

# HOTEL MANAGEMNET SYSTEM



## ITERATION 1

Date: 10/13/2019

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:

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 able to View the BILL.
R10	The Customer should be redirected to Payment gateway.

## NON-FUNCTIONAL REQUIREMENTS OF SYSTEM

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

Use case 1: Registration: This use case covers the functionality of registration of Manager/ Customer.

Actors: Manager/ Customer

Manager/Customer	System
	1. Display the registration option on terminal
2. Enter required credentials for registration like name, age, address, etc.	3. Perform data validation of data entered and prompt for required fields if any missing.
4. Enter required fields if prompted by system. 5. Click submit to register	6. Give success feedback

Post condition: New user is added

Use case 2: LOGIN: This use case covers the functionality of allowing a registered Manager/ Customer to login with their credentials.

Pre-condition: User must be registered with the system.

Actors: Manager/ Customer

Manager/Customer	System
	1. Display the login page on terminal asking for username and password.
2. User enters the required credentials and submits.	3. Check whether the user credentials match and give prompt to user to register if credentials not found.
4. Re-enter credentials and submit username and password. 6. User enters the system to perform its operations.	5. Give success feedback and gives right to the user according to its User Type.

Post condition: User is now logged in with system.

Use case 3: Add Listing: This use case covers the functionality of allowing a registered Manager to add a hotel listing for the customers to book.

Pre-condition: Manager must be registered and logged in with the system.

Actors: Manager

Manager	System
	1. Display the hotel details page.
2. Manager selects the option of adding a listing to the system.	3. Ask for credentials required to add this listing to the system.
4. Manager enters the credentials to register the listing of the hotel.	5. Perform data validation of the credentials entered by the Manager. If there is any missing field prompt the Manager of the same.
6. If there are any missing fields, Manager fills them accordingly. 7. Submits the final result.	8. The listing added by the manager is committed in the system.

Post condition: A new hotel listing is added in the system.

Use case 4: Delete Listing: This use case covers the functionality of allowing a registered Manager to delete a hotel listing so that the customers can no longer book that listing.

Pre-condition: Manager must be registered and logged in with the system.

Actors: Manager

Manager	System
	1. Display the hotel details page.
2. Manager selects the option of deleting a listing to the system.	3. Ask for a single credential Listing_id required to delete this listing from the system.
4. Manager enters the Listing_id to delete the listing from the hotel.	5. Check whether the listing is active and return error if no such listing present and exit. 6. If listing present delete the listing from system.
8. Manager receives acknowledgement of deletion of listing.	7. Send feedback to the Manager of the same.

Post condition: A hotel listing is deleted from the system.

Use case 5: Book Room: This use case covers the functionality of allowing a registered Customer to book a room from hotel listings available.

Pre-condition: Customer must be registered and logged in with the system.

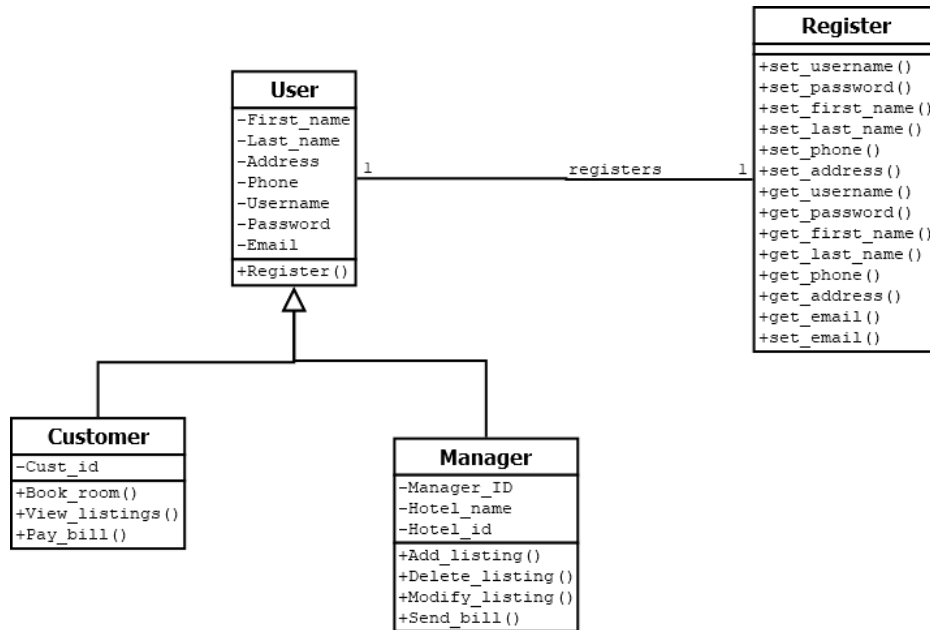
Actors: Customer

Customer	System
1. Customer selects the option of booking a room.	2. Display the hotel listings available.
3. Customer gives the hotel number that they want to book.	4. Retrieve the hotel listings for that hotel from the system. 5. Display all available listings in that hotel. 6. Ask for other details to book room like number of rooms, type, duration.
7. Customer enters their requirement for the rooms like number of rooms, type and duration.	8. Display summary of rooms to be booked and cost that will be incurred. 9. Ask Customer to Confirm or Abort Booking
10. Customer confirms booking. 11. Customer aborts booking.	12. If booking confirmed then send acknowledgment, remove listing from system and give Bill options to Customers. 13. If booking aborted then cancel booking and exit

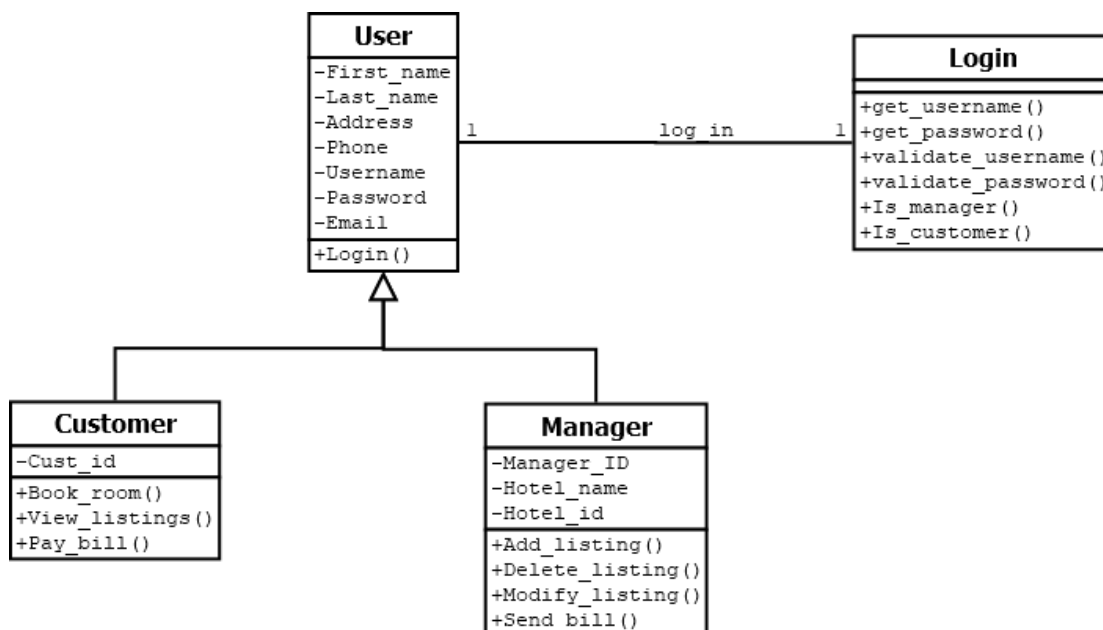
Post condition: A room is booked for the Customer and is made unavailable for other customers.

## Class Diagrams

1. Class Diagram for Registration of a new User with the system. The user can be any of the two actors. (Manager/Customer). This class diagram only focuses on the registration part of the system and not any other functionality. Other operations shown in the class diagram performed by the Manager and Customer are only for understanding purposes.

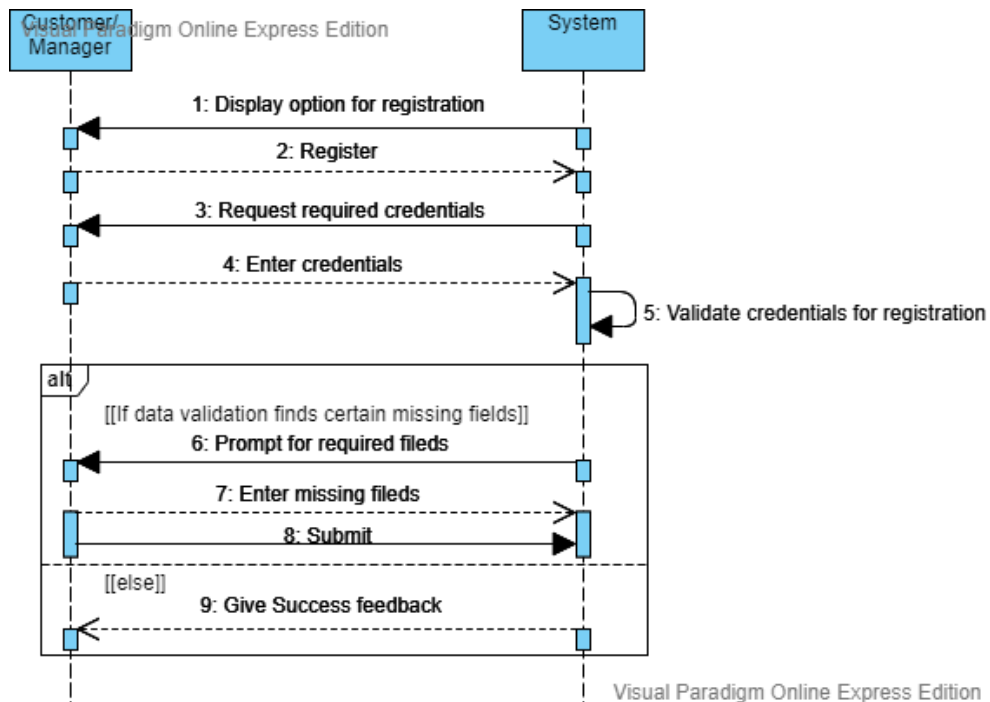


2. Class Diagram for Login of a registered User with the system. The user can be any of the two actors. (Manager/Customer). This class diagram only focuses on the login part of the system and not any other functionality. Other operations shown in the class diagram performed by the Manager and Customer are only for understanding purposes.

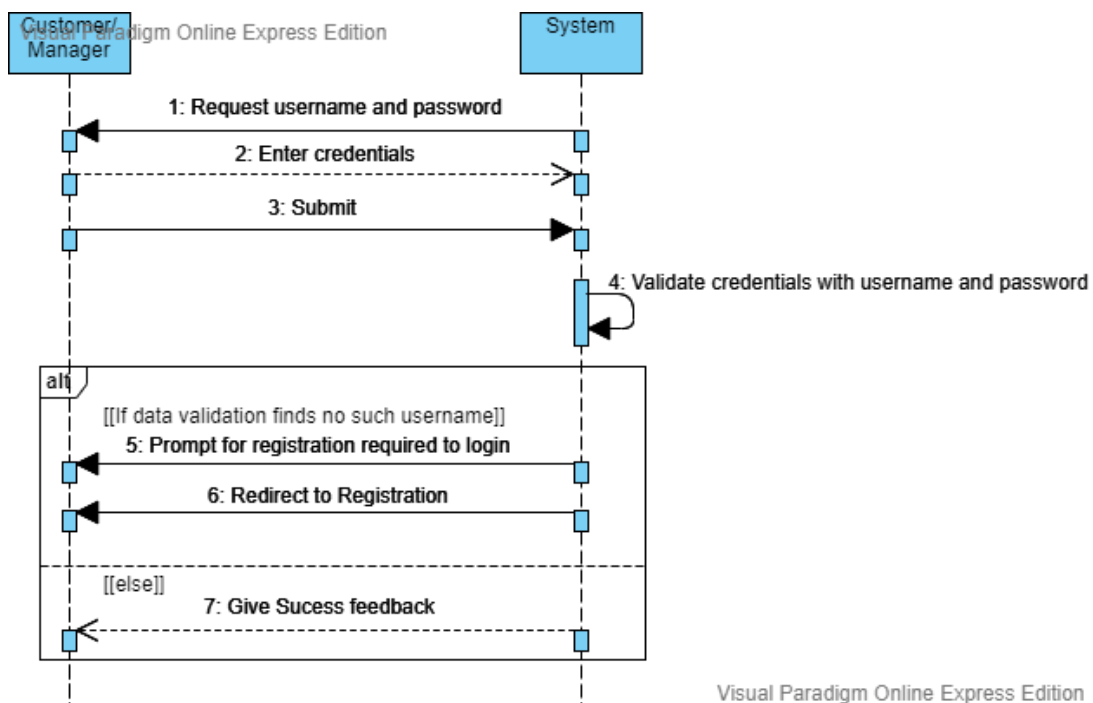


## Sequence Diagrams

1. Sequence Diagram for Registration of a new User with the system. The user can be any of the two actors. (Manager/Customer).



1. Sequence Diagram for Login of a registered User with the system. The user can be any of the two actors (Manager/Customer).



**Implementation of Use case Login:**

**Code is available on GitHub:** [https://github.com/UTDClassroom/CS6359002-Course\\_Project-team3](https://github.com/UTDClassroom/CS6359002-Course_Project-team3)

**The following are the classes available on GitHub:**

Customer.java

CustomerTest.java

Login.java

LoginTest.java

Manager.java

ManagerTest.java

User.java