Object Oriented Programming

Assessment-Coursework

Markos Galikas

University of Westminster

Student Number: w1895147

January 2023

Contents

List of Figures

1 Introduction

The focus of the current assignment is based on the design and development of a basic hotel booking management software system. The aim of this coursework is to assess the knowledge and skills acquired about Object Oriented Programming. Part of the coursework is to analyse the problem and identify a set of required system functionalities by providing a design for that system using a UML class diagram. A written explanation of each class and interface that was used as well as their relationships is also provided for better understanding.

2 Brief Explanation

We start from the class WestminsterHotel, which includes the main logic of the program. It incorporates two users,included in the 'User' enumeration type, with different menu options. For that reason the class implements two different interfaces to form a contract, the 'IHotelManager' and the 'IHotelCustomer'. It also has an attribute of type dictionary to accommodate the need for room storage in the system. We used this data structure to ensure that there are no duplicate entries of the rooms, as this structure fulfills this need. The program start by reading from the database the potential saved rooms and from displaying the customer menu.
This menu contains the methods useful for a simple user. To implement the contract there are three options(methods) plus an option to acquire admin privileges and one to quit the application. The options are selected based on the input of the user from the keyboard in the form of numbers. First, there is the option to list all available rooms of a given size and for dates specified(check-in, check-out). For the second options, we present the list of the rooms with the same parameters as option one, but with one more extra parameter, the max price the customer is willing to pay, Furthermore, the result are sorted on ascending price and we present only the rooms whose price is less than the maximum price. For the sorting we implement the 'IComparable' interface for the rooms with only one method. Lastly, there is the option to book a room. If the dates overlap with an existing 'Booking' for the same room we stop the process.
For the overlapping we implemented the 'IOverappable' interface with contains only one method for the contract. Logically, there is a relationship between the 'Room' and the 'Booking'. This is characterized as composition as a booking cannot exist without an available room for booking. For all dates in the program we check if the check-in is before the check-out date.
For the menu of the administrator user, the methods are different for the contract. There is the option to add a room while not allowing duplicate room numbers(unique), delete a room based on the number, list rooms based on the insertion sequence or based on a descending price. There is also the capability to generate a report, meaning writing to a file name 'ROOMS.json' all the information details for all the rooms in the system.

When the user, being either the admin or the customer, chooses to quit the state of the system is saved to a temporary file "db.json".
Finally the rooms are of two kinds and we used inheritance to create the entities. The 'Room' parent class contains the common attributes of the rooms(a unique number, a floor, a size, a price). The size is of type 'Size' enumeration and the values it can contain are only single, double, triple. The two children extending the base class are the 'StandardRoom'. and 'DeluxeRoom'. The StandardRoom includes further a window with value of at least one and the DeluxeRoom contains a balcony of a given size and a 'View' enumeration type with values seaview, mountainview, landmarkview. The WestminsterHotel may own many Rooms and their relationship is that of composition, because the rooms(weak entity) cannot exist without the hotel(strong entity).

3 UML diagram

For better understanding the below diagram(Figure 1) depicts the relationships and the entities which are used to implement the design of the system. It includes 3 classes, 2 sub-classes, 4 interfaces and 3 enumeration entities.

4 Video Link

https://drive.google.com/file/d/1g6kCmbq0tAJCAnMVl-JXLeGnqRsnQQPY/view?usp=sharing

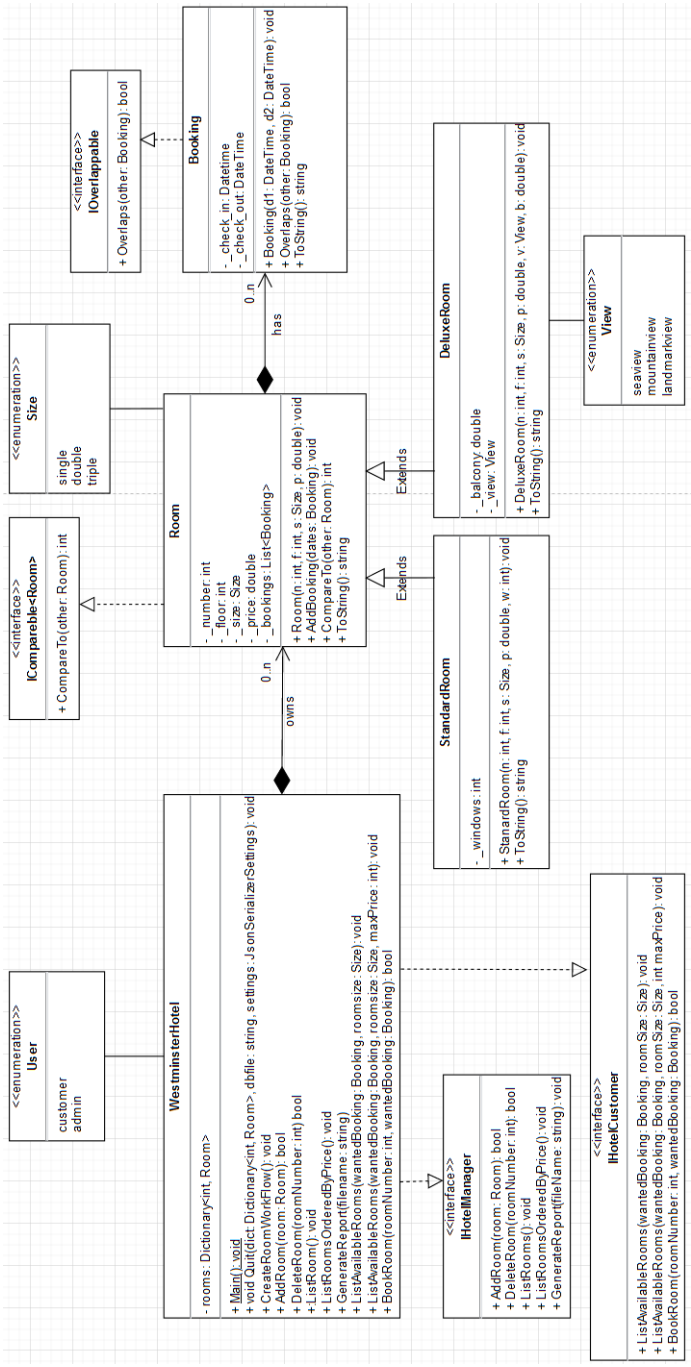

Figure 1: UML diagram