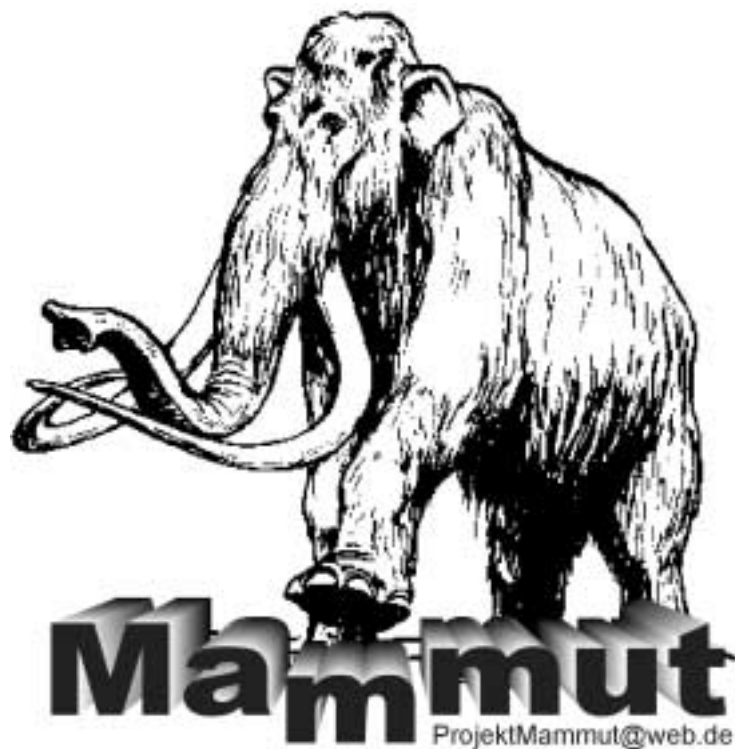


# Analyse



Autoren : Martin Amelsberg ([MartinAmelsberg@T-Online.de](mailto:MartinAmelsberg@T-Online.de))  
Daniel Finger ([Daniel.Finger@gmx.net](mailto:Daniel.Finger@gmx.net))  
Ralf Schmid ([Ralf-Schmid@web.de](mailto:Ralf-Schmid@web.de))  
Cord Buhlert ([Cord.Buhlert@gmx.de](mailto:Cord.Buhlert@gmx.de))

Version : 52

# Inhaltsverzeichnis

1.	UML-Diagramme	
1.1.	Anwendungsfall-Diagramme.....	3
1.1.1.	Nicht-registrierte User.....	4
1.1.2.	Registrierte User.....	5
1.1.3.	Privilegierter User.....	6
1.1.4.	Administration / Verwaltung.....	7
2.	Datenbankmodellierung	
2.1.	Entity-Relationship Modell.....	8
2.2.	Logische Datenmodellierung.....	9
2.2.1.	Nicht normalisierte Form.....	9
2.2.2.	Erste Normalform (1NF).....	10
2.2.3.	Zweite Normalform (2NF).....	10
2.2.4.	Dritte Normalform (3NF).....	10
3.	Prototyp der Oberfläche	
3.1.	User-Bereich .....	11
3.2.	Admin-Bereich .....	11

# 1. UML-Diagramme

Bei der Erstellung der UML-Diagramme wurde festgestellt, dass einige der sonst typischen UML-Diagramme nicht benötigt werden:

- Klassen-Diagramm (da nicht objektorientiert)
- Paket-Diagramm (da keine Klassendiagramme)
- Sequenz-Diagramm (da autarke Funktionen)
- Zustands-Diagramm (da statische Funktionen)

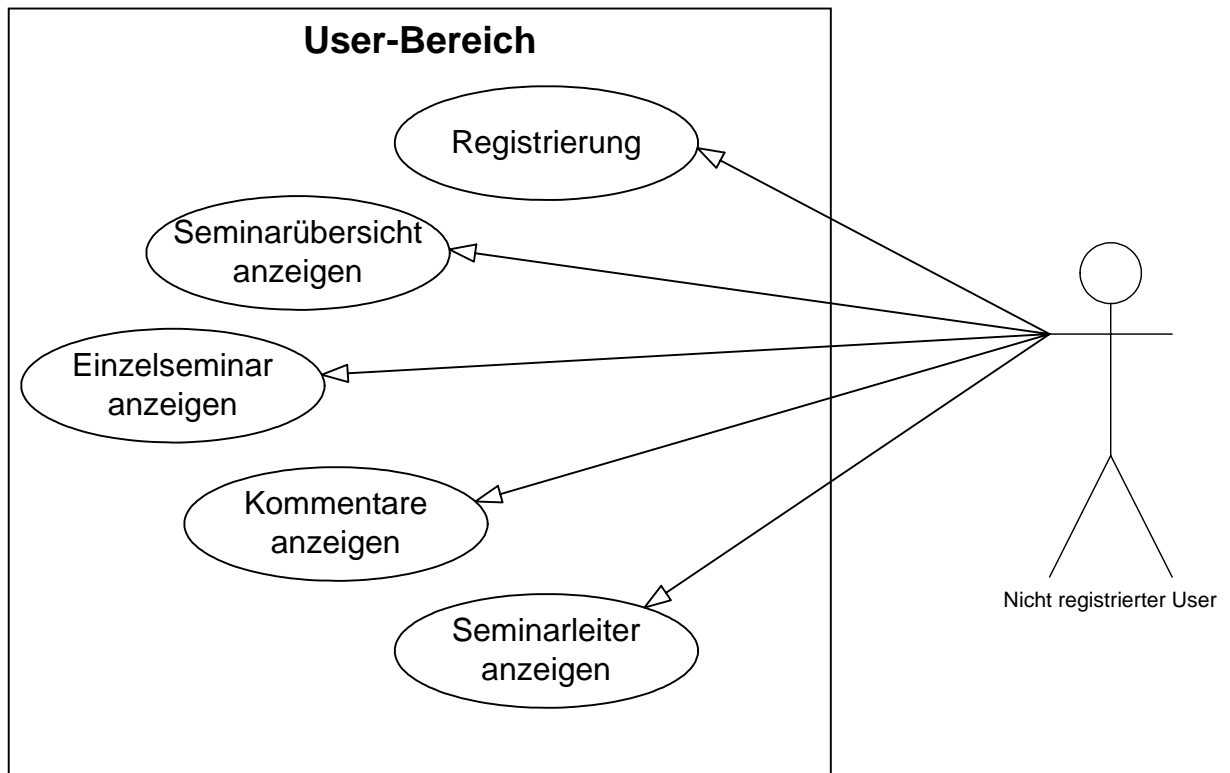
## 1.1. Anwendungsfall-Diagramme

Das Anwendungsfall - Diagramm beschreibt die Grenzen des zu realisierenden Systems. Dazu werden die erforderlichen dargestellt und mit den daran beteiligten außerhalb des Systems Anwendungsfälle Akteuren.

Wir haben unterteilt in:

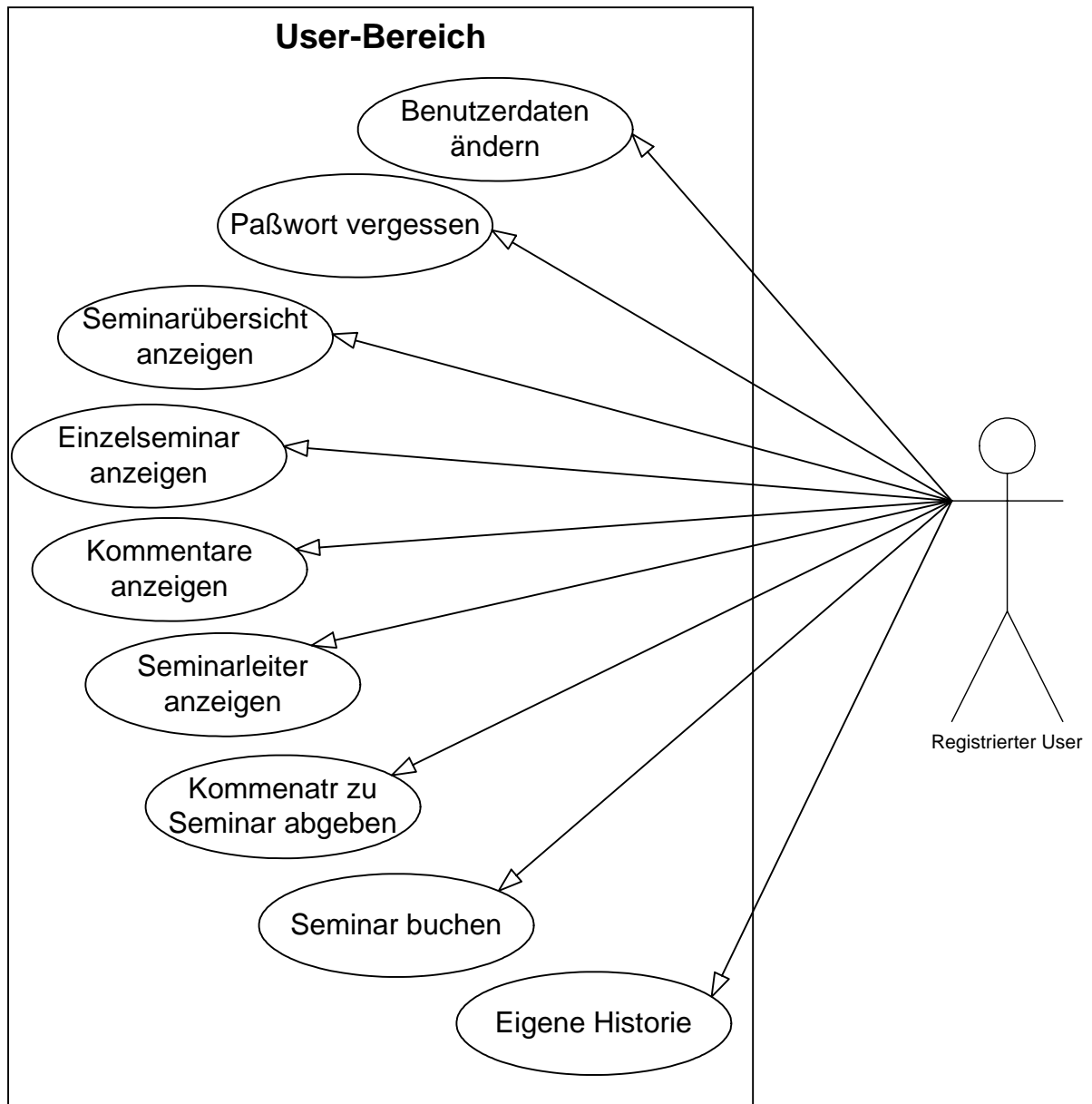
- nicht-registrierter User (Besucher der Homepage; dem System prinzipiell unbekannt)
- registrierter User (Daten des reg. Users sind in der Datenbank gespeichert)
- privilegierter User (Benutzer mit weitergehenden Rechten, wie z.B. Seminarleiter)
- Administration (Besitzt alle Rechte und verwaltet die DB/Seminarverwaltung)

### 1.1.1. Nicht-registrierte User



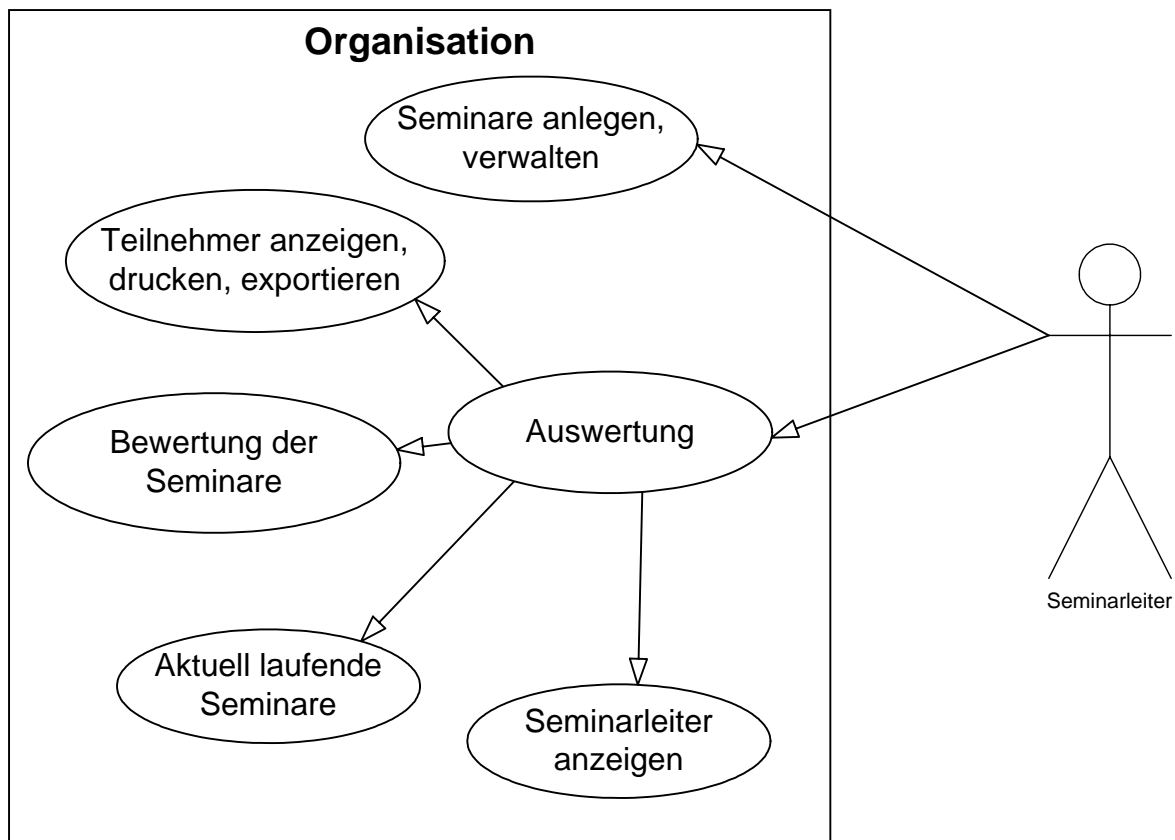
Ein Internet-Nutzer kann sich grundsätzlich über das Angebot informieren, aber nicht aktiv wirken, wie zum Beispiel Seminare buchen, Kommentare zu Seminaren abgeben.

## 1.1.2. Registrierte User



Aufbauend auf den Rechten des nicht-registrierten Users kann ein registrierter User zusätzlich Seminare buchen und Kommentare zu besuchten Seminaren abgeben. Des weiteren kann der registrierte User seine eigenen Benutzerdaten verwalten.

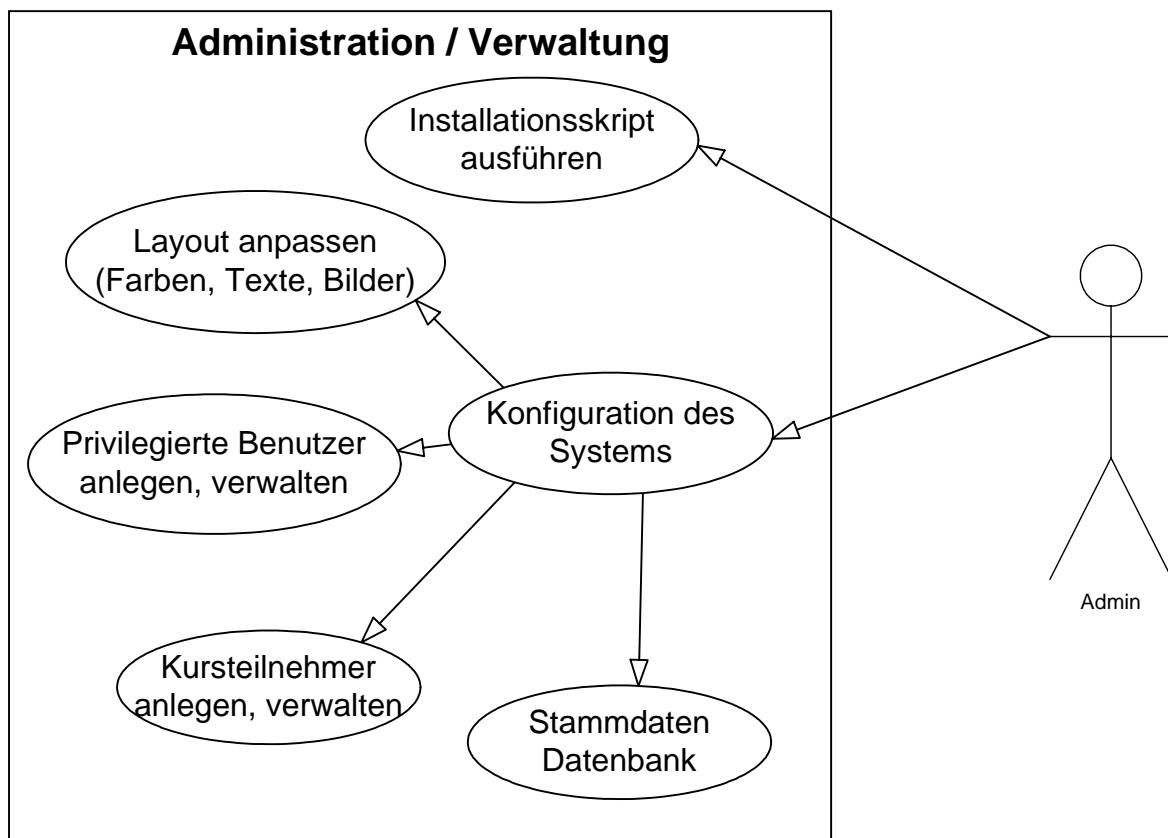
### 1.1.3. Privilegierter User



Zusätzlich zu den oben aufgeführten Personengruppen existieren ggf. noch andere autorisierte Mitarbeiter, die organisatorische Arbeiten im System durchführen dürfen, ohne dabei die vollen Administrationsrechte besitzen zu müssen.

Insbesondere trifft dies auf Seminarleiter und Mitarbeiter in der Telefonzentrale zu, die telefonisch Seminaranmeldungen entgegennehmen.

## 1.1.4. Administration



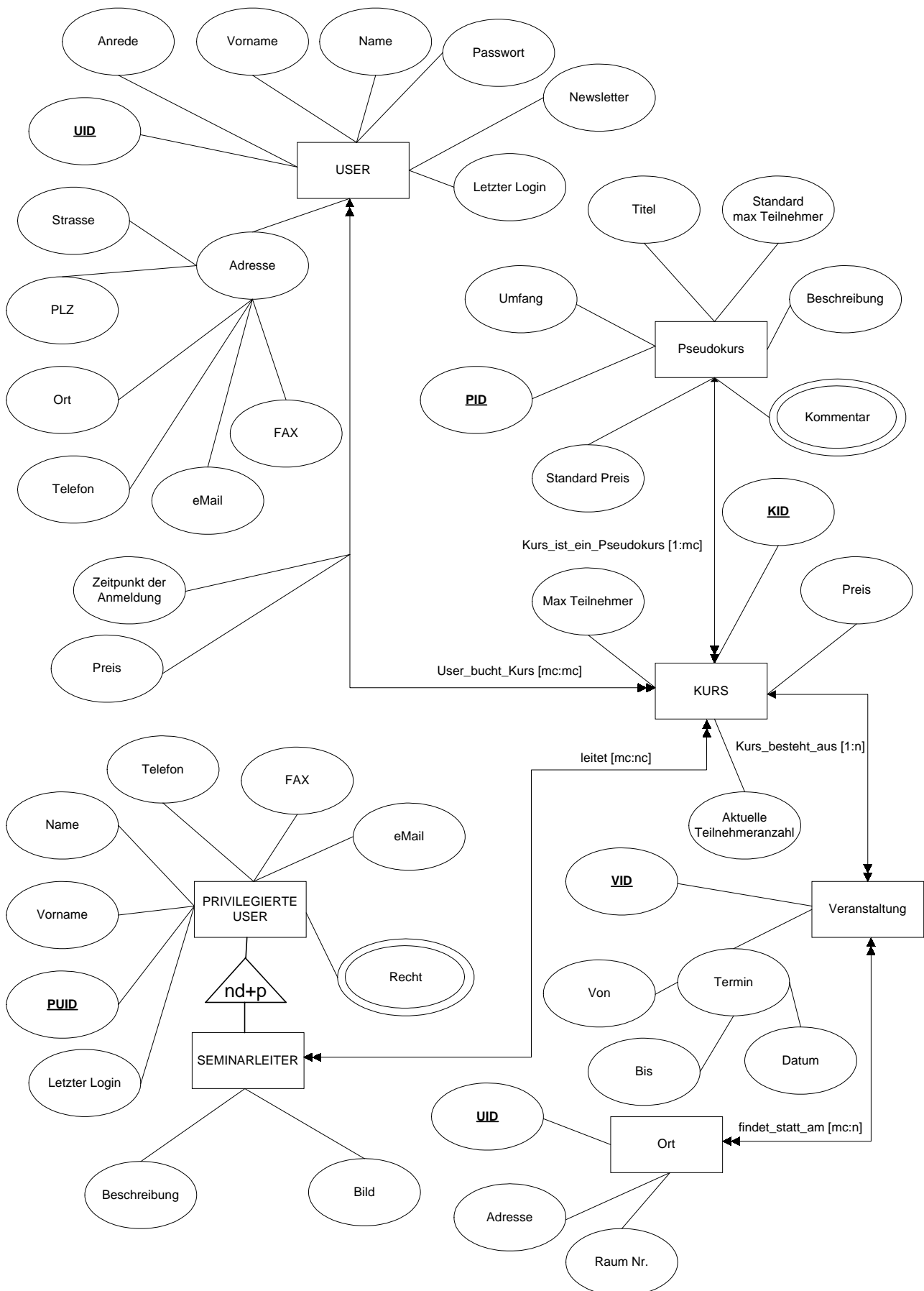
Ein Administrator hat weitergehende Rechte zur Administration des Systems. Insbesondere unterliegt ihm die Ausführung des Installationscriptes und die Konfiguration des Systems.

## 2. Datenbankmodellierung

Grundlage unserer Anwendung ist eine Datenbank. Um die einzelnen Tabellen und Attribute zu bestimmen, wurde zuerst ein ER-Modell erstellt und darauf aufbauend der logische Datenbank-Entwurf. Dieser wurde dann schrittweise normalisiert.



## 2.1. Entity-Relationship Modell



## 2.2. Logische Datenmodellieren

Aufbauend auf das ER-Modell wird nun schrittweise das logische Datenmodell in die dritte Normalform gebracht. Somit werden die Daten für die Verarbeitung im Rechner aufbereitet.

### 2.2.1. Nicht normalisierte Form

User	=	( <u>UID</u> , Anrede, Vorname, Name, Passwort, Newsletter, letzterLogin, Adresse)
mit Adresse	=	(Strasse, PLZ, Ort, Telefon, Fax, eMail)
Pseudokurs	=	( <u>PID</u> , Titel, Beschreibung, StdMaxTeilnehmer, StdPreis, Kommentar*, Umfang)
Kurs	=	( <u>KID</u> , Preis, AktTeilnehmer, MaxTeilnehmer)
Veranstaltung	=	( <u>VID</u> , Termin)
mit Termin	=	(Datum, von, bis)
Ort	=	( <u>OID</u> , Adresse, RaumNr)
PrivilegierteUser	=	( <u>PUID</u> , Vorname, Name, Telefon, Fax, eMail, letzterLogin, Recht*)
Seminarleiter	=	( <u>PUID</u> , Beschreibung, Bild)
R_UserBuchtKurs	=	( <u>UID, KID</u> , ZeitpunktAnmeldung, Preis)
R_KursIstPseudokurs	=	( <u>KID, PID</u> )
R_SeminarleiterLeitetKurs	=	( <u>PUID, KID</u> )
R_KursBestehtAusVeranstaltung	=	( <u>KID, VID, LNr</u> )
R_KursIstInOrt	=	( <u>VID, OID, LNr</u> )

### 2.2.2. Erste Normalform (1NF)

User	=	( <u>UID</u> , Anrede, Vorname, Name, Passwort, Newsletter, letzterLogin, Strasse, PLZ, Ort, Telefon, Fax, eMail)
Pseudokurs	=	( <u>PID</u> , Titel, Beschreibung, StdMaxTeilnehmer, StdPreis, Umfang)
Pseudo_Kommentar	=	( <u>PID, LNr</u> , Kommentar)
Kurs	=	( <u>KID</u> , Preis, AktTeilnehmer, MaxTeilnehmer)
Veranstaltung	=	( <u>VID</u> , von, bis, Datum)
Ort	=	( <u>OID</u> , Adresse, RaumNr)
PrivilegierteUser	=	( <u>PUID</u> , Vorname, Name, Telefon, Fax, eMail, letzterLogin)
PrivUser_Recht	=	( <u>PUID, LNr</u> , Recht)
Seminarleiter	=	( <u>PUID</u> , Beschreibung, Bild)
R_UserBuchtKurs	=	( <u>UID, KID</u> , ZeitpunktAnmeldung, Preis)
R_KursIstPseudokurs	=	( <u>KID, PID</u> )
R_SeminarleiterLeitetKurs	=	( <u>PUID, KID</u> )
R_KursBestehtAusVeranstaltung	=	( <u>KID, VID, LNr</u> )
R_KursIstInOrt	=	( <u>VID, OID, LNr</u> )

### 2.2.3. Zweite Normalform (2NF)

- siehe 2.2.2 - Die Relation ist bereits in zweiter Normalform

### 2.2.4. Dritte Normalform (3NF)

User	=	( <u>UID</u> , Anrede, Vorname, Name, Passwort, Newsletter, letzterLogin, Strasse, PLZ, Ort, Telefon, Fax, eMail)
Pseudokurs	=	( <u>PID</u> , Titel)
Pseudo_Titel	=	( <u>Titel</u> , Beschreibung)
Pseudo_Titel_Besch	=	( <u>Beschreibung</u> , StdMaxTeilnehmer, StdPreis, Umfang)
Pseudo_Kommentar	=	( <u>PID</u> , <u>LNr</u> , Kommentar)
Kurs	=	( <u>KID</u> , Preis, AktTeilnehmer, MaxTeilnehmer)
Veranstaltung	=	( <u>VID</u> , von, bis, Datum)
Ort	=	( <u>OID</u> , Adresse, RaumNr)
PrivilegierteUser	=	( <u>PUID</u> , Vorname, Name, Telefon, Fax, eMail, letzterLogin)
PrivUser_Recht	=	( <u>PUID</u> , <u>LNr</u> , Recht)
Seminarleiter	=	( <u>PUID</u> , Bild)
Sem_Beschreibung	=	( <u>Bild</u> , Beschreibung)
R_UserBuchtKurs	=	( <u>UID</u> , <u>KID</u> , ZeitpunktAnmeldung, Preis)
R_KursIstPseudokurs	=	( <u>KID</u> , <u>PID</u> )
R_SeminarleiterLeitetKurs	=	( <u>PUID</u> , <u>KID</u> )
R_KursBestehtAusVeranst	=	( <u>KID</u> , <u>VID</u> , <u>LNr</u> )
R_KursIstInOrt	=	( <u>VID</u> , <u>OID</u> , <u>LNr</u> )

Anmerkung: Die Relationen „Pseudokurs“, „Pseudo\_Titel“ und „Pseudo\_Titel\_Besch“ werden wieder zusammengeführt (wie in 2.2.3 / 2NF), da eine Kombination Titel/Beschreibung/etc... nur einmal auftaucht.

Es entstehen somit keine Redundanzen, und -wichtiger- es werden mehrere Joins vermieden. Dies spart Laufzeit und kommt dem Qualitätsmerkmal „QLade“ zu Gute.

Analog gilt dies für die Relationen „Seminarleiter“ und „Sem\_Beschreibung“.