



# Reservationstool

## Klassendiagramm

Dieses Dokument beinhaltet das Klassendiagramm, sowie eine kurze Beschreibung zu jeder Klasse.

Autoren: Elia Bösiger, Elias Schildknecht, Lukas Gestach, Patrik Aebischer

Datum: 20.10.2016

Version: 1.0

## Inhaltsverzeichnis

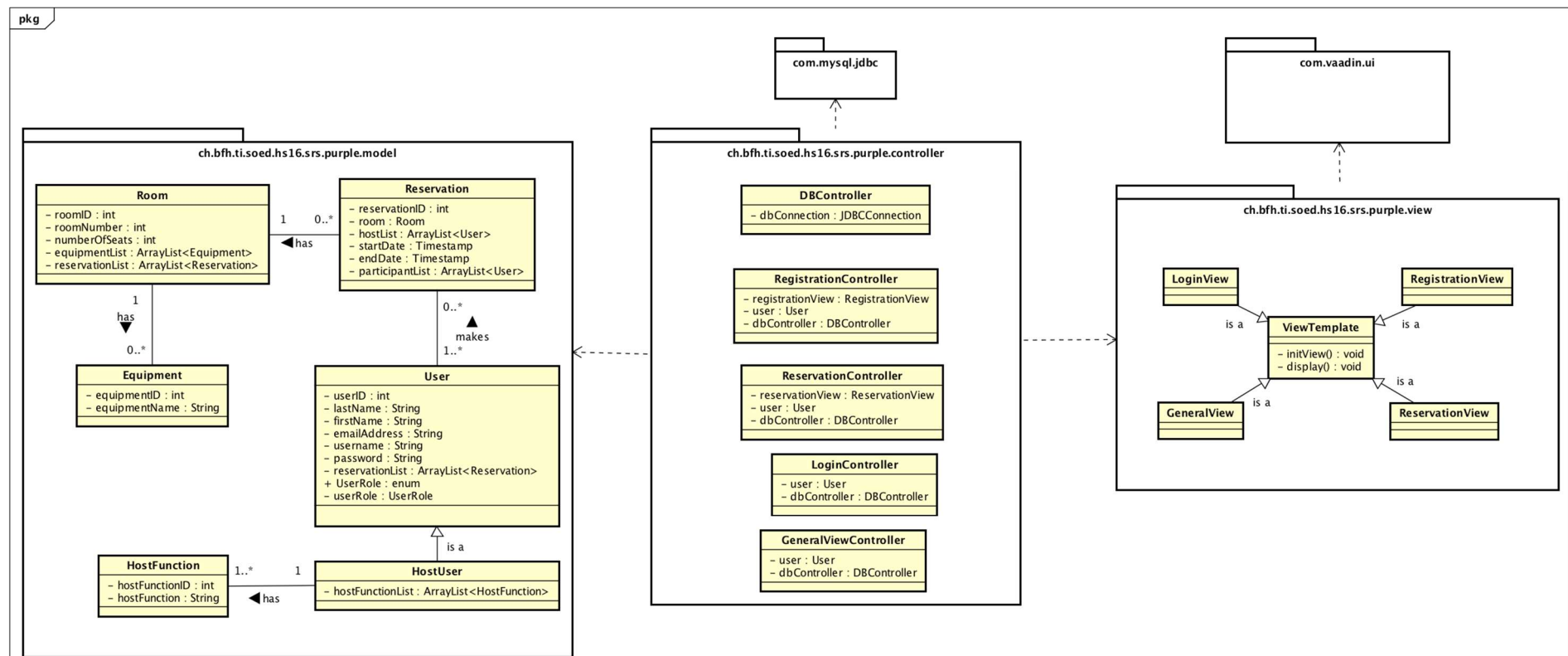
<b>1</b>	<b>UML-KLASSENDIAGRAMM</b>	<b>3</b>
1.1	KONZEPT	3
1.2	GESAMTÜBERSICHT	3
<b>2</b>	<b>KLASSEN</b>	<b>4</b>
2.1	MODEL	4
2.2	VIEW	6
2.3	CONTROLLER	7

# 1 UML-Klassendiagramm

## 1.1 Konzept

Wir haben uns nach dem Model-View-Controllern (MVC) Konzept gerichtet. Das MVC Konzept ermöglicht ein flexibles Programmdesign.

## 1.2 Gesamtübersicht

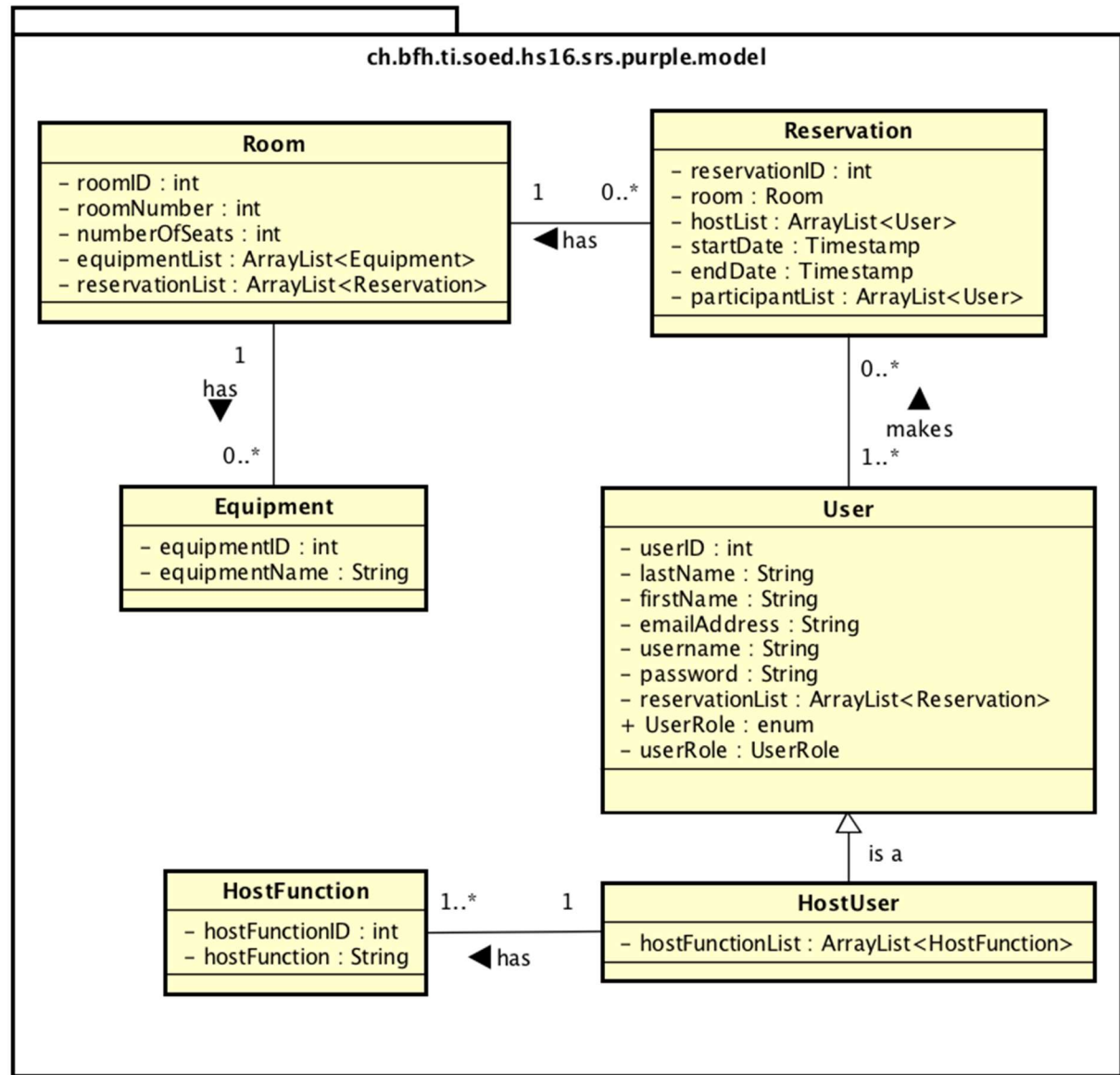


powered by Astah

## 2 Klassen

### 2.1 Model

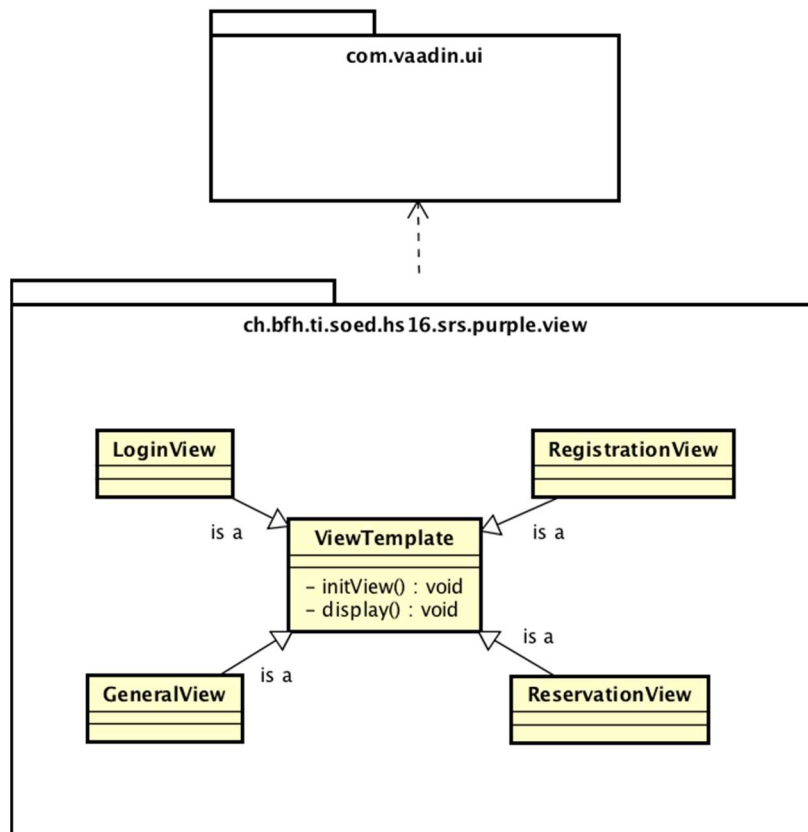
Die Model-Klassen repräsentieren eine Tabelle aus der Datenbank. Alle Model-Klassen besitzen eine ID mit dem Integer Variabel Typ. Diese ID entspricht dem Primary-Key vom Datensatz in der Datenbank.



Klasse	Beschreibung
<b>Room</b>	Die Klasse „Room“ ist ein Raum, der reserviert werden kann. Ein Raum besitzt eine Raumnummer, eine Anzahl an Sitzplätzen sowie unterschiedliche Gegenstände.
<b>Equipment</b>	Die Klasse „Equipment“ beschreibt ein Gegenstand in einem Raum. Das Attribut „equipmentName“ beschreibt den Gegenstand. Ein Beispiel hierfür wäre ein Beamer. Das Equipment eines Raumes soll dem Veranstalter helfen ein passendes Zimmer zu finden.
<b>Reservation</b>	Die Klasse „Reservation“ definiert eine Reservation, die von einem Veranstalter erstellt wurde. Eine Reservation beinhaltet einen Raum , ein Start- und End-Datum, sowie die Veranstalter und die Teilnehmer.
<b>User</b>	<p>Die Klasse „User“ ist ein Benutzer des Systems. Ein User registriert sich mit den folgenden Attributen:</p> <ul style="list-style-type: none"> <li>• Nachname</li> <li>• Vorname</li> <li>• Email-Adresse</li> <li>• Username</li> <li>• Passwort</li> </ul> <p>Die Rolle des Benutzers wird in der Klasse „UserRole“ definiert. Die Reservations-Applikation erlaubt die folgenden drei Benutzerrollen:</p> <ul style="list-style-type: none"> <li>• <b>USER_ROLE_HOST</b>: Der Veranstalter kann Reservationen tätigen.</li> <li>• <b>USER_ROLE_PARTICIPANT</b>: Ist der Teilnehmer einer Reservation/Veranstaltung.</li> <li>• <b>USER_ROLE_ADMIN</b>: Der Admin hat alle Rechte und wird verwendet um einem User die Veranstaltungs-Rechte (USER_ROLE_HOST) zu geben.</li> </ul>
<b>HostUser</b>	Die Klasse „HostUser“ ist von der „User“-Klasse abgeleitet. Sie verfügt zusätzlich über die „hostFunctionList“, welche die Funktionen des Veranstalters definiert.
<b>HostFunction</b>	Die Klasse „HostFunction“ definiert eine Funktion eines Veranstalters. Beispiel für eine Funktion eines Veranstalters ist „Physikprofessor“.

## 2.2 View

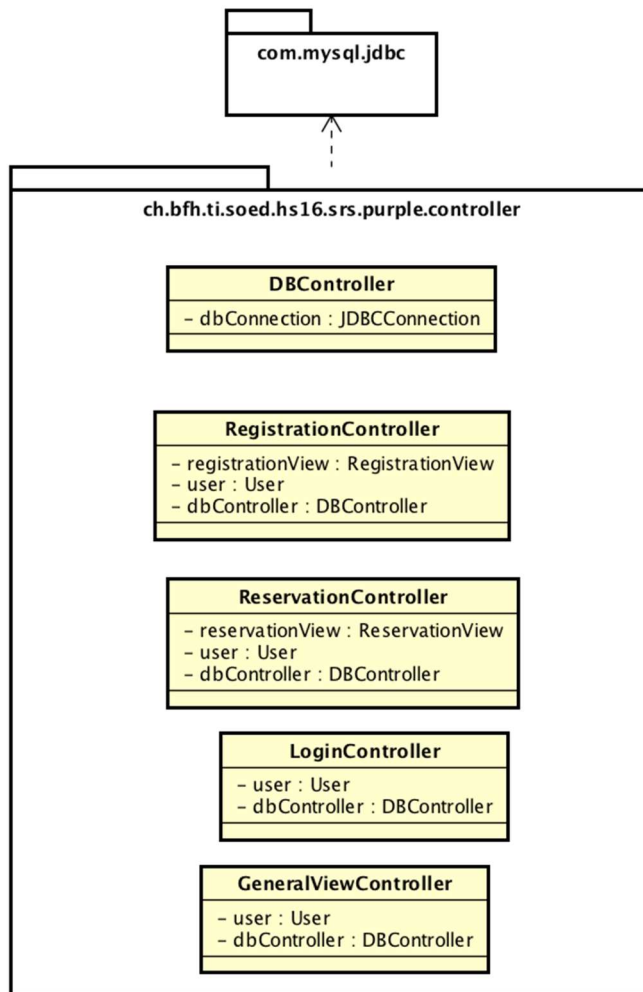
Die View-Klassen definieren die grafische Benutzeroberfläche.



Klasse	Beschreibung
<b>ViewTemplate</b>	Die Klasse „ViewTemplate“ definiert die Überklasse aller grafischen Benutzeroberflächen. Sie gibt folgende Funktionen vor: <ul style="list-style-type: none"> <li>• <b>initGUI</b>: Funktion für die Initialisierung der Komponenten auf der grafischen Benutzeroberfläche</li> <li>• <b>display</b>: Funktion für das Darstellen der grafischen Benutzeroberfläche</li> </ul>
<b>LoginView</b>	Die Klasse „LoginView“ beinhaltet alle grafischen Elemente für das Login Formular.
<b>RegistratioView</b>	Die Klasse „RegistrationView“ beinhaltet alle grafischen Elemente für die Registration.
<b>GeneralView</b>	Die Klasse „GeneralView“ ist die grafische Benutzeroberfläche, die einem User nach dem einloggen erscheint. Es ist eine Übersicht aller aktuellen Reservationen.
<b>ReservationView</b>	Die Klasse „ReservationView“ beinhaltet alle grafischen Elemente für die Erstellung und Mutation einer Reservation.

## 2.3 Controller

Die Controller-Klassen beinhaltet die Logik, welche aus den View-Klassen umgesetzt werden muss. Sie bearbeiten die Daten um sie schliesslich auf der Datenbank abzusetzen.



Klasse	Beschreibung
<b>DBController</b>	Die Klasse „DBController“ ist zuständig, die Daten aus der Datenbank zur Verfügung zu stellen und auf die Datenbank zu schreiben.
<b>RegistrationController</b>	Die Klasse „RegistrationController“ beinhaltet die Logik der Klasse „RegistrationView“.
<b>ReservationController</b>	Die Klasse „ReservationController“ beinhaltet die Logik der Klasse „ReservationView“.
<b>LoginController</b>	Die Klasse „LoginController“ beinhaltet die Logik der Klasse „LoginView“.
<b>GeneralViewController</b>	Die Klasse „GeneralViewController“ beinhaltet die Logik der Klasse „GeneralView“.