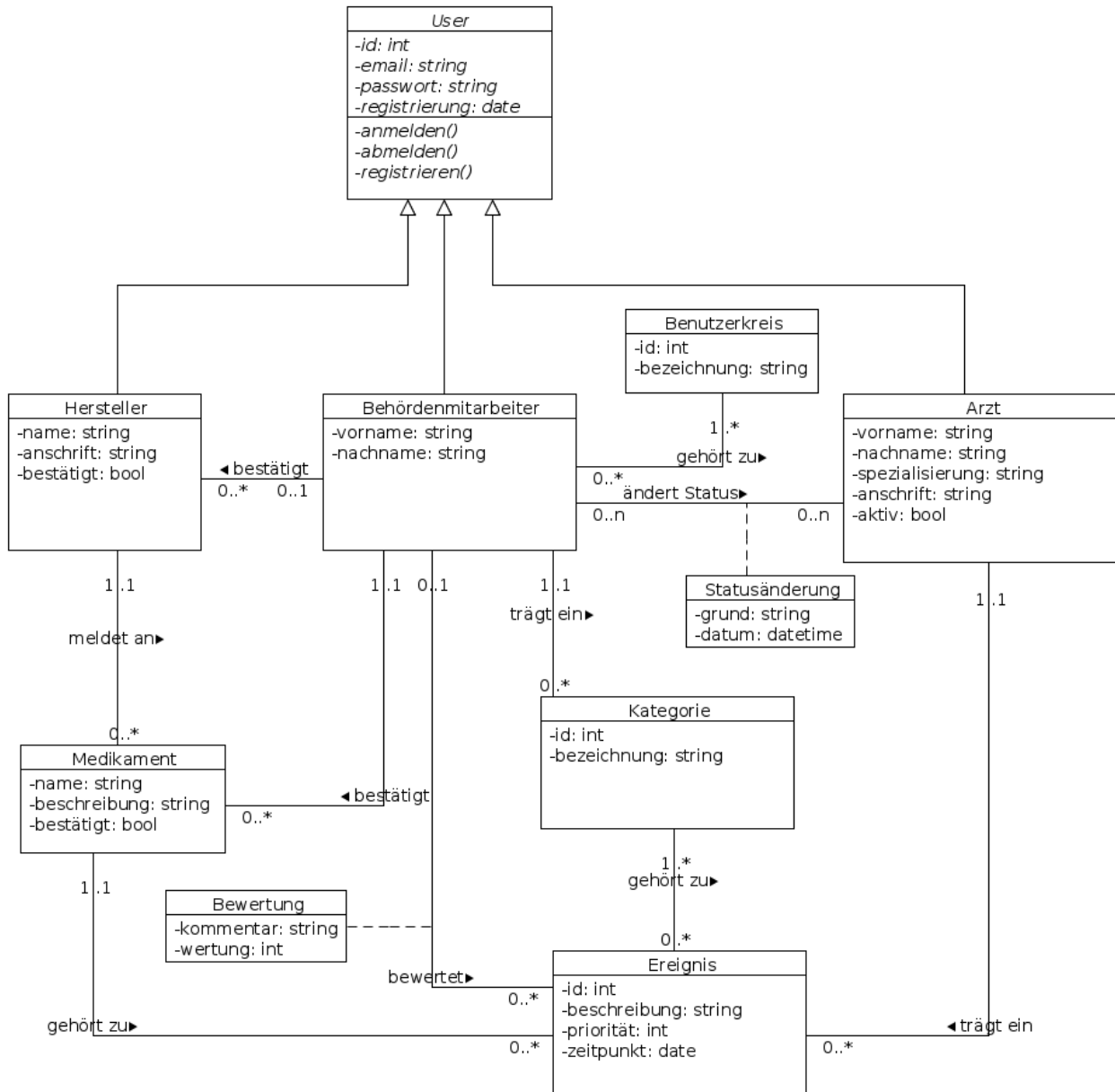




# 1 Systementwurf

## 1.1 Datenbasis



Klassendiagramm

## 1.2 Beschreibung

Im Folgenden werden die einzelnen Klassen jeweils näher erläutert.

## 1.2.1 User

Die Klasse *User* dient als abstrakte Basisklasse für die drei möglichen Rollen im Gesamtsystem. Zur Authentifizierung werden eine E-Mail und ein Passwort benötigt, zusätzlich wird das Registrierungsdatum vermerkt.

## 1.2.2 Behördenmitarbeiter

Behördenmitarbeiter erben von *User* und speichern zusätzlich Vor- und Nachname.

Ein Mitarbeiter kann im System folgende Funktionen ausüben:

1. Bestätigung der von den Herstellern angemeldeten Medikamente
2. Bewertung der von den Ärzten eingetragenen Ereignisse und dahingehend Bewertung der Medikamente
3. (De)aktivierung der Ärzte-Accounts
4. Bestätigen der Hersteller
5. Datenpflege
6. Administrative Aufgaben

Je nach Befugnis werden einem Behördenmitarbeiter ein oder mehrere Benutzerkreis(e) zugewiesen, welche unterschiedliche Rechte verleihen (s. unten).

## 1.2.3 Benutzerkreis

Einem Behördenmitarbeiter muss mindestens ein Benutzerkreis zugeordnet werden. Es existieren Benutzerkreise für die im Abschnitt *Behördenmitarbeiter* genannten Funktionen. Die Benutzerkreise können beliebig kombiniert werden.

## 1.2.4 Hersteller

Hersteller erben von *User* und implementieren zusätzlich Namen und Anschrift.

Ein Hersteller kann beliebig viele Medikamente anmelden. Um mit dem System interagieren zu können, muss der *Hersteller-Account* von einem *Behördenmitarbeiter* aktiviert worden sein.

## 1.2.5 Arzt

Ärzte erben von *User* und implementieren zusätzlich Vorname, Nachname, Spezialisierung und Anschrift.

Ein Arzt ist dafür zuständig, Ereignisse zu Medikamenten einzutragen. Sein Status kann von Behördenmitarbeitern mit entsprechenden Rechten unter Angabe eines Grundes aktiviert oder deaktiviert werden.

## 1.2.6 Medikament

Ein Medikament wird von *einem* Hersteller eingetragen und kann anschließend von einem Behördenmitarbeiter bestätigt werden. Weiterhin kann ein Medikament beliebig vielen *Ereignissen* zugeordnet werden.

## 1.2.7 Ereignis

Ein Ereignis wird von einem *Arzt* eingetragen. Neben einer Beschreibung wird mit der Priorität eine Einordnung bezüglich der Relevanz vorgenommen. Es gilt: Je größer der Wert, desto negativer ist das Ereignis. Jedem Ereignis wird genau ein *Medikament* und mindestens eine *Kategorie* zugeordnet.

Eingetragene Ereignisse können von einem Behördenmitarbeiter bewertet werden.

## 1.2.8 Bewertung

Pro Medikament kann von einem *Behördenmitarbeiter* eine Bewertung eingetragen werden. Diese wird anhand der vorliegenden *Ereignisse* festgelegt. Die *Bewertung* besteht aus einem Kommentar und einer Wertung.

## 1.2.9 Kategorie

Eine *Kategorie* spezifiziert ein *Ereignis* und wird von einem *Behördenmitarbeiter* eingetragen.

## 1.2.10 Statusänderung

Eine *Statusänderung* kann von einem *Behördenmitarbeiter* hinzugefügt werden und bezieht sich immer auf einen Arzt. Es bewirkt die Änderung des aktiv-Flags. Für die Statusänderung muss ein Grund angegeben werden.

# 1.3 Umwandlung in relationales Datenmodell

## 1.3.1 User

Attribut	Typ
id	int
rolle_id	int
email	string
password	string
registrierung	datetime

## 1.3.2 Benutzerkreis

Attribut	Typ
id	int
bezeichnung	string

## 1.3.3 User\_Benutzerkreis

Attribut	Typ
user_id	int
benutzerkreis_id	int

### 1.3.4 Hersteller

Attribut	Typ
id	int
name	string
anschrift	string
bestätigt	bool

### 1.3.5 Arzt

Attribut	Typ
id	int
vorname	string
nachname	string
spezialisierung	string
anschrift	string
aktiv	bool

### 1.3.6 Medikament

Attribut	Typ
id	int
hersteller_id	int
name	string
beschreibung	string
bestätigt	bool

### 1.3.7 Ereignis

Attribut	Typ
id	int
medikament_id	int
arzt_id	int
Beschreibung	string

Attribut	Typ
priorität	int
zeitpunkt	date

### 1.3.8 Bewertung

Attribut	Typ
ereignis_id	int
mitarbeiter_id	int
kommentar	string
wertung	int

### 1.3.9 Kategorie

Attribut	Typ
id	int
bezeichnung	string
mitarbeiter_id	int

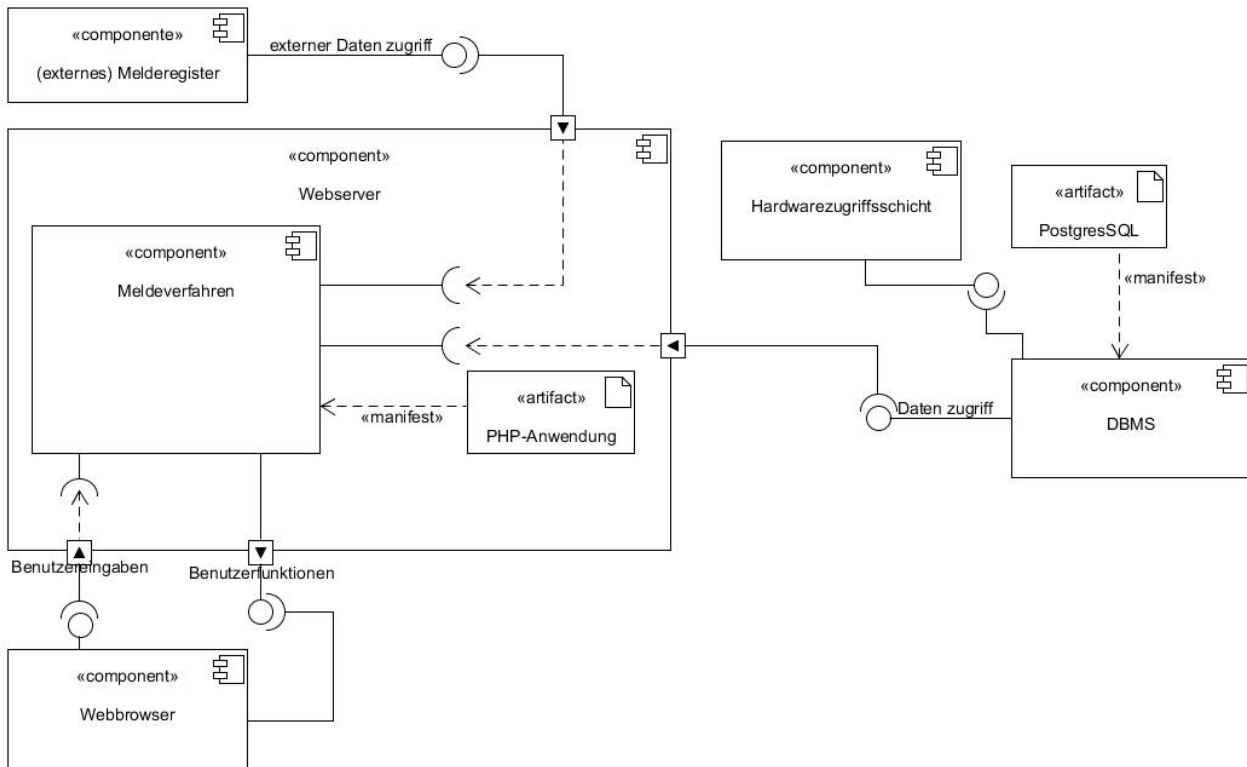
### 1.3.10 Ereignis\_Kategorie

Attribut	Typ
ereignis_id	int
kategorie_id	int

### 1.3.11 Statusänderung

Attribut	Typ
id	int
mitarbeiter_id	int
arzt_id	int
grund	string
datum	datetime

## 1.4 Komponenten

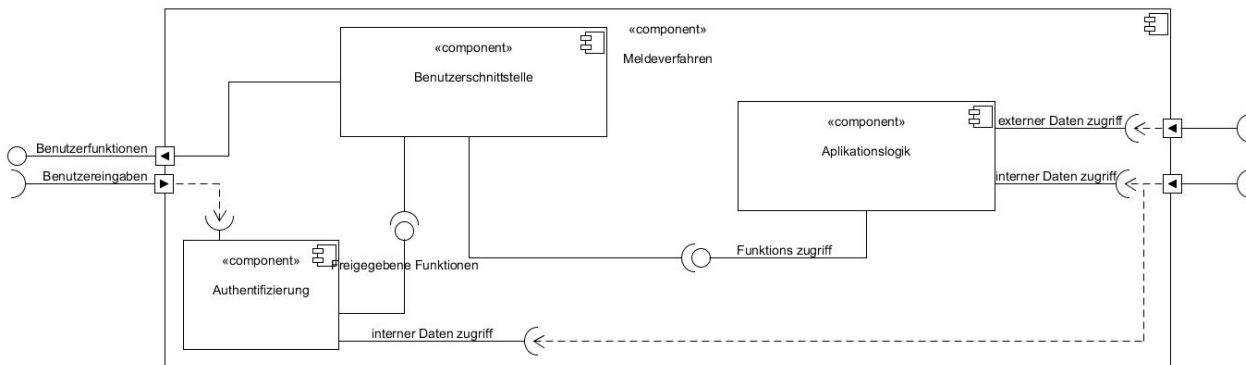


### Komponenten

#### 1.4.1 Beschreibung

- (externes) Melderegister: Stellt die zum Abgleichen benötigten externen Daten zur Verfügung.
- Hardwarezugriffsschicht: Gewährleistet den Hardwarezugriff auf die Daten.
- DBMS: Verwaltet die Datenbank per PostgreSQL. Es sind Standard Zugriffe möglich, wie lesen, schreiben und löschen.
- Webserver: Stellt die Webapplikation Meldeverfahren zur Verfügung und leiten die externen/internen Datenzugriffe an sie weiter.
- Meldeverfahren: Die Webapplikation, welche dem Benutzer die gewünschten Funktionen zur Verfügung stellt und seine Eingaben verarbeitet.
- Webbrowser: Beliebiger Webbrowser, der die Benutzereingaben an den Webserver schickt und die gewünschten Funktionen für den Benutzer zur Darstellung bringt.

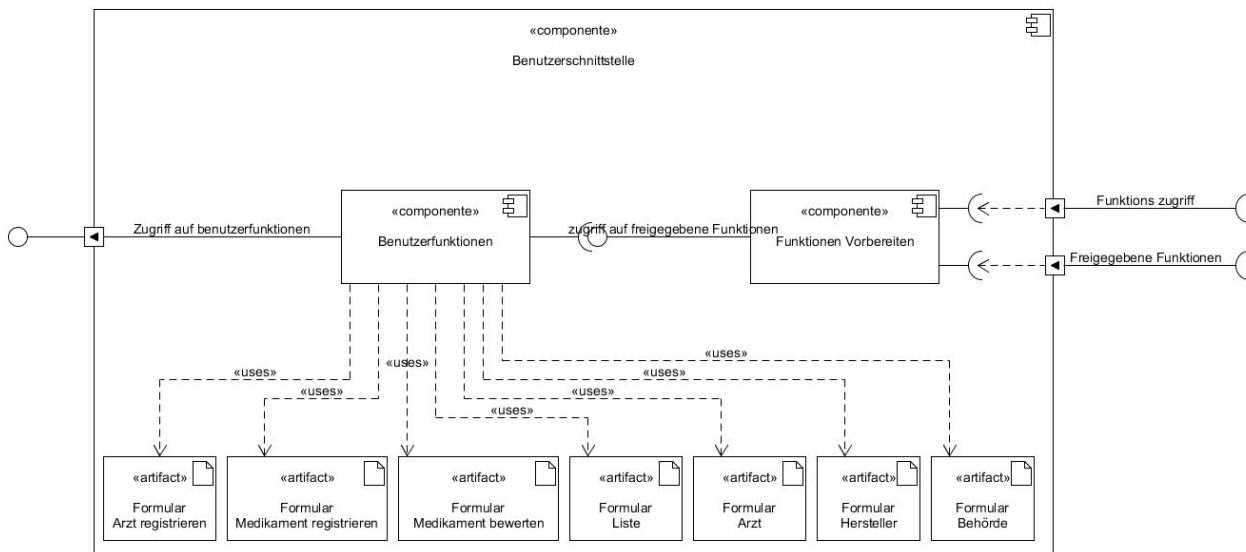
## 1.4.2 Komponenten des Meldeverfahrens



### Komponenten

- **Benutzerschnittstelle:** Liefert die Benutzerfunktionen durch der von der Authentifizierung gelieferten für den aktuellen Benutzer freigegebenen Funktionen und greift durch die Applikationslogik auf diese zu.
- **Applikationslogik:** Liefert Funktions zugriff auf die von der benutzerschnittstelle benötigten Funktionen.
- **Authentifizierung:** Authentifiziert den benutzer und gibt der Benutzerschnittstelle an welche Funktionen dieser zur Verfügung hat.

## 1.4.3 Komponenten der Benutzerschnittstelle



### Komponenten

- **Funktionen vorbereiten**
- **Benutzerfunktion**

Webbrowser

# Registrierung für Ärzte

Vorname:

Nachname:

Spezialisierung:

Adresse:

Stadt:  PLZ:

E-Mail:

Passwort:

Passwort wdh. :

Abbrechen

Bestätigen

Komponenten



Webbrowser

-

+

×

# Ereignis melden

Priorität:

Zeitpunkt:  :   .  .

Beschreibung:

Komponenten

Webbrowser

—

+

×

# Medikament Registrieren

Name:

Beschreibung:

Abbrechen

OK

Komponenten

- externer Daten zugriff
- interne Daten verwalten
- Funktionen Medikamente
- Funktionen Datenpflege
- Funktionen DV

- Funktionen