedoch durch die Erzeugung durch die Servlets komplett dynamisch gestaltet werden kann. Besonders bei einem Projekt wie diesem, wo unterschiedliche Benutzer mit unterschiedlichen Rechten und Sichten auf die Daten unterschieden werden müssen, macht sich der Einsatz dieser Technologie dadurch bezahlt.
Ein anderer Vorteil bei Mehrbenutzer Systemen, die Servlets in ihrer Basisform bieten ist das einfache Sessiontracking. Dies ist schon eine grundlegende Technologie und wird schon für einfache Servletanwendungen unterstützt. Durch die Verwendung von Cookies ist eine ständige Neuauthentifizierung des Benutzers überflüssig. Für jeden eingeloggten Benutzer wird eine Session mit eigenem Sessionkontext erstellt. Das heißt, das Servlet kann immer wieder während des Betriebes auf die Informationen, die schon gespeichert wurden zugreifen. Der Name oder eventuelle Rolleninformationen, bzw. Rechteinformationen gehen nicht verloren.
Der Sessionkontext ermöglicht das Speichern von Datenobjekten, was die Erhaltung der wichtigen Daten stark vereinfacht.
Diese Eigenschaften der Servlettechnologie sind der Grund, warum eben diese Technik für die Realisierung der Software eingesetzt wurde.

Verwendung einer DB:
Da die Daten, die für diese Projekt verwaltet werden müssen, je nach Anwendungsfall sehr umfangreich werden können, hat man sich für die Verwendung einer Datenbank entschieden. Die Vorteile für die Entwicklung liegen hier wieder auf der Hand: Es ist möglich die Datenbank auf einem separaten System zu halten, was die Administration von WebAssign wesentlich von der der DB entkoppelt. Das Austauschen der Datenbank wird erleichtert. Außerdem sind einige Daten schon in externen Datenbanken vorhanden und können auf diesem Weg mit verwendet werden. Beispielsweise sind Studentendaten schon in der Universitätsdatenbank enthalten und diese können auf die Weise weiterverwendet werden.

Verwundung von Schnittstellen:
Für eine bessere Erweiterbarkeit und eine einheitliche Kommunikationsgrundlage existieren Schnittstellen, die für das Softwareprojekt verwendet werden. Die Systeme auf die damit zugegriffen wird sind im Allgemeinen sehr verbreitet, weshalb eine einfache Anpassung der Applikation auf gegebene Umstände sehr einfach wird. Für eventuell andere Basissysteme können die Schnittstellen angepasst oder erweitert werden.
Die wichtigsten Verwendetet Schnittstellen sind CORBA, LDAP und die DB-Schnittstelle.

Finale Version Gruppe: ueb 11
Bearbeitet von: Raphael Doehring, Patrick Wabnitz 25.05.03

Designbeschreibung WebAssign

CORBA :
Zur Kommunikation zwischen den einzelnen Komponenten der Software sind klare Schnittstellen notwendig. Da die Kommunikation nicht nur auf ein System beschränkt ist, sondern aus unterschiedlichen Gründen auch auf über ein Netzwerk verteilte Services zugreifen muss, haben sich die Programmierer für die CORBA-Schnittstelle entschieden.
Die WebAssignAppikation kann nun sowohl als Dienstgeber oder Dienstnehmer auftreten und nutzt dabei jeweils diese Schnittstelle. Die genaue Spezifikation der Kommunikation sind Datenstrukturen festgelegt, die eingehalten werden. Diese sind über ein Interface festgelegt, welches von jeder dienst gebenden Applikation implementiert werden muss. Dieses Interface heißt WebAssignCorbaServer und enthält alle für die Kommunikation wichtigen Funktionen.

LDAP:
Da an der Universität ein Verzeichnisserver auf LDAP-Basis existiert, über den die Speicherung von Benutzerdaten geschieht, bietet es sich natürlich an diesen zur Authentifizierung der Nutzer zu benutzen, um diese Informationen nicht duplizieren und redundant speichern, sowie an zwei Stellen aktualisieren zu müssen. WebAssign authentifiziert sich am LDAP Server und kann bei Erfolg dann Daten wie Vorname, Nachname und E-Mailadresse abfragen.

JDBC:
Die Verwendung diese Schnittstelle bietet sich bei Java Projekten an, da sie schon implementiert ist und somit leicht benutzt und erweitert werden kann.
Sie bietet auch die Möglichkeit eine Änderung der Datenbank ohne großen Aufwand zu gestalten, da diese Schnittstelle einheitlich für unterschiedliche Datenbanken verwendet werden kann.

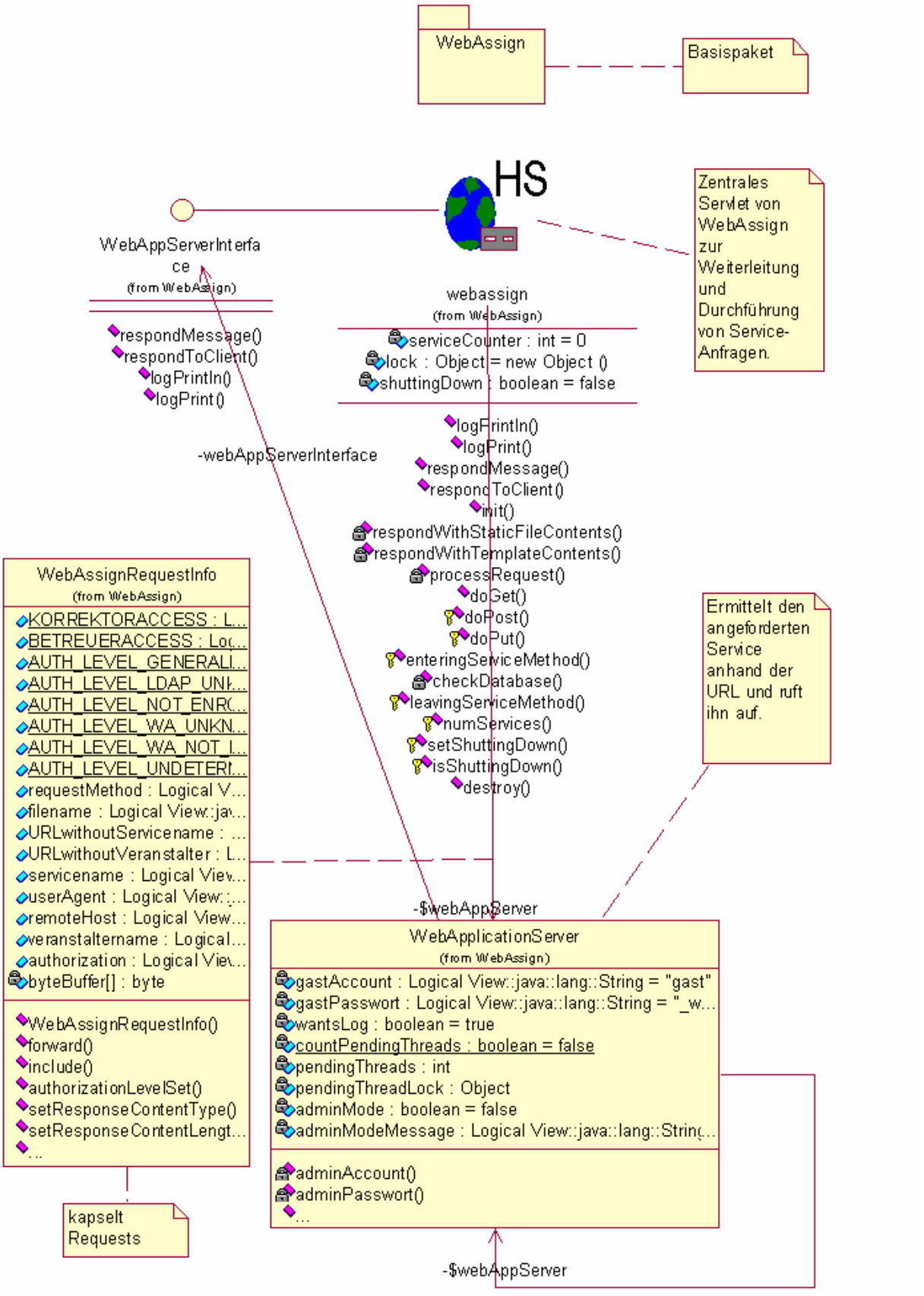
Dynamische Nutzerschnittstelle über HTML:
Die Verwendung von HTML-Seiten als Nutzerschnittstelle liegt bei einer Servlet Umgebung natürlich auf der Hand. Nun ist das ein wenig statisch und der Benutzer hat keinen Einfluss auf die Menu's oder den Inhalt der Seiten. Dieses Problem wurde eben dadurch gelöst, dass der Übungsleiter die WebSeiten zur Übung verändern und mit Variablen dynamisch gestalten kann. Das bietet jedem Benutzer dieser Software die Möglichkeit sie wesentlich besser an seine Bedürfnisse anzupassen. Das bringt im universitären Umfeld große Vorteile, wenn man bedenkt, wie unterschiedlich die Ansprüche der verschiedenen Fakultäten sind und wie schwer es ist dafür eine einheitliche Oberfläche zu schaffen.

Klassenstruktur:

weitere / bessere Qualität siehe *.mdl Datei

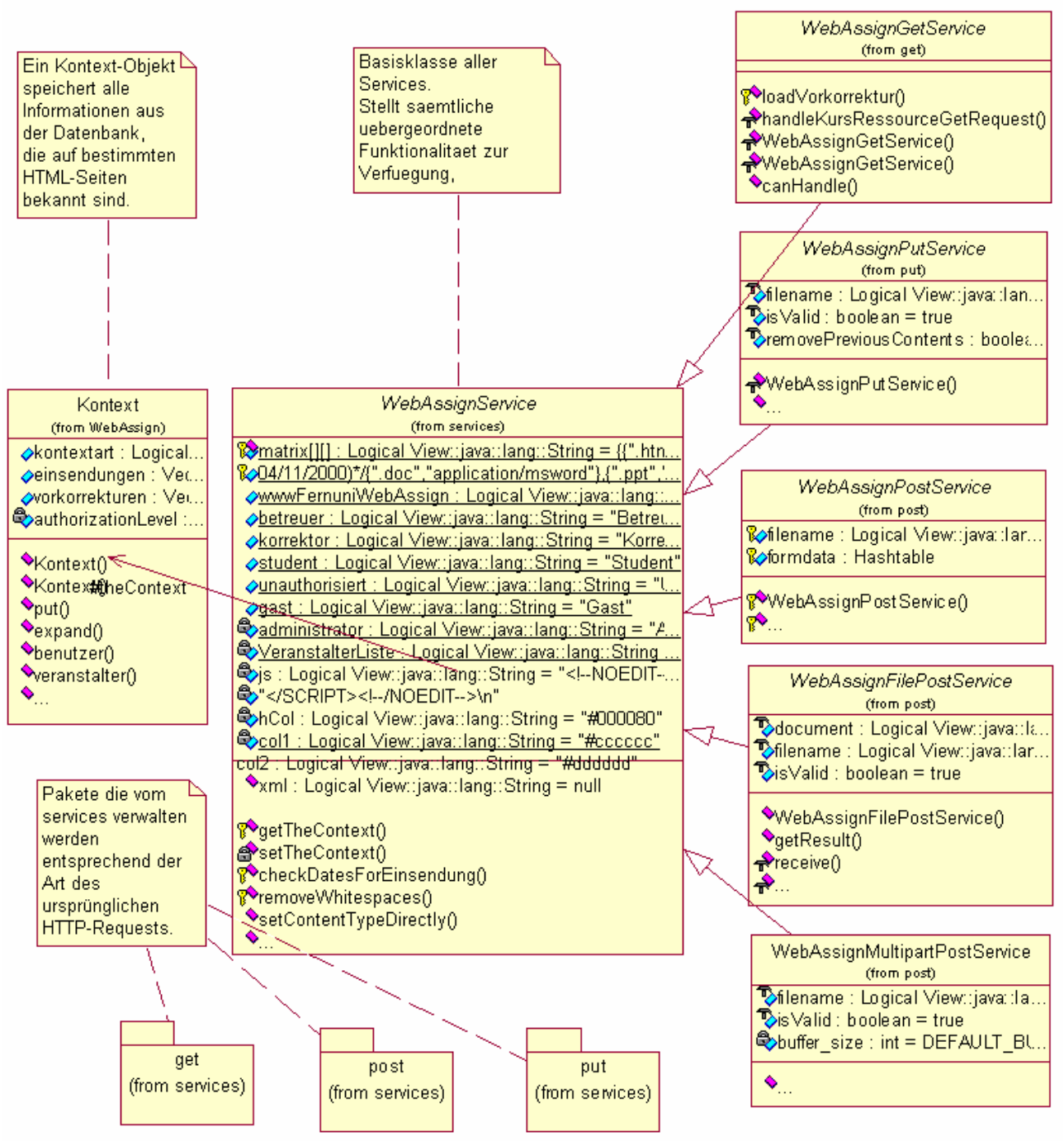
Finale Version Gruppe: ueb 11
Bearbeitet von: Raphael Doehring, Patrick Wabnitz 25.05.03

Designbeschreibung WebAssign



Finale Version Gruppe: ueb 11
Bearbeitet von: Raphael Doehring, Patrick Wabnitz 25.05.03

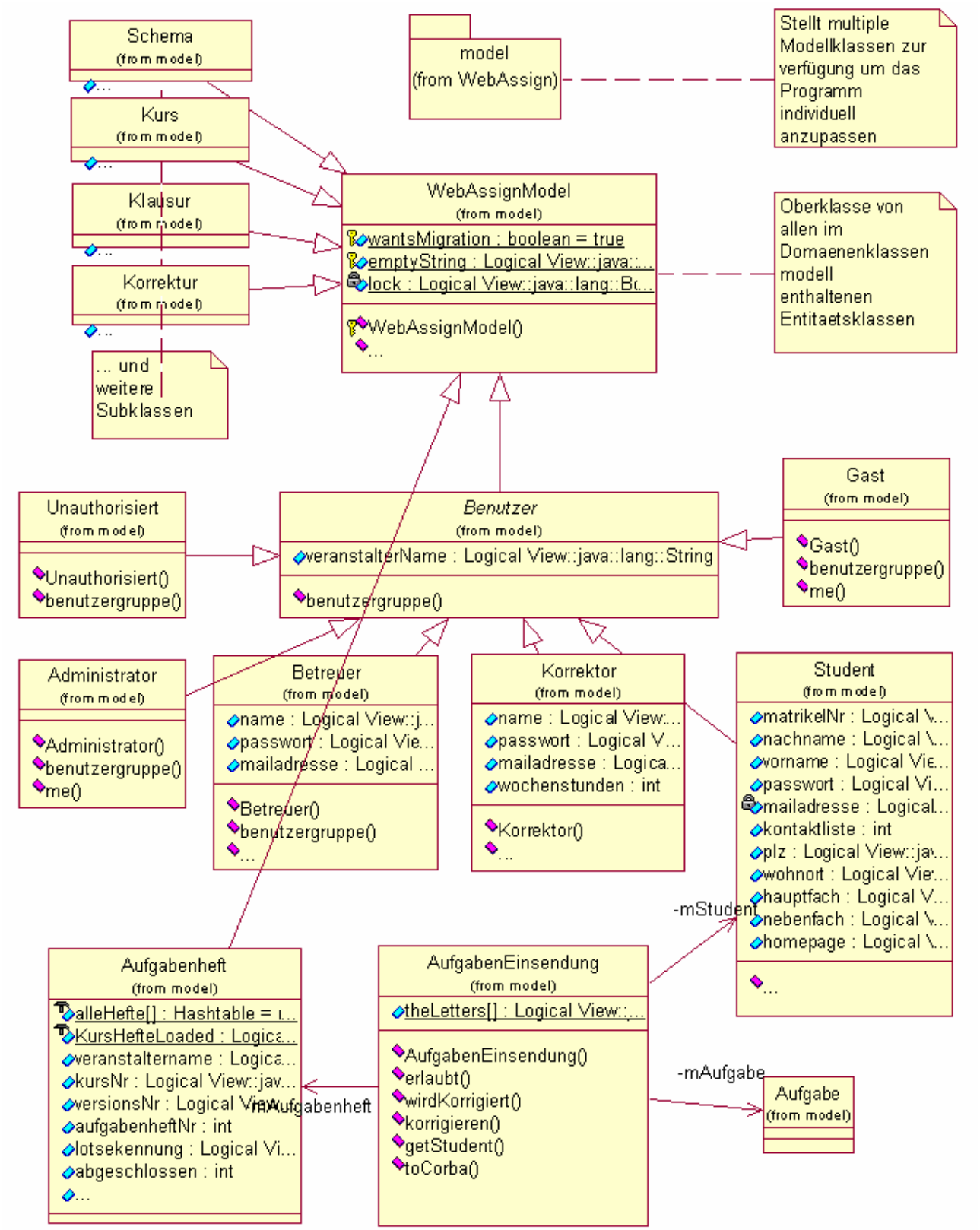
Designbeschreibung WebAssign
Paket services verwaltet von WebApplicationServer



Finale Version Gruppe: ueb 11
Bearbeitet von: Raphael Doehring, Patrick Wabnitz 25.05.03

Designbeschreibung WebAssign

von WebAssignPostService wird u.a. folgendes Paket verwaltet:



Finale Version Gruppe: ueb 11
Bearbeitet von: Raphael Doehring, Patrick Wabnitz 25.05.03

Designbeschreibung WebAssign

Klassen für Datenbankzugriffe: