

Testcases

Orbit Hotel

Login-Seite (Index.jsp):

- Testcase Username- und Passwort-Feld bleiben leer
 - Fehlermeldung erscheint mit „Please enter username!“
- Testcase Username wird hingeschrieben, jedoch bleibt das Passwort-Feld leer
 - Fehlermeldung erscheint mit „Please enter password!“
- Testcase Passwort wird hingeschrieben, jedoch bleibt das Username-Feld leer
 - Fehlermeldung erscheint mit „Please enter password!“
- Testcase Falscher Username oder Passwort wird eingegeben
 - Fehlermeldung erscheint mit „Falsche Eingabe!“
- Testcase Richtiger (registrierter) Username und Passwort wird eingegeben
 - Weiterleitung zum richtigen Interface (Customer-, Hotelier- oder Analyst-Interface)
- Testcase Button „Register“ wird angeklickt
 - Weiterleitung auf Register-Seite

Erwarteter Input:

1)

Username: Customer
Password: 1234567
Klick auf „Login“

2) Klick auf „Register“

Register-Seite (Register.jsp):

- Testcase Button „Back“ wird angeklickt
 - Weiterleitung auf die Login-Seite
- Testcase schon verwendeter Username wird bei der Registrierung benutzt
 - Fehlermeldung erscheint mit „User bereits vorhanden!“
- Testcase Username-Feld wird freigelassen
 - Fehlermeldung erscheint mit „Please enter valid username!“
- Testcase Passwort-Feld wird freigelassen
 - Fehlermeldung erscheint mit „Please enter password!“
- Testcase FullName-Feld wird freigelassen
 - Fehlermeldung erscheint mit „Please enter full name!“
- Testcase E-Mail-Feld wird freigelassen
 - Fehlermeldung erscheint mit „Please enter email!“
- Testcase E-Mail wird ohne ein @____.____ eingegeben
 - Fehlermeldung erscheint mit „Not an e-mail-adress!“
- Testcase Birthday-Feld wird freigelassen
 - Fehlermeldung erscheint mit „Invalid date format:“
- Testcase Birthday wird im falschen Format hingeschrieben
 - Fehlermeldung erscheint mit „Invalid date format: ____ (so wie man es falsch eingetippt hat)!“
- Testcase Alle Felder sinngemäß ausgefüllt
 - Benutzer wird angelegt, Weiterleitung auf Login-Seite

Erwarteter Input:

1)

User-Category: <Hotelier>
Username: Hotelier
Password: 1234567
FullName: HotelierFullName
E-Mail: hotelier@email.com
Sex: <sex>
Birthday: 01.01.1972
Klick auf „Register“

Customer Interface:

- Testcase Button „Back“ wird angeklickt
 - Weiterleitung auf die Login-Seite

Search Room (CustomerInterface.jsp):

- Testcase Preis-Feld wird freigelassen
 - Fehlermeldung erscheint mit „Please enter a price!“
- Testcase Alle Felder sinngemäß ausgefüllt
 - Weiterleitung auf Requested Rooms-Seite, wo alle verfügbaren Zimmer mit den ausgewählten Kriterien angezeigt werden

Erwarteter Input:

1)

Price: 1000
Equipment: <Equipment>
Number of Persons: <Nr of Persons>
Klick auf „SearchRoom“

Searched Rooms (CustomerSearchRooms.jsp):

- Testcase Button „Back“ wird angeklickt
 - Weiterleitung auf Customer Interface
- Testcase Zimmer wird gebucht
 - Weiterleitung auf Booking-Seite

Erwarteter Input:

1)

Klick auf „Book“

Booking (CustomerBook.jsp):

- Testcase Button „Back“ wird angeklickt
 - Weiterleitung auf Requested Rooms-Seite
- Testcase Bookingstart oder Bookingend werden ausgelassen oder falsch eingegeben
 - Fehlermeldung erscheint mit „Invalid date format: ____ (so wie man es falsch eingetippt hat)!“
- Testcase Zimmer sind nicht verfügbar für die gewählte Zeitperiode
 - Fehlermeldung erscheint mit „Room is not available during the period you entered / Number of Persons exceeds limit!“
- Testcase Alle Felder sinngemäß ausgefüllt
 - Meldung erscheint mit „Booking Successful!“, Weiterleitung auf Customer Interface

Erwarteter Input:

1)

Bookingstart: 01.03.2014
Bookingend: 04.02.2014
Payment-Method: <Payment-Method>
Number of Persons: <Nr of Persons>
Klick auf „FinishBooking“

Rate Booking (CustomerInterface.jsp):

- Testcase Booking number-Feld wird freigelassen
 - Fehlermeldung erscheint mit „Please enter a booking number!“
- Testcase Rating text-Feld wird freigelassen
 - Fehlermeldung erscheint mit „Please enter a rating message!“
- Testcase Alle Felder sinngemäß ausgefüllt
 - Meldung erscheint mit „Operation succeeded!“

Erwarteter Input:

1)

Booking Number: 1
Rating Text: <text>
Klick auf „RateBooking“

Rate Booking (CustomerInterface.jsp):

- Testcase Button „MyBookings“ wird angeklickt
 - Weiterleitung auf Requested Rooms

Erwarteter Input:

- 1) Klick auf „MyBookings“

Booked Rooms (CustomerCancelBooking.jsp):

- Testcase Button „Back“ wird angeklickt
 - Weiterleitung auf Customer Interface
- Testcase Buchung stornieren, Button „CancelBooking“ wird angeklickt
 - Meldung erscheint mit „Operation succeeded!“, Buchung wird storniert, Weiterleitung auf Customer Interface

Erwarteter Input:

- 1) Klick auf „CancelBooking“

Hotelier Interface:

- Testcase Button „Back“ wird angeklickt
 - Weiterleitung auf die Login-Seite

Show all Rooms (HotelierInterface.jsp):

- Testcase Button „Show all Rooms“ wird angeklickt
 - Weiterleitung auf All Rooms-Seite

Erwarteter Input:

- 1) Klick auf „Show all Rooms“

All Rooms (HotelierShowAllRooms.jsp):

- Testcase Button „Back“ wird angeklickt
 - Weiterleitung auf Hotelier Interface
- Testcase „EditRoom“ wird angeklickt
 - Weiterleitung auf Hotelier-Seite

Erwarteter Input:

- 1) Klick auf „EditRoom“

Edit Room (HotelierEditRoom.jsp):

- Testcase Button „Back“ wird angeklickt
 - Weiterleitung auf All Rooms-Seite
- Testcase Price/Night-Feld wird freigelassen
 - Fehlermeldung erscheint mit „Please enter a valid price!“
- Testcase Alle Felder sinngemäß ausgefüllt
 - Durchführung der Änderungen, Weiterleitung auf Hotelier Interface

Erwarteter Input:

1)

Price / Night: 500
Equipment: <Equipment>
Number of Persons: <Nr of Persons>
Klick auf „FinishEdit“

Show Rating (HotelierInterface.jsp):

- Testcase Button „ShowRating“ wird angeklickt
 - Fehlermeldung erscheint mit „Please enter a roomnumber!“
- Testcase Alle Felder sinngemäß ausgefüllt
 - Weiterleitung auf Rating for Roomnumber-Seite, wo die Ratings angezeigt werden

Erwarteter Input:

1)

Room Number: 1
Klick auf „ShowRating“

Rating for Roomnumber (HotelierInterface.jsp / HotelierShowRating.jsp):

- Testcase Button „Back“ wird angeklickt
 - Weiterleitung auf HotelierInterface.jsp

Erwarteter Input:

1)

Room number: 1
Klick auf „Show Rating“

2)

Klick auf „Back“

New Room (HotelierInterface.jsp):

- Testcase Room number-Feld wird freigelassen
 - Fehlermeldung erscheint mit „Please enter a roomnumber!“
- Testcase Price/Night-Feld wird freigelassen
 - Fehlermeldung erscheint mit „Please enter a valid price!“
- Testcase Alle Felder sinngemäß ausgefüllt
 - Neues Zimmer wird erstellt, Meldung erscheint mit „New Offer created!“

Erwarteter Input:

1)

Room number:	2
Price / Night:	450
Equipment:	<Equipment>
Number of Persons:	<Number of Persons>
Klick auf „Create“	

Analyst Interface:

Show all Rooms (AnalystInterface.jsp):

- Testcase Button „Show all Rooms“ wird angeklickt
 - Weiterleitung auf All Rooms-Seite

Erwarteter Input:

1)

Klick auf „Show all Rooms“

Make Price Offer (AnalystShowAllRooms.jsp):

- Testcase Ein neues Price Offer wird erstellt
 - Meldung erscheint mit „Price Offer succeeded!“, Price Offer wird erstellt

Erwarteter Input:

1)

Price Offer:	450
Klick auf „MakePriceOffer“	

Season Statistic (AnalystInterface.jsp / AnalystSeason.jsp):

- Testcase Statistik für ein bestimmtes Jahr wird ausgewählt
 - Weiterleitung auf SeasonStatistic-Seite, wo die Statistik angezeigt wird
- Testcase Button „Back“ wird angeklickt
 - Weiterleitung auf Analyst Interface

Erwarteter Input:

1)

Year <Year>
Klick auf „SeasonStatistic“

Price Statistic (AnalystInterface.jsp / AnalystPrice.jsp):

- Testcase Statistik für ein bestimmtes Jahr wird ausgewählt
 - Weiterleitung auf PriceStatistic-Seite, wo die Statistik angezeigt wird
- Testcase Button „Back“ wird angeklickt
 - Weiterleitung auf Analyst Interface

Erwarteter Input:

1)

Year <Year>
Klick auf „PriceStatistic“

Occupancy Statistic (total) (AnalystInterface.jsp / AnalystInterface.jsp):

- Testcase
 - Weiterleitung auf OccupancyStatistic-Seite, wo die Statistik angezeigt wird
- Testcase Button „Back“ wird angeklickt
 - Weiterleitung auf Analyst Interface

Erwarteter Input:

1)

Klick auf „OccupancyStatistic“

Rating for Roomnumber (AnalystInterface.jsp / AnalystShowRating.jsp):

- Testcase Button „Back“ wird angeklickt
 - Weiterleitung auf AnalystInterface.jsp

Erwarteter Input:

- 1) Room number: 1
Klick auf „Show Rating“
- 2) Klick auf „Back“