

Grasshopper - Designmodell

Mit einem Sprung zum besten Urlaub

Anreiter, Simon Kocman, Andreas Moser, Victoria



.LocationDAO

- -- speichert & lest Locations
- zu den Locations gehören Städte und Hotels
- ist das Interface welches sich um die Speicherung von Location kümmert

SQLLocationDAO

- Implementiert Location DAO und speichert die Daten in eine SQL Datenbank

.UserDAO

- Interface welches alle user daten speichert und lest

.SQLUserDAO

- Implementiert UserDAO und speichert die Daten in eine SQL Datenbank



Bookings

- Buchungen für einen bestimmten Raum
 - Es wird angegeben wer die Buchung durchführt und das Datum von-bis
- Buchungen werden nach Datum sortiert und in der SQL Datenbank gespeichert
- Jeder Raum hat eine ArrayList wo die Buchungen für sich gespeichert werden

.City

- extends Location
 - Jede Stadt hat ein Land zu welchem es gehört
 - Hotels welche in einer Stadt sind werden in einem Array gespeichert



.Hotel

- extends Location
 - Jedes Hotel hat Räume welche in einem ArrayList gespeichert werden

.Location

- Abstrakte Überklasse von Hotel und City

.Review

- Objektwertig
 - Die Reviews werden direkt in die SQL Datenbank gespeichert
 - Man kann für jede Location, Hotel oder Stadt kann man alle Reviews suchen
- Beim speichern kann jeder User zu einer Location immer nur ein Review schreiben
- Sollte ein User zu einer Location ein erneutes Review schreiben, wird das vorherige überschrieben
- Bei Kunden wird für das Hotelreview vorher überprüft, ob eine Buchung für



.Statistik

- Statistik hat 3 Instanzvariablen: Beginn- EndDatum der Statistik und ein Array
- Im Array wird gespeichert, wieviele Buchungen an dem Tag stattgefunden haben
- Statistiken werden jeweils für bestimmte Tage erstellt
- Die Hotelstatistik wird aus den Statistiken der Räume dieses Hotels erstellt
- Die Stadtstatistiken werden aus den Statistiken der Hotels welche sich in dieser Stadt befinden erstellt

.HotellierManagement

.CustomerManagement

- extends UserManagement



Session

- Gibt die Pages an den Browser weiter

.TAManagement

- extends UserManagement

.UserManagement

- Sind die Funktionen die alle User ausführen können

·Customer, Hotellier, TA

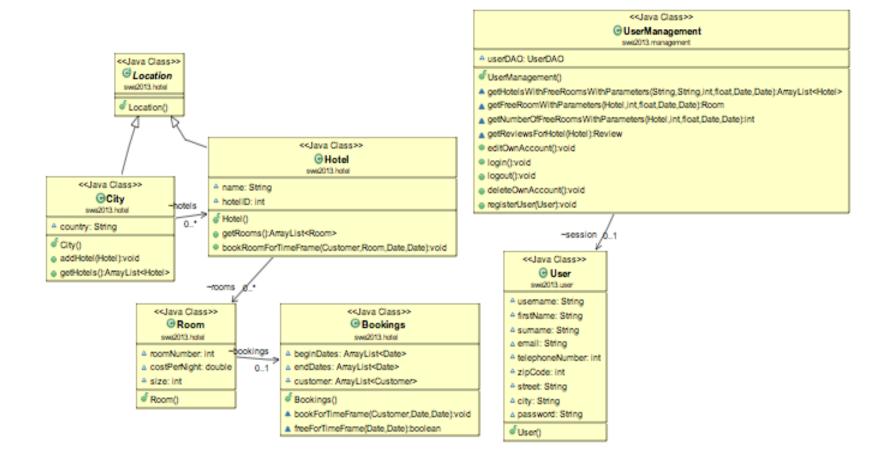
- extends User

.User

- public abstract class

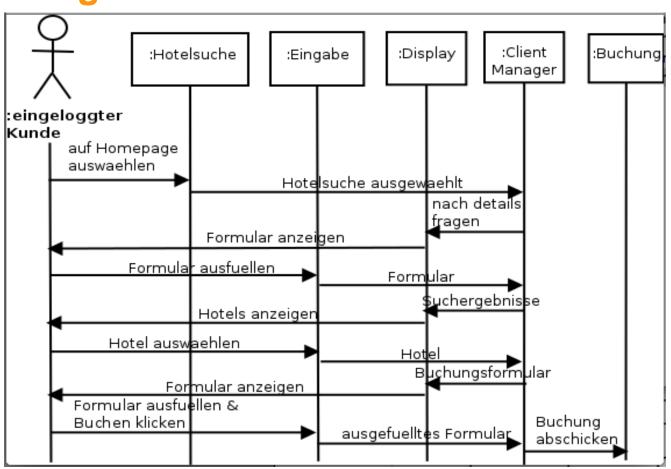


Buchung





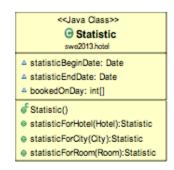
Buchung

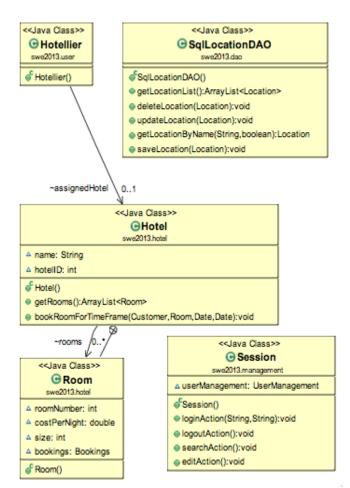




Statistik einsehen Hotellier

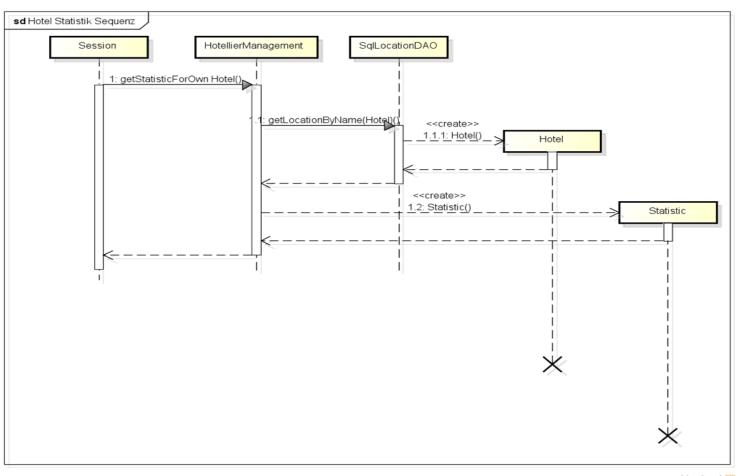






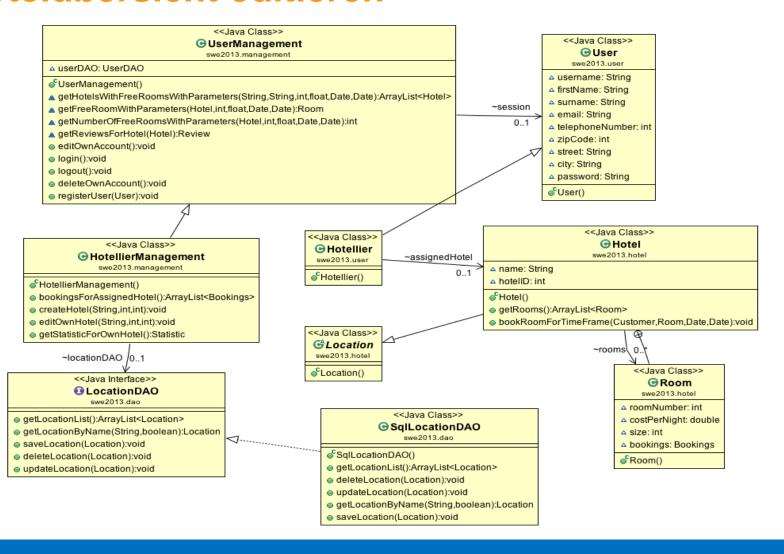


Statistik einsehen Hotellier





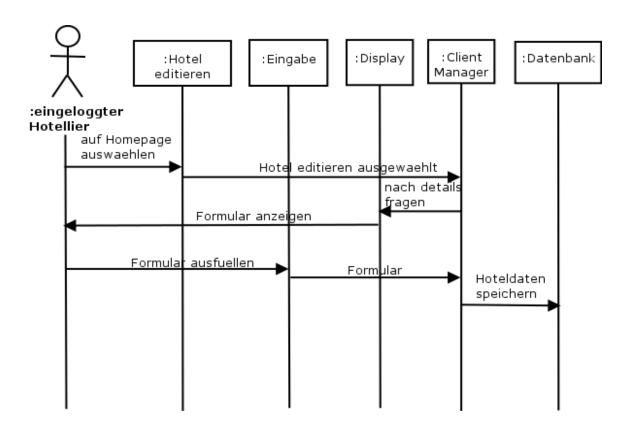
Hotelübersicht editieren







Hotelübersicht editieren



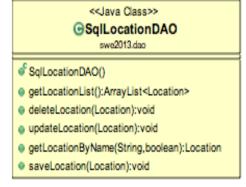


Hoteldaten editieren



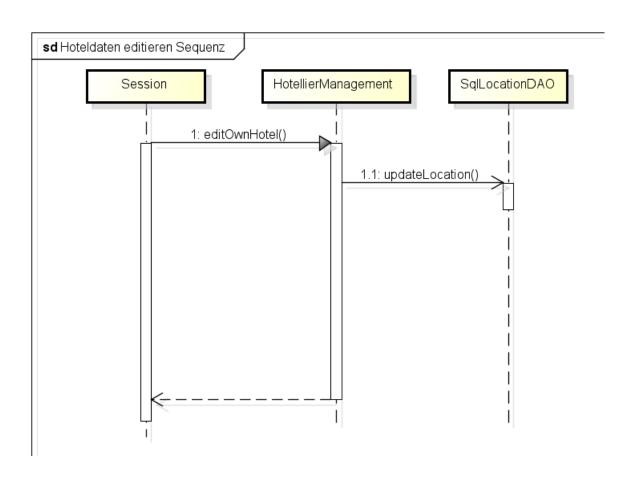






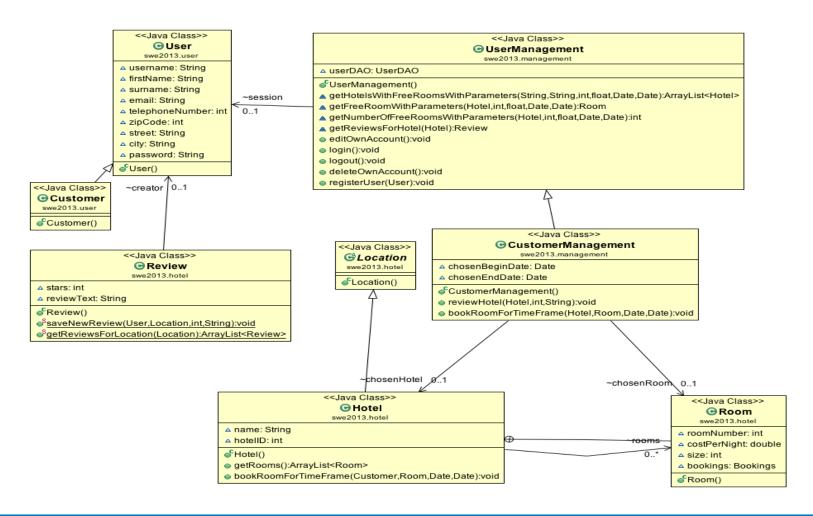


Hoteldaten editieren





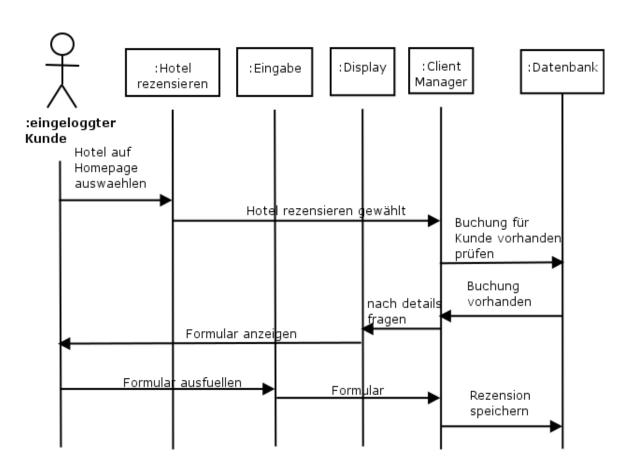
Rezenssion schreiben





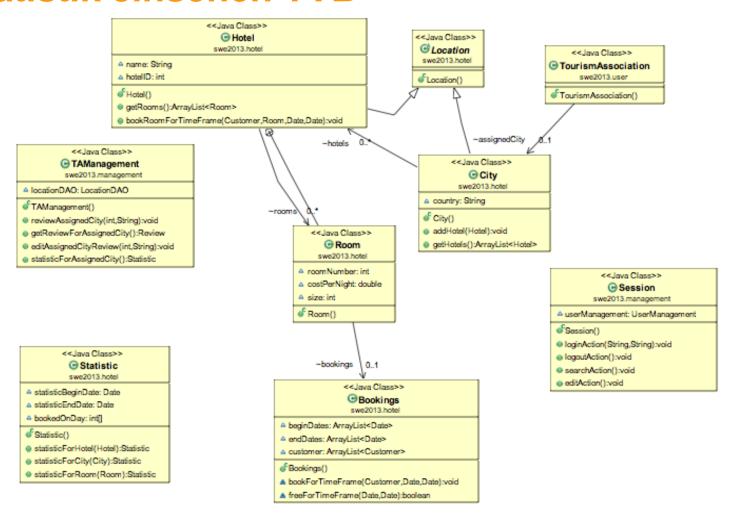


Rezenssion schreiben



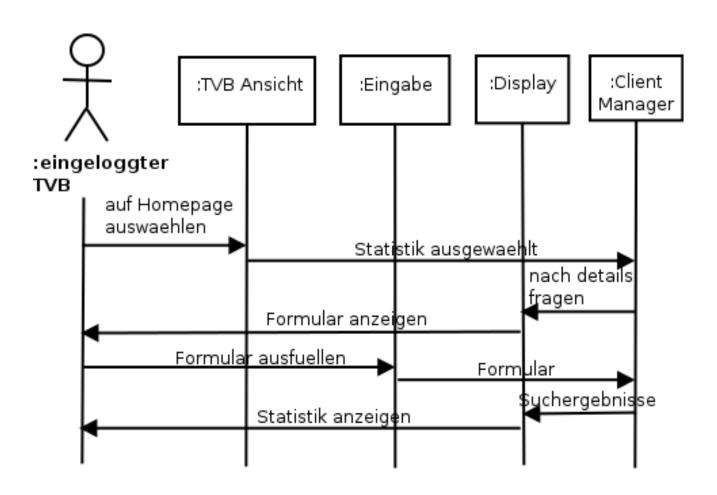


Statistik einsehen TVB



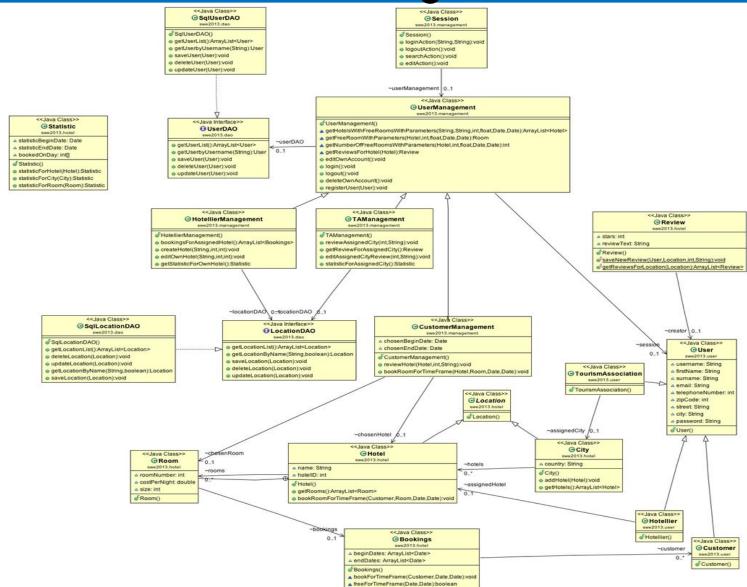


Statistik einsehen TVB



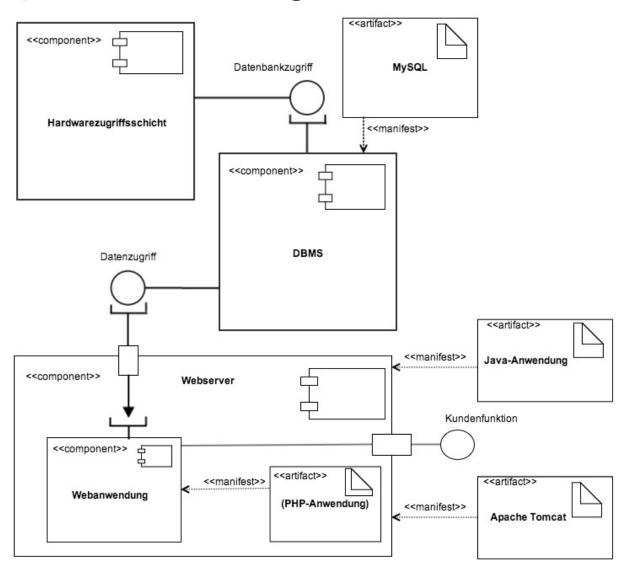
Übersichtsklassendiagramm





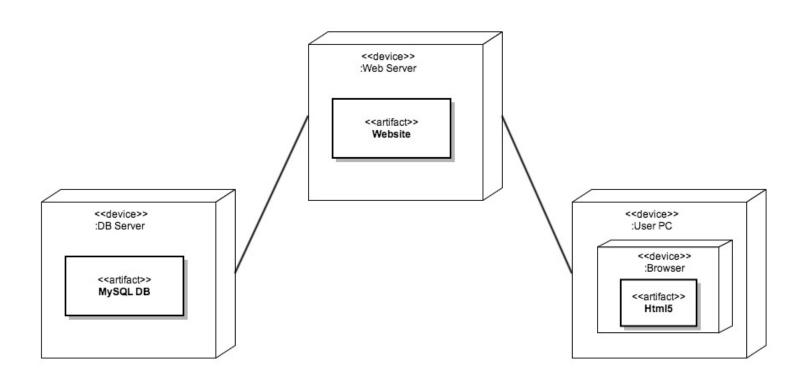


Komponentendiagramm









Datenspeicherung



User, Hotel und Buchungsdaten werden im Rahmen unserer Aufgabenstellung persistent mittels MySQL in einer Datenbank gespeichert. Die nötige Infrastruktur wird durch den ZID zur Verfügung gestellt (http://zid.univie.ac.at/mysql/).

Programmintern werden die Zugriffe auf die SQL Datenbank im Rahmen der Klassen SqlUserDAO und SqlLocationDAO mittels der *Java Database Connectivity (JDBC) API* Version 3.0 realisiert.



Fragen? Anmerkungen?

Vielen Dank für Eure Aufmerksamkeit!