Programming in Java
Prep Term, 2018-19
Lab 3

We are developing a simple prototype for QuickRooms, an online “store” that allows customers to book hotel rooms. While such a solution will need a web or other front-end, we will first look at the core functionality of the back-end of such a solution. We will focus on a few key features:

- A customer can see the list of available rooms (assume this is for a specific day)
- The customer can see details of each available set of rooms – e.g. hotel name, price, number of beds, and rating (0-5 stars)
- The customer can ask for the list to be displayed as sorted by any of the parameters above

QuickRooms works with hotel chains to get them to list their products on this site. Each such chain can list multiple rooms, and for each room, provide the information above, including the price. They can change any of these at will, even during run time. For the purposes of this work, they each provide one or more Java classes to QuickRooms, who then add it to their solution. Since they want to keep their computations hidden, they provide only .class files to QuickRooms

To simplify their development efforts, QuickRooms turns to VirtualStore, a platform software company, that provides a platform (a set of classes) that allows multiple hotel chains to be integrated, and provides interfaces that would support the QuickRooms application: adding a hotel chain to the solution, getting the list of rooms from all the chains, sorting them based on the parameter chosen by the customer, etc. This allows QuickRooms to focus on the outermost layer of the application.

Part 1: Design a set of classes that would model this approach, and allow any number of chains to be added to the solution.

Part 2: One of the hotel chains likes this functionality, and wants to create their own website, and decide to use the same platform from VirtualStore. However, they would want to display additional features of hotels and hotel rooms (whether they have TV, WiFi, is breakfast included etc). To what extent would you be able to reuse the design from Part 1? Can VirtualStore software be used without any modifications?

Part 3: VirtualStore figures out that the functionality they have is quite generic, and they can now sell this to other kinds of “stores” such as QuickBus, a company that allows online booking of bus tickets. Again, how much of the original software would VirtualStore be able to re-use? All of it?

You can put these together as a Java program with a command line interface that the customer would use to type in their requirements.

For those wishing to go further, you could look at how this functionality could be wrapped as a web service (preferably REST services running on tomcat), and have one or more front-ends that connect over http and are returned the results in JSON format. You would be pleasantly surprised by the small amount of additional work you would need to accomplish this. Look at Java Jersey/JAX-RS, for example, or http://www.vogella.com/tutorials/REST/article.html. For the front-end, you could initially use something like Postman.