

# Software Engineering Übung, LVNr: 8\_H2

Übungsleiter: Hans Moritsch

Dokument: Design 2

## Projekttitel: Orbit Hotel

Projekthomepage:

<http://www.unet.univie.ac.at/~a1200993/>

Gruppenmitglieder:

MatNr:	Nachname:	Vorname:	e-mail:
1203094	Christl	Korbinian	korbson@gmx.net
1206994	Lazarus	Michael	michaellazarus@hotmail.de
1200993	Lomasow	Veronika	veronika.lomasow@gmx.at
1247136	Chlup	Sebastian	Seb.c21@gmx.at

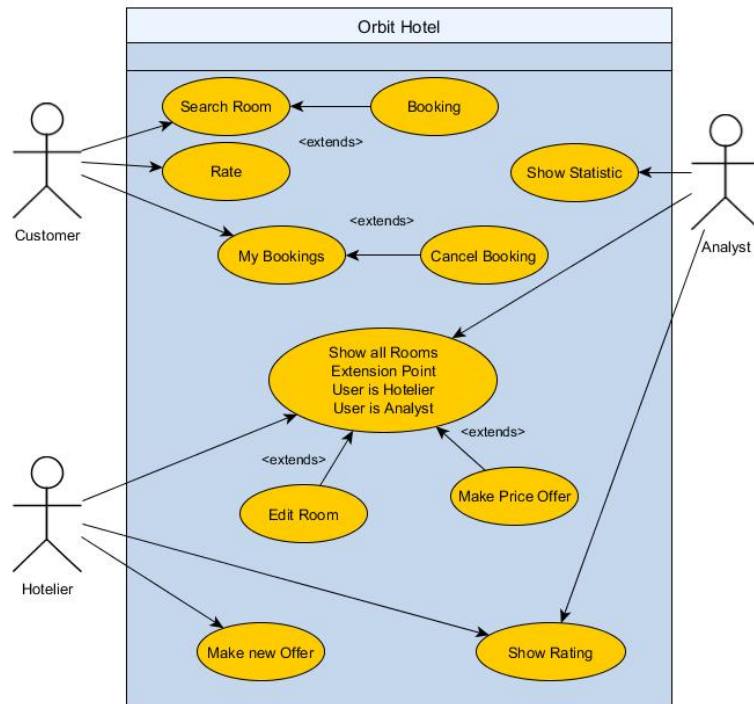
Git Repository:

<https://github.com/MichiLazarus/Hotelreservierung>

Datum: 23.11.2013

# 1 Use Cases, Sequenzdiagramme und Use Cases Realization Design

## 1.1 Use Cases



### 1.1.1 Use Case 1

- Use Case Name: „Search Room”
- Ziel:
  - o Ausgabe der verfügbaren Zimmer, Zimmerpreise, Ausstattung, Maximalbelegung und Rating dieser. Ermöglichung des Booking Vorgangs.
- Kategorie:
  - o Primär
- Vorbedingung:
  - o Der Customer muss zum Ausführen des Use Cases eingeloggt bzw. registriert sein.
- Nachbedingung bei Erfolg:
  - o Customer kann entweder ein Zimmer buchen oder eine erneute Zimmerabfrage erstellen.
- Nachbedingung bei Fehlschlag:
  - o Customer kann eine erneute Zimmerabfrage starten.
- Beteiligte Akteure:
  - o Customer

### 1.1.2 Use Case 1 Beschreibung

- Auslösendes Ereignis:
  - o Klick auf den Suchbutton inkl. Eingabe der Suchkriterien.
- Beschreibung Basisablauf:
  - o Customer gibt Suchkriterien wie Preis, Personenanzahl oder Ausstattung ein und klickt auf den Suchbutton. Bei keiner Eingabe von Suchkriterien werden alle Zimmer angezeigt. Danach werden treffende Zimmer angezeigt. Weiters werden Informationen über das jeweilige Zimmer angegeben. Der Customer hat darauf die Möglichkeit das jeweilige Zimmer zu buchen.

### 1.1.3 Use Case 2

- Use Case Name:
  - o "Rate Booking"
- Ziel:
  - o Bewertung der Zimmer.
- Kategorie:
  - o Primär
- Vorbedingung:
  - o Customer muss registriert und eingeloggt sein.
- Nachbedingung bei Erfolg:
  - o Customer kann Wünsche, Anregungen, Beschwerden mitteilen.
- Nachbedingung bei Fehlschlag:
  - o Erneuter Versuch Bewertung abzugeben.
- Beteiligte Akteure:
  - o Customer

### 1.1.4 Use Case 2 Beschreibung

- Auslösendes Ereignis:
  - o Nach eingeben, der Buchungsnummer und des Bewertungstextes kann man auf "Rate" klicken.
- Beschreibung Basisablauf:
  - o Customer gibt Buchungsnummer ein und verfasst einen Bewertungstext. Im Anschluss an den Klick auf den "Rate" Button wird die Bewertung abgeschickt und in ein Textdokument des zugehörigen Zimmers gespeichert.

### 1.1.5 Use Case 3

- Use Case Name:
  - o "Cancel Booking"
- Ziel:
  - o Stornierung eines zuvor gebuchten Zimmers.

- Kategorie:
  - o Primär
- Vorbedingung:
  - o Customer muss registriert und eingeloggt sein und ein Zimmer gebucht haben.
- Nachbedingung bei Erfolg:
  - o Zimmer wird storniert.
- Nachbedingung bei Fehlschlag:
  - o Zimmer wurde nicht storniert und Buchungsseite wird angezeigt.
- Beteiligte Akteure:
  - o Customer

#### 1.1.6 Use Case 3 Beschreibung

- Auslösendes Ereignis:
  - o Customer klickt auf die zu stornierende Buchung.
- Beschreibung Basisablauf:
  - o Customer sieht sich seine Buchungen über „My Bookings“ an und klickt auf den „CancelBooking“ Button bei der zu stornierenden Buchung. Daraufhin wird bei gültiger Eingabe die Buchung storniert. Der Zimmerstatus wird auf "Verfügbar" gesetzt.

#### 1.1.7 Use Case 4 Beschreibung

- Use Case Name:
  - o "Booking"
- Ziel:
  - o Buchung eines Zimmers
- Kategorie:
  - o Optional
- Vorbedingung:
  - o Der Customer muss registriert und eingeloggt sein. Das Zimmer muss mittels „Search Room“ gesucht werden.
- Nachbedingung bei Erfolg:
  - o Das Zimmer ist gebucht. Buchungsbestätigung mit Buchungsnummer erscheint.
- Nachbedingung bei Fehlschlag:
  - o Zimmer wurde nicht gebucht und kann erneut gebucht werden.
- Beteiligte Akteure:
  - o Customer

#### 1.1.8 Use Case 4 Beschreibung

- Auslösendes Ereignis:
  - o Nach Eingabe der Suchkriterien und klick auf den "Search" Button, wird Liste mit allen zur Verfügung stehenden Räumen, welche den Suchkriterien entsprechen ausgegeben. Der Customer wählt eines

dieser Zimmer und kommt anschließend in die Buchungsmaske wo die Daten zur Buchung bestimmt werden.

- Beschreibung Basisablauf:
  - o In der Buchungsmaske ist es essenziell, dass der User alle Informationen wie Datum, Zahlungsart und Personenanzahl eingibt. Im Anschluss klickt er auf den "Buchen" Button wodurch der Status des Zimmers auf "belegt" geändert wird. Der Customer kann sich seine Buchungen nun über „My Bookings“ ansehen.

#### 1.1.9 Use Case 5

- Use Case Name:
  - o "Show Rating"
- Ziel:
  - o Einsehen der Bewertungen der Customer.
- Kategorie:
  - o Primär
- Vorbedingung:
  - o Hotelier oder Analyst muss eingeloggt sein und Bewertungen müssen vorhanden sein.
- Nachbedingung bei Erfolg:
  - o Hotelier oder Analyst kann Bewertungen eines bestimmten Zimmers einsehen.
- Nachbedingung bei Fehlschlag:
  - o Keine Bewertung vorhanden („No rating available“) oder neuer Versuch Bewertung einzusehen.
- Beteiligte Akteure:
  - o Hotelier, Analyst

#### 1.1.10 Use Case 5 Beschreibung

- Auslösendes Ereignis:
  - o Klick auf "ShowRating" Button bei Show Rating nach Eingabe der Raumnummer.
- Beschreibung Basisablauf:
  - o Hotelier oder Analyst gibt im Feld "Show Rating" eine bestimmte Raumnummer ein und klickt auf den "ShowRating" Button. Im Anschluss werden die Bewertungen für dieses Zimmer angezeigt.

#### 1.1.11 Use Case 6

- Use Case Name:
  - o "Alle Zimmer anzeigen"
- Ziel:
  - o Ausgabe der verfügbaren Zimmer, Zimmerpreise, Ausstattung, Maximalbelegung und Rating dieser.
- Kategorie:
  - o Primär
- Vorbedingung:
  - o User muss eingeloggt sein.
- Nachbedingung bei Erfolg:
  - o Dem User werden alle Zimmer angezeigt.
- Nachbedingung bei Fehlschlag:
  - o Erneuter Versuch Zimmer einzusehen bzw. zurückkehren auf das jeweilige "Hauptmenü" (HotelierInterface/AnalystInterface).
- Beteiligte Akteure:
  - o Hotelier, Analyst

#### 1.1.12 Use Case 6 Beschreibung

- Auslösendes Ereignis:
  - o Klick auf den "Show" Button im Feld "Show all Rooms"
- Beschreibung Basisablauf:
  - o User klickt auf den "Show" Button im Feld "Show all Rooms". Er bekommt eine Tabelle zurückgeliefert mit dem Inhalt "Rommnummer", "Equipment", "Number of Persons", "Price", „OfferedPrice“, „Rating“ und für Hoteliers die Möglichkeit, Zimmer zu editieren bzw. für Analysten die Möglichkeit ein Preisangebot zu erstellen.

#### 1.1.13 Use Case 7

- Use Case Name:
  - o "Zimmer Editieren"
- Ziel:
  - o Editierung von "Price", "Equipment" und "Number of Persons" eines Zimmers.
- Kategorie:
  - o Optional
- Vorbedingung:
  - o Hotelier muss eingeloggt sein.
- Nachbedingung bei Erfolg:
  - o Zimmer wurde editiert und weiter Zimmer können editiert werden.
- Nachbedingung bei Fehlschlag:
  - o Erneuter Versuch das selbe Zimmer zu editieren.
- Beteiligte Akteure:
  - o Hotelier

#### 1.1.14 Use Case 7 Beschreibung

- Auslösendes Ereignis:
  - o Klick auf den "Edit" Button in der Maske "Show all Rooms".
- Beschreibung Basisablauf:
  - o Über das Interface, welches Alle Räume anzeigt, kann der Hotelier auf den "EditRoom" Button eines bestimmten Zimmers klicken, wodurch er in die Maske zum Editieren des Zimmers gelangt und kann dadurch "Price", "Equipment", "Number of Persons" und editieren. Die veränderten Informationen werden gespeichert.

#### 1.1.15 Use Case 8

- Use Case Name:
  - o "Make new Offer"
- Ziel:
  - o Erstellen eines neues Angebotes
- Vorbedingung:
  - o Hotelier muss eingeloggt sein.
- Nachbedingung bei Erfolg:
  - o Angebot wird erfolgreich erstellt.
- Nachbedingung bei Fehlschlag:
  - o Angebot konnte nicht erstellt werden, erneuter Versuch.
- Beteiligte Akteure:
  - o Hotelier

#### 1.1.16 Use Case 8 Beschreibung

- Auslösendes Ereignis:
  - o Beim Feld "New Room", "Create" Button klicken.
- Beschreibung Basisablauf:
  - o Hotelier gibt "Rommnumber", "Price", "Equipment" und "Number of Persons" ein und klickt auf den "Create" Button. Das Angebot wurde erstellt.

#### 1.1.17 Use Case 9

- Use Case Name:
  - o „Show Statistic“
- Ziel:
  - o Einblick in öffentliche Statistik.
- Kategorie:
  - o Primär
- Vorbedingung:
  - o Login, Vorhandensein von Reservierungs- und Kundendaten

- Nachbedingung bei Erfolg:
  - Statistik vorhanden
- Beteiligte Akteure:
  - Analyst

#### 1.1.18 Use Case 9 Beschreibung

- Auslösendes Ereignis:
  - Klicken auf die jeweilige Statistik bzw. Auswahl der Kriterien wie „Year“ welche für die Eingabe benötigt werden.
  - Beschreibung Basisablauf:  
Der Analyst wählt die bevorzugten Kriterien der zu anzeigenden Statistik ein und klickt dann auf den dazugehörigen Button (Button: „SeasonStatistic“ zur Saisonalen Statistik, „SeasonStatistic“). Falls abrufbare Daten vorhanden sind, wird eine passende Statistik zu Beliebtheit des Angebotes oder saisonalen Reservierungen autogeneriert.

#### 1.1.19 Use Case 10

- Use Case Name:
  - "Make Price Offer"
- Ziel:
  - Erstellen eines Preisvorschlages
- Kategorie:
  - Primär
- Vorbedingung:
  - Analyst muss eingeloggt sein
- Nachbedingung bei Erfolg:
  - Preisvorschlag wurde erstellt
- Nachbedingung bei Fehlschlag:
  - Der Analyst hat erneut die Chance ein Zimmer zu wählen und einen Preisvorschlag zu machen.
- Beteiligte Akteure:
  - Analyst

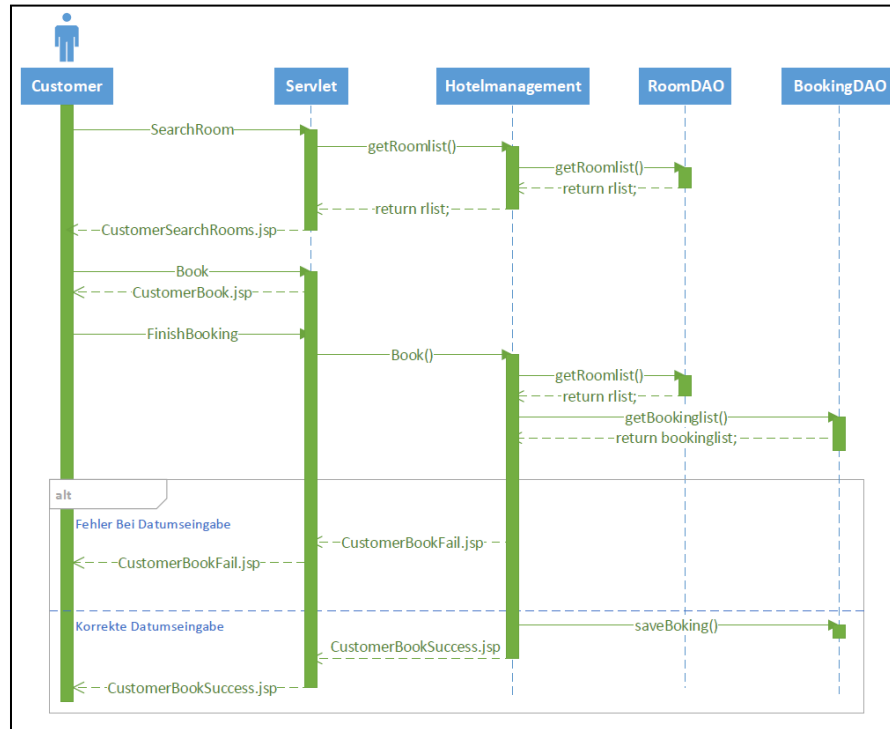
#### 1.1.20 Use Case 10 Beschreibung

- Auslösendes Ereignis:
  - Klick auf den "MakePriceOffer" Button.
- Beschreibung Basisablauf:
  - Der Analyst wählt zunächst die Option, sich alle Zimmer mittels „Show all Rooms“ anzeigen zu lassen und hat dort die Möglichkeit, in einem Textfenster einen Preisvorschlag, welcher mit „MakePriceOffer“ gespeichert wird, einzugeben.

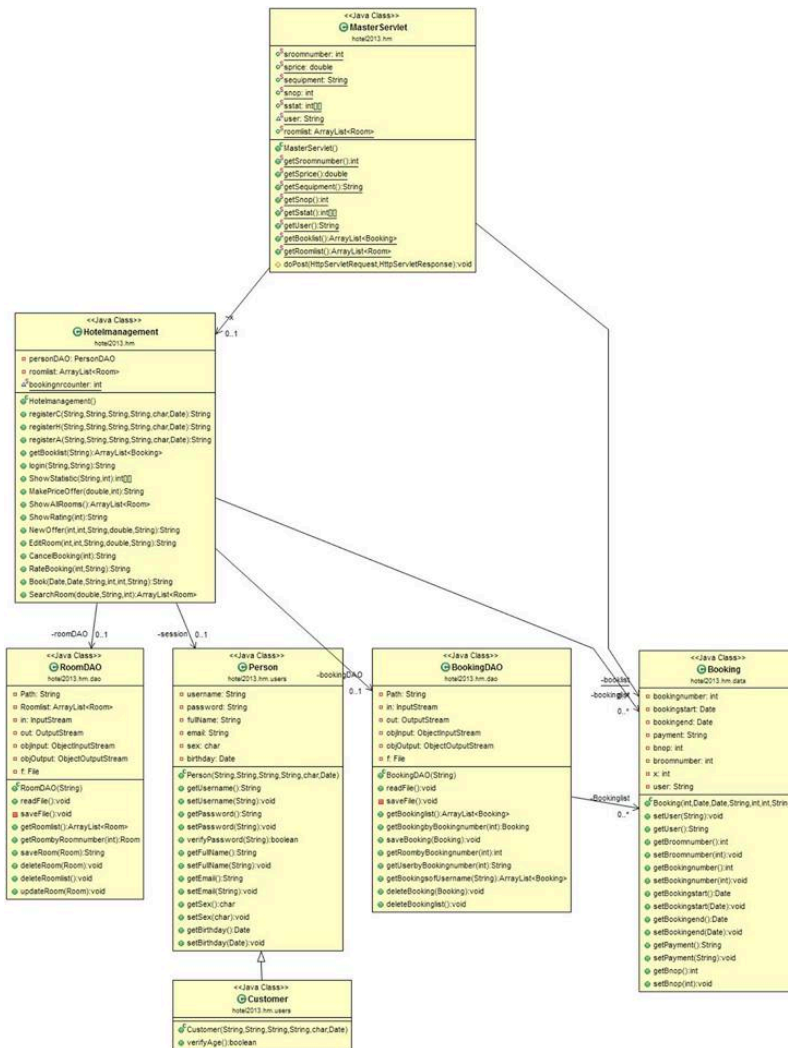


## 1.2 Use Case Realization Design

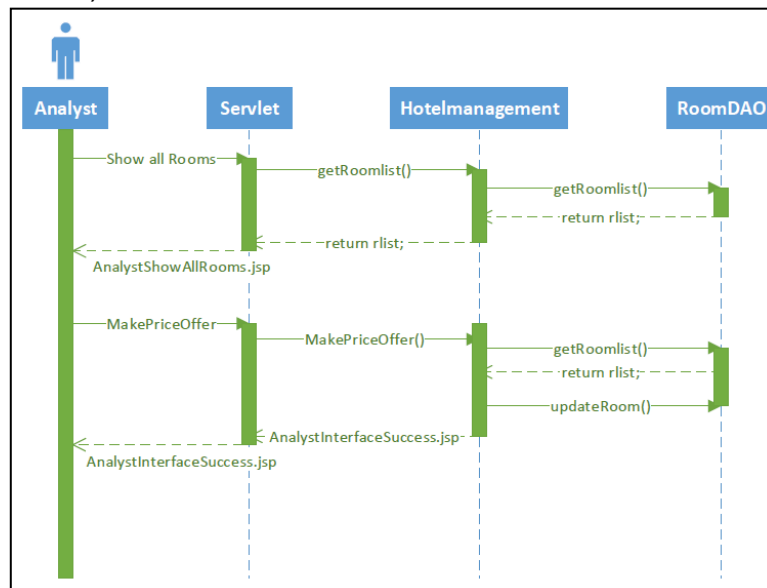
### 1.2.1 Sequenzdiagramm 1 (UseCase „Booking“ & „Search Room“)



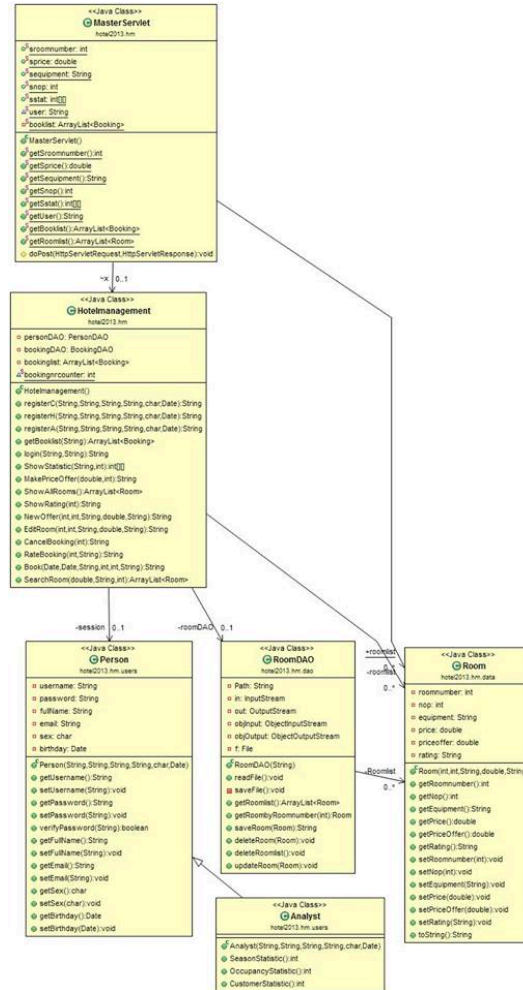
## 1.2.2 Klassendiagramm 1 (Use Case „Booking“ & „Search Room“)



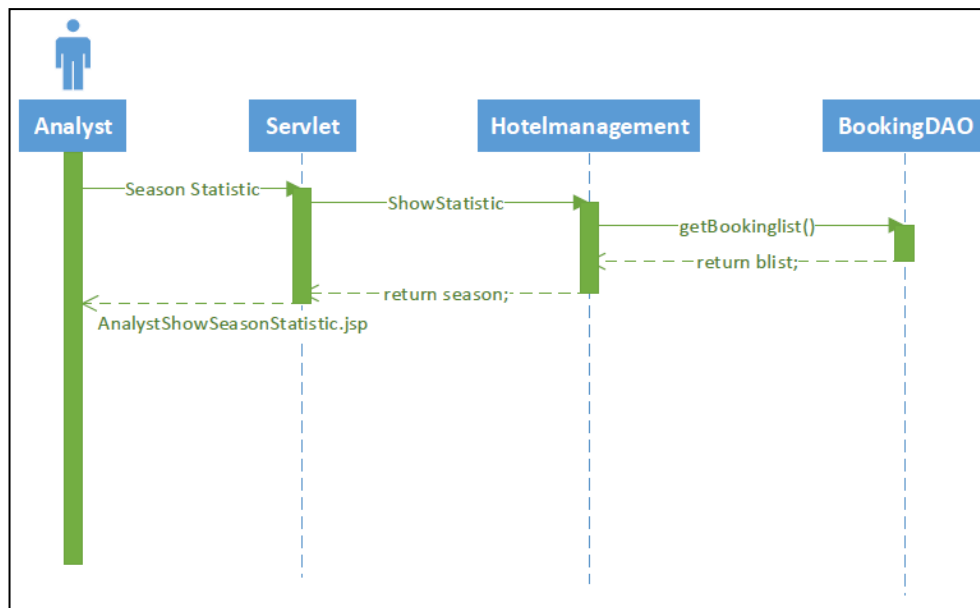
## 1.2.3 Sequenzdiagramm 2 (Use Case “Make Price Offer” & “Show all Rooms”)



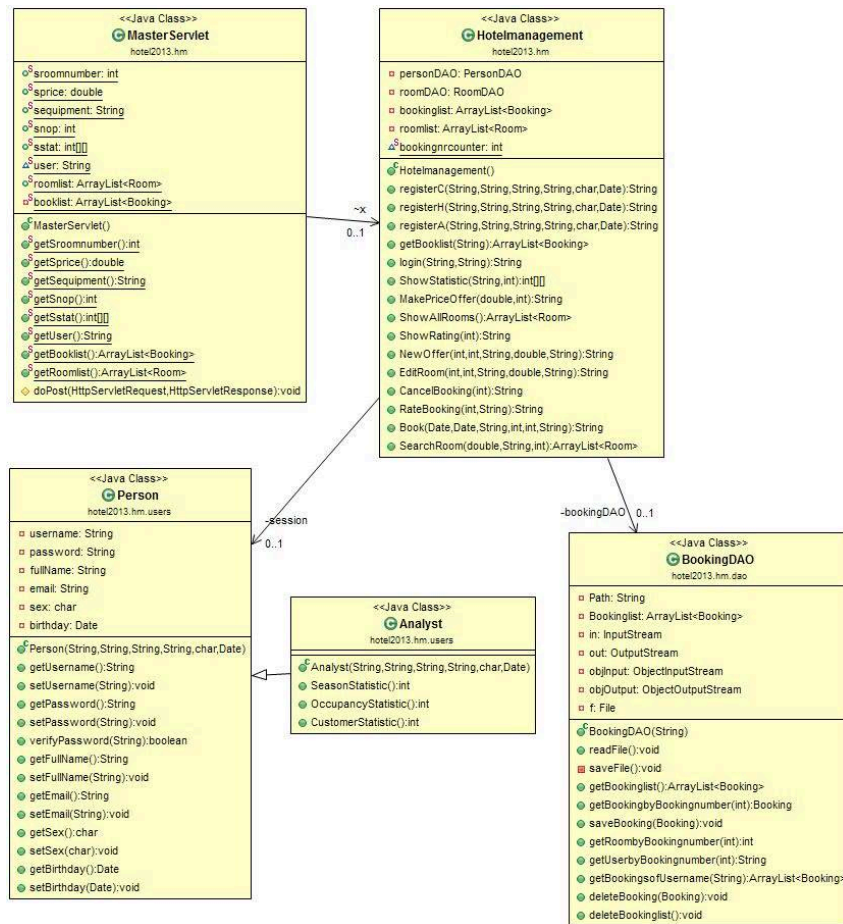
## 1.2.4 Klassendiagramm 2 (Use Case “Make Price Offer” & “Show all Rooms”)



## 1.2.5 Sequenzdiagramm 3 (Use Case “Show Statistic”)



## 1.2.6 Klassendiagramm 3 (Use Case Make Price Offer)



## 1.3 Use Case Realization Beschreibung

### 1.3.1 Beschreibung 1

#### Sequenzdiagramm 1 (Use Case „Booking“ & „Search Room“)

Der Customer sendet den Befehl „Search Room“ an das Servlet, welches an das Hotelmanagement die Nachricht bzw. die Methode „getRoomlist()“ schickt, um die Zimmer des Hotels zu erhalten. Das Hotelmanagement sendet anschließend an die Klasse RoomDAO die Methode getRoomlist(), um aus der .ser Datei die Zimmer zu lesen. Anschließend sendet das RoomDAO die rlist mit den Zimmern des Hotels an das Hotelmanagement, welches diese an das Servlet sendet. Das Servlet schickt an den Customer die „CustomerSearchRooms.jsp“ Verlinkung.

Um anschließend eine Buchung durchzuführen, schickt der Customer die Nachricht „Book“ an das Servlet, dieses speichert sich allerdings die roomlist temporär ab und kann somit sofort mit der Verlinkung „CustomerBook.jsp“ antworten. Damit die Buchung beendet wird, schickt der Customer die eingegebenen Daten mit „FinishBooking“ an das Servlet, dieses ruft die Methode Book im Hotelmanagement auf. Anschließend wird die Nachricht „getRoomlist()“ an das RoomDAO gesendet und weiters die Nachricht „getBookinglist()“ an das BookingDAO. Findet das Hotelmanagement nach dem Erhalt der beiden Listen einen Fehler in der Datumseingabe, sendet dieses die Nachricht „CustomerBookFail.jsp“ an das Servlet zurück. Dieses Antwortet dann eben mit dieser Verlinkung dem Customer. Ist die Datumseingabe korrekt, so sendet das Hotelmanagement die Nachricht „CustomerBookSuccess.jsp“ an das Servlet und dieses die Verlinkung mit derselben Nachricht an den Customer.

### 1.3.2 Beschreibung 2

#### Sequenzdiagramm 2 (Use Case „Make Price Offer“ & „Show all Rooms“)

Die Interaktion beginnt damit, dass der Analyst die Nachricht „Show all Rooms“ an das Servlet sendet. Dieses schickt die Methode „getRoomlist()“ an das Hotelmanagement, welches sich mit derselben Methode das Array mit allen Raumobjekten aus der Klasse RoomDAO holt. Das RoomDAO und das Hotelmanagement senden das Array weiter an das Servlet und dieses sendet die Verlinkung „AnalystShowAllRooms.jsp“ an den Analysten.

Um dann ein Preisangebot abzugeben sendet der Analyst die Nachricht „MakePriceOffer“ an das Servlet, dieses schickt die Meldung „MakePriceOffer()“ an das Hotelmanagement, welches sich erneut die roomlist aus dem RoomDAO holt. Nachdem das Array mit allen Zimmern nun beim Hotelmanagement angekommen ist, speichert das Hotelmanagement den neuen „offeredPrice“ in das jeweilige Zimmer und sendet die Nachricht „UpdateRoom()“ an die Klasse RoomDAO. Anschließend sendet das Hotelmanagement die Antwortnachricht „AnalystInterfaceSuccess.jsp“ an das Servlet und dieses dieselbe Nachricht an den Analysten.

### 1.3.3 Beschreibung 3

Sequenzdiagramm 3 (Use Case „Show Statistic“)

Um eine die saisonale Statistik zu erhalten sendet der Analyst die Nachricht „Season Statistic“ an das Servlet, welches die Methode „ShowStatistic“ an das Hotelmanagement sendet. Das Hotelmanagement sendet die Nachricht getBooklist() an das BookingDAO, welches die Bookinglist „blist“ returned. Darauf sendet das Hotelmanagement ein Array „Season“ an das Servlet zurück, welches die Verlinkung „AnalystShowSeasonStatistic.jsp“ an den Analysten sendet.

## 2 Klassendesign

### 2.1 Klassenbeschreibungen

#### 2.1.1 Klasse 1

##### Person:

Person ist die Upperclass von Analyst, Hotelier und Customer. Sie beinhaltet alle notwendigen privaten Attribute, um sich einzuloggen oder zu registrieren. Des Weiteren sind hier auch alle notwendigen public get- und set-Methoden implementiert, um auf die Variablen zugreifen zu können bzw. sie zu verändern. Dazu kommt noch die Methode „verifyPassword“, welche zur Überprüfung der Gültigkeit der Eingabe verwendet wird. Diese Klasse wird benötigt, damit alle unterschiedlichen Subclassen von ihr erben können.

#### 2.1.2 Klasse 2

##### Customer:

Customer ist eine SubClass der Klasse Person. Dies bedeutet, dass sie sämtliche Methoden und Variablen dieser erbt. Mit dieser Klasse wird der Otto-

Normalverbraucher dargestellt. Zusätzlich enthält er noch die public Methode „verifyAge“, welche sicherstellt, dass der sich registrierende Benutzer volljährig ist.

### 2.1.3 Klasse 3

#### Analyst:

Customer ist eine SubClass der Klasse Person. Dies bedeutet, dass sie sämtliche Methoden und Variablen dieser erbt. Diese Klasse repräsentiert den Mitarbeiter eines Unternehmens, welches sich auf Analysen für Hotelstatistiken spezialisiert hat. Um die verschiedenen Statistiken zu erstellen, besitzt er die public Methoden „SeasonStatistic“, „OccupancyStatistic“ und „CustomerStatistic“. Diese greifen auf die bereits gespeicherten Daten zu und autogenerieren je nach Anforderung die gewünschte Statistik.

### 2.2.4 Klasse 4

#### Hotelier:

Hotelier ist eine SubClass der Klasse Person. Dies bedeutet, dass sie sämtliche Methoden und Variablen dieser erbt. Diese Klasse repräsentiert den SuperUser, mit dem Zimmerangebote erstellt bzw. bearbeitet werden können. Der Hotelier wird im Usermanagement benötigt, um Rechte für Analyst und Customer einzuschränken.

### 2.2.5 Klasse 5

#### SaveUsers:

Diese Klasse dient der Speicherung, sowie dem Auslesen der registrierten und angelegten Personen. Sie beinhaltet die dazu benötigten In- und Outputstreams. Auf diese wird mittels der public Methoden „readFile“, auslesen der ArrayList in der alle UserObjekte gespeichert sind, „saveFile“ zur Speicherung der ObjektArrayList in das File, „getUserList“ zum Aufrufen der ausgelesenen ArrayList und „getUserByUsername“ um ein bestimmtes Object aus der Liste zu suchen.

### 2.2.6 Klasse 6

#### Booking:

Hierbei wird das Objekt einer Buchung erzeugt. Diese enthält die privaten Variablen „bookingnumber“, „bookingstart“ (ab wann eine Buchung gültig ist), „bookingend“ (wann die Buchung endet), „payment“ (Art der Bezahlung, mit Kreditkarte oder in Bar) und „bnop“ (booking number of persons). Zu diesen Variablen hinzu, kommen die public get- und set- Methoden, um auf die Variablen zugreifen zu können bzw. diese zu ändern.

### 2.2.7 Klasse 7

### SaveBooking:

Diese Klasse wird dazu verwendet, dass erfolgte Buchungen in ein File gespeichert werden und wieder ausgelesen werden können. Auf diese wird mittels der public Methoden „readFile“, auslesen der ArrayList in der alle BuchungsObjekte gespeichert sind, „saveFile“ zur Speicherung der ObjektArrayList in das File, „getBookingList“ zum Aufrufen der ausgelesenen ArrayList und „getBookingbyBookingnumber“ um ein bestimmtes Object aus der Liste zu suchen.

## 2.2.8 Klasse 8

### Room:

Diese Klasse repräsentiert das anzulegende Objekt Raum. Sie besteht sowohl aus den private Variablen „roomnumber“, „nop“(number of persons), „equipment“(besondere zusätzliche Ausstattung), „price“, „priceoffer“(vom Analysten vorgeschlagener Preis), „rating“(vom Kunden erstellte Bewertung der Zimmer) und „status“(ist Zimmer frei oder belegt), als auch den benötigten public get- und set- Methoden.

## 2.2.9 Klasse 9

### SaveRooms:

Diese Klasse wird dazu verwendet, um erzeugte Objekte vom Typ Room in ein File zu speichern und diese wieder auszulesen. Auf diese wird mittels der public Methoden „readFile“ aus der ArrayList zugegriffen, in der alle RoomObjekte gespeichert sind, „saveFile“ zur Speicherung der ObjektArrayList in das File, „getRoomList“ zum Aufrufen der ausgelesenen ArrayList und „getRoombyRoomnumber“ um ein bestimmtes Objekt aus der Liste zu suchen.

## 2.2.10 Klasse 10

### Hotelmanagement:

Stellt die Verwaltungsklasse unseres Projektes dar. Von dieser Klasse ausgehend wird auf alle anderen zugegriffen. Zusätzlich sind die ihr zugewiesenen Methoden nur für die spezifischen Subklassen von Person ausführbar. Für den Customer : „register“, „login“, „searchroom“, „Book“, „CancelBooking“, „RateBooking“. Dem Hotelier stehen die public Methoden: „ShowAllRooms“, „ShowRating“, „NewOffer“ und „Edit“ zur Verfügung. Schlussendlich hat der Analyst die Möglichkeit auf: „ShowAllRooms“, „ShowStatistic“ und „MakePriceOffer“ zuzugreifen. Diese Klasse wird benötigt um alle notwendigen Methoden zu verwalten.



### 2.2.11 Klasse 11

#### MasterServlet:

Dient als Schnittstelle zwischen JSP und dem Java Programm. Mittels `request.getParameter("")` werden die vom JSP übergebenen Werte übernommen und im Anschluss daran an die Klasse Hotelmanagement übergeben. Des Weiteren werden auch Variablen, welche in den JSPs ausgegeben werden müssen fixiert und get-Methoden hierfür implementiert. Anhand der Variable "check" werden vom Hotelmanagement zurückgelieferte JSPs übernommen und im Anschluss daran aufgerufen.

## 2.2 Wichtige Designentscheidungen

Die Klasse Person wurde gewählt, um eine einfache Erweiterbarkeit des Hotelreservierungssystems zu ermöglichen. In dieser Klasse wurden die wichtigsten Attribute und Methoden definiert, welche in jeder Subklasse benötigt werden.

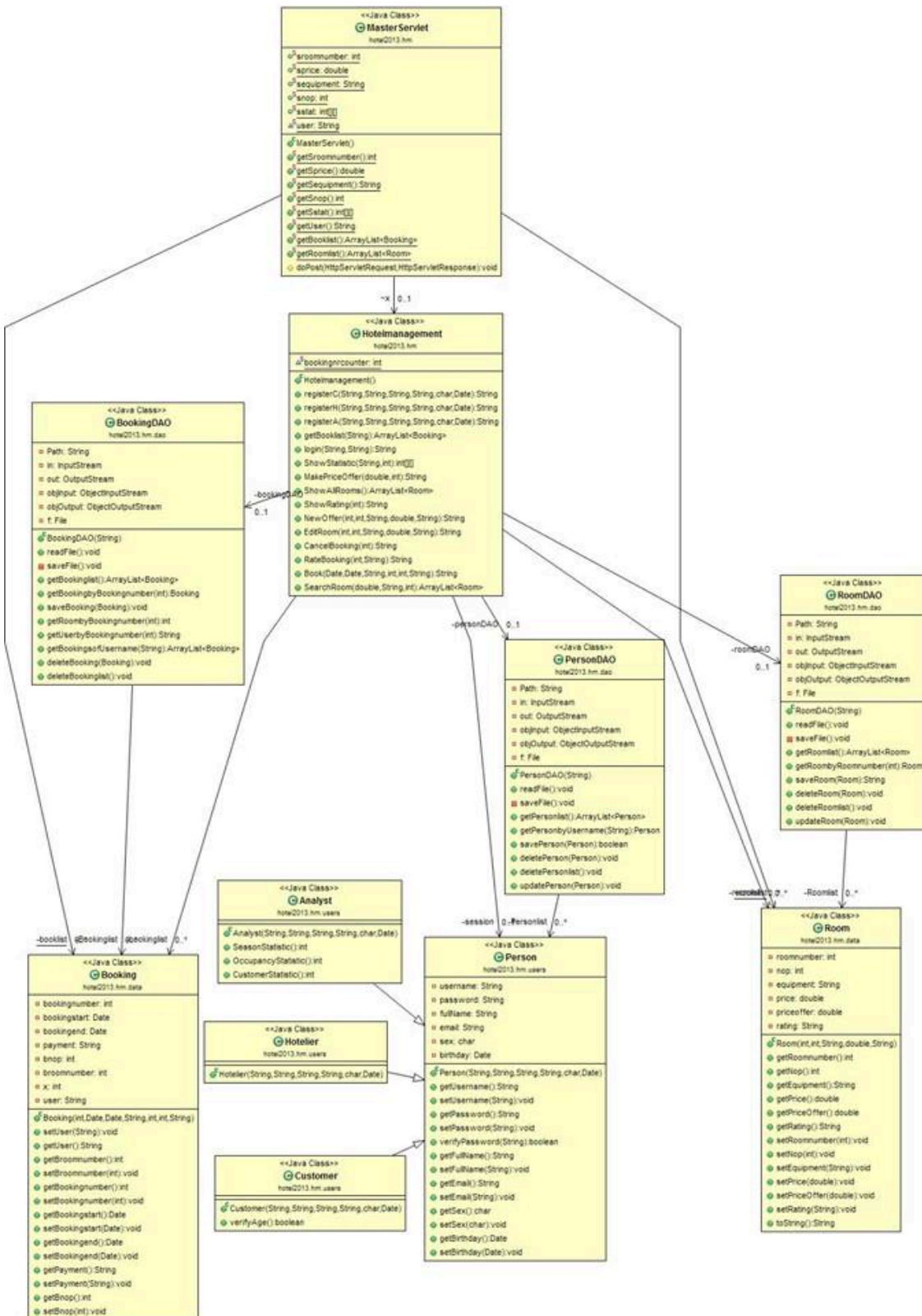
Der Hotelier wird im Hotelmanagement wichtig, da dort mittels „instanceof“ unterschieden wird, welche Methoden und Aktionen von den unterschiedlichen Usern ausgeführt werden dürfen. Ansonsten hat der Hotelier keine zusätzlichen Funktionen.

Um eine Vereinfachung und Übersichtlichkeit der Datenspeicherung zu ermöglichen, wurden 3 unterschiedliche Klassen gewählt, welche in 3 unterschiedliche Dateien serialisieren.

Das Hotelmanagement agiert als zentrale Verbindung zwischen den verschiedenen Klassen des Systems. Über das Hotelmanagement wird auf alle wichtigen Methoden zugegriffen und weiters werden Berechtigungen geprüft.

Die Klasse MasterServlet dient als Schnittstelle zwischen JSP und Hotelmanagement.

# Architekturbeschreibung



## 2.3 Klassendiagramm

## 2.4 Paketstruktur

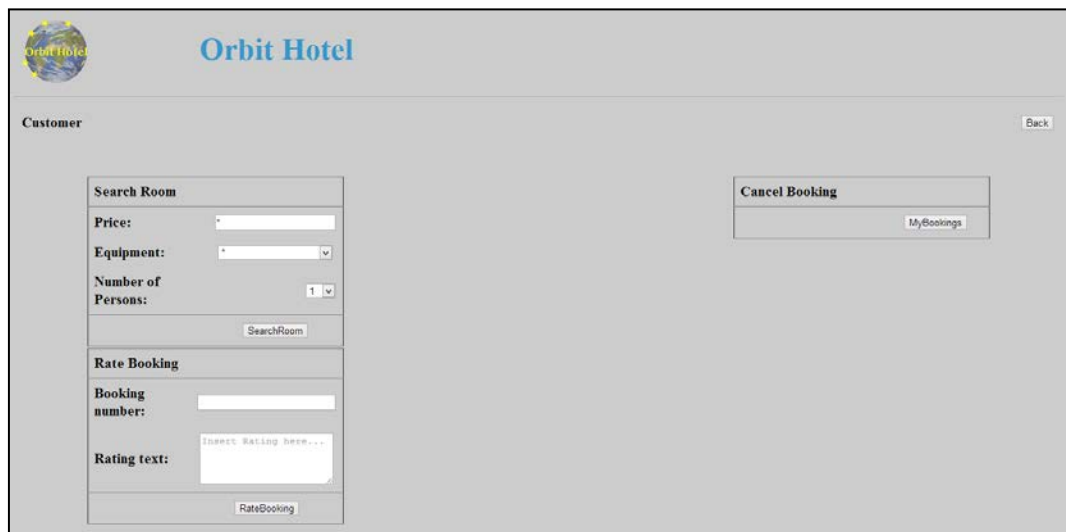
Um die Übersicht über unser Projekt zu behalten haben wir uns dazu entschlossen es in vier verschiedene Pakete zu unterteilen. Diese Pakete beinhalten jeweils verschiedene Arten von Klassen. Die Klassen eines Paketes interagieren untereinander und zueinander. In unserem Oberpaket „hotel2013.hm“ legen wir die Klasse Hotelmanagement an, da in dieser Klasse die wichtigsten Operationen durchgeführt werden, welche alle auf die Klassen der Unterpakete zugreifen. Unser erstes Unterpaket „hotel2013.hm.dao“ beinhaltet die Klassen zur Speicherung bzw. zum Auslesen der Daten „SaveRooms“, „SaveUsers“ und „SaveBookings“. In dem Paket „hotel2013.hm.data“ sind folgende Klassen enthalten: „Room“ und „Booking“, welche zu verwaltende Objekte des Hotels darstellen und welche zur Statistikerstellung notwendig sind. Das letzte Subpaket enthält die User, deren Oberklasse die Klasse „Person“ repräsentiert. Die erbenenden Klassen lauten „Customer“, „Hotelier“ und „Analyst“.

## User Interface Beschreibung

## 2.5 GUI Screenshot / User Interface Skizzen



The screenshot shows the login interface of the Orbit Hotel. At the top left is a small globe icon with the text "Orbit Hotel". To its right, the text "Orbit Hotel" is displayed in a blue font. Below this, there is a central login form. The form has a title "Login" and contains two input fields: "Username" and "Password". Below the password field is a "Login" button. At the bottom of the form is a "Register" button.



The screenshot shows the customer interface of the Orbit Hotel. At the top left is a small globe icon with the text "Orbit Hotel". To its right, the text "Orbit Hotel" is displayed in a blue font. Below this, the word "Customer" is displayed on the left, and a "Back" button is on the right. The main content area is divided into two columns. The left column contains two sections: "Search Room" and "Rate Booking". The "Search Room" section has input fields for "Price:", "Equipment:", and "Number of Persons:", followed by a "SearchRoom" button. The "Rate Booking" section has input fields for "Booking number:" and "Rating text:", followed by a "RateBooking" button. The right column contains a "Cancel Booking" section with a "MyBookings" button.




# Orbit Hotel

Hoteller Back

**Show all Rooms**

**Show Rating**
Room number:

**New Room**
Room number:   
Price / Night:   
Equipment:   
Number of Persons:



# Orbit Hotel

Requested Rooms Back

Roomnr	Equipment	NrOfPersons	Price	Rating	Book
1	Business Suite	2	450.0 Euro	null...	<input type="button" value="Book"/>
2	Economy Class Room	3	25.0 Euro	No rating available...	<input type="button" value="Book"/>
3	Business Suite	2	450.0 Euro	No rating available...	<input type="button" value="Book"/>
4	Superior Suite	1	100.0 Euro	No rating available...	<input type="button" value="Book"/>
5	Superior Suite	1	200.0 Euro	No rating available...	<input type="button" value="Book"/>
6	Business Suite	2	50.0 Euro	No rating available...	<input type="button" value="Book"/>
7	High Class Room	2	30.0 Euro	No rating available...	<input type="button" value="Book"/>
8	Economy Class Room	1	25.0 Euro	No rating available...	<input type="button" value="Book"/>
9	Superior Suite	2	1000.0 Euro	No rating available...	<input type="button" value="Book"/>

Orbit Hotel


Analyst:

1. Show Rating  
Roomnumber:

2. Show Statistic  
Show Statistics:

3. Make Price Offer  
Roomnumber :  Price:

4. Show all Rooms



# Orbit Hotel

Hoteller Back

**Edit Room**
Roomnumber: 1  
Price / Night:  Current: 1300.0  
Equipment:  Current: Superior  
Number of Persons:  Current: 1



## Orbit Hotel

All Rooms

Back

Roomnr	Equipment	NrOfPersons	Price	OfferedPrice	Rating	Edit
1	Business Suite	2	450.0 Euro	-	null...	EditRoom
2	Economy Class Room	3	25.0 Euro	-	No rating available...	EditRoom
3	Business Suite	2	450.0 Euro	-	No rating available...	EditRoom
4	Superior Suite	1	100.0 Euro	-	No rating available...	EditRoom
5	Superior Suite	1	200.0 Euro	-	No rating available...	EditRoom
6	Business Suite	2	50.0 Euro	-	No rating available...	EditRoom
7	High Class Room	2	30.0 Euro	-	No rating available...	EditRoom
8	Economy Class Room	1	25.0 Euro	-	No rating available...	EditRoom
9	Superior Suite	2	1000.0 Euro	-	No rating available...	EditRoom



## Orbit Hotel

Rating for Roomnumber:

Back

3  
No rating available



## Orbit Hotel

Booking

Back

Please enter your booking data:

Roomnr: 2  
Bookingstart: DD.MM.YYYY  
Bookingend: DD.MM.YYYY  
Payment-Method: Bar  
Number of Persons: 1  
FinishBooking Cancel

## Orbit Hotel

Analyst:

Back


1. Statistic:


	Jan	Feb	Mar	Apr	May	June	July	Aug	Sep	Oct	Nov	Dec
Bookings												
Cancelations												
Customer Number												

 <b>Orbit Hotel</b>						
<b>All Rooms</b> <span>Back</span>						
Roomnr	Equipment	NrOfPersons	Price	OfferedPrice	Rating	PriceOffer
1	Business Suite	2	450.0 Euro	-	null...	<input type="text"/> MakePriceOffer
2	Economy Class Room	3	25.0 Euro	-	No rating available...	<input type="text"/> MakePriceOffer
3	Business Suite	2	450.0 Euro	-	No rating available...	<input type="text"/> MakePriceOffer
4	Superior Suite	1	100.0 Euro	-	No rating available...	<input type="text"/> MakePriceOffer
5	Superior Suite	1	200.0 Euro	-	No rating available...	<input type="text"/> MakePriceOffer
6	Business Suite	2	50.0 Euro	-	No rating available...	<input type="text"/> MakePriceOffer
7	High Class Room	2	30.0 Euro	-	No rating available...	<input type="text"/> MakePriceOffer
8	Economy Class Room	1	25.0 Euro	-	No rating available...	<input type="text"/> MakePriceOffer
9	Superior Suite	2	1000.0 Euro	-	No rating available...	<input type="text"/> MakePriceOffer

<b>Orbit Hotel</b>	
Analyst:	Back
1. Statistic:	
<input type="text"/>	
Customers average age:	
<input type="text"/>	
Customers Sex:	Male: <input type="text"/> Female: <input type="text"/>
Average Duration of Stay:	
<input type="text"/>	

<b>Orbit Hotel</b>	
Analyst:	Back
1. Statistic:	
<input type="text"/>	
Roomnumber	Occupancy
<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>
Total:	
<input type="text"/>	

 <b>Orbit Hotel</b>	
<b>Register</b> <span>Back</span>	
Please enter your register data:	
User-Category:	<input type="text" value="Customer"/>
Username:	<input type="text"/>
Password:	<input type="text"/>
FullName:	<input type="text"/>
E-Mail:	<input type="text"/>
Sex:	<input type="text" value="Female"/>
Birthday: (DD.MM.YYYY)	<input type="text"/>
<input type="button" value="Register"/> <input type="button" value="Cancel"/>	



# Orbit Hotel

**Requested Rooms** Back

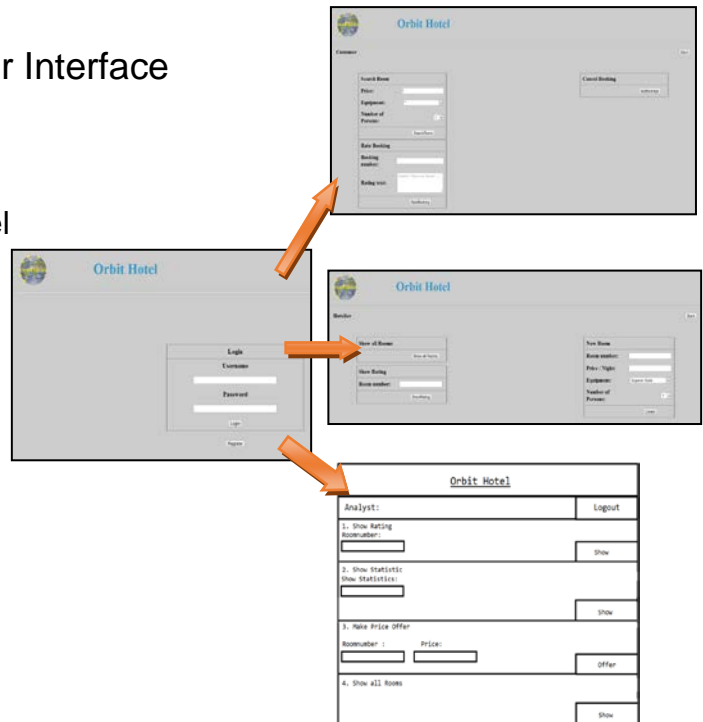
BookingNr	Roomnr	Start	End	NrOfPersons	Payment	CancelBooking
1000	2	1.5.2014	6.5.2014	1	Bar	<a href="#">CancelBooking</a>

## 2.6 Beschreibung des GUI / User Interface

### Login / Registrierung:

Zu Beginn jedes Aufrufs der Orbit Hotel Startseite steht jedem Nutzer ein Login bzw. eine Registrierung bevor. Dabei muss der User seinen Username und Password eingeben und anschließend auf „Confirm“ drücken.

Ist der User nicht registriert, kann er sich mittels „Register“ registrieren.



Der User kann sich im jeweiligen Teil seines GUI neben seinen gegebenen Aktionen ausloggen. Mittels drücken des „Back“ Buttons kommt man wieder zurück auf den Login Bildschirm.



## Customer:

Um ein bestimmtes Zimmer zu suchen, kann der Kunde die Suchkriterien „Price“, „Equipment“ und „Number of Persons“ eingeben und anschließend auf „Search“ drücken. Anschließend werden den Kriterien entsprechend verfügbare Zimmer angezeigt.

Klickt der Customer anschließend auf das „Book“ Feld, wird das Buchungsfenster angezeigt, bei dem der Kunde noch die Bezahl Methode „Method of Payment“ und die Daten zur Reservierung „From“ und „Until“ eingeben kann.

Mit Drücken des „FinishBooking“ Buttons bestätigt er die Reservierung.

**Requested Rooms**

Roomnr	Equipment	NrOfPersons	Price	Rating	Book
1	Business Suite	2	450.0 Euro	un...	Book
2	Economy Class Room	2	25.0 Euro	No rating available...	Book
3	Business Suite	2	450.0 Euro	No rating available...	Book
4	Superior Suite	1	100.0 Euro	No rating available...	Book
5	Superior Suite	1	200.0 Euro	No rating available...	Book
6	Business Suite	2	50.0 Euro	No rating available...	Book
7	High Class Room	2	80.0 Euro	No rating available...	Book
8	Economy Class Room	1	25.0 Euro	No rating available...	Book
9	Superior Suite	2	1000.0 Euro	No rating available...	Book

**Booking**

Please enter your booking data:

Roomnr: 2

Bookingstart: 10.08.2014

Bookingend: 10.08.2014

Payment-Method: Debit

Number of Persons: 2

FinishBooking Cancel

Um eine bereits erfolgte Buchung zu stornieren kann der Customer auf „My Bookings“ klicken um dort seine derzeitigen Buchungen anzusehen. Dort kann er dann auf den Button „CancelBooking“ drücken, um seine Buchung zu stornieren.

**My Bookings**

BookingNr	Roomnr	Start	End	NrOfPersons	Payment	CancelBooking
1000	2	1.5.2014	6.5.2014	1	Bar	CancelBooking



Weiters kann der Customer eine Zimmerbewertung abgeben indem er seine Buchungsnummer in das Feld „Booking Number“ eingibt, eine Bewertung in das dafür vorgesehene Feld eingibt und auf den Button „RateBooking“ klickt.

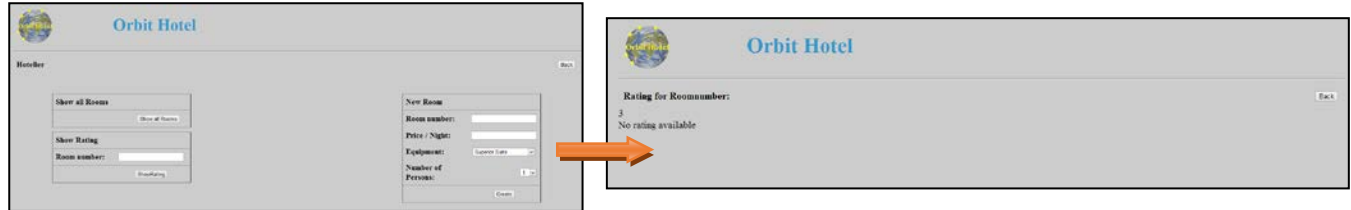
### Hotelier:

Um als Hotelier alle Zimmer anzuzeigen, kann man den Button „Show all Rooms“ drücken und bekommt eine Tabelle mit allen Zimmern. Um anschließend ein Zimmer zu editieren muss der Button „Edit“ drücken. Dann werden die Zimmerinformationen angezeigt, welche durch den Hotelier geändert werden können. Die Eingabe wird wiederum mit „FinishEdit“ bestätigt.

Room	Equipment	SelfPerson	Price	Occupancy	Rating	Edit
1	Superior Suite	2	250.0 Euro	1	Not available	edit
2	Superior Suite	2	250.0 Euro	1	Not available	edit
3	Superior Suite	2	250.0 Euro	1	Not available	edit
4	Superior Suite	1	100.0 Euro	1	Not available	edit
5	Superior Suite	1	200.0 Euro	1	Not available	edit
6	Superior Suite	2	90.0 Euro	1	Not available	edit
7	Single Class Room	2	40.0 Euro	1	Not available	edit
8	Superior Suite	1	25.0 Euro	1	Not available	edit
9	Superior Suite	2	100.0 Euro	1	Not available	edit

Um ein neues Zimmer freizugeben, muss der Hotelier eine Raumnummer unter „Roomnummer“, einen Preis (Price), die Ausstattung (Equipment) und die Anzahl der maximalen Personenbelegung pro Zimmer (Number of Persons) eingeben. Darauf drückt er auf „Create“ um die Eingabe zu bestätigen bzw. um ein neues Zimmer zu erstellen.

Zuletzt kann der Hotelier noch das Rating für ein gewisses Zimmer anzeigen lassen indem er die Zimmernummer im Feld „Roomnumber“ eingibt und anschließend den Button „ShowRating“ drückt.



### Analyst:

Der Analyst hat wie der Hotelier die Möglichkeit sich alle Zimmer anzeigen zu lassen indem er auf „Show all Rooms“ drückt.

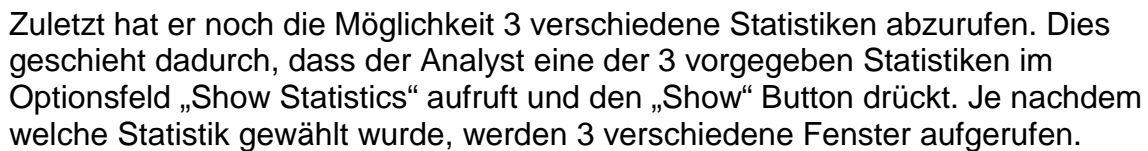


Um ein bestimmtes Rating anzusehen, muss der Analyst die Raumnummer (Roomnumber) des zu betrachtenden Zimmers eingeben.



The screenshot displays the 'Orbit Hotel' application interface. On the left, the 'Make Price Offer' form is visible, featuring fields for 'Roomnumber' and 'Price', and an 'Offer' button. On the right, the 'All Rooms' table is shown, listing various room types and their prices. An orange arrow highlights the 'Price Offer' column in the table, indicating the target for the offer.

Roomnr	Equipment	NoOfPersons	Price	OfferedPrice	Rating	Price Offer
1	Business Suite	2	650.0 Euro	-	null	<input type="text"/>
2	Economy Class Room	3	25.0 Euro	-	No rating available	<input type="text"/>
3	Business Suite	2	450.0 Euro	-	No rating available	<input type="text"/>
4	Superior Suite	1	100.0 Euro	-	No rating available	<input type="text"/>
5	Superior Suite	1	200.0 Euro	-	No rating available	<input type="text"/>
6	Business Suite	2	50.0 Euro	-	No rating available	<input type="text"/>
*	Mini Club Room	2	30.0 Euro	-	No rating available	<input type="text"/>



27

Files („Users.ser“, „Rooms.ser“ und „Bookings.ser“) geschrieben. Bei jedem Speichervorgang werden die alten Daten komplett durch die Neuen ersetzt. Die Speicherklassen dienen ebenfalls dem Auslesen zuvor gespeicherter Datensätze aus den Files und übertragen diese Objekte dann wieder in eine ArrayList.