

# ENTWURF

## Aufgabenblatt 3 – Entwurfsdokument

### Softwarepraktikum Sommersemester 2015

Gruppenmitglieder	Matrikelnummer
Christoph Jachmann	2820042
Hannes Fischer	2862505
Lena Maier	2870610
Kevin Kruse	2876685
Michael Hackenberg	2908184

# Inhaltsverzeichnis

1. Allgemeine Ziele des Entwurfs .....	1
2. Überblick über die Projektarchitektur .....	2
3. Beschreibung der Komponenten und ihrer Schnittstellen .....	4
4. Verteilung der Anwendungskomponenten .....	12
5. Beschreibung der Datenhaltung.....	13
6. Benutzerrechte und deren Verwaltung.....	15
Abkürzungsverzeichnis.....	17

# **1. Allgemeine Ziele des Entwurfs**

Bei dem Softwareentwurf wird die technische Lösungsstruktur der Lernplattform entworfen und dabei festgelegt, wie die Realisierung der Systemfunktionalität erfolgen soll.

Um die Anforderungen des Kunden aufzudecken, wurde diesem zunächst ein Fragenkatalog mit kategorisierten Fragen zu dem Softwareprojekt gestellt. In einem ausführlichen Kundengespräch wurden dann sämtliche Fragen beantwortet und die Anforderungen dokumentiert.

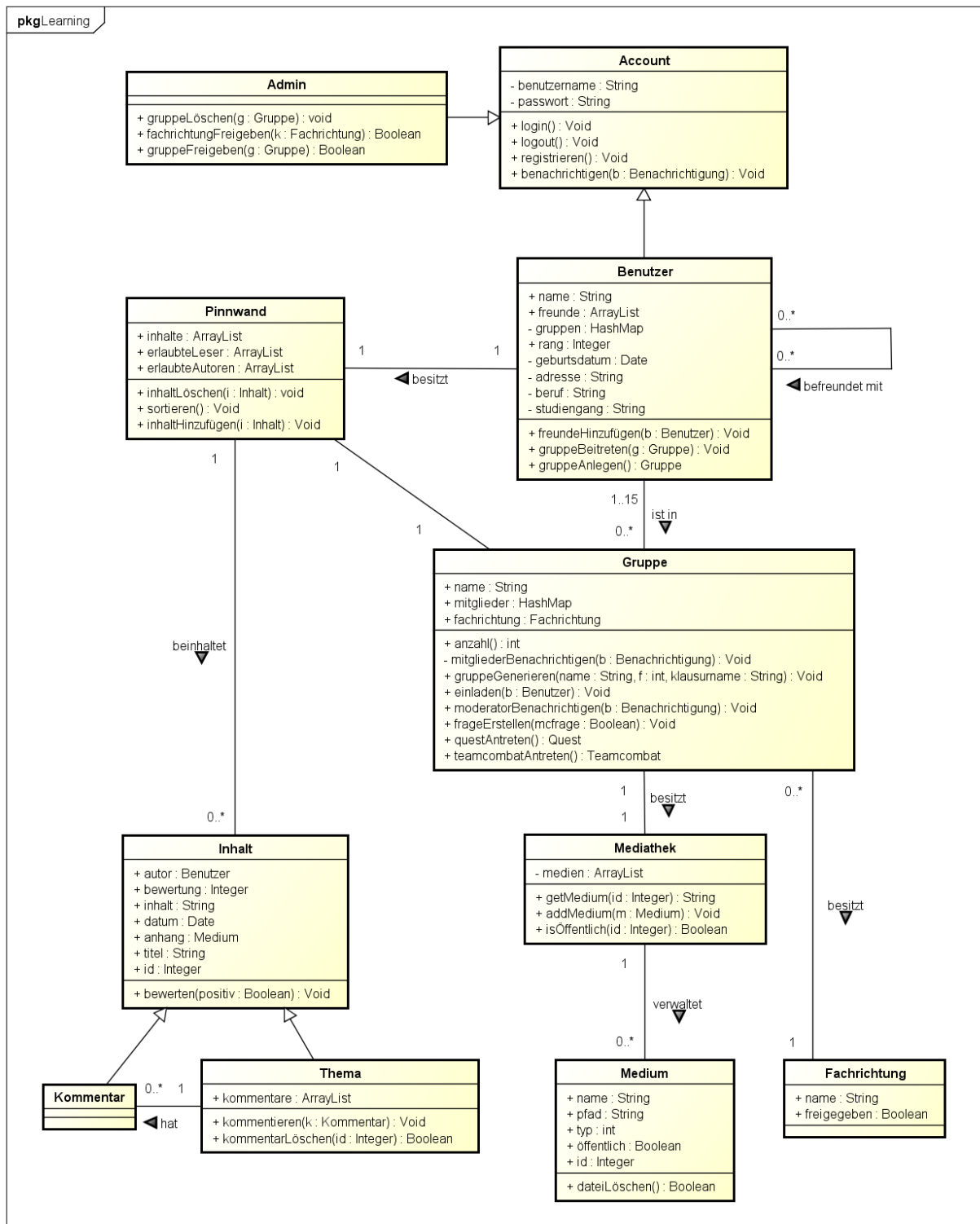
Ausgehend von der Anforderungsanalyse und Spezifikation werden im Entwurf die Infrastruktur und Architektur der Software festgelegt, zudem werden Komponenten und Klassen entworfen. Das Gesamtsystem wird in Teilsysteme zerlegt und die Wechselwirkungen zwischen den Teilsystemen werden beschrieben.

Bei der Erstellung des Entwurfs wurde große Sorgfalt auf die Erfüllung der funktionalen Anforderungen gelegt. Große Zusammengehörigkeit der Bestandteile einer Komponente und geringe Koppelung zwischen den Komponenten als Qualitätskriterien führen zu erleichtertem Verständnis und besserer Wartbarkeit.

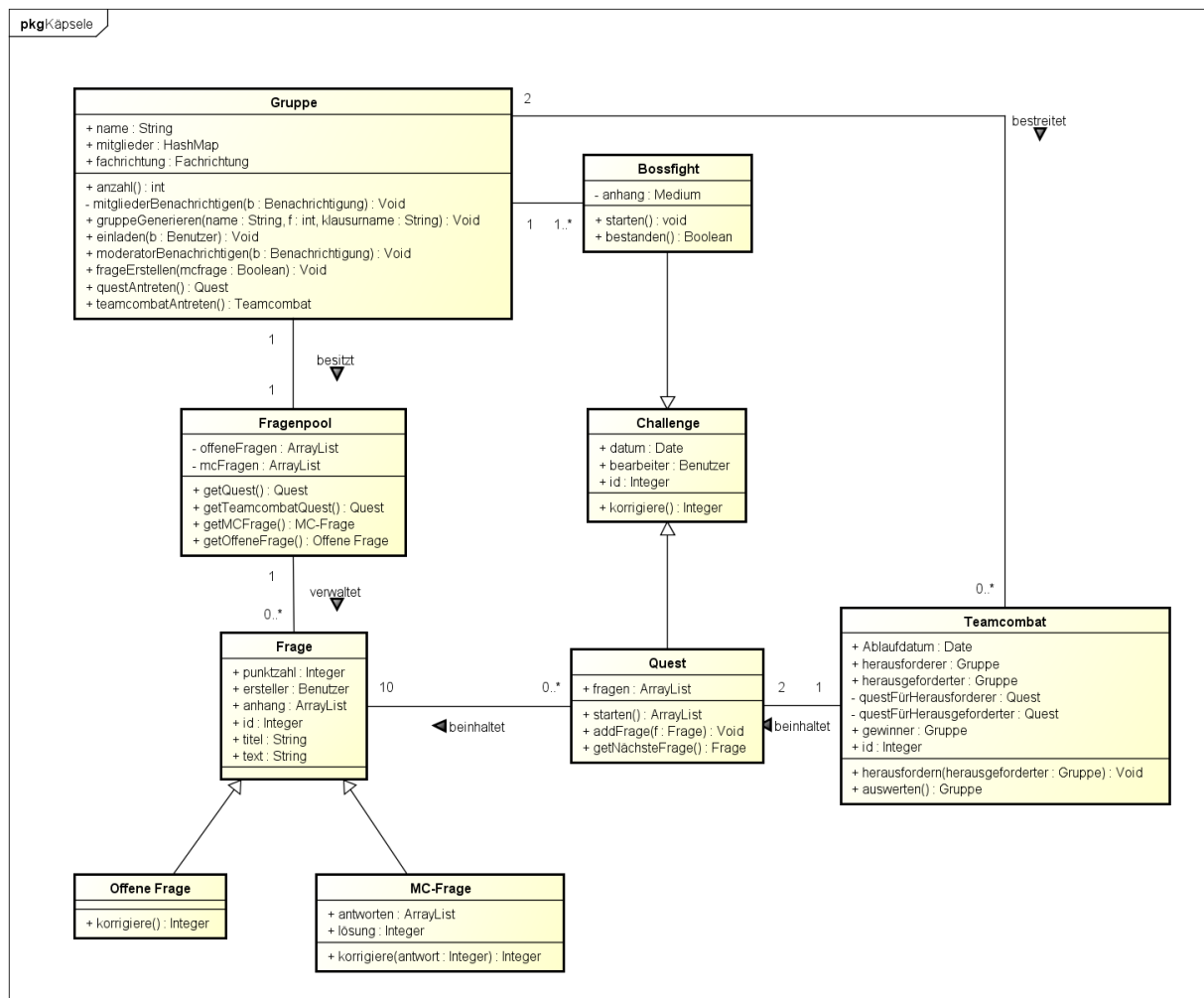
## 2. Überblick über die Projektarchitektur

Im Folgenden werden die Klassen im Paket „Learning“ detaillierter dargestellt. Dabei wird das Paket aus Gründen der Übersichtlichkeit in zwei Klassendiagramme dargestellt, wobei der Fokus einmal auf der Verwaltung und einmal auf den Gamifications liegt.

Klassendiagramm: *Verwaltungsklassen des Pakets „Learning“*



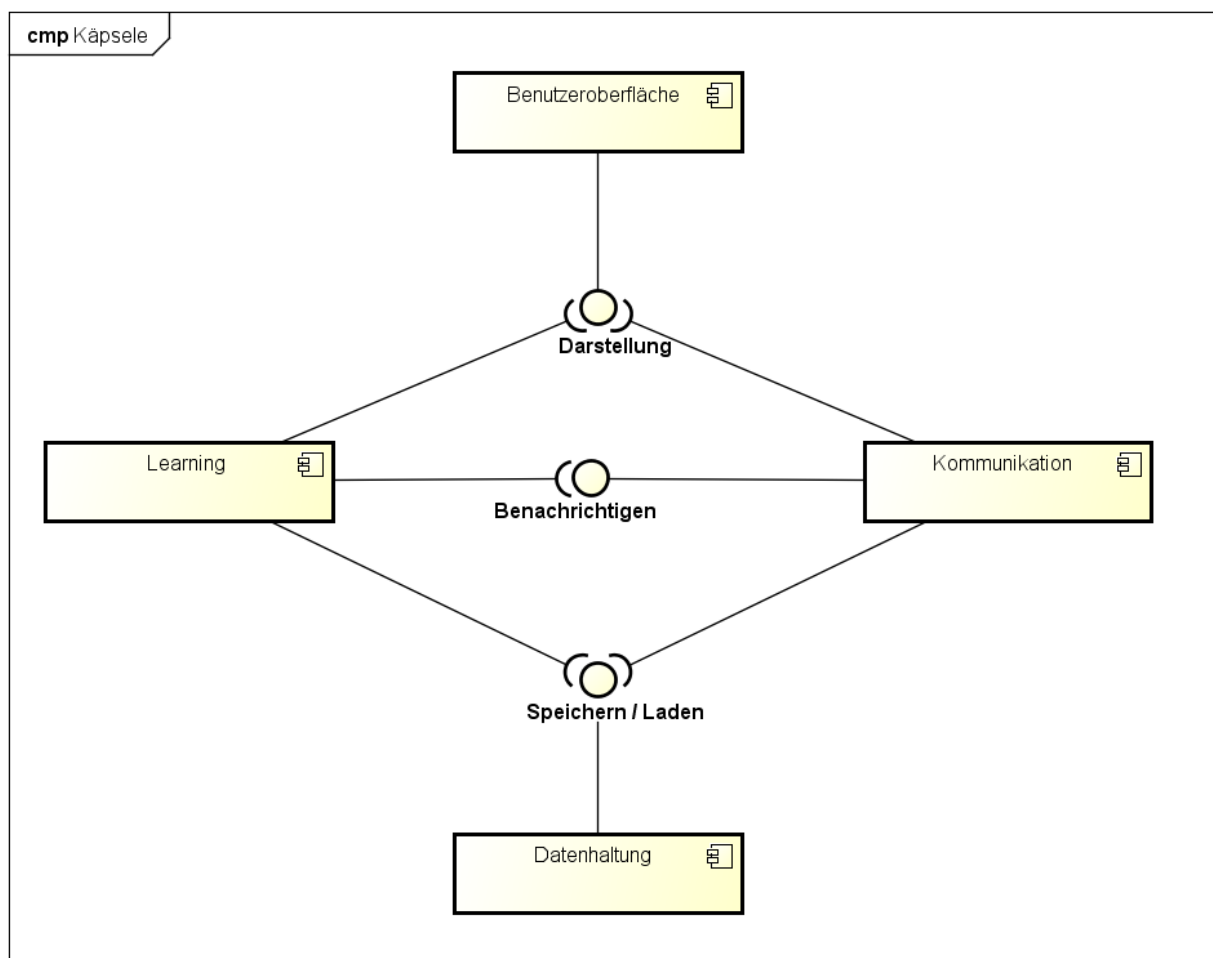
## Klassendiagramm: Gamificationklassen des Pakets „Learning“



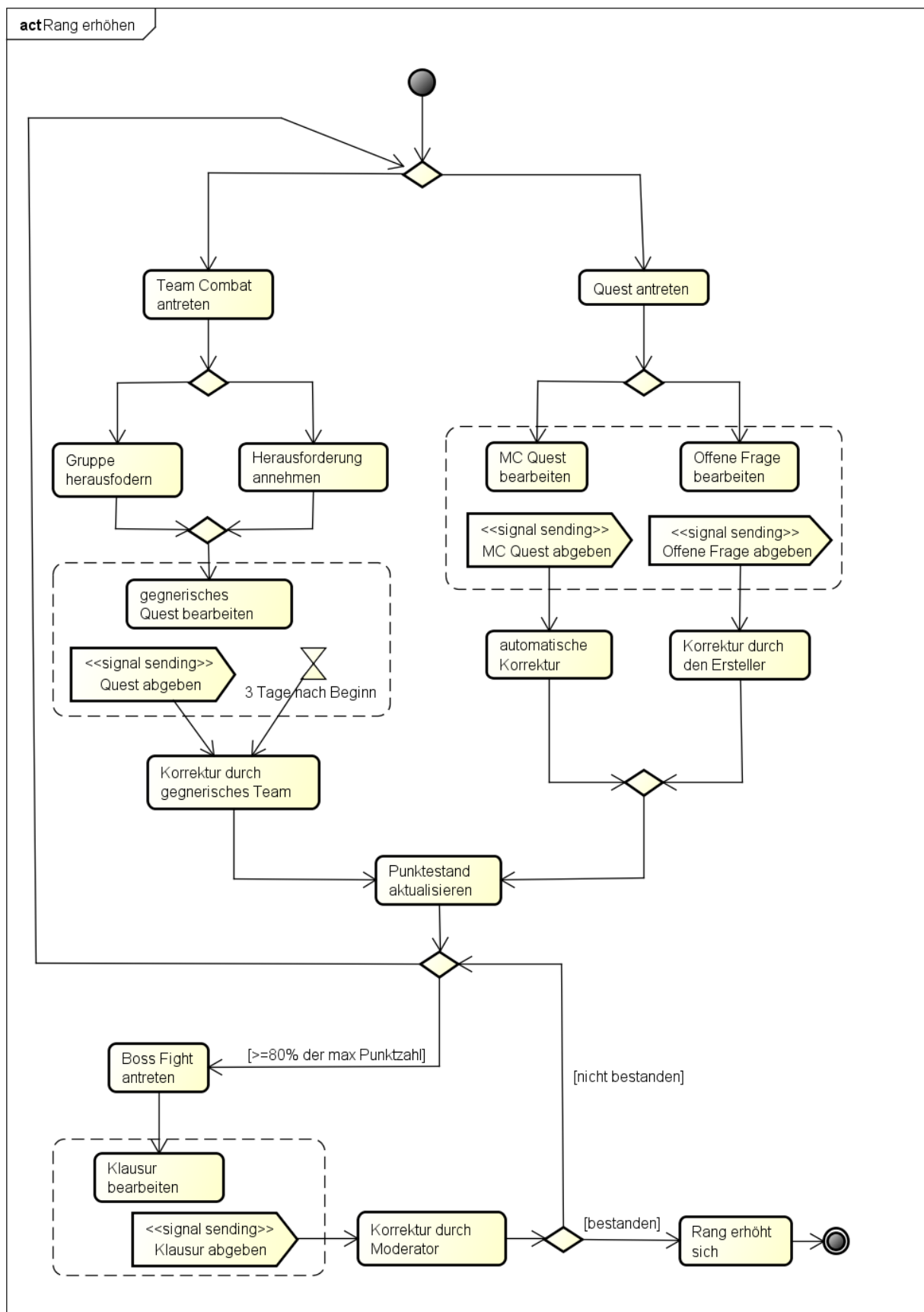
### 3. Beschreibung der Komponenten und ihrer Schnittstellen

Die Architektur der Software wird grundsätzlich gemäß dem Model-View-Controller-Prinzip in drei Bereiche aufgeteilt. Das Paket „Datenhaltung“ ist dabei für das Speichern und Laden sämtlicher Daten zuständig und stellt dafür eine entsprechende Schnittstelle zur Verfügung. Diese wird von den beiden Controller-Paketen implementiert. Davon ist das Paket „Learning“ für die grundsätzliche Verwaltung sämtlicher Funktionen (insb. Benutzer und Gruppen) zuständig. Es greift dabei auf die Schnittstelle des Pakets „Kommunikation“ zu, die insbesondere für den Emailversand und die Benachrichtigung von Benutzern Klassen zur Verfügung stellt. Schlussendlich werden die Informationen aus den beiden Controllerpaketen durch entsprechende Klassen im Paket „Benutzeroberfläche“ aufbereitet und dargestellt. Dafür wird von der Viewklasse eine entsprechende Schnittstelle zur Verfügung gestellt.

Komponentendiagramm: *Pakete der Software „Käpsele“*



## Aktivitätsdiagramm: *Rang erhöhen*



Beschreibung des Aktivitätsdiagramms *Rang erhöhen*:

Damit ein Benutzer seinen Rang erhöhen kann muss er ein Modul, also seine Lerngruppe erfolgreich abschließen. Dafür sind mehrere Schritte notwendig. Zunächst kann der Benutzer seinen Punktestand innerhalb der Gruppe mit Quests und Team Combats erhöhen.

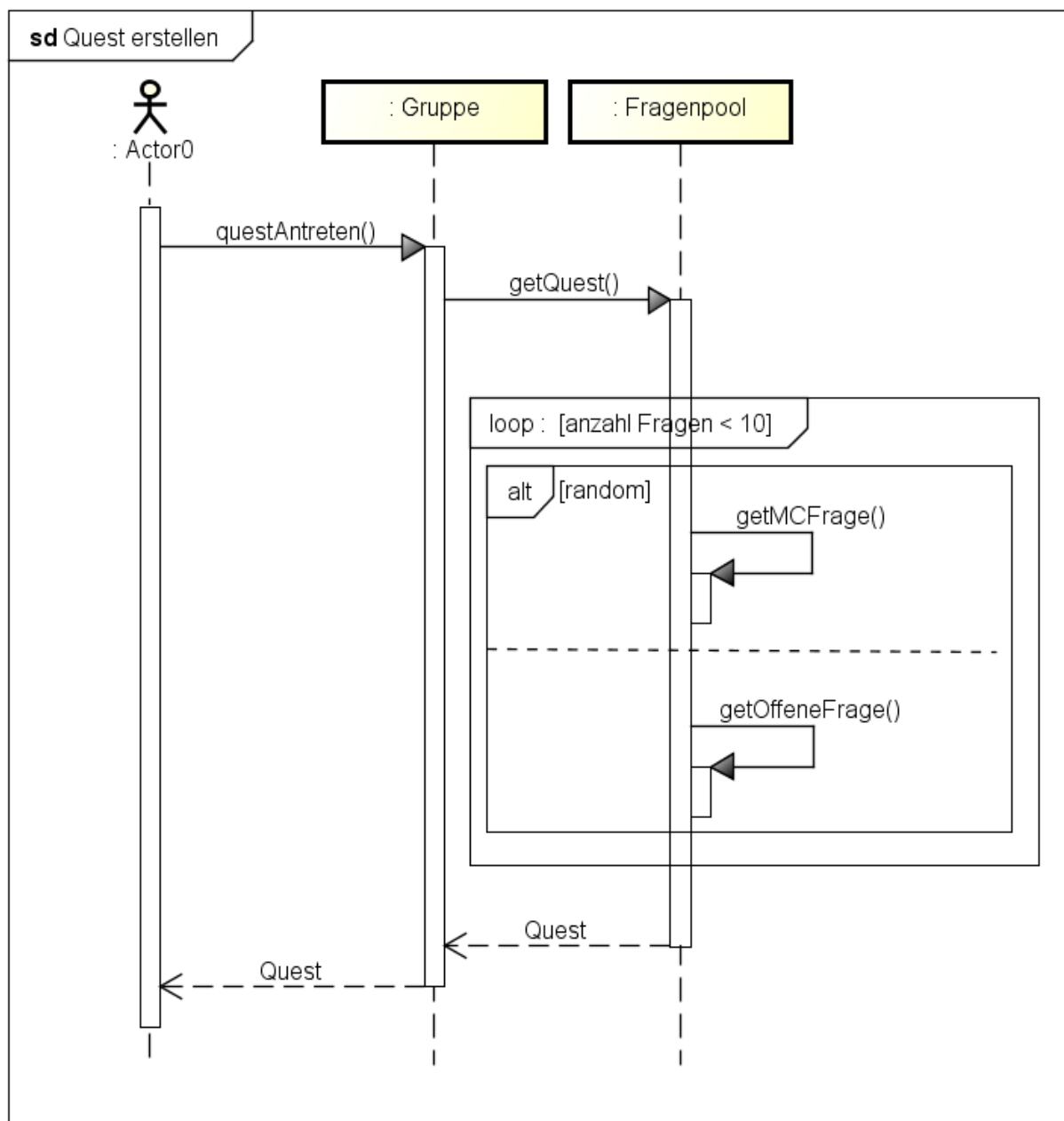
Quests beinhalten Multiple-Choice Fragen und offene Fragen. Hierbei werden MC Quests automatisiert korrigiert, die offenen Fragen werden jeweils vom Ersteller korrigiert.

Um ein Team Combat beginnen zu können muss entweder eine andere Gruppe herausgefordert werden oder es gibt bereits eine Anfrage einer anderen Gruppe die angenommen werden kann. Jedes Team erhält nun jeweils aus dem gegnerischen Fragenpool eine zusammengestellte Quest. Zur Bearbeitung der Quest bleiben maximal 3 Tage Zeit. Korrigiert werden die Quests von der gegnerischen Gruppe.

Der Benutzer muss die Team Combats oder Quests solange wiederholen bis er mindestens 80% der innerhalb der Gruppe zu erreichenden Punktzahl erreicht hat, so dass er im Boss Fight antreten kann. Der Boss Fight besteht aus einer Altklausur früherer Jahre. Der Moderator der Gruppe korrigiert die Klausuren mithilfe von Musterlösungen. Falls der Benutzer die Klausur besteht erhält er die restlichen Punkte die es in der Gruppe gibt, schließt das Modul ab und steigt einen Rang auf. Falls er die Klausur nicht besteht kann der Benutzer direkt wieder in einem Boss Fight antreten oder vorher noch ein paar Quests oder Team Combats zur Übung machen.



## Sequenzdiagramm: *Quest erstellen*



Beschreibung des Sequenzdiagramms *Quest erstellen*:

**questAntreten():** Die Methode `questAntreten()` führt die Methode `getQuest()` aus und liefert eine Quest zurück.

**getQuest():** `getQuest()` instanziert eine neue Quest und liefert diese zurück.

**Repeat:** Damit die Quest 10 Fragen beinhaltet, wird der Vorgang 10 mal wiederholt.

**Alternative:** Durch Zufall wird ausgewählt ob eine MC-Frage oder eine offene Frage zur Quest hinzugefügt werden soll.

Entweder:

getMCFrage():

Die Methode liefert eine MC-Frage zurück, die zur ArrayList hinzugefügt wird.

Oder:

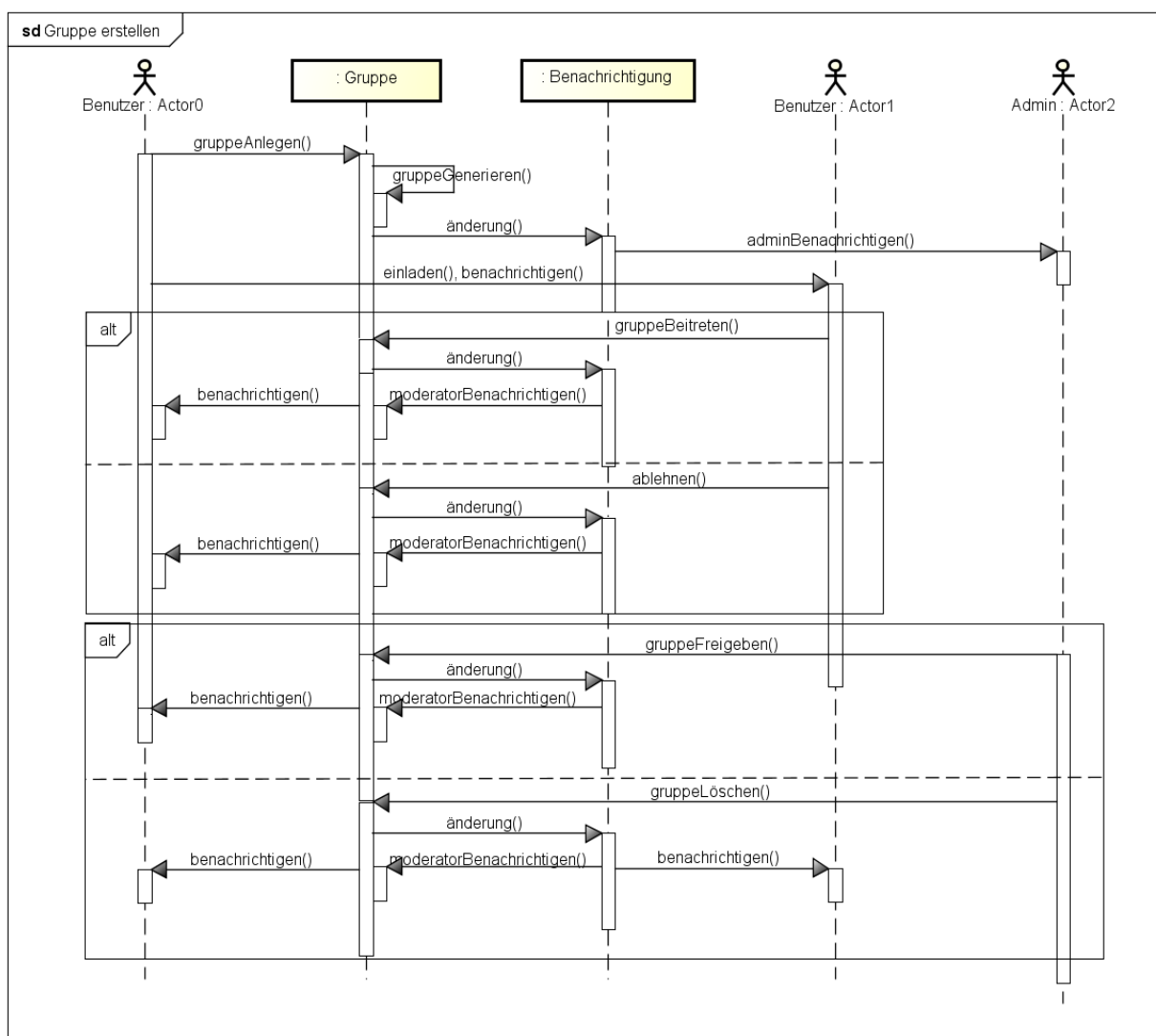
getOffeneFrage():

Die Methode liefert eine offene Frage zurück, die zur ArrayList hinzugefügt wird.

Quest:

Eine Quest besteht aus 10 Fragen, die in einer ArrayList gespeichert sind.

Sequenzdiagramm: *Gruppe erstellen*



Die Klasse Benachrichtigung befindet sich im Paket Kommunikation (siehe Komponentendiagramm: *Pakete der Software „Käpsele“*) und ist deshalb nicht im Klassendiagramm zu finden.

### Ablaufbeschreibung des Sequenzdiagramms *Gruppe erstellen*:

gruppeAnlegen():	Zum Anlegen einer Gruppe legt der Benutzer einen Gruppennamen, Fachrichtung und Klausurnamen fest
gruppeGenerieren():	Es wird das Profil der Gruppe generiert (Konstruktor)
ändern():	Neue Gruppe wird registriert
adminBenachrichtigen():	Der Admin wird über die neue Gruppe benachrichtigt
einladen(), benachrichtigen():	Beim Anlegen der Gruppe können immer direkt Benutzer eingeladen werden, die somit eine Benachrichtigung erhalten

#### Entweder:

gruppeBeitreten():	Der eingeladene Benutzer kann die Einladung annehmen
ändern():	Neues Gruppenmitglied wird registriert
moderatorBenachrichtigen():	Es wird eine Info an die Klasse Gruppe ausgegeben, dass der Moderator benachrichtigt werden muss
benachrichtigen():	Der Moderator wird jetzt mit Hilfe seiner persönlichen benachrichtigen() Methode über das neue Gruppenmitglied informiert

#### Oder:

ablehnen():	Der eingeladene Benutzer kann die Einladung ablehnen
ändern():	Ablehnung der Einladung zur Gruppe wird registriert
moderatorBenachrichtigen():	Es wird eine Info an die Klasse Gruppe ausgegeben, dass der Moderator benachrichtigt werden muss
benachrichtigen():	Der Moderator wird jetzt mit Hilfe seiner persönlichen benachrichtigen() Methode über die Ablehnung der Einladung informiert

#### Entweder:

gruppeFreigeben():	Der Administrator gibt die Gruppe frei
ändern():	Freigabe der Gruppe wird registriert
moderatorBenachrichtigen():	Es wird eine Info an die Klasse Gruppe ausgegeben, dass der Moderator benachrichtigt werden muss
benachrichtigen():	Der Moderator wird jetzt mit Hilfe seiner persönlichen benachrichtigen() Methode über die Freigabe der Gruppe informiert

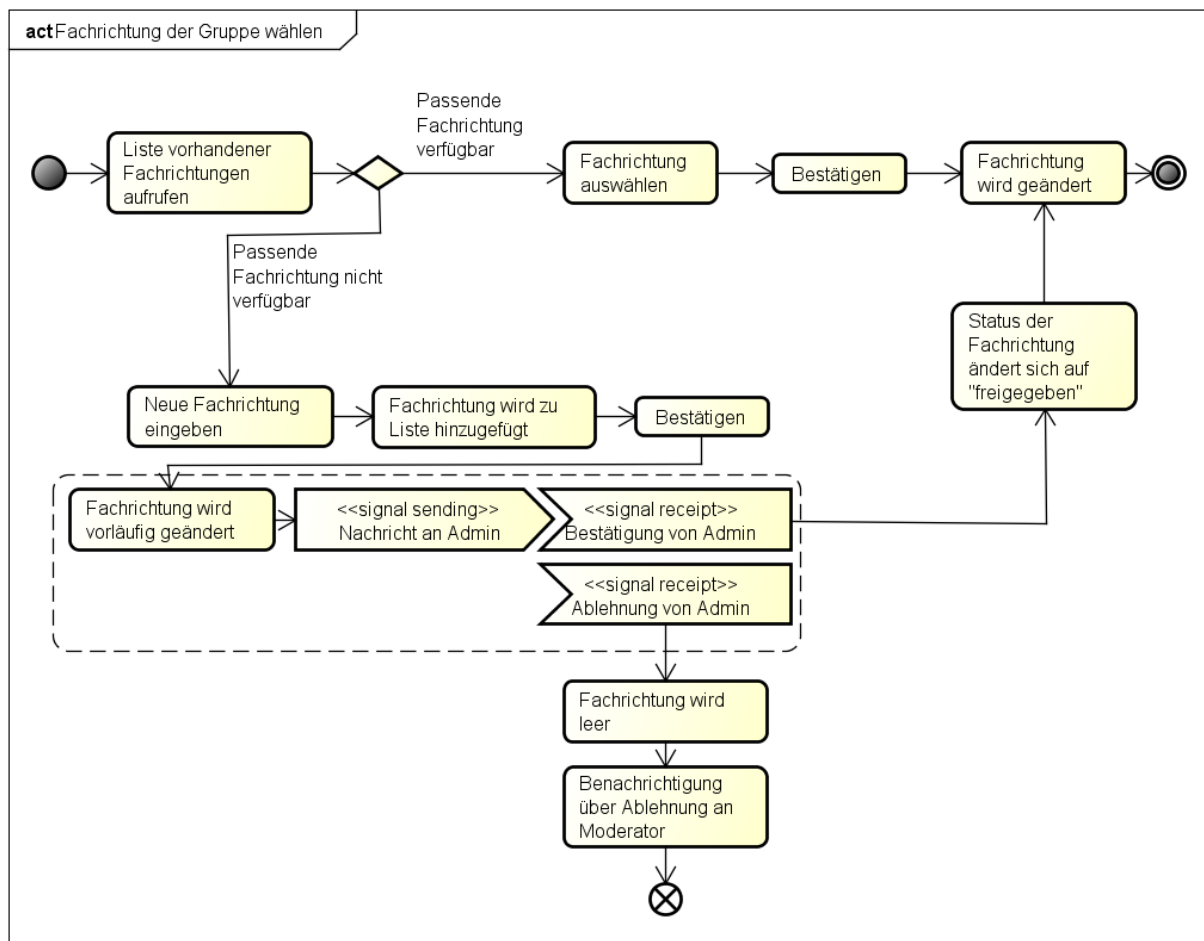
#### Oder:

gruppeLöschen():	Der Administrator lehnt die Gruppe ab, die somit gelöscht wird
ändern():	Gruppe wurde gelöscht wird registriert
moderatorBenachrichtigen():	Es wird eine Info an die Klasse Gruppe ausgegeben, dass der Moderator benachrichtigt werden muss

benachrichtigen(): Der Moderator wird jetzt mit Hilfe seiner persönlichen benachrichtigen() Methode über die Freigabe der Gruppe informiert

benachrichtigen(): Es werden zusätzlich alle Gruppenmitglieder darüber informiert, das die Gruppe gelöscht wurde

### Aktivitätsdiagramm: *Fachrichtung der Gruppe wählen*



#### Beschreibung des Aktivitätsdiagramms *Fachrichtung der Gruppe wählen*:

Eine Gruppe hat eine Fachrichtung um später mehrere Gruppen in einer Kategorie zu ordnen, um nach Gruppen einer bestimmten Fachrichtung suchen zu können. Dafür wird eine Liste vorhandener Fachrichtungen mit allen Optionen angezeigt.

Falls die passende Fachrichtung verfügbar ist, kann der Benutzer diese in der Liste auswählen. Danach muss er Nutzer seine Auswahl bestätigen.

Falls die passende Fachrichtung nicht verfügbar ist, kann der Benutzer eine neue Fachrichtung eingeben. Die Fachrichtung wird zur Liste aller Fachrichtungen hinzugefügt, allerdings bleibt der Status bei „nicht freigegeben“ (Fachrichtung.freigegeben = false). Nach einer erneuten Bestätigung wird die

Fachrichtung vorläufig geändert und eine Benachrichtigung an den Admin gesendet. Sobald der Admin seine Bestätigung zur neuen Fachrichtung gibt, ändert sich der Status der Fachrichtung auf freigegeben und wird somit in der Auswahlseite auf der Website angezeigt.

Falls der Admin jedoch die neue Fachrichtung ablehnt, wird das Attribut der Fachrichtung wieder geleert und eine Benachrichtigung über die Ablehnung an den Moderator der Gruppe gesendet. Danach wird der Prozess abgebrochen und er muss erneut gestartet werden um die Fachrichtung zu ändern.

Anschließend wird das Attribut „Fachrichtung“ der Gruppe zur ausgewählten Fachrichtung geändert und der Prozess ist beendet.

## 4. Verteilung der Anwendungskomponenten

Sowohl die Anwendung „Käpsele“ selbst als auch deren Datenhaltung in einer SQL-Datenbank (siehe Kapitel 5) werden auf dem selben Server laufen, jedoch intern voneinander nach dem Model-View-Controller-Prinzip getrennt agieren (siehe Kapitel 3).

Um einen unterbrechungsfreien und reibungslosen Betrieb der Software gewährleisten zu können, ist ein grundlegendes Betriebssystem mit Java Runtime Environment der Version Java 1.7 zwingend voraussetzend.

Zudem wird empfohlen, aus Datensicherheitsgründen zu jeder Zeit ein RAID 1 bei der Datenspeicherung zu verwenden, um einem Datenverlust bei Festplattenausfall vorzubeugen. Darüber hinaus sollte in geringen Zeitintervallen eine komplette / inkrementelle Sicherung der Daten an einen externen Backupserver vorgenommen werden.

Um die im Lastenheft festgehaltenen Qualitätsmerkmale einhalten zu können, ist insbesondere ausreichend Arbeitsspeicher, Prozessorgeschwindigkeit und Bandbreite für ein performantes Verhalten der Website zwingend nötig. Da in der aktuellen Phase noch nicht abgeschätzt werden kann, wie groß die Nutzerzahlen zu Stoßzeiten langfristig sein werden, sollte eine modulare Ergänzung der Serverleistung möglich sein.

## 5. Beschreibung der Datenhaltung

Sämtliche in den Objekten der Klassen vorhandenen Attribute werden in einer SQL-Datenbank gespeichert.

Für jede Klasse wird dabei, so wie sie im Klassendiagramm dargestellt ist, eine Tabelle erstellt. Jede Instanz einer Klasse wird dann in der dazugehörigen Tabelle dargestellt. Ausnahmen stellen die Generalisierungen dar. Diese sind abstrakt und werden weder instanziiert noch gespeichert. Zudem werden Klassen, die in 1:1 Beziehungen stehen gemäß ihrer Entitätsbeziehungen in einer Tabelle gemeinsam gespeichert (z.B. Pinnwandattribute in der Benutzertabelle).

An dieser Stelle werden beispielhaft ausgewählte Klassen des Pakets „Learning“ in Ihrer Tabellenform dargestellt:

**Benutzer (Benutzername, Passwort, Name, Freunde, Gruppen, Rang, Geburtsdatum, Adresse, Beruf, Studiengang, Inhalte, ErlaubteLeser, ErlaubteAutoren)**

Das Passwort wird selbstverständlich gehasht in der Tabelle abgespeichert. Alle Attribute, die im Klassendiagramm vom Datentyp String sind werden als VARCHAR abgespeichert. Attribute vom Datentyp ArrayList oder HashMap werden als BLOB abgespeichert. Alle anderen Attribute behalten ihren Datentyp bei.

Die Einträge der Listen von „Freunde“, „ErlaubteLeser“ und „ErlaubteAutoren“ sind Fremdschlüssel und verweisen wieder auf die Tabelle Benutzer. Die Einträge der Hashmap, die für das Attribut „Gruppen“ gespeichert wird, verweisen als Fremdschlüssel auf die Tabelle „Gruppe“. Die Einträge der Liste von „Inhalten“ verweisen auf die Tabelle „Inhalt“.

**Gruppe (Name, Mitglieder, Fachrichtung)**

Name wird als VARCHAR abgespeichert. Die Liste der Mitglieder wird als BLOB gespeichert. Von der Fachrichtung wird der Name gespeichert, der als Fremdschlüssel dient und auf die Tabelle „Fachrichtung“ verweist.

**Quest (ID, Datum, Bearbeiter, Fragen)**

Die ID ist ein generisch erzeugter Integer. Vom Bearbeiter wird nur der Benutzername gespeichert und verweist als Fremdschlüssel auf die Tabelle „Benutzer“. Die Liste der Fragen wird als BLOB gespeichert.

**Bossfight(ID, Datum, Bearbeiter, Anhang)**

Die ID ist ein generisch erzeugter Integer. Vom Bearbeiter wird nur der Benutzername gespeichert und verweist als Fremdschlüssel auf die Tabelle „Benutzer“. Für den Anhang wird die ID gespeichert, die dann auf als Fremdschlüssel auf die Tabelle „Medium“ verweist.

**Teamcombat(ID, Ablaufdatum, Herausforderer, Herausgeforderter, QuestFürHerausforderer, QuestFürHerausgeforderter, Gewinner)**

Die ID ist ein generisch erzeugter Integer. Von den Attributen „Herausforderer“, „Herausgeforderter“ und „Gewinner“ werden jeweils die Namen der Gruppen gespeichert. Diese verweisen dann als Fremdschlüssel auf die Tabelle „Gruppe“. Für „QuestFürHerausforderer“ und „QuestFürHerausgeforderter“ werden die IDs der Quests gespeichert und verweisen als Fremdschlüssel auf die Tabelle „Quest“.

**Fachrichtung(Name, Freigegeben)**

Um Fachrichtungen abspeichern zu können wurde eigens eine Entitätsklasse angelegt, die keine Funktionen hat und einzig und allein der Datenhaltung dient. Das Attribut „Name“ wird als VARCHAR gespeichert, das Attribut „Freigegeben“ als Boolean.



## 6. Benutzerrechte und deren Verwaltung

In der Anwendung existieren zwei Rollen: Es gibt Administratoren und Benutzer. Zudem können Benutzer die Rolle des Moderators einnehmen.

Der Administratoraccount dient in der Anwendung ausschließlich zur administrativen Arbeit, Benutzerfunktionen stehen ihm nicht zur Verfügung.

Ein Administrator kann folgende Tätigkeiten ausüben:

- Login, Logout
- Registrierung
- Ein neue Fachrichtung freigeben
- Freigabe einer neuen Gruppe
- Gruppen löschen, die nicht mehr genutzt werden oder nicht freigegeben werden
- Account löschen

Ein Benutzer kann diese Tätigkeiten ausüben:

- Login, Logout
- Registrierung
- Freunde hinzufügen
- Pinnwandinhalt löschen
- An eigene Pinnwand posten
- An Pinnwände und Foren posten, bei denen er in der Liste „erlaubteAutoren“ steht
- Pinnwandeinträge und Forenbeiträge lesen, bei denen er in der Liste „erlaubterLeser“ steht
- Datei hochladen
- Datei runterladen
- Thema bewerten
- Thema kommentieren
- Gruppen beitreten
- Gruppen anlegen
- Gruppe verlassen

- Gruppenmitgliedern eine Nachricht senden
- Fragen innerhalb seiner Gruppe erstellen
- Fragen beantworten
- Quests bearbeiten
- Quest korrigieren
- Benutzer in eine Gruppe einladen
- Bossfights starten
- Team Combats starten
- Offene Fragen korrigieren

Ein Moderator ist der Verwalter einer Gruppe. Er besitzt zusätzlich zu den Benutzerrechten in seiner Gruppe folgende Rechte:

- Hochgeladene Dateien löschen
- Kommentare löschen
- Thema löschen
- Boss Fights korrigieren
- Gruppenmitglieder entfernen
- Löschung der Gruppe beantragen

# Abkürzungsverzeichnis

## RAID 1

Ein RAID 1 (Redundant Array of Independent Disks) stellt eine Speicherungs-methode dar, bei der sämtliche Datensätze parallel auf zwei Speichermedien abgelegt werden. Durch diese Spiegelung kann bei einem Hardwareausfall unterbrechungsfrei auf das zweite Speichermedium gewechselt werden.

## SQL-Datenbank

Eine SQL (Structured Query Language) Datenbank ist eine Datenbank, in der die Datensätze in Tabellen gespeichert werden, die untereinander in relationalen Beziehungen stehen. In diesem Fall werden sämtliche nichtabstrakte Klassen während der Speicherung durch Tabellen dargestellt und in diesem Format mitsamt ihrer Beziehungen untereinander gespeichert. Mittels einer eigenen Abfragesprache (SQL) können gezielt Datensätze in der Datenbank abgerufen werden.

## MC-Fragen

Eine MC (Multiple Choice) Frage, ist ein Fragetyp, bei dem bereits eine bestimmte Anzahl von Antworten vorgegeben wird. Von diesen Antworten ist genau eine korrekt. Nur wenn ausschließlich diese ausgewählt wurde, gilt die Antwort als korrekt gelöst. Aufgrund der vordefinierten Musterlösung kann dieser Fragetyp automatisiert korrigiert werden.

## ID

Eine ID (Identifikationsnummer) stellt eine eindeutige laufende Nummer dar, über die Instanzen von Klassen eindeutig identifiziert werden können.