

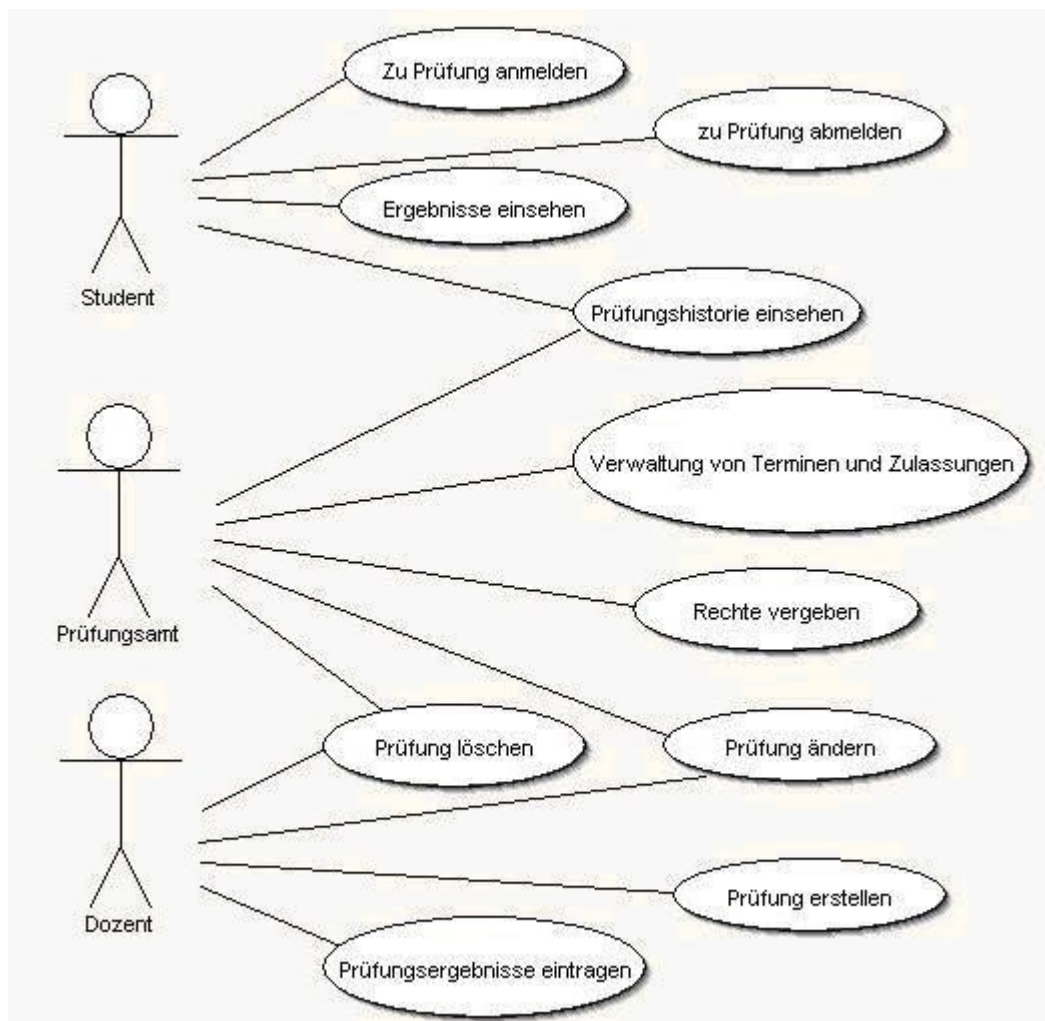
## Entwurfsbeschreibung

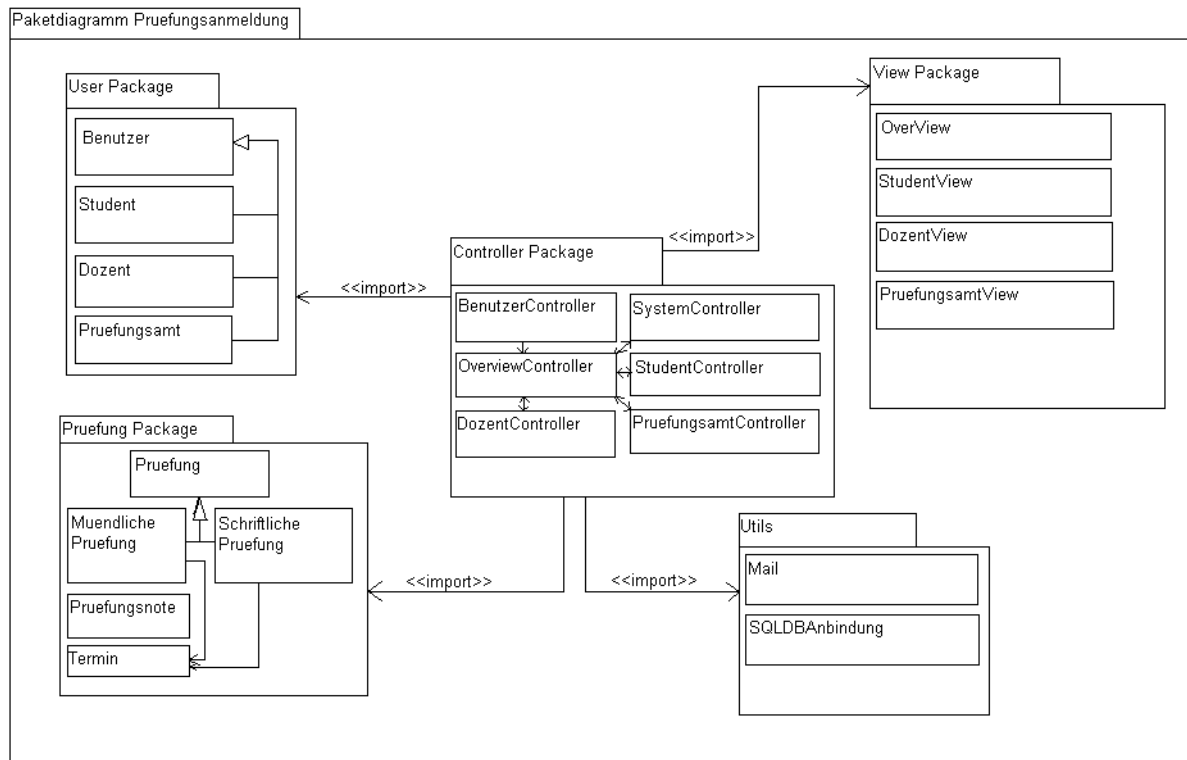
### 1. Allgemeines

Die Prüfungsanmeldung stellt eine funktionale Erweiterung zum Lernmanagement-System OLAT dar. Auf der Basis von OLAT als serverseitige Anwendung integriert, soll es in den Rollen Student, Dozent und Prüfungsamt das Verwalten von Prüfungen zu erstellen Lehrveranstaltungen ermöglichen. Die Prüfungsanmeldung ist dabei aus Servlets und Java-Klassen konstruiert, sowie Java-Server-Pages für die grafische Darstellung.

### 2. Produktübersicht

Die Prüfungsanmeldung stellt jeder der drei Rollen Student, Dozent und Prüfungsamt verschiedene Funktionalitäten zur Verfügung. Das Prüfungsamt ist der Akteur mit den meisten Rechten. Es kontrolliert die Akteure mithilfe umfassender Editier- und Löschfunktionen. Als Dozent kann man eigene Prüfungen anlegen, ändern und löschen, sowie die dafür angemeldeten Studenten verwalten und später Prüfungsergebnisse eintragen. Der Student steht ganz unten in der Hierarchie und hat nur die Möglichkeiten, sich zu bestehenden Prüfungen an- und abzumelden. Im Nachhinein kann er auch seine Prüfungsergebnisse einsehen.

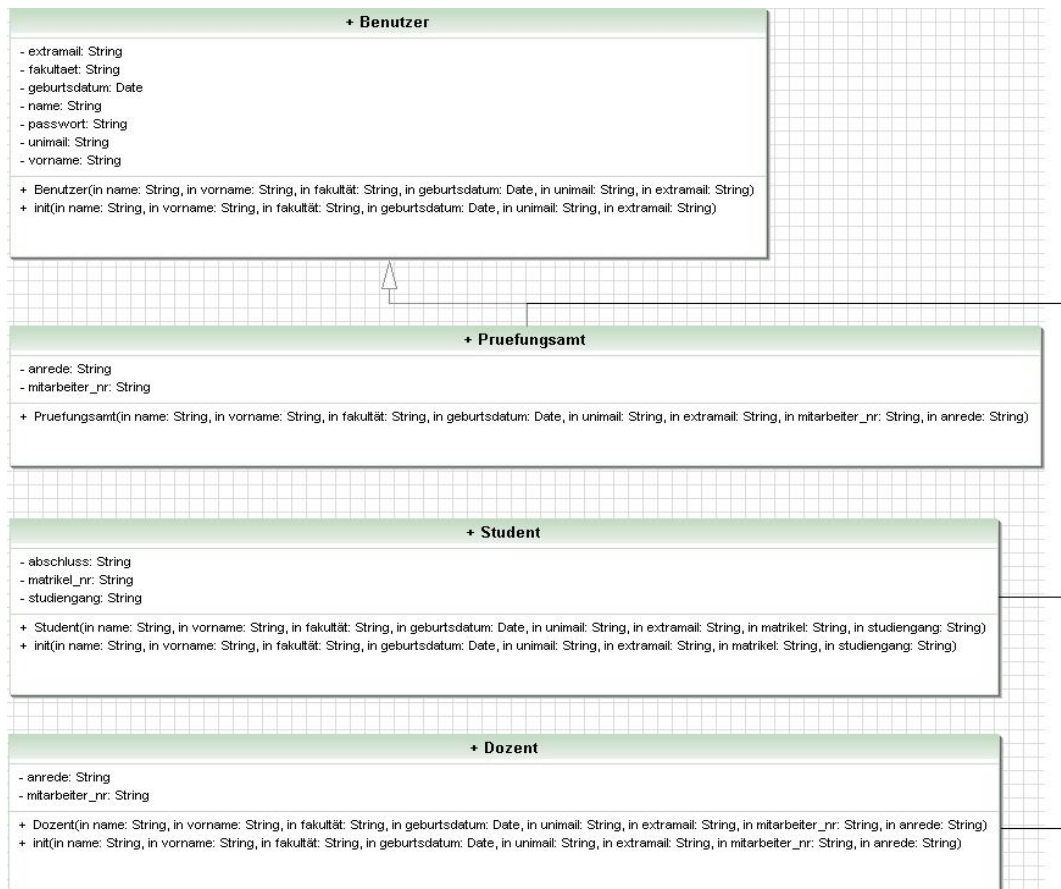


**Statisches Modell****3. Grundsätzliche Struktur- und Entwurfsprinzipien für das Gesamtsystem**

Der Gesamtcode des Systems setzt sich aus fünf Paketen zusammen. In dem Paket UserPackage werden die Benutzer des Systems festgelegt. Im Paket PruefungPackage werden die Pruefungen festgelegt und es wird zwischen mündlicher und schriftlicher Prüfung unterschieden. Das Paket ViewPackage definiert den gesamten View. Der View setzt sich aus Servlets zusammen die das Velocity-Framework und Java-Server-Pages nutzen. In dem Paket Utils werden zusätzliche Funktionen, wie der Mailversand sowie die Datenbanksteuerung definiert. Das zentrale Paket welches für die gesamte Steuerung der Anwendung verantwortlich ist, ist das Paket ControllerPackage. In diesem Paket gibt es einen OverviewController, welcher für die Steuerung gemeinsam genutzter Fenster aller Benutzer verantwortlich ist. So z.B. die Erzeugung der Hauptübersicht. StudentController, DozentController und PruefungsamtController erzeugen und steuern die Views für einzelne benutzerspezifische Interaktionen wie z.B. Anmeldung für eine Prüfung für Student oder Erstellung einer Prüfung für Dozent. SystemController ist der Controller der die Systemdienste steuert. Deshalb muss eine Instanz dieser Klasse ständig auf dem Server laufen.

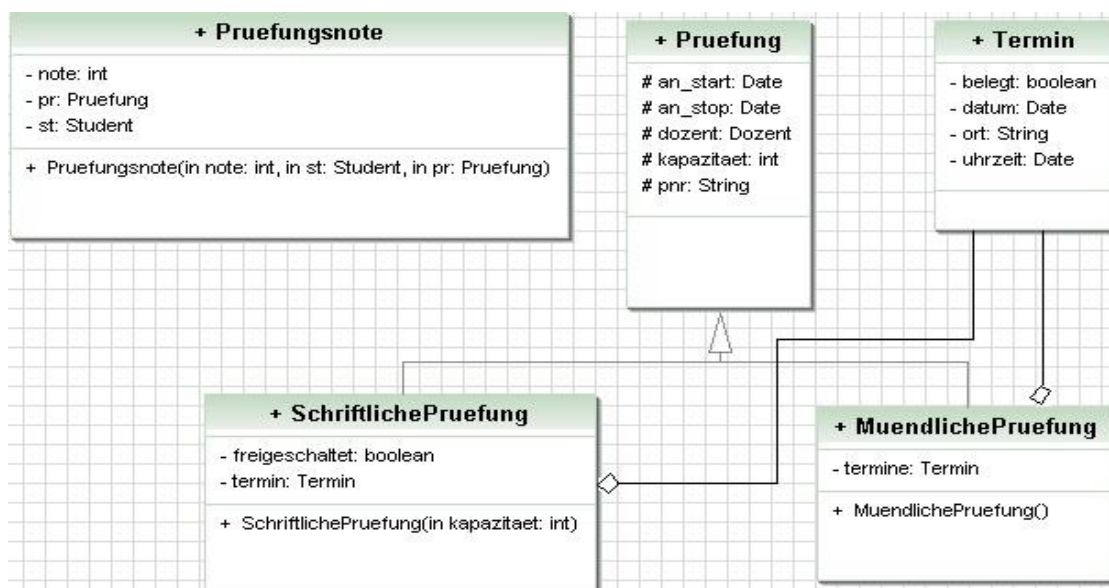
**4. Grundsätzliche Struktur- und Entwurfsprinzipien der einzelnen Pakete****Paket UserPackage**

**Softwaretechnik-Praktikum SS 2007**



In Benutzer werden die Eigenschaften aller Nutzer des Systems definiert. Danach werden die weiteren Eigenschaften der einzelnen Nutzer in den jeweiligen Klassen definiert.

Paket PruefungPackage

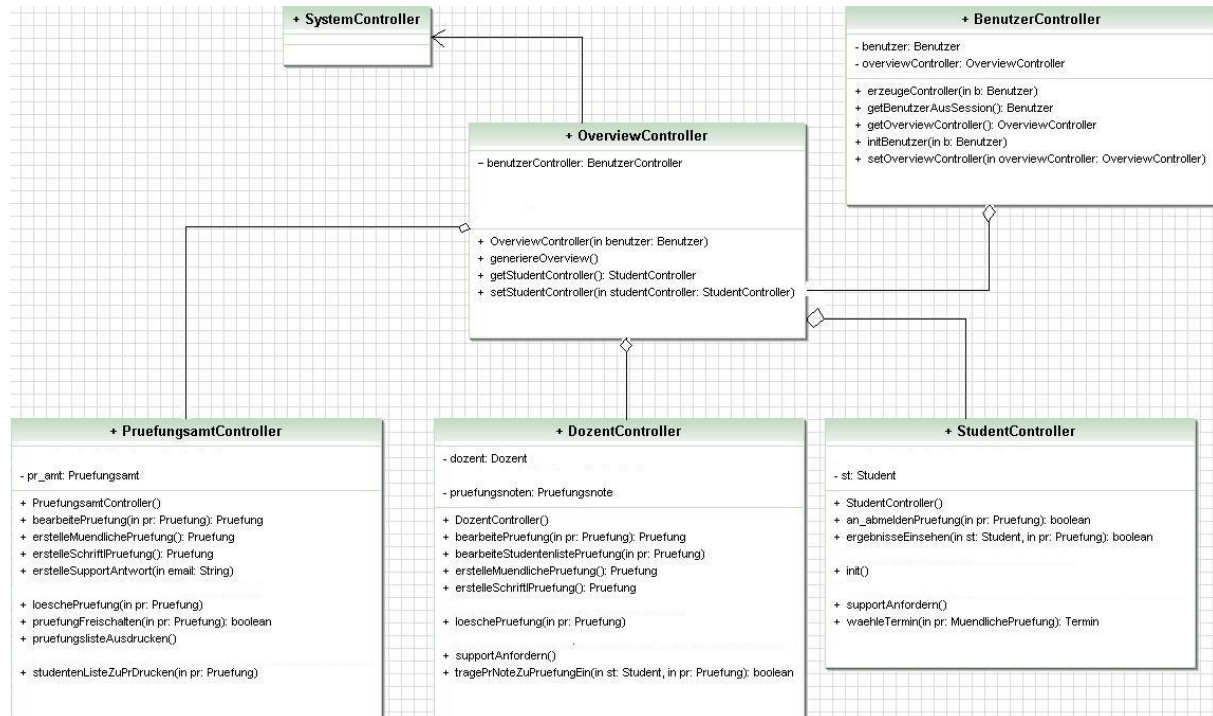


In der Klasse Pruefung werden generelle Eigenschaften einer Prüfung festgelegt die sowohl für schriftliche als auch für mündliche Prüfungen gilt. In den Klassen SchriftlichePruefung und MuendlichePruefung werden danach weitere Eigenschaften der einzelnen

**Softwaretechnik-Praktikum SS 2007**

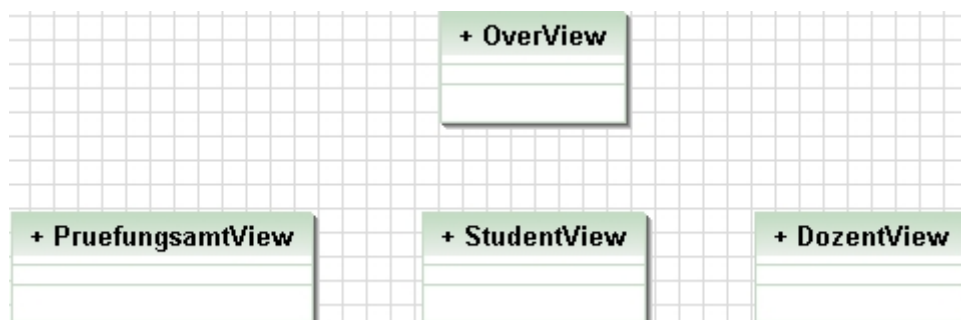
Prüfungsformen spezifiziert. Eine weitere wichtige Klasse des Pakets ist Pruefungsnote, diese dient der Verknüpfung von Student, Pruefung und Prüfungsnote. Somit wird durch die Klasse eine Relation erstellt, die dann die leichte Speicherung in der SQL-Datenbank ermöglicht.

Paket ControllerPackage



Das Paket ControllerPackage ist die zentrale Steuereinheit der Anwendung. Über die Klassen BenutzerController wird der Benutzer von OLAT aus den Sessions ausgelesen und noch durch fehlende Angaben ergänzt. Danach wird er je nach Recht an den jeweiligen zugehörigen Controller durch OverviewController übergeben. OverviewController erzeugt den Overview der eine Gesamtübersicht der Prüfungen darstellt. Desweiteren kann er je nach Recht bestimmte Funktionen im View z.B. Anmelden an einer Prüfung freischalten. Der SystemController verwaltet und steuert Systemprozesse, welche noch näher spezifiziert werden müssen. StudentController, DozentController und PruefungsamtController sind selbsterklärend.

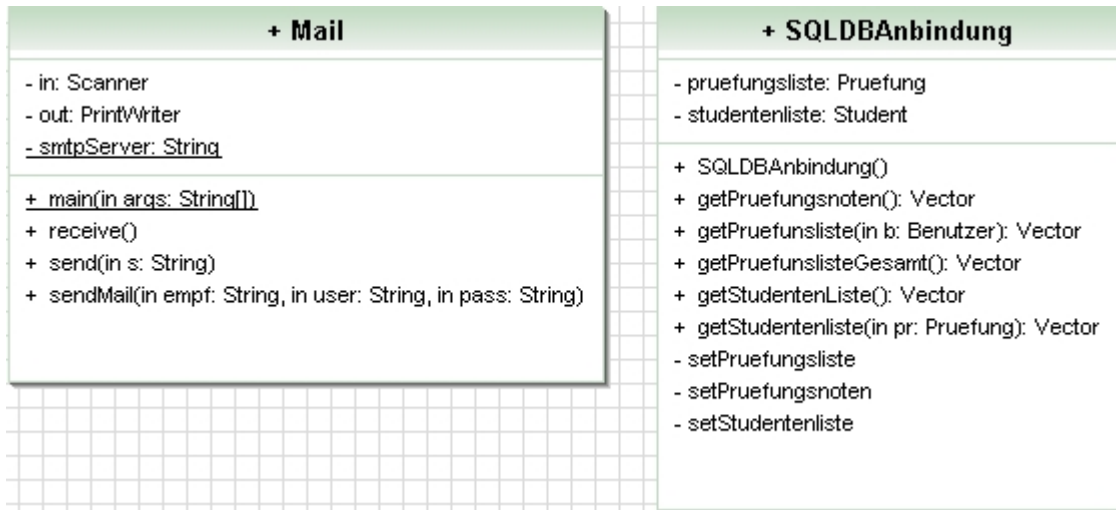
Paket ViewPackage



**Softwaretechnik-Praktikum SS 2007**

Die Klassen des Views wurden für die erste Entwursbeschreibung nur grob erfasst, da bei Beginn der Implementierung die beste Variante für die Erstellung der Views erst erfasst wird. Daher ob JSPs oder Velocity oder beides in Kombination verwendet wird.

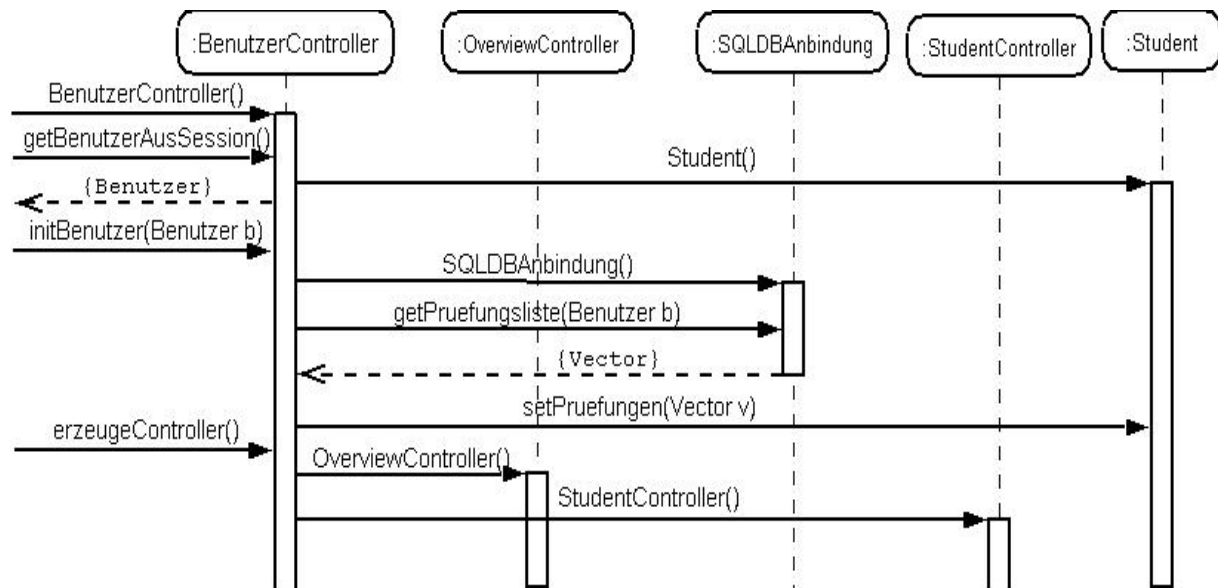
Paket Utils



Das Paket Utils enthält die Klassen Mail und SQLDBAnbindung welche sowohl für den Mailverkehr sowie die Datenbanksteuerung benötigt werden.

**Dynamisches Modell**

Folgendes Diagramm zeigt den Ablauf der Initialisierung der Controller und des Users des Plugins, nachdem der in Olat eingebundene Button „Prüfungsanmeldung“ ausgewählt wurde.



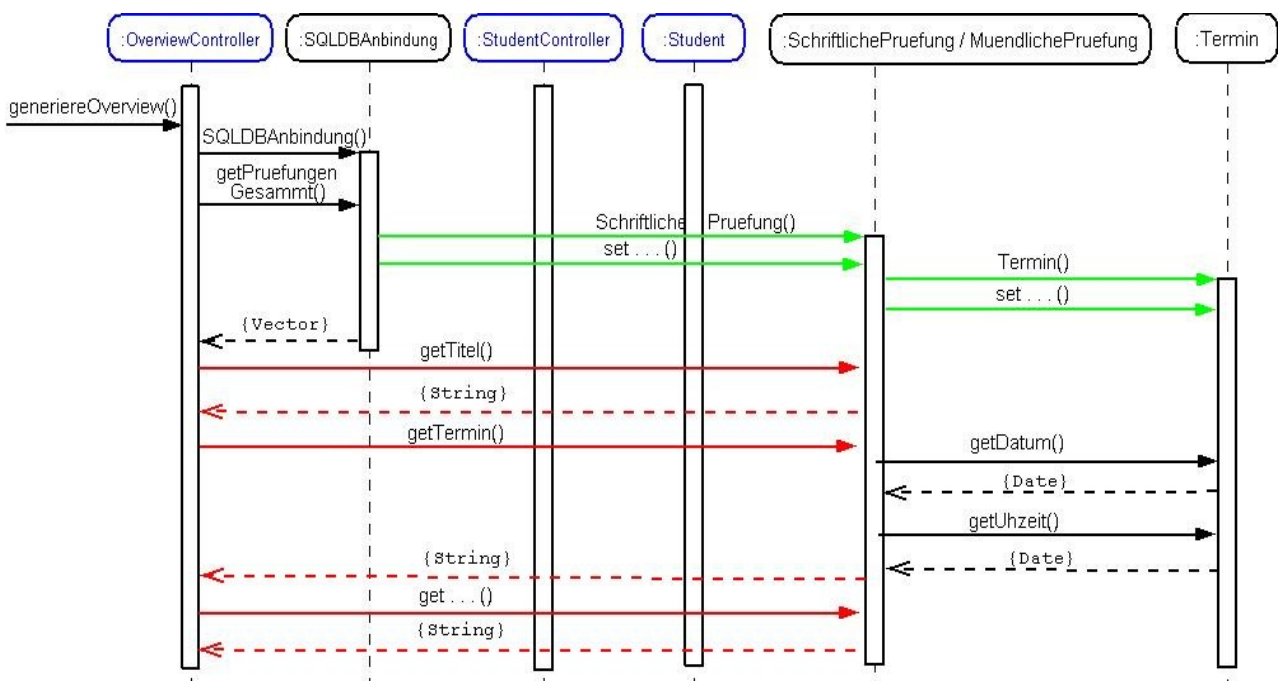
zunächst wird der BenutzerController ausgeführt, der aus der Sessiondatei den aktiven Nutzer ausliest und mit den Daten aus der Olat-Datenbank über die User ein Objekt von Benutzer anlegt. Je nachdem, welche Rechte der Nutzer hat, bekommt er eine der 3 Rollen zugesprochen: Student, Dozent oder Prüfungsamt.

**Softwaretechnik-Praktikum SS 2007**

Aus der Datenbank des Plugins werden nun die eigenen Prüfungen geladen und an das Objekt als Attribut gebunden. Bei Studenten sind das Die Prüfungstermine, für die er angemeldet ist. Beim Prüfungsamt und bei Dozenten sind dies die Prüfungen, die sie erstellt haben.

Danach werden die Controller ( OverviewController und rollenspezifischer Controller, hier StudentController) geladen.

Nach der Initialisierung der Controller werden automatisch die Prüfungen angezeigt. Für Studenten mit der Einschränkung, dass die Prüfung noch nicht voll sein darf und vom Prüfungsamt bereits freigeschalten sein muss. Folgendes Szenario spielt sich ab, wenn der Request „Pruefungen anzeigen“ beim Controller ankommt:.



Anmerkung: (get ...() und set...()) sind Abkürzungen für die nötigen get- und set-Methoden, die aufgerufen werden müssen.

Sollen nur die eigenen Prüfungen angezeigt werden, so funktioniert dies genauso, nur dass die Prüfung angezeigt wird, wenn auch im Vector Pruefungsliste des Benutzers ein Eintrag steht, dass er angemeldet ist bzw. er die Prüfung erstellt hat.

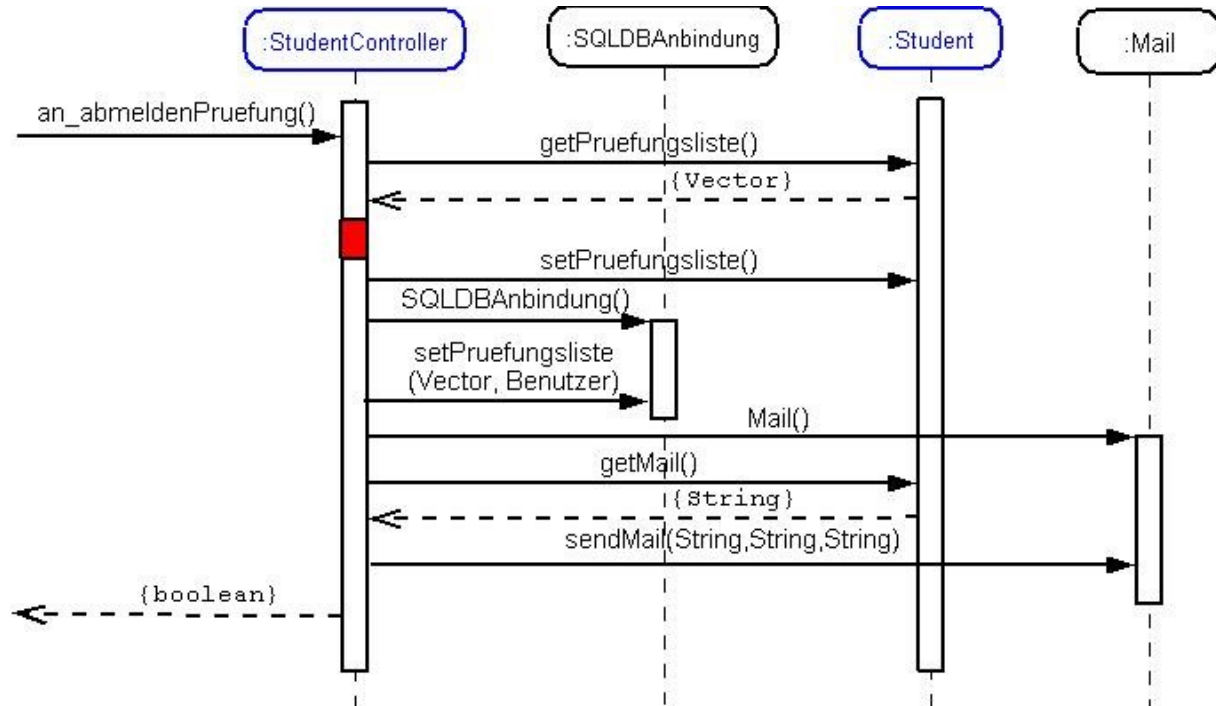
Die blauen Objekte sind bereits Instanziert und sind folglich bei Beginn des Szenarios schon vorhanden. Nach dem Request werden die ganzen Prüfungen in einer Schleife ausgelesen (grün gekennzeichnet – Beispiel für eine Prüfung) und in Form eines Vektors zurückgegeben. In einer zweiten Schleife werden alle anzuzeigenden Prüfungen ausgelesen (rot markiert – Beispiel für eine Prüfung) und angezeigt.

Dieses Beispiel bezieht sich auf eine schriftliche Prüfung, gilt aber auch für eine mündliche Prüfung. Der einzige Unterschied ist das Attribut Termin/Termine... bei schriftlichen Prüfung ist nur ein Termin erlaubt – wohingegen bei mündlichen Prüfungen mehrere Termine erstellt werden können.

**Softwaretechnik-Praktikum SS 2007**

Betrachten wir nun die Operationen des Studenten:

Er kann sich für schriftliche und mündliche Prüfungen an- und abmelden.



Bei diesem Beispiel wird das An- und Abmeldeszenario für eine schriftliche Prüfung durchgespielt. Je nachdem ob der Student schon angemeldet ist oder nicht, gibt es zu jeder Prüfung einen Anmelde- oder Abmeldebutton. Dem StudentController wird die Prüfung übergeben und er verarbeitet den Request. Hierbei wird die Prüfung dem Prüfungsvector hinzugefügt bzw. entfernt (rot) und sowohl dem Benutzer zugeteilt als auch in der Datenbank gespeichert. Danach wird eine automatisch generierte Mail als Bestätigung der Anmeldung an den Student geschickt. War alles erfolgreich wird ein „true“, zurückgegeben und auf dem Bildschirm erscheint eine Bestätigung.

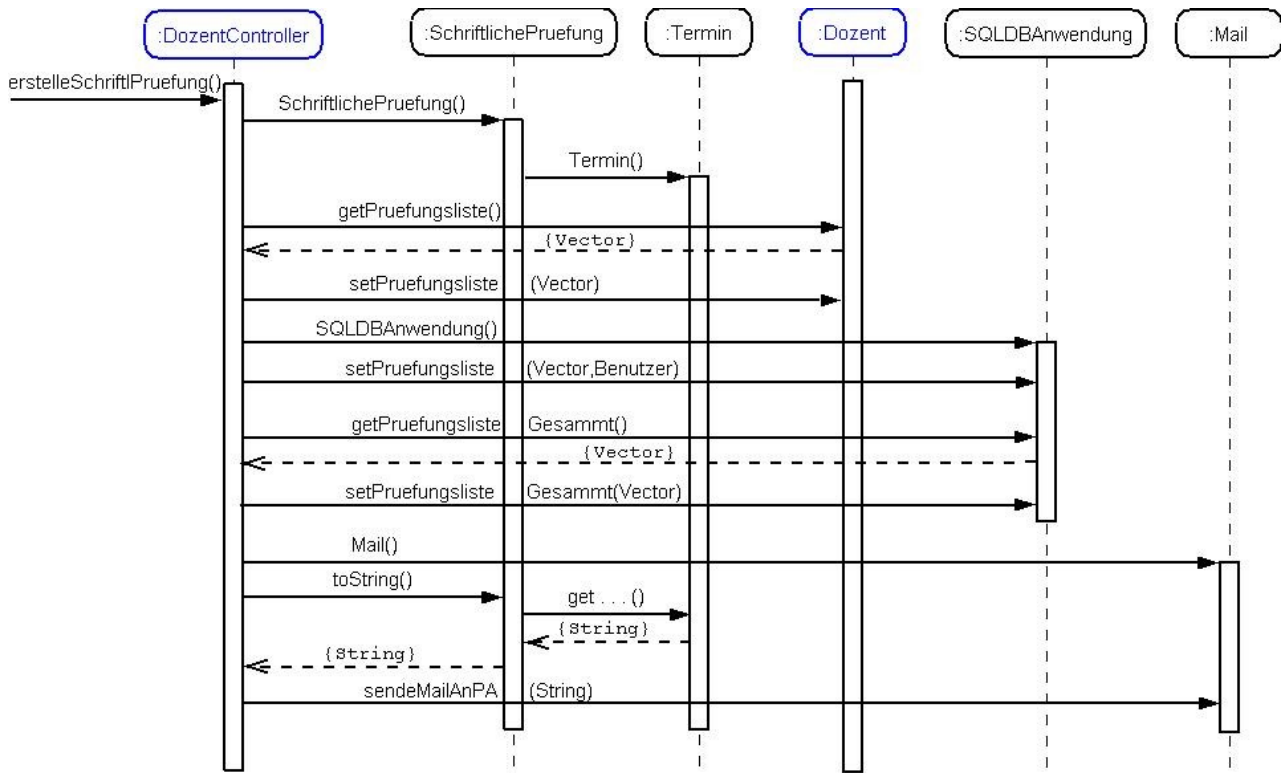
Blaue Objekte sind zu Beginn der Session bereits erstellt wurden und in diesem Szenario vor Beginn schon instanziiert.

Die Methode „wähleTermin()“ zur Anmeldung an mündliche Prüfungen, läuft sehr ähnlich ab. Der Unterschied ist, dass noch ein Termin von den angebotenen gewählt wird. Ist man für einen Termin eingeschrieben, muss man sich erst abmelden, bevor man einen Neuen wählen darf.

Betrachten wir nun Prüfungsamt und Dozent:

**Softwaretechnik-Praktikum SS 2007**

Sie dürfen Prüfungen erstellen, bearbeiten und löschen. Der Unterschied ist, dass Dozenten nur auf eigene Prüfungen Zugriffsrecht haben, wohingegen das Prüfungsamt auf jede Prüfung Einfluss nehmen darf.



Das Erstellen einer Prüfung läuft für Dozenten und Prüfungsamt gleich ab. Es müssen Daten, wie maximale Belegung, Termin, Ort, usw., angegeben werden und das Objekt zur Prüfung wird erstellt. Dann wird die Prüfung an die Prüfungsliste angehängen und beim Dozent/ Prüfungsamt und in der Datenbank gespeichert. Ebenso wird der Prüfungsübersichtsliste die Prüfung angehängen. Eine Mail mit den Daten der Prüfung geht nun an das Prüfungsamt.

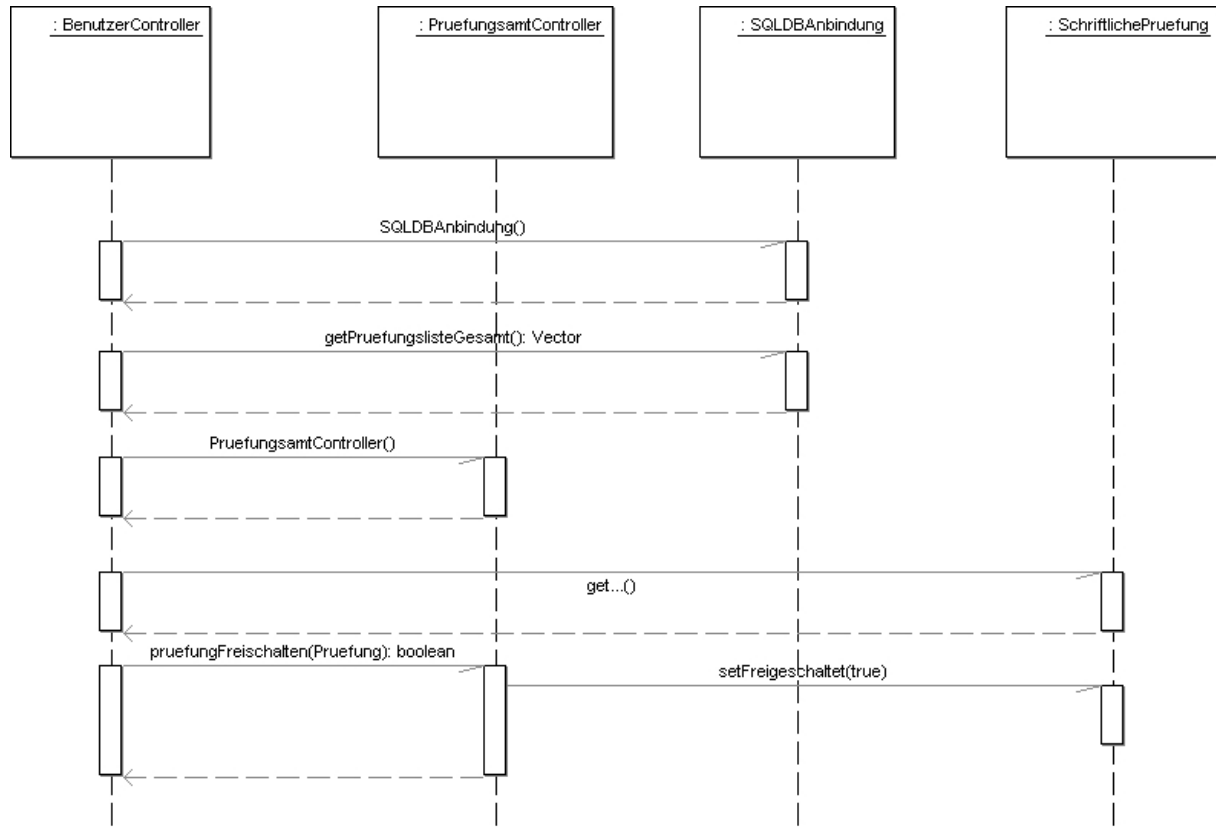
Will man eine Prüfung bearbeiten, läuft dies ähnlich ab. Nur das vorher alle Details geladen werden und dann die Prüfung mit der Änderung ersetzt wird. Die Benachrichtigung des Prüfungsamtes ist dann optional, da Dinge wie Raumänderung dem Prüfungsamt egal sein können.

Das Löschen einer Prüfung läuft über die gleichen Objekte ab. Die Prüfung wird aus dem Vector entfernt. Die Stornierung einer Prüfung wird dem Prüfungsamt per Mail mitgeteilt. Die Möglichkeit der Angabe eines Grundes soll vorhanden sein.

Bei dem Erstellen einer mündlichen Prüfung kann der Dozent mehrere Termine festlegen. Es ist geplant eine Unterstützung bei der Erstellung anzubieten, so dass der Dozent beispielsweise nur angibt: „9-13 und 14.30-18.30; stündlich; Montag bis Mittwoch“ und die Menge der Termine wird dann automatisch erstellt. Bei einer mündlichen Prüfung ist es nicht üblich das Prüfungsamt zu informieren, jedoch ist es optional, dass eine Mail an jenes geschickt wird.



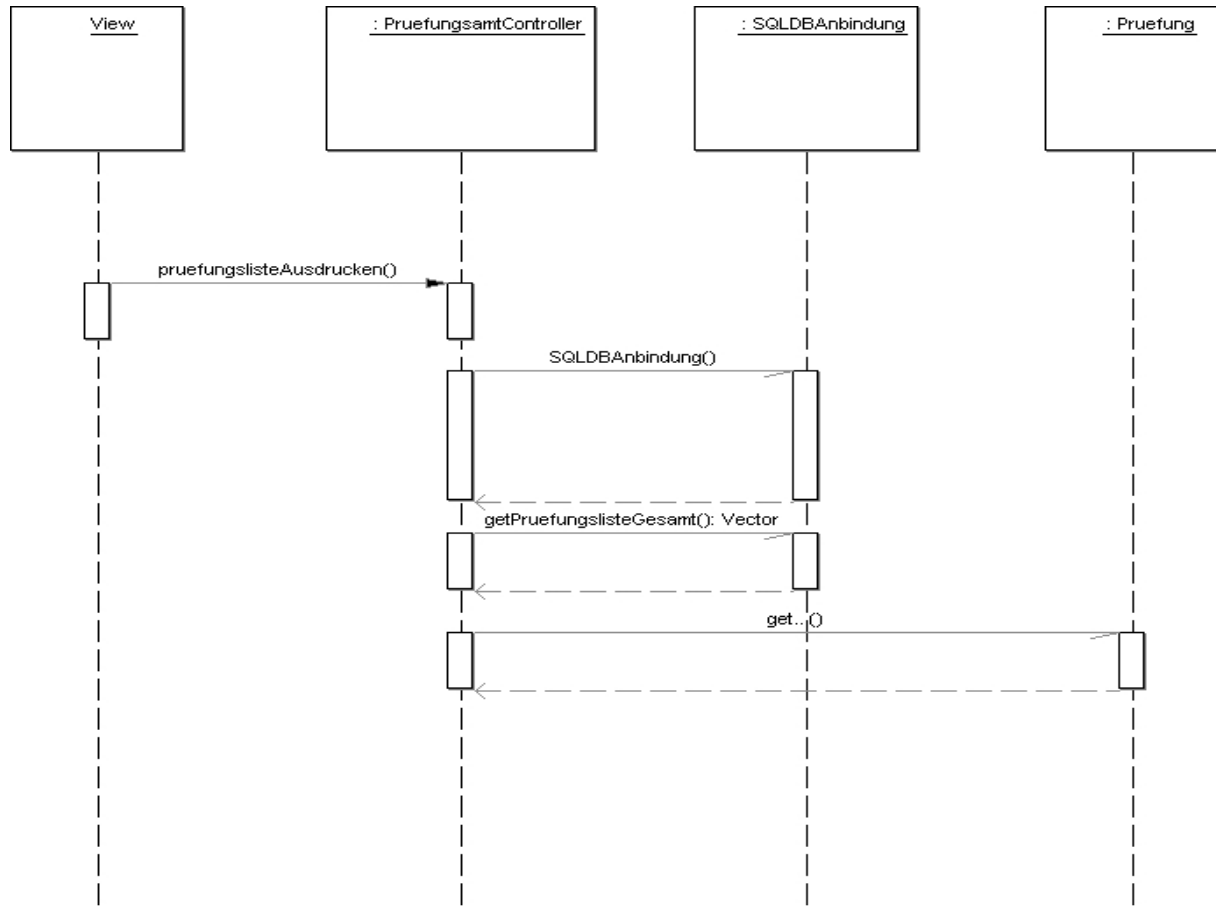
Prüfungen freischalten



Nachdem der BenutzerController via SQLAnbindung eine Liste der Prüfungen ermittelt hat, wird der PruefungsamtController erzeugt, falls der Benutzer als Prüfungsamt erkannt wurde. Danach liest der Benutzercontroller diverse Informationen der gewählten Prüfung aus. Das sind insbesondere Termin und Ort der Prüfung, die das PA wissen muss, um Kollisionen mit anderen Prüfungen zu verhindern. Hat das PA keine Einwände, schaltet der PruefungsamtController die Prüfung frei, das heißt setzt den entsprechenden Wert "freigeschaltet" auf true.

(Der dazu natürlich nötige View, bleibt in diesem Diagramm der Übersicht halber außen vor. get...() bedeutet, alle nötigen Getter werden ausgeführt)

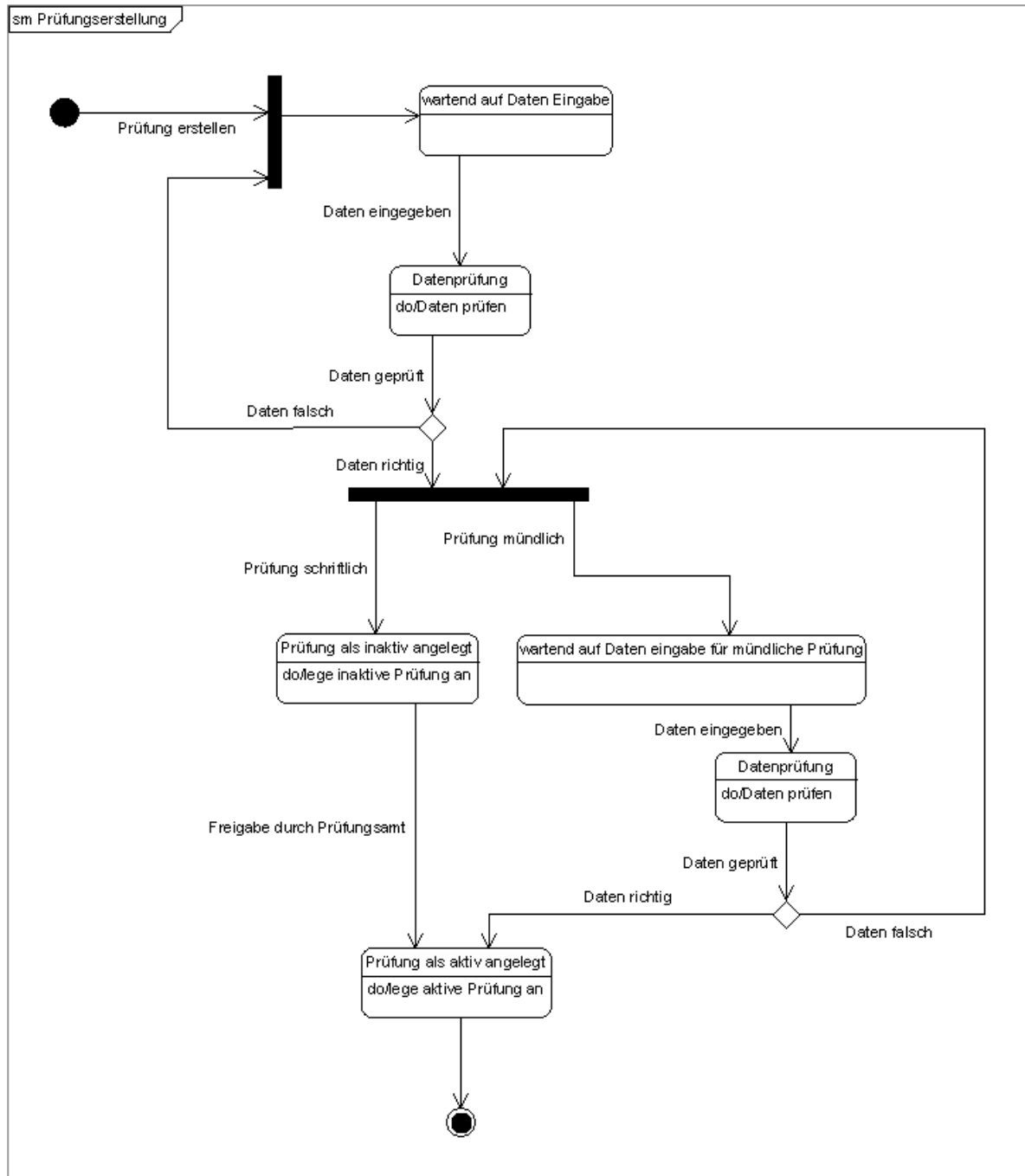
Prüfungsliste drucken



Das Diagramm geht vereinfachend davon aus, dass das Prüfungsamt bereits als solches authentifiziert und der Controller gemäß den Sequenzdiagrammen weiter oben initialisiert wurde.

Die SQL-Datenbank wird über SQLAnbindung abgefragt und eine Liste der Prüfungen angefordert. Die Elemente der Liste werden ALLE einzeln abgefragt und deren Werte bestimmt. (im Diagramm symbolisch durch ein einzelnes get...()) dargestellt)

Prüfung erstellen mit allen Sonderfällen



Der Zustandsautomat zeigt die Erstellung einer Prüfung, bis zur Speicherung der Prüfung in die Datenbank.

Dabei ist die Sicht des Dozenten betrachtet worden.

Nachdem die Anweisung zum Prüfung erstellen getroffen ist, werden Prüfungsspezifische Dateneingaben erwartet. Danach wird geprüft, ob die eingegebenen Daten korrekt sind und

**Softwaretechnik-Praktikum SS 2007**

es wird ausgewertet ob es sich um eine mündliche oder eine schriftliche Prüfung handelt. Sind die Daten unvollständig oder falsch, werden diese wiederholt zur Eingabe erwartet.

Handelt es sich um eine mündliche Prüfung so wird auf die Eingabe weiterer Daten gewartet, diese Daten sind die entsprechenden Termine, zu denen die Prüfung stattfinden kann. Nach der Eingabe dieser Daten wird abermals auf Korrektheit geprüft.

Waren die Daten nicht richtig, so wird erneut auf die Eingabe der zusätzlichen Daten der mündlichen Prüfung gewartet.

Waren die Daten korrekt so wird die mündliche Prüfung angelegt, d.h. die mündliche Prüfung wird als aktiv gespeichert.

Handelt es sich um eine schriftliche Prüfung und die Daten sind korrekt, so wird diese Prüfung angelegt, d.h. sie wird als inaktiv gespeichert.

Die schriftliche Prüfung wird als aktiv gespeichert, nachdem das Prüfungsamt diese Prüfung freigegeben hat.

(aktiv bedeutet in diesem Zusammenhang, dass die Prüfung in der Datenbank gespeichert wurde, jedoch noch nicht zur Anmeldung für Studenten freigegeben ist. Dies geschieht erst, wenn die Anmeldefrist für die Prüfung startet, dieser "Zyklus" ist im obigen Zustandsautomaten nicht enthalten)