



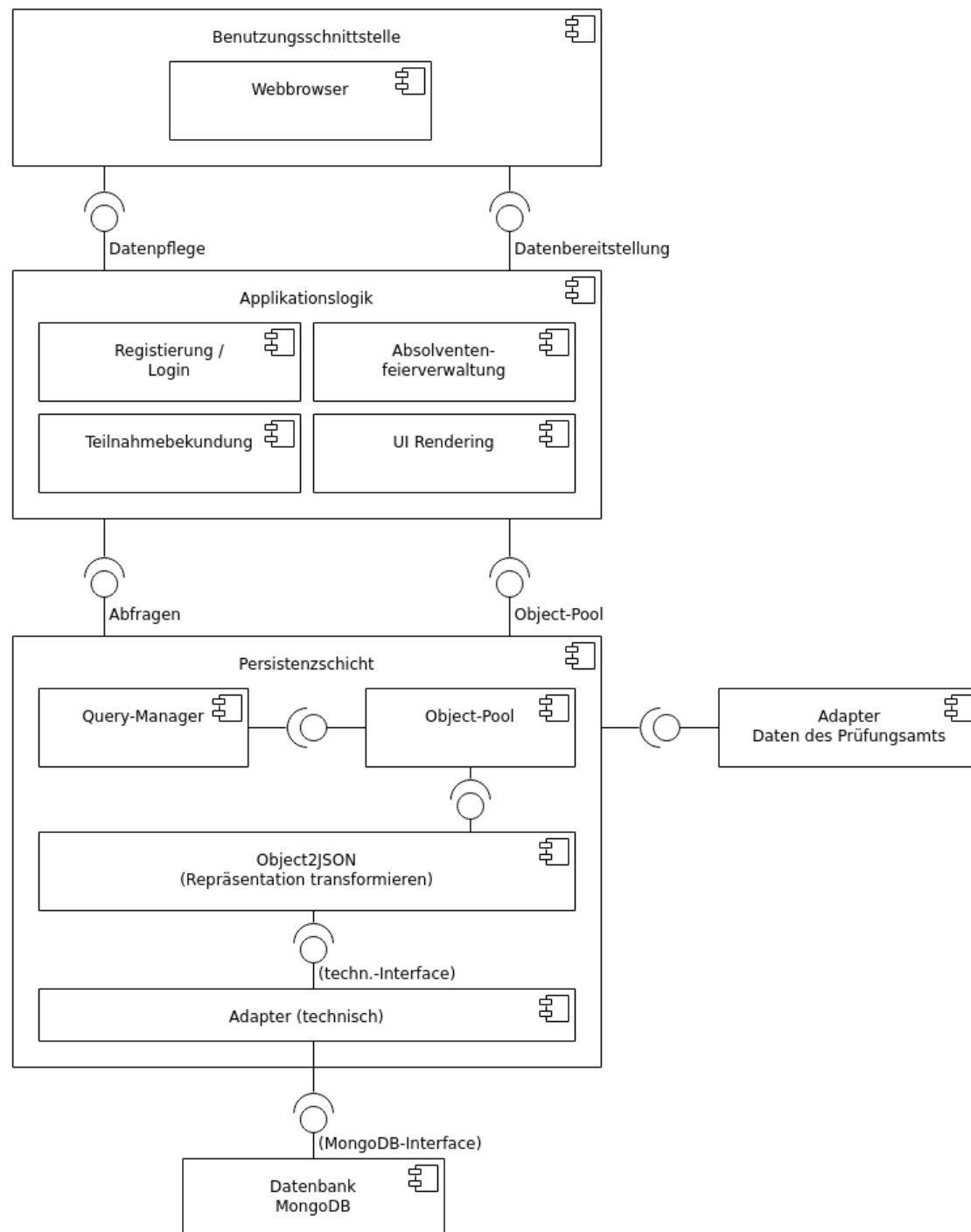
1 Allgemeine Beschreibung

Das Absolventenfeiersystem kurz AFS nutzt eine Server-Client-Architektur. Bei dem Client handelt es sich um ein Thin-Client, da das Rendering der Darstellungen auf dem Server durchgeführt wird. Da es sich um eine Webapplikation handelt, ist der genutzte Client ein Webbrowser. Über diesen werden die Nutzerinteraktionen an den Server gesendet. Die Persistierung wird von einer Persistierungsschnittstelle im Zusammenspiel mit mongoDB durchgeführt. Die Absolventendaten werden über einen Adapter von Prüfungsamt bezogen.

2 Systemstruktur

2.1 Komponentendiagramm

2.1.1 Übersicht

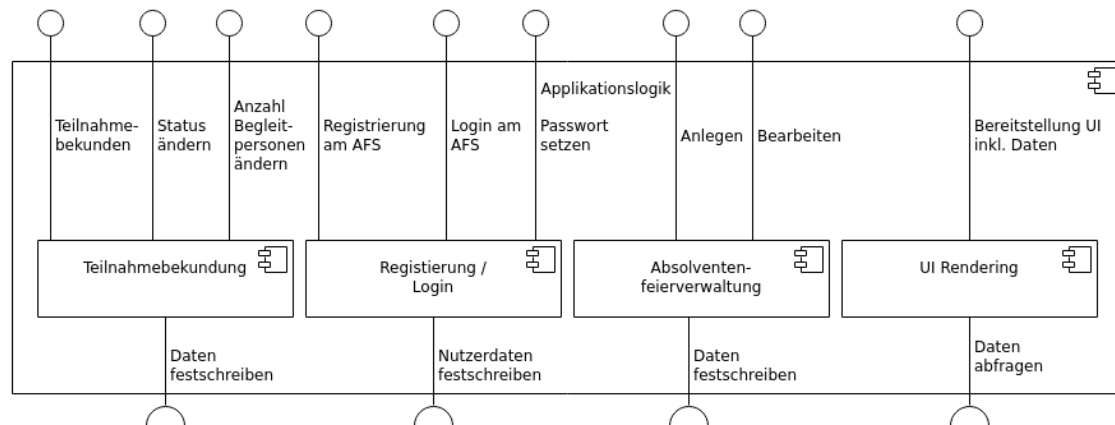


[1] Übersicht UML-Komponenten-Diagramm für Absolventenfeiersystem

Die Benutzungsschnittstelle greift auf Funktionen der Applikationslogik zu um Daten an die Applikation zu senden. Desweiteren erfolgt das Rendern der HTML-Seite auf dem Server, sodass der Client/Web-Browser auch auf diese Funktionalität zugreift.

Die Applikationslogik wiederum, benötigt für das Speichern sowie für das Abrufen von Informationen/Daten die Persistenzschicht. Die Persistenzsicht dient als Mediator zwischen Applikationslogik und Datenbank (MongoDB).

2.1.2 Applikationslogik verfeinert



[2] Applikationslogik UML-Komponenten-Diagramm für Absolventenfeiersystem

Die Applikationslogik ist im oberen Bild verfeinert, da diese in der Übersicht nur grob eingezeichnet ist. Die vier verschiedenen Komponenten bieten verschiedene Export- und Import-Schnittstellen an, welche im folgenden genauer erklärt werden.

2.2 Komponenten

2.2.1 Benutzungsschnittstelle

2.2.1.1 Webbrowser

Der Webbrowser wird genutzt, damit die Nutzer die Applikation verwenden können. Der Browser muss HTML und CSS anzeigen können. Je nach Use-Case und Benutzerrolle werden verschiedene HTML-Seiten angezeigt, welche von der Applikation gerendert worden sind. Der Webbrowser muss Formulardaten an den Server senden können.

2.2.1.1.1 Import-Schnittstelle

2.2.2 Applikationslogik

2.2.2.1 Registrierung / Login

Diese Komponente ermöglicht die Registrierung und Anmeldung am AFS, um so Absolventen und Mitarbeitern des Prüfungsausschuss die Möglichkeit zu geben, Daten innerhalb der Anwendung zu verwalten und zu ändern.

2.2.2.1.1 Export-Schnittstelle

Die Komponente Registrierung / Login bietet die folgenden Exportschnittstellen an: * Registrierung am AFS * Login am AFS * Passwort setzen

2.2.2.1.1.1 Registrierung am AFS

Registrierung am AFS bietet einem Mitarbeiter des Fachbereiches(Prüfungsausschuss) die Möglichkeit Zugangsdaten am AFS zu hinterlegen, um so im Zugriff auf das Backend des AFS zu erhalten. Diese Funktion darf nur von Berechtigten, also Mitgliedern des Prüfungsausschusses genutzt werden.

Datenstruktur:

```
new_login ::= #email + password;  
email ::= string;  
password ::= string;
```

2.2.2.1.1.2 Login am AFS

Der Login am AFS ermöglicht den Absolventen und den Mitarbeitern des Prüfungsausschusses sich am AFS an zu melden, um so Funktionen der Rolle wahrzunehmen.

Datenstruktur:

```
login ::= #email + password;  
email ::= string;  
password ::= string;
```

2.2.2.1.1.3 Passwort setzen

Absolventen ohne Nutzeraccount müssen vor Nutzung des AFS ein Passwort festlegen, damit diese Möglichkeit ist, exportiert die Registrierung/Login Komponente die Möglichkeit ein Passwort zu vergeben. Nach dem ein Passwort vergeben worden ist, kann der Absolvent das AFS nutzen.

Datenstruktur:

```
new_login ::= #email + password;  
email ::= string;  
password ::= string;
```

2.2.2.1.2 Import-Schnittstelle

2.2.2.1.2.1 Nutzerdaten festschreiben

Das Festschreiben der Nutzerdaten wird aus der Persistenzsicht importiert, um das Abspeichern von Daten zu ermöglichen. Die Daten umfassen die Nutzereingaben, welche durch die Export-Schnittstellen an die Komponente gesendet werden können. Die Aktionen auf der Datenbank ist entweder eine Abfrage oder ein Einfügen.

Datenstruktur:

```
login ::= #email + password;  
email ::= string;
```

```
passwort ::= string;
```

2.2.2.2 Teilnahmebekundung

Diese Komponente ermöglicht den Benutzern eine Anmeldung für die Absolventenfeier auszusprechen. Desweiteren ermöglicht die Komponente Absolventen die Anzahl der Begleitpersonen sowie ihre Zusage zu ändern.

2.2.2.2.1 Export-Schnittstelle

Die Komponente Teilnahmebekundung bietet die folgenden Exportschnittstellen an: * Teilnahmebekunden * Status ändern * Anzahl Begleitpersonen ändern

2.2.2.2.1.1 Teilnahmebekunden

Diese Export-Schnittstelle bietet den Mitarbeitern des Fachbereichs die Möglichkeit ihr kommen zur Absolventenfeier anzumelden.

Datenstruktur:

```
login ::= #email + name + vorname;
```

```
email ::= string;
```

```
name ::= string;
```

```
vorname ::= string;
```

2.2.2.2.1.2 Status ändern

Durch diese Export-Schnittstelle, ist es dem Absolventen möglich, sein kommen anzumelden bzw. zu widerrufen.

Datenstruktur:

```
login ::= #email + name + teilnahmestatus;
```

```
teilnahmestatus ::= boolean;
```

2.2.2.2.1.3 Anzahl Begleitpersonen ändern

Diese Export-Schnittstelle ermöglicht es den Absolventen, die Anzahl an Begleitpersonen, welche diesen zur Absolventenfeier begleiten zu ändern.

Datenstruktur:

```
login ::= #email + name + anzahl;
```

```
anzahl ::= number;
```

2.2.2.2.2 Import-Schnittstelle

2.2.2.2.2.1 Daten festschreiben

Das Festschreiben der Daten wird aus der Persistenzsicht importiert, um das Abspeichern von Daten zu ermöglichen. Die Daten umfassen die Nutzereingaben, welche durch die Export-Schnittstellen an die Komponente gesendet werden können. Die Aktionen auf der Datenbank ist entweder ein Einfügen oder ein Update auf bestehende Datenbestände.

Datenstruktur:

```
teilnahmebekunden ::= #email + name + vorname;  
status_ändern ::= #email + teilnahmestatus; anzahl_begleitpersonen ::= #email + anzahl;  
  
email ::= string;  
name ::= string; vorname ::= string;  
teilnahmestatus ::= boolean; anzahl ::= number;
```

2.2.2.3 Absolventenfeierverwaltung

Die Komponente Absolventenfeierverwaltung ermöglicht es eine Absolventenfeier anzulegen bzw. diese zu bearbeiten.

2.2.2.3.1 Export-Schnittstelle

2.2.2.3.1.1 Anlegen

Durch diese Exportschnittstelle haben die Mitglieder des Prüfungsausschusses die Möglichkeit eine Absolventenfeier im AFS zu hinterlegen. Sollte bereits eine andere Feier angelegt sein, wird diese überschrieben.

Datenstruktur:

```
neue_feier ::= name + #datum + uhrzeit + ort;  
  
name ::= string;  
datum ::= date;  
uhrzeit ::= time;  
ort ::= string;
```

2.2.2.3.1.2 Bearbeiten

Durch die Exportschnittstelle "Bearbeiten" hat der Prüfungsausschuss die Möglichkeit die Daten der Absolventenfeier im AFS zu ändern.

Datenstruktur:

```
feier ::= name + #datum + uhrzeit + ort;
```

```
name ::= string;  
datum ::= date;  
uhrzeit ::= time;  
ort ::= string;
```

2.2.2.3.2 Import-Schnittstelle

Die Import-Schnittstelle ermöglicht, das eine Erstellung einer Absolventenfeier sowie die Änderungen an einer Absolventenfeier in die Datenbank geschrieben werden und somit persistent sind.

Datenstruktur:

```
feier ::= name + #datum + uhrzeit + ort;
```

```
name ::= string;  
datum ::= date;  
uhrzeit ::= time;  
ort ::= string;
```

2.2.2.4 UI Rendering

Diese Komponente umfasst ein server seitiges rendern der Ansicht, sowie eine Auslieferung an den Client.

2.2.2.4.1 Export-Schnittstelle

Die Export-Schnittstelle liefert dem Browser/Client eine gerenderte HTML-Seite mit Informationen aus der Datenbank, sofern diese benötigt werden.

Datenstruktur: Auslieferung einer HTML/CSS-Seite

2.2.2.4.2 Import-Schnittstelle

Die Import-Schnittstelle wird benötigt, damit HTML-Seiten mit unter anderem Datenbank inhalt gerendert werden können.

Datenstruktur: Abhängig von der angeforderten Seite

2.2.3 Persistenzschicht

Die Persistenzsicht hat die folgenden Aufgaben: Sie kümmert sich um das Transofmieren der Daten in eines für die Datenbank nutzbare Format. Desweiteren ermöglicht diese, SQL Abfragen auf vorhandene Daten. Neben diesem stellt die Persitenzsicht auch eine Möglichkeit bereit mit Hilfe eines Adapters auf die benötigten Daten des Prüfungsamtes zuzugreifen

2.2.3.1 Query-Manager

Der Query-Manager ermöglicht es SQL Datenbank abfragen auszuführen.

2.2.3.2 Object-Pool

Der Object-Pool verwaltet die Objekte welche in der Datenbank vorhanden sind bzw. welche in die Datenbank geschrieben werden sollen.

2.2.3.3 Object2JSON

Diese Komponente wandelt die von der Applikation übergebenden Objekte in das JSON Format aus.

2.2.3.4 Adapter (technisch)

Der Adapter ermöglicht es die umgewandelten Daten in der MongoDB Datenbank abzuspeichern.

2.2.4 Weiters

2.2.4.1 Datenbank - MongoDB

Die Datenbank erhält von der Persistenzsicht Daten, welche abgespeichert werden sollen. Desweiteren gibt die Datenbank Daten auf Anforderung Daten an die Persistenzsicht.

2.2.4.2 Adapter - Daten ders Prüfungsamts

Diese Komponente stellt einen Adapter für die benötigten Daten des Prüfungsamtes dar, sodass nur benötigte Daten an die Persistenzschicht weiter gegeben werden.

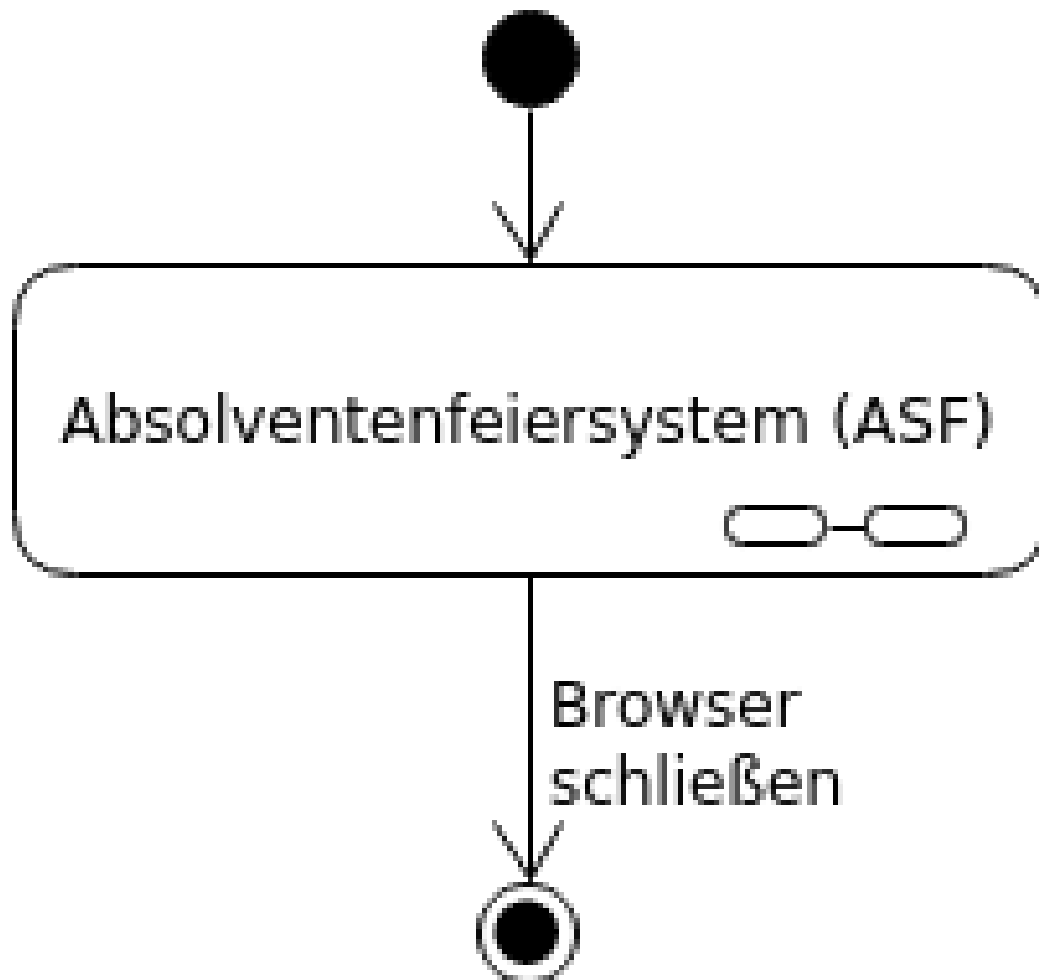
2.3 Benutzungsschnittstelle

Geben Sie die Elemente der Benutzungsschnittstelle an; verwenden Sie Wireframes zur Visualisierung. Beschreiben Sie einheitliche Bedienabläufe.

2.4 Systemverhalten

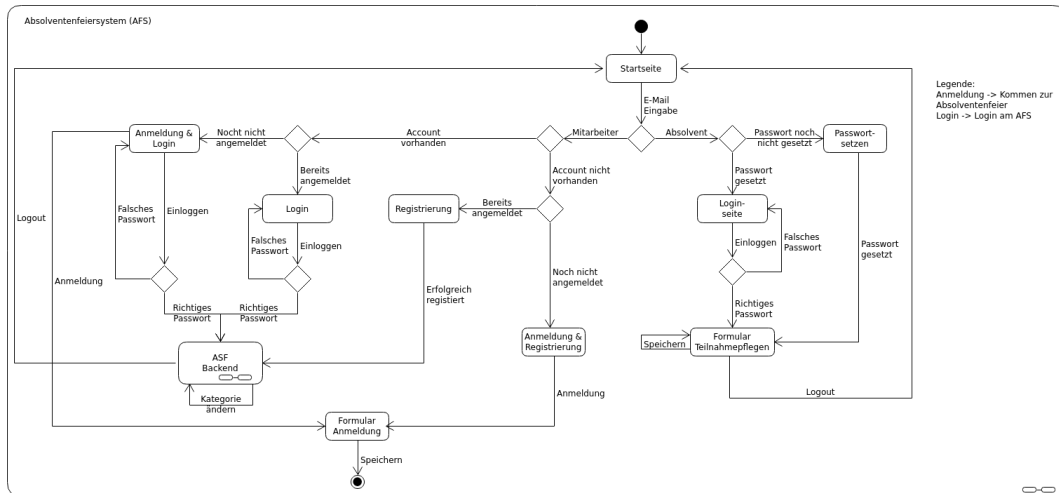
2.4.1 UML Zustandsdiagramme

2.4.1.1 Übersicht



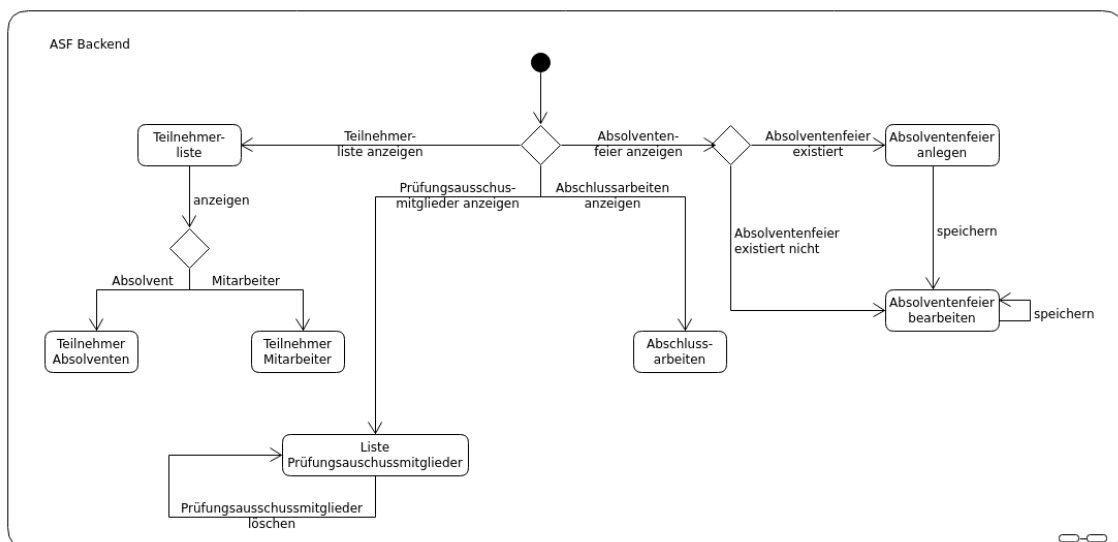
[3] Übersicht UML-Zustandsdiagramm-Diagramm für Absolventenfeiersystem

2.4.1.2 Absolventenfeiersystem (ASF)



[4] ASF UML-Zustandsdiagramm-Diagramm für Absolventenfeiersystem

2.4.1.3 ASF Backend



[4] ASF-Backend UML-Zustandsdiagramm-Diagramm für Absolventenfeiersystem

2.5 Datenbasis

2.5.1 Klassenmodell

Geben Sie hier UML-Klassendiagramme an und erläutern Sie diese.

2.5.2 Nutzung MongoDB

Geben Sie hier an, wie das Klassenmodell auf Container und Referenzen zwischen den Containern abgebildet wird.

2.6 Verwendung von Entwurfsmustern

2.6.1 Singleton

Die Applikationslogik soll nur über einen Pool von Verbindungen zur Datenbank verfügen. Hierzu wird das Entwurfsmuster Singleton genutzt.

2.6.2 Adapter