



# MODELLIERUNG

MHC-PMS Team YELLOW

Michelle Lüscher, Simon Oppliger,  
Yaron Walter, Lucien Heuzeveldt,  
Nadine Siegfried

VERSION	BESCHREIBUNG	NAME	DATUM
0.1	Initiales Dokument	Simon Oppliger	24.04.2019
0.2	Domain Model	Lucien Heuzeveldt	26.04.2019
0.3	Sequenzdiagramm – Seminar erstellen	Yaron Walter	27.04.2019
0.4	Sequenzdiagramm – Seminare filtern	Michelle Lüscher	28.04.2019
0.5	Klassenmodell	Nadine Siegfried	28.04.2019
1.0	Finale Anpassung und Freigabe	Simon Oppliger und Yaron Walter	29.04.2019

## Inhalt

1. Vorwort .....	2
2. Glossar.....	2
3. Domain Model .....	3
4. Sequenzdiagramme .....	4
4.1 Seminar erstellen .....	4
4.2 Seminare filtern .....	5
5. Klassendiagramme.....	6
5.1 Klassendiagramm ohne Methoden.....	7
5.2 Repository-Klassen.....	6
5.3 Manager-Klassen.....	6
6. Anhang .....	8
6.1 Abbildungsverzeichnis .....	8
6.2 Tabellenverzeichnis.....	8

## 1. Vorwort

In diesem Dokument wird Task 4 aus dem «Software Engineering and Design»-Kurs der BFH abgehandelt. Im Rahmen dieses Tasks wurde folgendes gefordert:

For your application, design:

1. a UML domain model
  - a. covering the main classes
  - b. plus association, inheritance, and aggregation relations
  - c. observe the general modeling guidelines
  - d. make use of Responsibility Driven Design (RDD) or CRC-Cards
2. two UML sequence diagrams
  - a. based on your domain model
  - b. one each for the following situations (each should be non-trivial)
  - c. when the user updates some specific information
  - d. when the system alerts the user about some specific event
3. extend your UML domain model into a UML class model (first version)
  - a. update or add classes and relations as needed
  - b. include attributes and operations needed for your sequence diagrams

Die Erzeugnisse dieser Aufgabe werden hier in derselben Reihenfolge präsentiert.

Das Dokument richtet sich an technische Fachleute, es werden Fachwörter verwendet. Das Wissen aus den vorhergehenden Task-Dokumenten wird ebenfalls vorausgesetzt.

## 2. Glossar

Begriff	Erklärung
<b>PMS</b>	Patientenmanagementsystem
<b>MHC</b>	Mental Health Care
<b>SoED</b>	Software Engineering and Design (Modul an der BFH-Bern)
<b>BFH</b>	Berner Fachhochschule
<b>Vaadin</b>	Framework für Web-Applikationen in Java
<b>UML</b>	Unified Modelling Language – Ein Standard für Diagramme im Bereich der Software-Modellierung.
<b>Use Case</b>	Anwendungsfall einer Applikation
<b>Framework</b>	Programmiergerüst für eine Applikation mit vielen Hilfsmethoden

Tabelle 1: Glossar

### 3. Domain Model

Das UML-Diagramm zeigt die verschiedenen Komponenten unserer Applikation. Dabei wurde auf einen möglichst einfachen Aufbau geachtet. Man kann deswegen auch gut die Unterteilung in die drei Module (SeminarFinder, Forum und Wiki) sehen.

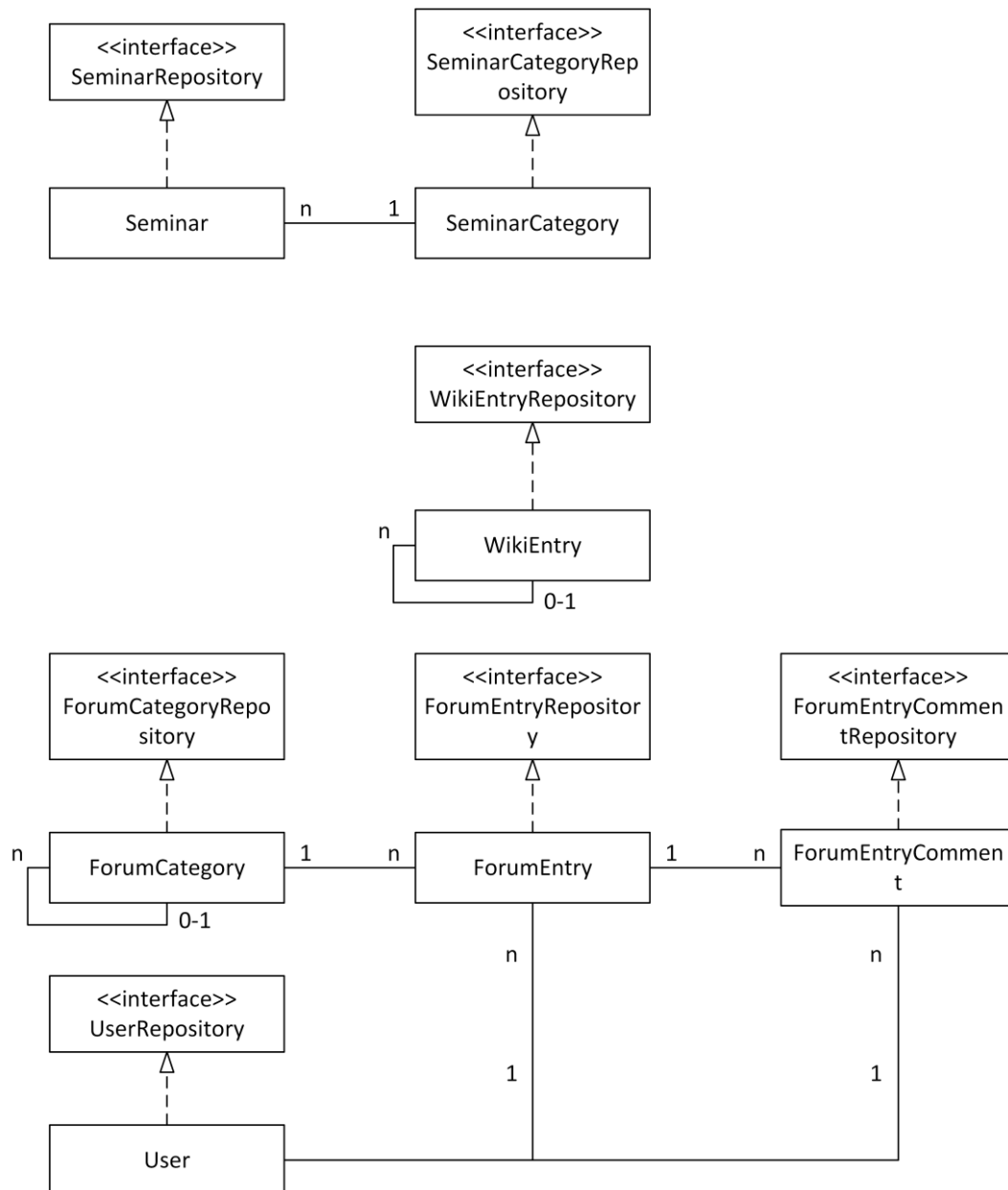


Abbildung 1: Domain Model Diagramm

Ergänzungen:

- Bei der Erstellung wurde das Repository-Pattern berücksichtigt.
- Die Verwendung des Actor-Role-Patterns wurde für die Applikation geprüft. Es wurde entschieden, dieses nicht umzusetzen, da die Verwendung im spezifischen Fall nicht zielführend ist. Das einfache, strikt-hierarchische Rollenkonzept erlaubt eine Umsetzung mit Integer-Werten von 1-4 auf dem Benutzerobjekt in der Session. Verschiedene Objekte je Rolle machen unserer Erachtens Sinn, wenn Rollen mehrere Aufgaben in unterschiedlicher Kontrastierung erledigen müssen. Deswegen wird auf die entsprechenden Klassen verzichtet.

## 4. Sequenzdiagramme

### 4.1 Seminar erstellen

Dieses Sequenzdiagramm beschreibt den Fall, wenn ein Benutzer Daten dem System hinzufügt.

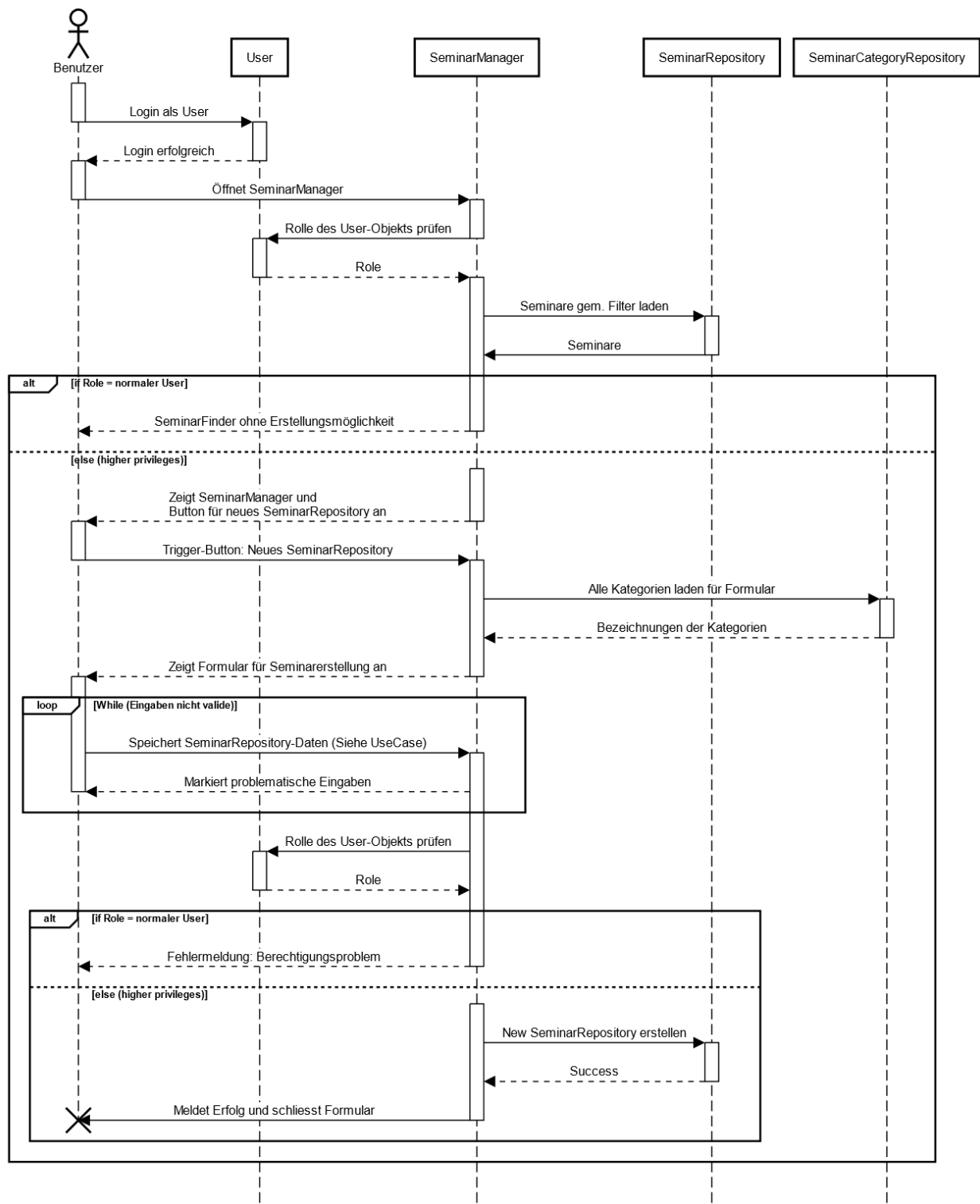


Abbildung 2: Sequenzdiagramm - Seminar hinzufügen

## 4.2 Seminare filtern

Hier handelt es sich um die Ausarbeitung eines Sequenzdiagramms, bei dem das System einen Benutzer «benachrichtigt». In Ermangelung von Anforderungen zu einem spezifischen Messaging oder Alert-Feature mit dem System als Aktor, wurde beschlossen, einen möglichst nahestehenden Anwendungsfall zu modellieren.

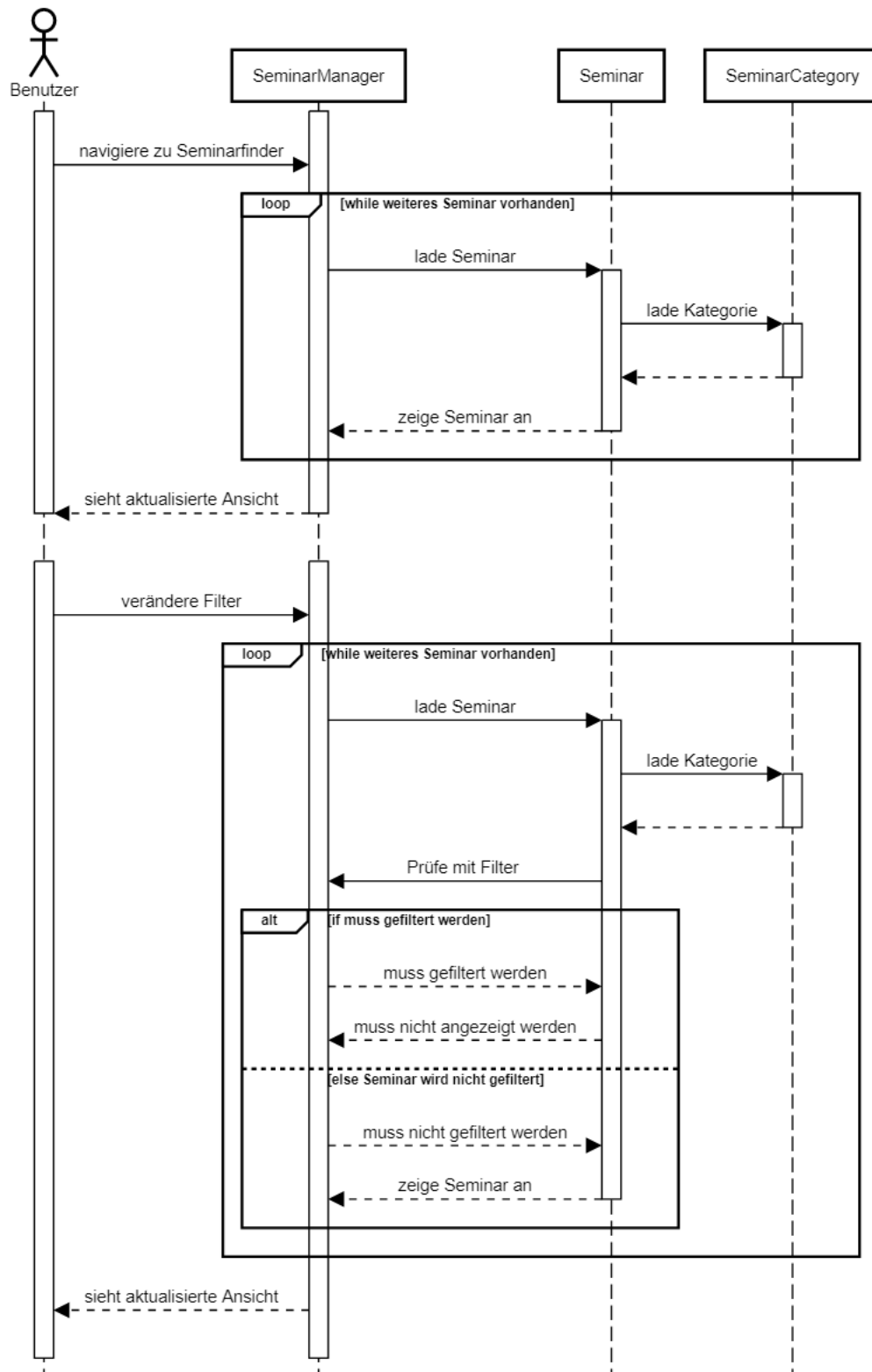


Abbildung 3: Sequenzdiagramm - Seminare Filtern

## 5. Klassendiagramme

Das Klassendiagramm wurde ohne Kenntnisse zu Vaadin gemacht. Dadurch kann es später zu Änderungen kommen, wenn Wissen um das Framework angeeignet wurde. Gemäss Aufgabenbeschrieb der Lehrpersonen handelt es sich um die erste Iteration.

### 5.1 Repository-Klassen

Dieses Klassendiagramm beinhaltet alle Repository-Klassen für die Anwendung des Repository-Patterns.

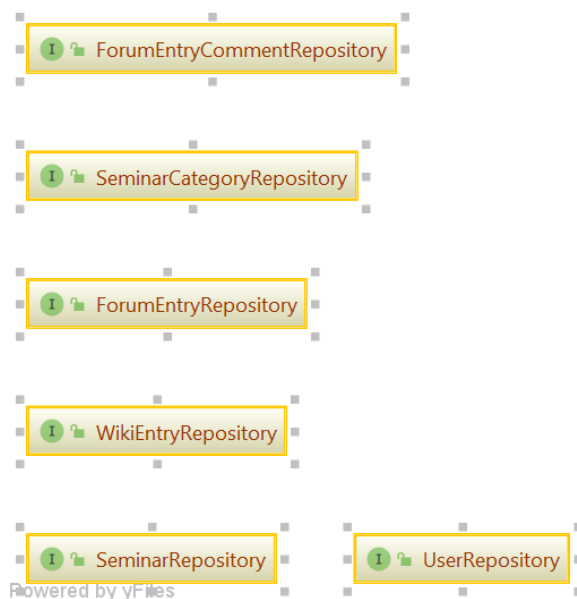


Abbildung 4: Repository-Klassendiagramm

### 5.2 Manager-Klassen

Pro Feature wird eine Manager-Klasse implementiert. Diese kümmert sich um Erstellung oder Veränderung zugehöriger Objekte.

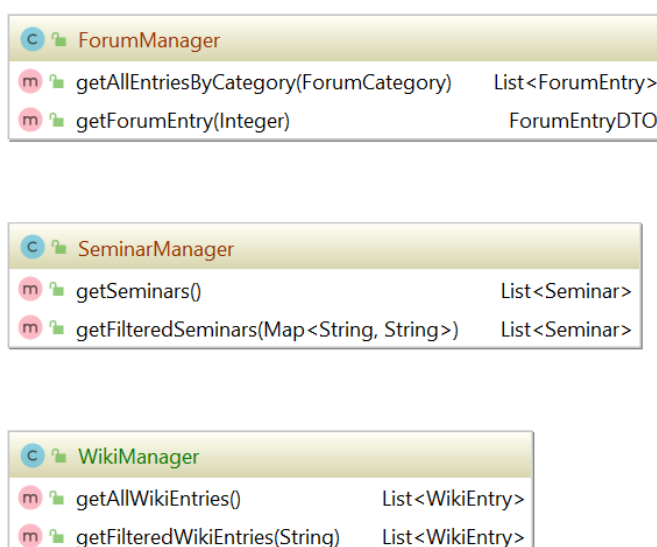


Abbildung 5: Manager-Klassendiagramm

## 5.3 Klassendiagramm mit Methoden

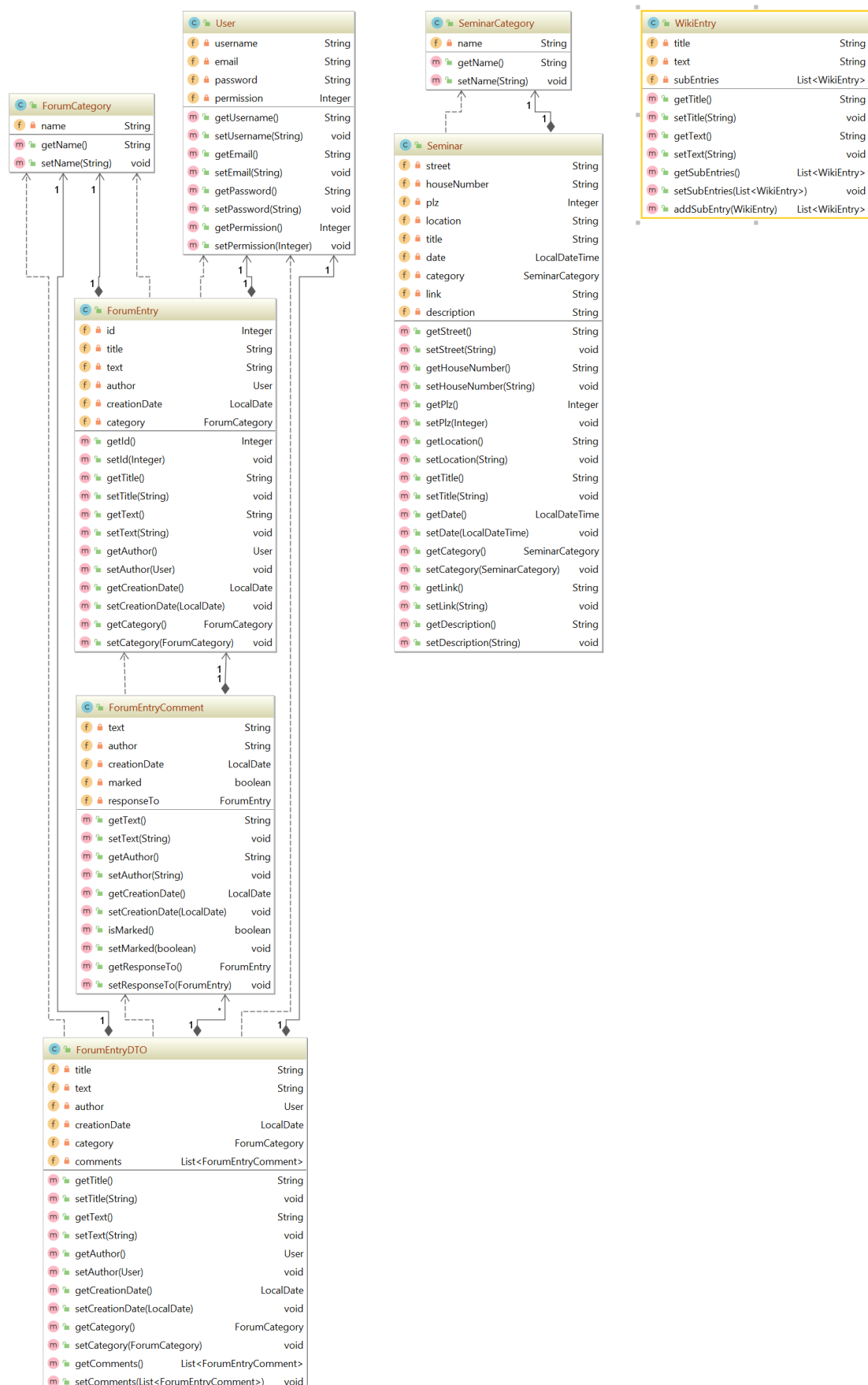


Abbildung 6: Klassendiagramm mit Methoden



## 6. Anhang

### 6.1 Abbildungsverzeichnis

Abbildung 1: Domain Model Diagramm .....	3
Abbildung 2: Sequenzdiagramm - Seminar hinzufügen .....	4
Abbildung 3: Sequenzdiagramm - Seminare Filtern .....	5
Abbildung 4: Repository-Klassendiagramm .....	6
Abbildung 5: Manager-Klassendiagramm .....	6
Abbildung 6: Klassendiagramm mit Methoden .....	7
Abbildung 7: Klassendiagramm - ohne Methoden .....	8

### 6.2 Tabellenverzeichnis

Tabelle 1: Glossar .....	2
--------------------------	---

### 6.3 Klassendiagramm ohne Methoden (Präsentationsansicht)

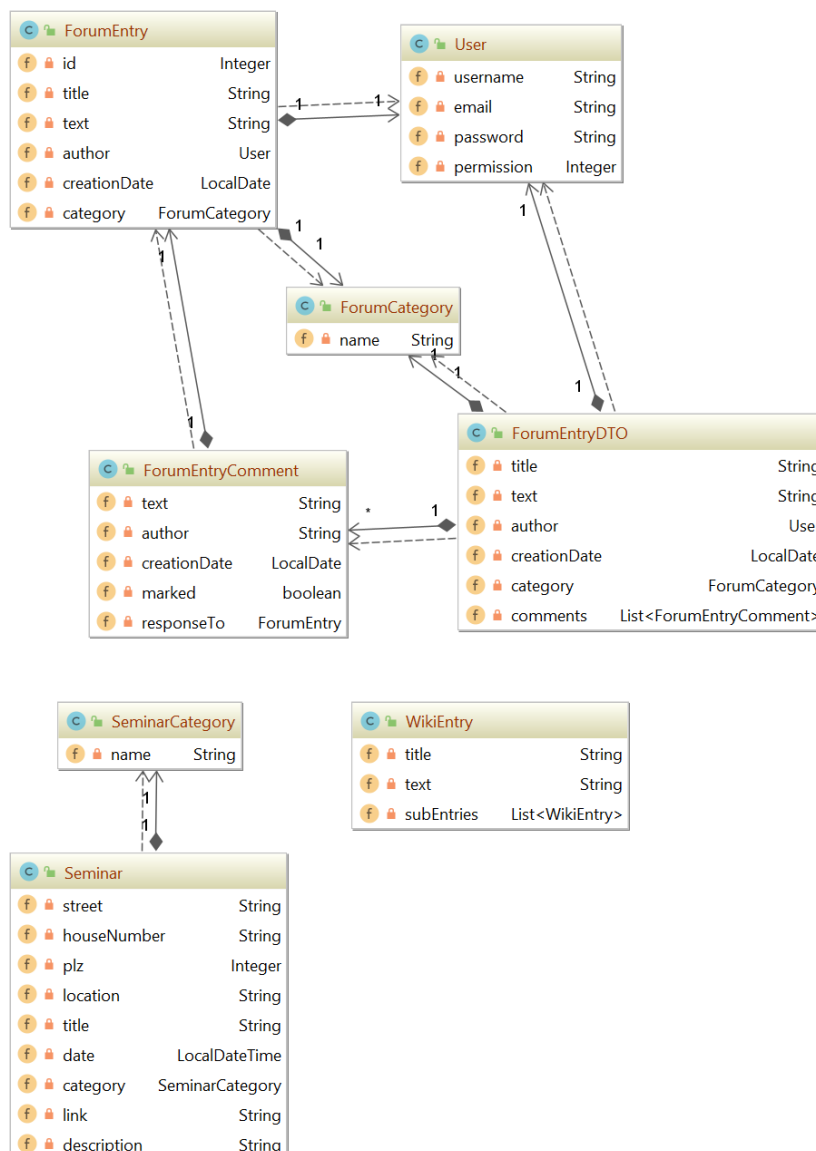


Abbildung 7: Klassendiagramm - ohne Methoden