5th assignment

A tourist location offers a series of accommodations, subdivided into two categories: apartments and hotels. Both categories are characterized by a name, a daily price and an availability.
Extra info for hotels are: half board price, number of stars, number of available places.
Extra info for apartments are: price for final cleaning, max number of guests.

Apartments are the following:

NAME	PRICE	FINAL CLEANING	MAX N. PERSONS
Pietra Bianca	40	15	4
Sapore Di Sale	80	20	8
Tenuta Di Artimino	60	12	6

Hotel are the following:

NAME	PRICE	EXTRA HALF BOARD	STARS	PLACES
Artemide	100	20	4	60
Majestic	65	15	3	50
Palace	200	30	5	25
Zenith	70	18	3	40

Every accommodation is associated with an availability information, that for every day specifies how many places are available (i.e., for hotels total number of places minus number of reserved places, while for apartments only be 0 – unavailable or 1- available, regardless of the number of hosted persons).

Write a routine to initialize the DB with:
- Random hotel occupancy between 90 and 100% of the total places for every single day (independently for every day) in the period February 1-28.
- Random apartment occupancy: every apartment is occupied in 4 random days in the period February 1-28 (independently for every apartment).

The name of mock guests is always Mock Guest.

Write a routine to fully clean the occupancy data.

The actions a user can perform via Web are:

1) Give a start date, an end date and a number of persons to get one of the following:
- list of all the available accommodations (compatible with the request)
- list of all the available hotels (compatible with the request)
- list of all the available apartments (compatible with the request)
2) View his/her reservation (simply specifying name and surname: it is not requested to use authentication). We do not worry about possible homonymity.

On the resulting set, the user can choose one, and get the total price (or leave). When choosing a hotel, users have to specify if half board is requested.
The total price for a hotel is the daily price (optionally + half board) times number of guests times number of days.

The total price of an apartment is given by the daily price times the number of days, plus the cleaning price (applied only once): it does not depend on the number of guests, nor on the number of days

After getting the total price, the user can confirm by providing name, surname and credit card number, or go back to the list of results.
After a user ends this process, s/he gets a reservation confirmation (the reservation is saved in the database, availability is adjusted).

From every page, users can go back to the beginning.

Since the focus of the exercise is on the business logic part, no special care is requested in presenting the view (a very basic presentation is ok, as long as the functionalities are there).

The business logic part must be developed using EJB using EBJ patterns, and transactions must be managed by container. The database (H2) must not be embedded.

The web part must be deployed outside of Wildfly, in a separate project.

Write a detailed report using the following structure:

- **Title Page** containing date, title , your name
- **Introduction** (problem statement, description of the domain)
- **Design:** detail the DB structure, the Entity mapping, the role of the beans, the transactional behaviour.
- **Deployment:** screen shots of your app running, documenting the various steps.
- **Comments and notes** (any problems encountered during the project development, any other comment)

Deliver on Moodle: report, DB, zipped projects (you can throw away the target section, so as to reduce the needed space if necessary).

Delivery dates:

For the January 11 exam, deadline is January 6 (strongly suggested anyway also for following exam dates).
A second deadline, valid for the February 17 exam as well as for all the summer sessions, is February 11.