

# HOTEL MANAGEMNET SYSTEM



## ITERATION 3

Date: 11/17/2019

Team Name: CS 6359.002 Team 3

Team Members:

1. Ajay Kishorkumar Danda
2. Bingjie Yan
3. Elijah Snow

## FUNCTIONAL REQUIREMENTS OF SYSTEM

### Functional Requirements:

We had 10 functional requirements initially and then we identified 2 more additional requirements in Iteration 2. Now for the final and third iteration there are no new requirements that were added or refined.

Requirement ID	Functional Requirements
R1	The Customer should be able to Register.
R2	The Customer should be able to LOGIN.
R3	The Manager should be able to Register.
R4	The Manager should be able to LOGIN.
R5	The Manager should be allowed to Add Listings.
R6	The manager should be allowed to Modify Listings.
R7	The Manager should be allowed to Delete Listings.
R8	The Customer should be allowed to Book Room.
R9	The Customer should be allowed to Modify its Room Booking
R10	The Customer should be allowed to Delete a Room Booking.
R11	The Customer should be able to View the BILL.
R12	The Customer should be redirected to Payment gateway.

## REQUIREMENT PRIORITY

There are no changes in the requirement priority for the final Iteration.

Requirement ID	Priority	Functional Requirements
R1	5	The Customer should be able to Register.
R2	5	The Customer should be able to LOGIN.
R3	5	The Manager should be able to Register.
R4	5	The Manager should be able to LOGIN.
R5	4	The Manager should be allowed to Add Listings.

R6	3	The manager should be allowed to Modify Listings.
R7	3	The Manager should be allowed to Delete Listings.
R8	4	The Customer should be allowed to Book Room.
R9	3	The Customer should be allowed to Modify its Room Booking
R10	3	The Customer should be allowed to Delete a Room Booking.
R11	4	The Customer should be able to View the BILL.
R12	4	The Customer should be redirected to Payment gateway.

### USE CASE REALIZATION AND TRACEABILITY MATRIX

The following are the use cases that are realized after refining and identifying all functional requirements in the final iteration:

Use Case 1: Registration

Use Case 2: Login

Use Case 3: Add Listing

Use Case 4: Delete Listing

Use Case 5: Modify Listing

Use Case 6: Book Room

Use Case 7: Delete Booking

Use Case 8: Modify Booking

Use Case 9: Bill Generation

Use Case 10: Payment Redirection

	Priority	TC1	TC2	TC3	TC4	TC5	TC6	TC7	TC8	TC9	TC10
R1	5	X									
R2	5		X								
R3	5	X									
R4	5		X								
R5	4			X							
R6	3					X					
R7	3				X						
R8	4						X				
R9	3								X		
R10	3							X			
R11	4									X	
R12	4										X
Total		10	10	4	3	3	4	3	3	4	4

## NON-FUNCTIONAL REQUIREMENTS OF SYSTEM

There are no additional non-functional requirements identified in the final iteration.

The following are some of the non-functional requirements of the system:

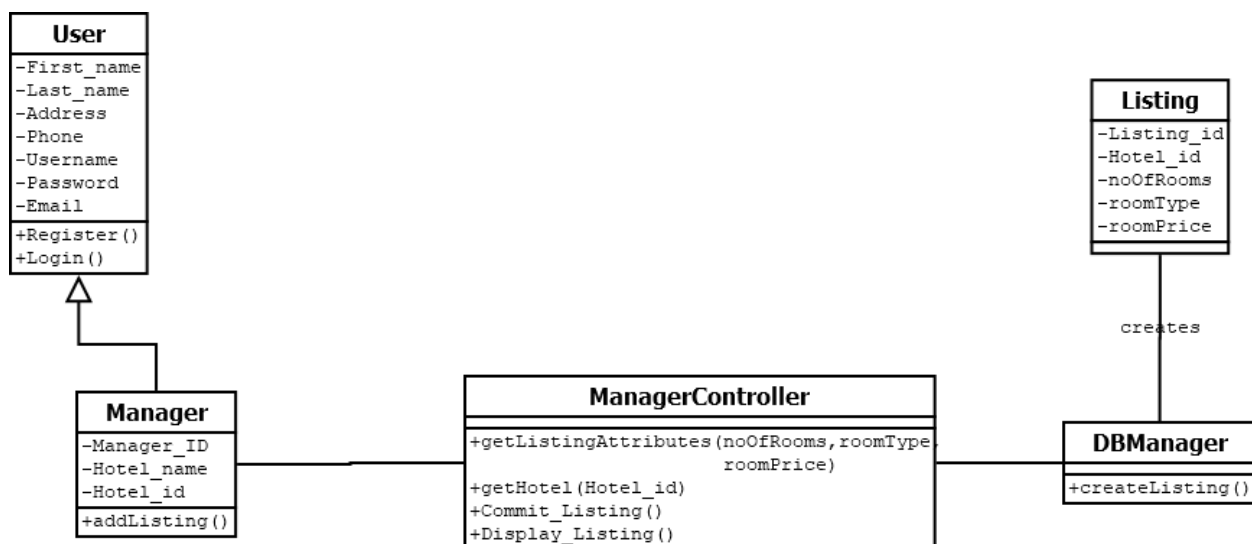
- Performance: The system will perform at its peak performance in most of the situations. The system will have fast throughput and utilization.
- Capacity: There is no limit to the number of customers the system can handle. However, the system will have only one manager for one hotel.
- Security: The system will have enough security so that no attacker can access the hotel and customer's private credentials.
- Availability: The system will be available to the customers for bookings.
- Maintainability: The system will be developed using all the important OOPs design patterns and hence will have good maintainability.

## Use Cases

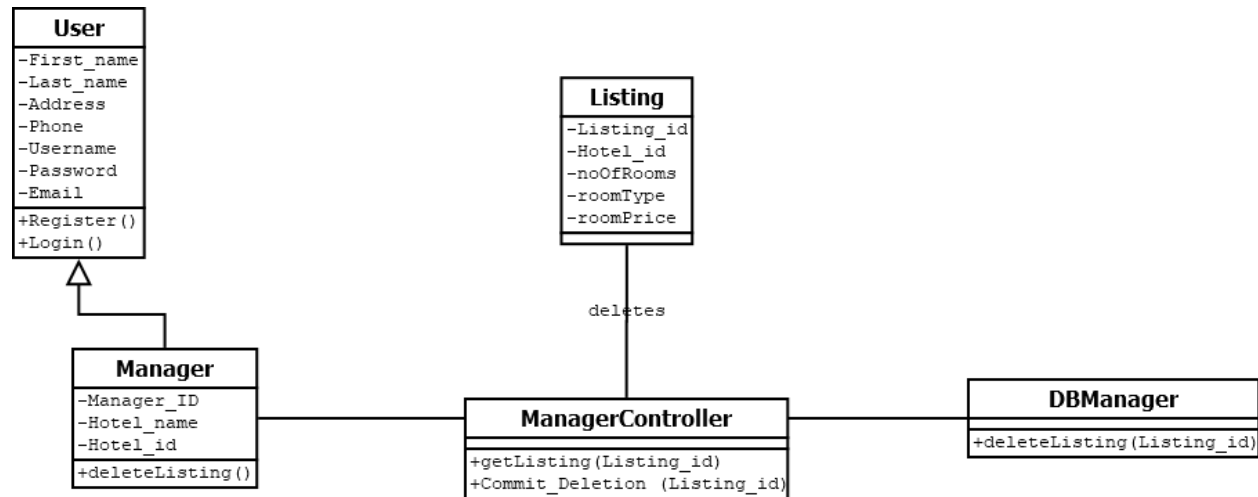
All the Use cases of the Hotel Management System are already realized in the previous two iterations. There are no new use cases that were identified in this iteration and neither any of them refined. All the use cases, user stories and actors of each of those use cases are realized in the first two iterations. There are only 3 Class diagrams and 3 Sequence Diagrams that are yet to be realized for the system.

## Class Diagrams

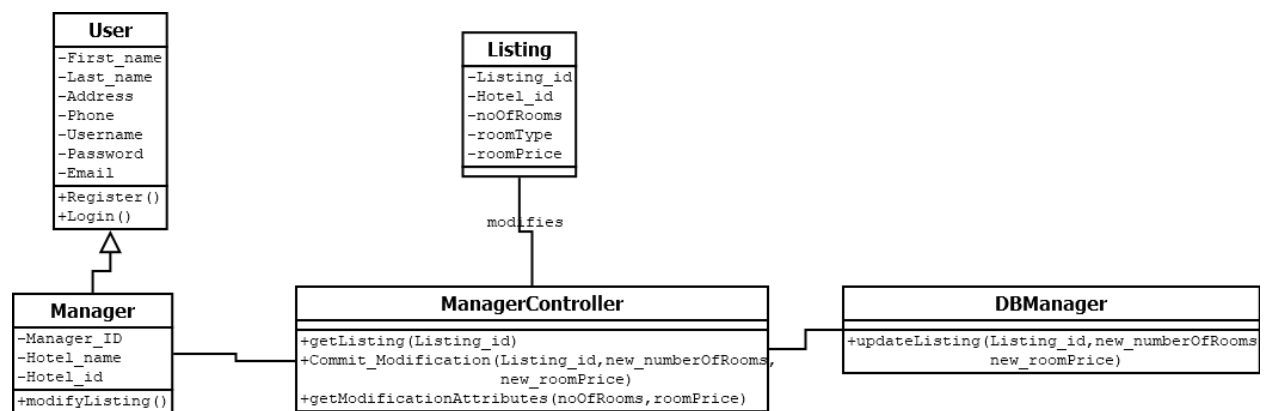
1. Class Diagram for Add Listing: The Manager is the only user in this use case. This class diagram only focuses on the add listing for room booking part of the system and not any other functionality.



2. Class Diagram for Delete Listing. The Manager is the only user in this use case. This class diagram only focuses on the deletion of listing part of the system and not any other functionality.

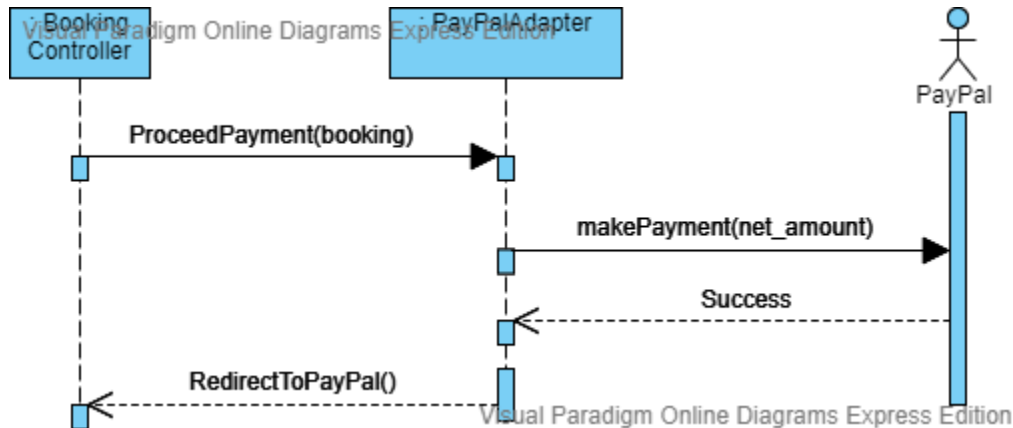


3. Class Diagram for Modify Listing. The Manager is the only user in this use case. This class diagram only focuses on the modification of number of rooms available for each room type of existing listing and not any other functionality.

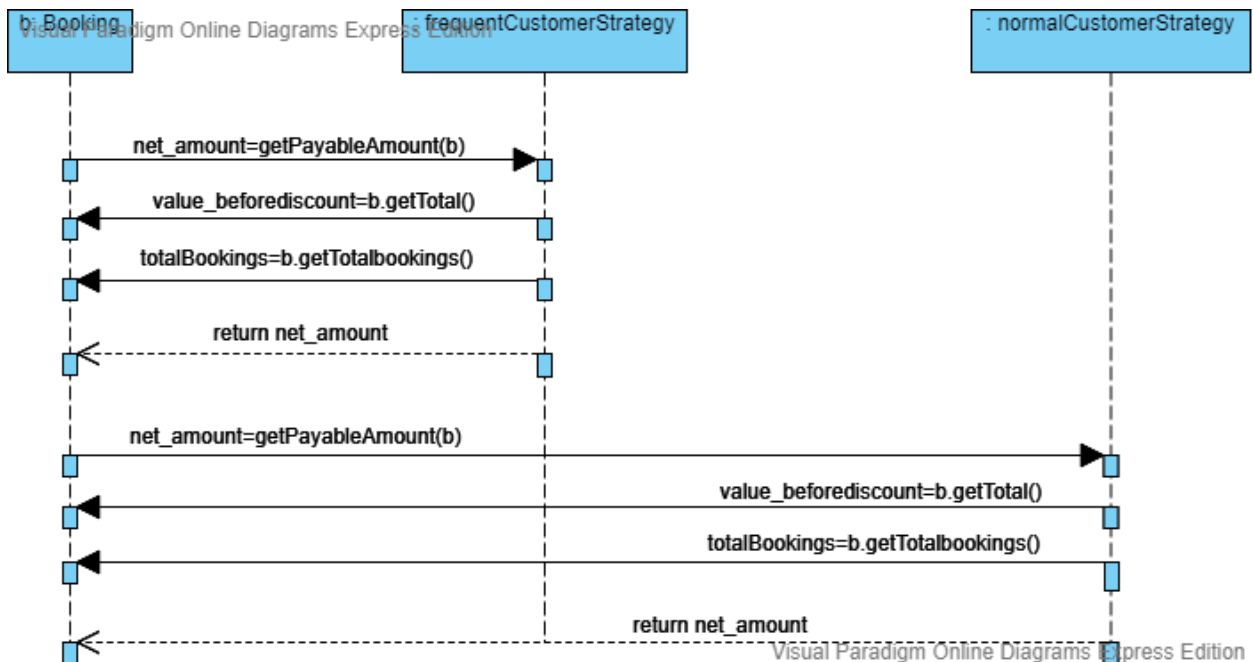


## Sequence Diagrams

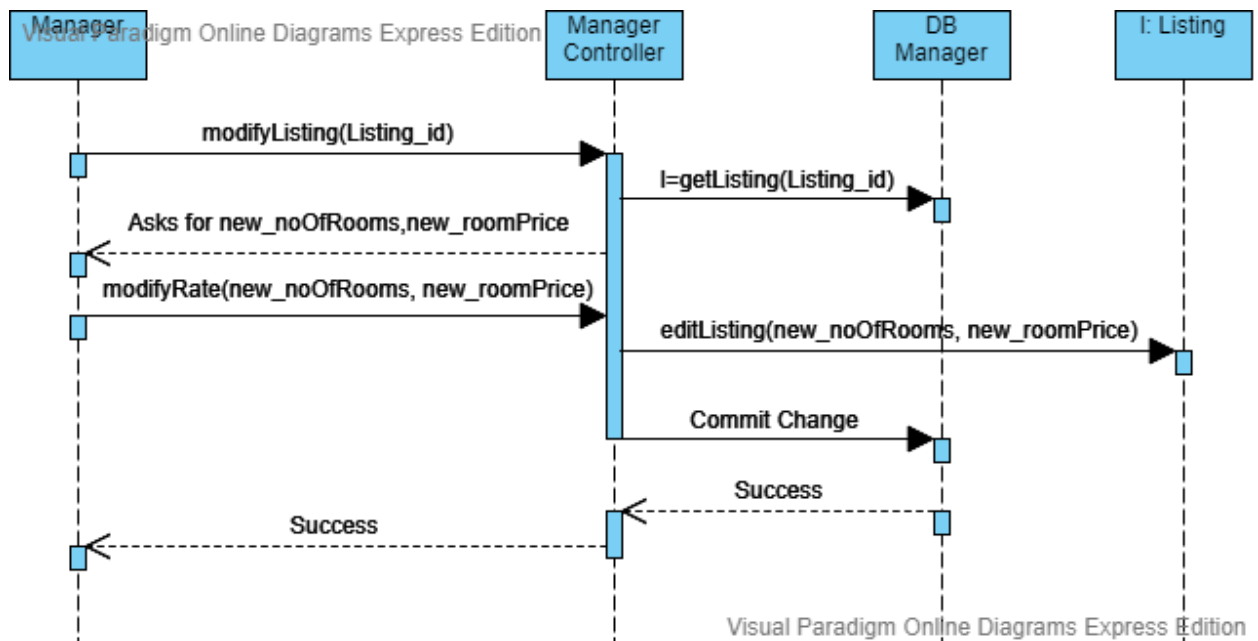
1. Sequence Diagram for Payment Redirection. The PayPalAdapter will disintegrate the entire object and send only the net\_amount to PayPal redirecting the Customer to PayPal for payment.



2. Sequence diagram for Bill Generation. System checks whether the Customer has already more than 5 bookings with the system. If yes, then the frequentCustomerStrategy is applied on the net\_amount where the Customer gets a 10% discount. If no, then the normalCustomerStrategy is applied where there is no discount on net\_amount.



3. Sequence diagram for Modify Listing. Here, there is only one actor Manager that has the rights to do so.



Code is available on GitHub: <https://github.com/UTDClassroom/CS6359002-Course-Project-team3>