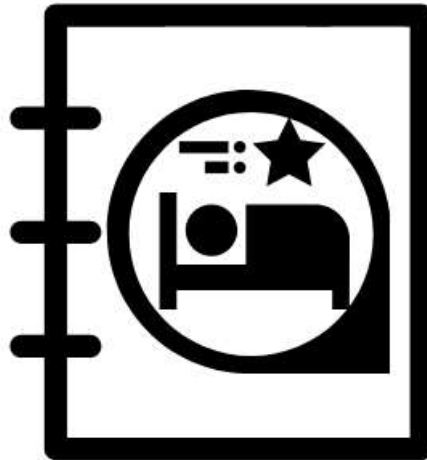


California State University Fullerton

CPSC 462



Object Oriented Software Design Domain Model for the



Hotel Reservation System

Josh Ibad

Chief Software Architect

joshcibad@csu.fullerton.edu

Revision History:

Version	Date	Summary of Changes	Author
1.0	2021-11-15	• Initial Release	Josh Ibad

Table of Contents

1 Domain's Major Concepts.....	1
1.1 Hotel.....	1
1.2 Room.....	1
1.3 RoomType.....	1
1.4 BedType.....	1
1.5 Date.....	1
1.6 Reservation.....	1
1.7 Transaction.....	1
1.8 BillingMethod.....	1
1.9 CardType.....	1
1.10 User.....	1
1.11 EmployeeUser.....	2
1.12 ManagerUser.....	2
1.13 GuestUser.....	2
1.14 Account.....	2
2 Domain Model Diagram.....	3

1 Domain's Major Concepts

1.1 Hotel

A Hotel represents a single facility that belongs to the overarching hotel company that uses the system. The Hotel facility will serve as a container of Rooms and is managed by a manager using a ManagerUser given access to the Hotel.

1.2 Room

A Room represent a single rentable room that resides in a Hotel. The Room is managed by a manager using a ManagerUser and can be reserved by Guests through a Reservation. A Room gets its own description and price for renting. A Room is also defined to be of a RoomType and to have a set amount of a BedType.

1.3 RoomType

A RoomType represents a set of types that a Room may be categorized as. RoomTypes are purely for organization to categorize the Rooms of a Hotel and is set by a Manager.

1.4 BedType

A BedType represents the types, usually the sizing or frame type, of a bed. Rooms will have a set amount of beds, which all are identified by BedType.

1.5 Date

A Date represents a single instance of time and date. A Date is used to keep track of the duration of a Reservation, through the start and end Dates of Reservation. The reservation start and end Dates will allow the determination of a Room's availability. A Date is also used to keep track of when a Transaction is paid. The Date can be used in the future to track the time of any activities or transactions for accounting purposes or for further tracking of availability.

1.6 Reservation

A Reservation represents a singular reservation of a singular Room. This Reservation is managed by an Employee through an EmployeeUser to make any adjustments based on a Guest's true check in and check out, as well as for any adjustments made during a Guest's accomodation. In the future, any other purchasables can be tracked on a Reservation by an Employee to centralize expenses associated with the Guest's stay.

1.7 Transaction

A Transaction represents the instance by which a Reservation is paid for. The Transaction is paid using a BillingMethod and is paid on a certain Date. Transactions can be aggregated for accounting purposes

1.8 BillingMethod

A BillingMethod represents the means used to pay a Transaction. The BillingMethod stores billing information such as card number, CardType, expiration date, and cvv. Guests pays transactions with the BillingMethod stored on their Account.

1.9 CardType

A CardType represents the type of card that is being used to pay a Transaction. Examples of CardTypes include Visa, MasterCard, American Express, and Discover. CardTypes are mainly informational rather than having different behaviours or stored information. They plainly determine which payment network to route a Transaction through using the BillingMethod containing the CardType.

1.10 User

A User represents the abstract object that an actor uses to interact with domain objects. The User is **not the actor**, but is merely an abstraction of the actor's system account. This User domain object stores user information, other domain objects, and explicit user permissions or roles by which the actor may use the User object to interact with other domain objects.

1.11 EmployeeUser

The EmployeeUser is a User that represents the system account used by an employee actor. Again, it is **not the actor** (refer to 1.10 - User). The EmployeeUser is responsible for managing Reservation objects to adjust them according to true check-in and check-out times of guests, as well as any adjustments made in the real world that must be reflected in the system.

1.12 ManagerUser

The ManagerUser is a User that represents the system account used by a hotel manager actor. Again, it is **not the actor** (refer to 1.10 - User). The ManagerUser is an EmployeeUser, and thus has the same abilities to adjust Reservation objects. Furthermore, ManagerUsers are assigned access to Hotels and their Rooms and are able to manage them, changing names, descriptions, adding, updating, and removing Rooms, etc.

1.13 GuestUser

The GuestUser is a User that represents the system account used by a guest actor. Again, it is **not the actor** (refer to 1.10 - User). The GuestUser has no access to hotel management functionalities, but all GuestUsers have an Account and are the only Users to have an Account. (Refer to 1.14 for the definition of Account).

1.14 Account

An Account represents a container for Reservations held by guest actors through their GuestUser, as well as billing information. Specifically, they also hold the guest actor's BillingMethods. Through the Account, a guest is able to make a Reservation as well as pay for Transactions using a stored BillingMethod. The Account is more visible and accessible to the user, in comparison to the GuestUser which has info for system use such as role and authentication (and thus, hidden from the User). The Account however, is still relevant to the domain due to its relationship with other domain objects.