Meilenstein 5:
Datenbankentwurf:

TABELLE	Kurs
BESCHREIBUNG	Speichert Informationen über einen Kurs
VERWALTET	Kurs, Benutzer
SCHLÜSSEL	ID: Int
FELD	Name: Varchar
FELD	Einschreibeschlüssel: Varchar
FELD	Öffentlich: Tinyint
FELD	ErstellerID: Int

TABELLE	Benutzer
BESCHREIBUNG	Speichert die Benutzer des Systems
VERWALTET	Benutzer
SCHLÜSSEL	ID: Int
FELD	Name: Varchar
FELD	Passwort: Varchar
FELD	Matrikelnummer: Int
FELD	Email: Varchar
FELD	istDozent: Tinyint

TABELLE	Rollen
BESCHREIBUNG	Speichert Rollen
VERWALTET	Rollen
SCHLÜSSEL	ID: Int
FELD	Name: Varchar
FELD	Eröffnen: Tinyint
FELD	Diskutieren: Tinyint
FELD	Bewerten: Tinyint
FELD	Löschen: Tinyint
FELD	Zulassen: Tinyint
FELD	Ändern: Tinyint
FELD	Verwalten: Tinyint

TABELLE	BenutzerKurs
BESCHREIBUNG	Speichert die Zusammenhänge zwischen Benutzern, Kursen und Rollen
VERWALTET	Benutzer, Kurs
SCHLÜSSEL	ID: Int
FELD	KursID: Int
FELD	BenutzerID: Int

TABELLE	KursRolle
BESCHREIBUNG	Speichert die Zusammenhänge zwischen Kursen und Rollen
VERWALTET	Kurs, Rolle
SCHLÜSSEL	ID: Int
FELD	KursID: Int
FELD	BenutzerID: Int

Methoden:

Kurs:

Um Attribute zu bekommen oder ändern, werden getter und setter Methoden verwendet, diese sind im Durchstich aber nicht aufgeführt.

Kurs(name:String, Schluessel:String, Oeffentlich:boolean, ersteller:Benutzer)

Hier wird ein Kurs angelegt. Dazu muss ein Name übergeben werden, welcher dann dem Kursnamen entspricht. Der Schlüssel wird dazu übergerben um unbefugten das einschreiben zu verbieten.Der Boolean Oeffentlich muss dann bevor es in der Datenbank gespeichert wird noch “umgecastet” werden, da die Datenbank kein boolean aktzeptiert Der Benutzer der den Kurs anlegt, ist der Ersteller und bekommt dann auch alle Rechte (Dozent)

Vorbedingung:	Ersteller muss registrierter Benutzer sein. Kursname muss “Unique” (also nicht schon vergeben) sein.
Nachbedingung:	keine

benutzerHinzufuegen (student:Benutzer)

Diese Methode ist dafür da um alle Benutzer die sich korrekt eingeschrieben haben, in den Kurs hinzuzufügen. Das könnte manuell von einem Verantwortlichen gemacht werden, oder ein automatisch geschehen.

Vorbedingung:	Kurs muss existieren (erstellt worden sein)
Nachbedingung:	Kurs verwaltet Benutzer

benutzerLoeschen (student:Benutzer)

Um einen Benutzer zu löschen. Muss manuell von einem Verantworlichen gemacht werden.

Vorbedingung:	Benutzer muss existieren.
Nachbedingung:	Alle Referenzen und Pointer gehen somit verloren

rolleHinzufuegen (rolle : Rolle)

Kreiert eine neue Rolle in der Datenbank.

Vorbedingung: Kurs muss existieren (da auf Kurs verwaltet)
Nachbedingung: Kurs verwaltet Rolle.

rolleEntfernen(rolle: Rolle)

Löscht Rolle in der Datenbank

Vorbedingung: Rolle muss existieren. Darf nicht einzige Rolle sein.
Nachbedingung: Alle Referenzen und Pointer gehen verloren. Benutzer mit dieser Rolle bekommt Defaultrolle “Student”

loeschen ()

Löscht Kurs und schließt Veranstaltung

Vorbedingung: Sicherheitsabfrage bestätigt
Nachbedingung: Alle Daten werden gelöscht

einschreiben (student: Benutzer, schluessel: String)

Verifizierung eines Benutzer. Prüft ob Benutzer Zugang zum Kurs bekommt

Vorbedingung: Benutzer muss existieren. Kurs muss existieren
Nachbedingung: Benutzer bekommt Defaultrolle “Student” zugeteilt

Benutzer:

Auch hier sind Getter und Setter unerwähnt.

Benutzer(name:String, matrikelnummer:int, passwort:String, email:String, istDozent:boolean)

Legt neuen Benutzer in der Datenbank an. istDozent muss ebenfalls als Int der Datenbank übergeben werden (siehe Oeffentlich). Matrikelnummer wird aufgrund der Länge als Int übergeben, maximale Stellenlänge liegt bei 10 Stellen (allg. aber ausreichend).

Vorbedingung: Name, E-Mail, Matrikelnummer muss Unique sein.
Nachbedingung: Benutzer angelegt. Noch ohne Schnittmengen

loeschen ()

Löscht Benutzer aus der Datenbank. Wird vom Benutzer gesteuert.

Vorbedingung: Sicherheitsabfrage muss bestätigt werden.
Nachbedingung: Alle Refenrenzen und Pointer gehen verloren.

Rolle:

Hat nur Getter und Setter und den Konstruktor zum Erstellen einer Rolle:

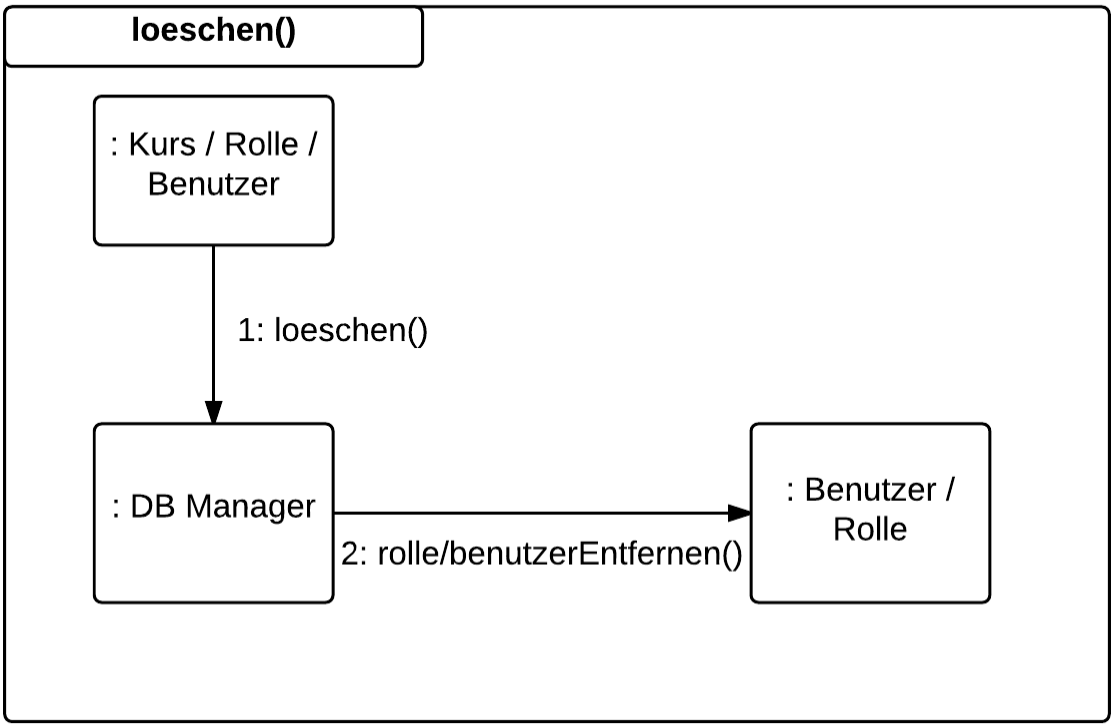
Rolle(name: String, darfDiskussionEroeffnen, darfDiskutieren, darfPostBewerten, darfPostLoeschen, darfZuKursZulassen, darfSkriptAendern, darfRecheVerwalten : boolean)

Rolle wird angelegt, mit true oder false Attributen. True entspricht “darf tun”. False entspricht “darf nicht tun”

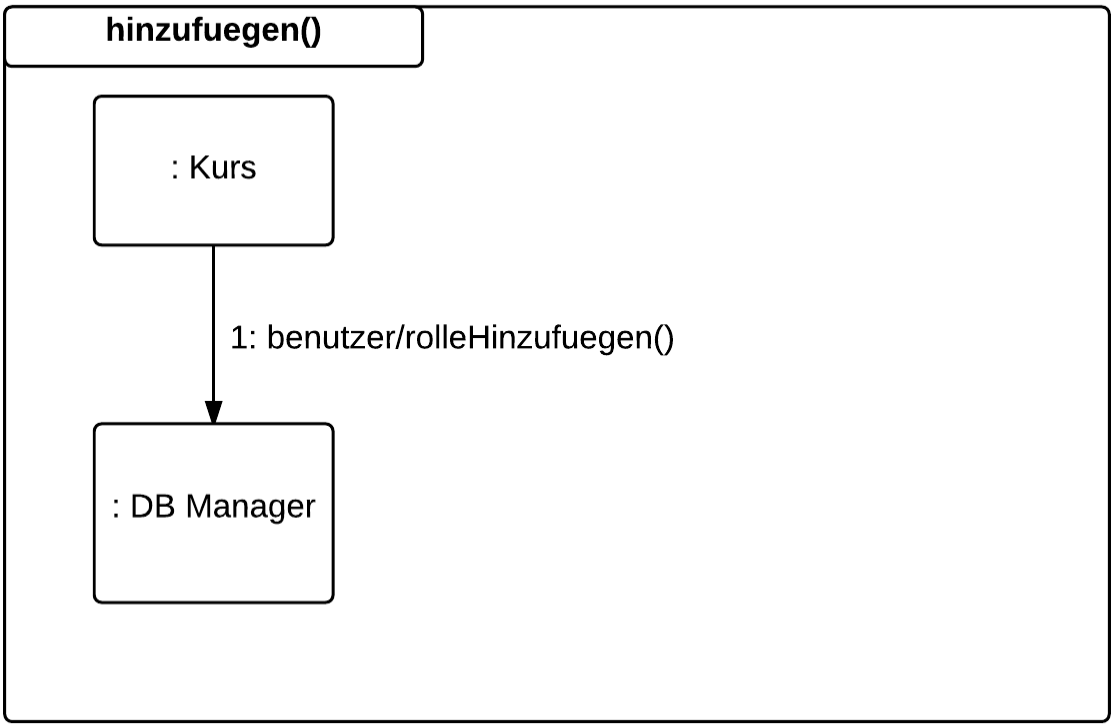
Vorbedingung: Rollenname muss Unique sein
Nachbedingung: Rolle wurde angelegt. Noch ohne Schnittmengen

Kommunikationsdiagramme

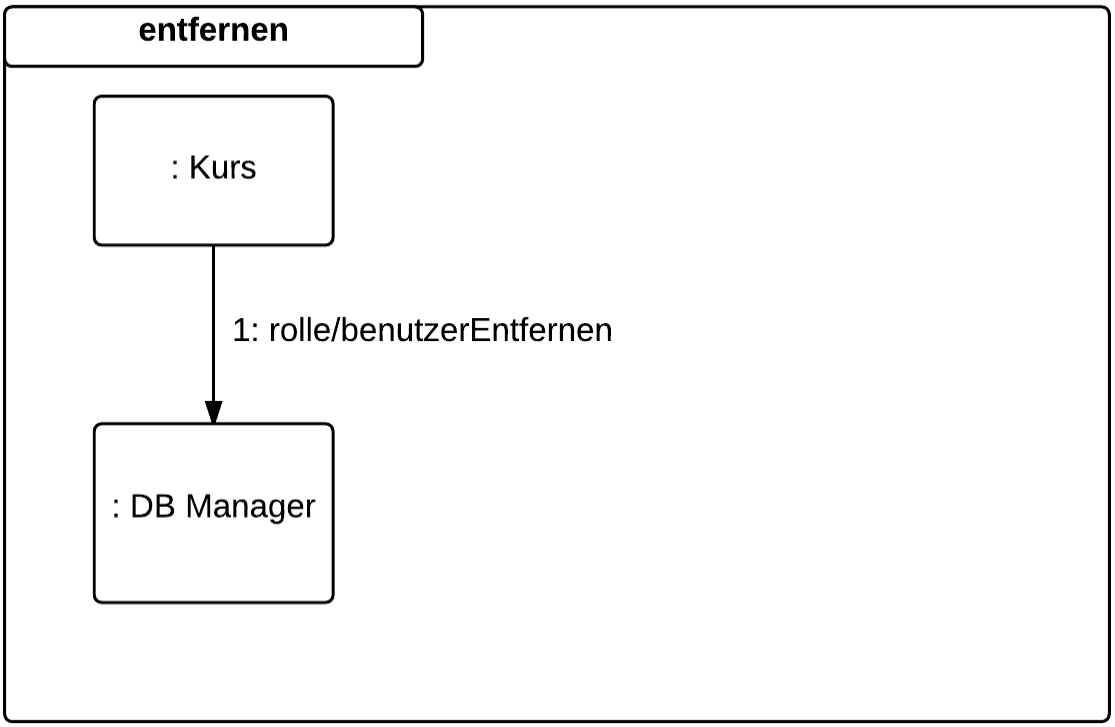
Kurs / Benutzer / Rolle löschen:



Rolle / Benutzer hinzufügen:



Rolle / Benutzer entfernen:



Klassendiagramm

Im Klassendiagramm werden alle Methoden, Klassen und Attribute/Parameter dargestellt und deren Abhängigkeit zueinander. Der Ablauf wird dabei vereinfacht dargestellt.
Der Kasten entspricht dabei einer Klasse. Unterhalb des Strichs sind alle notwendigen Methoden, und oberhalb somit die Attribute mit ihrem Datentyp (hier: Java Datentypen)