Functional and Non-Functional Requirements

FUNCTIONAL:

1.) The System shall present a customer with an option to make a room reservation with corresponding fields for entry and system validation.
Author: Sam Kim

2.) The System shall present the Admin with a prompt to check in a specific customer.
Author: Sam Kim

3.) The System shall present the option to modify or cancel a customer's reservation.
Author: Sam Kim

4.)The System shall assign a reservation guaranteed status if the customer's credit card was validated and the system shall assign a reservation non-guaranteed status if not.
Author: Sam Kim

5.) The System shall cancel reservations that are not guaranteed by a specific time before the reservation time.
Author: Marcus Domingo

6.) The System shall communicate with an external bank server to validate a customer's credit card and will display whether or not the transaction was successful.
Author: Courtney Olson

7.) The System shall scan available rooms and present a specific assignment at check-in for customers.
Author: Marcus Domingo

8.) The System shall print out reports on requested fields (such as check-in statements, check-out statements, and management reports).
Author: Marcus Domingo

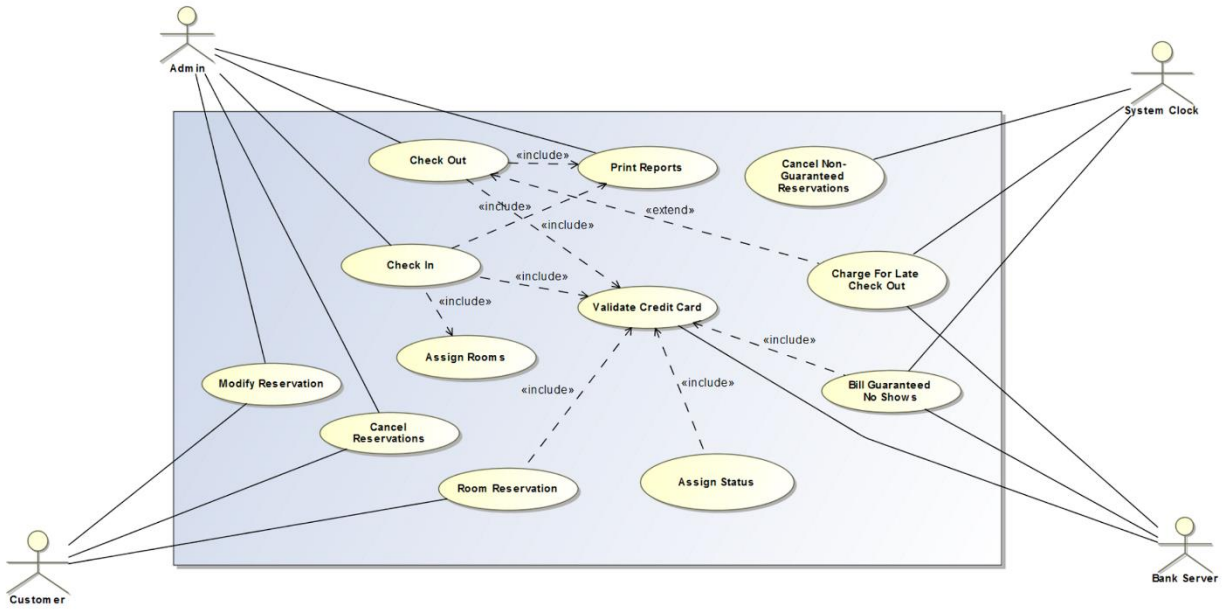
9.) The System shall prompt for a check-out procedure and execute the steps with a Hotel Staff's input. If the customer does not check out by check-out time, the System shall charge the customer for an additional night.
Author: Courtney Olson

10.) The System shall automatically bill no-show guaranteed reservation at a specified time.
Author: Courtney Olson

NON-FUNCTIONAL:

11.) The System shall take no longer than 10 seconds to authorize credit cards.
Author: Courtney Olson

12.) The System shall protect and secure all credit card and billing information.
Author: Marcus Domingo



Use Case Descriptions

Use Case Name: Create Customer Reservation
Use Case Author: Sam Kim
Summary: The admin creates a reservation for the customer
Dependency: Check room availability, Customer information
Actor(s): Administrator
Precondition(s): Customer made a reservation online
Description of Main Sequence:

1. Customer provides his name, address, etc
2. Searches for available rooms
3. Assigns the room to the customer

Description of Alternative Sequence(s):
N/A
Non-functional Condition(s):
Postcondition(s): A reservation is made for the customer

Use Case Name: Check In
Use Case Author: Sam Kim
Summary: The customer is checked in the system.

Dependency: Print reports, validate credit card, assign rooms
Actor(s): Admin
Precondition(s): The customer made a reservation and is now checking in
Description of Main Sequence:

1. The customer is checked in by the clerk
2. A room is assigned to the customer
3. The clerk requests the customers credit card
4. An external bank system will check the credit card
5. A sheet of the customer's name, room info, and check-in check-out date is printed.

Description of Alternative Sequence(s):
N/A
Non-functional Condition(s):
Postcondition(s): The customer is checked in, assigned a room, and credit card is checked.

Use Case Name: Modify Reservation
Use Case Author: Sam Kim
Summary: Customer may request the admin to modify their checkout date and room.
Dependency:
Actor(s): Customer, Admin
Precondition(s): The customer made a reservation and wants to modify it
Description of Main Sequence:

1. Customer requests the admin to modify their checkout date or room
2. The admin modifies the reservation

Description of Alternative Sequence(s):
N/A
Non-functional Condition(s):
Postcondition(s): The customer's reservation is modified

Use Case Name: Cancel Reservation
Use Case Author: Sam Kim
Summary: The customer's reservation is cancelled
Dependency:
Actor(s): Customer, Admin
Precondition(s): The customer made a reservation and wants to cancel it
Description of Main Sequence:

1. Customer requests the admin to cancel their reservation
2. The admin cancels the customer's reservation

Description of Alternative Sequence(s):

N/A
Non-functional Condition(s): N/A
Postcondition(s): The customer's reservation is cancelled

Use Case Name: Check Out
Use Case Author: Courtney Olson
Summary: The system will first print out a check-out statement for the Customer then the Customer will pay their bill and will be checked out by the Hotel Staff.
Dependency: Validate Credit Card, Print Report
Actor(s): Customer, Hotel Staff
Precondition(s): Customer is checked-in successfully and it is the day of check-out.
Description of Main Sequence:

1. Customer's name and/or room number will be used for Customer look-up.
2. A request for a check-out statement will be initiated by the Hotel Staff and a check-out statement will be printed for the Customer.
3. If paying by credit, the Customer's credit card will be validated by the external Banking system.
4. The Customer's credit card will be billed and receipt will be printed out for the Customer.
5. The Hotel Staff will remove the patient from the checked-in status and move them to the checked-out status.

Description of Alternative Sequence(s):

1. If the Customer has passed the assigned check-out time, their bill will be updated with an additional charge for another night.
2. If a Customer's credit card is declined, a new prompt for another credit card or a different form of pay will occur.

Non-functional Condition(s): The system processes will take no longer than 30 seconds to complete.
Postcondition(s): The Customer has been set in the checked-out status and has paid their bill.

Use Case Name: Charge for Late Check-out
Use Case Author: Courtney Olson
Summary: When a Customer misses their assigned check-out time, the system will automatically charge an additional day to their bill.
Dependency: Validate Credit Card, Check-out
Actor(s): System clock
Precondition(s): The Customer is not present at the expiration of their checkout time.
Description of Main Sequence:

1. The System will access the Customer's bill and add an additional charge for another night under their current rate of pay
2. The System will create a new check-out time for the following day.

3. The System will set up a notification to alert of additional charge at time of Customers check-out

Description of Alternative Sequence(s):

N/A

Non-functional Condition(s): The System clock is operational 24 hours a day.

Postcondition(s): The Customer has been charge for an additional night.

Use Case Name: Validate Credit Card

Use Case Author: Courtney Olson

Summary: The system will communicate with an external banking system to validate the Customer's credit card.

Dependency: Create Customer Reservation

Actor(s): Customer or Hotel Staff, Bank Server

Precondition(s): The Customer is in the system.

Description of Main Sequence:

1. The user will be prompted to enter credit card information.
2. Credit card Information will be sent to external banking system to validate card information.
3. Message will appear with whether or not the credit card is valid.

Description of Alternative Sequence(s):

1. If fields have been improperly filled, will prompt user for correction before the system will continue.

Non-functional Condition(s): During the external banking validation step, the system will take no longer than 10 seconds to complete. The Customer's credit information will be completely secure during transmission and storage.

Postcondition(s): The Customer's credit card has been validated.

Use Case Name: Print Report

Use Case Author: Courtney Olson

Summary: Hotel Staff will request information fields for a report and the system will generate a report consisting of those fields.

Dependency: N/A

Actor(s): Hotel Staff

Precondition(s): All information requested is contained currently within the System.

Description of Main Sequence:

1. Hotel Staff will follow the prompts for which type of report they would like to print.
2. The System will gather and compile all requested information and generate a report to be printed out.
3. The System will print the report out.

Description of Alternative Sequence(s):

1. If a report is needed for customer check-in/check-out, the system will automatically queue a request for either or.

Non-functional Condition(s): The System will be capable of storing an exponentially large amount of information, depending upon scalability.
Postcondition(s): The System generated a printed report.

Use Case Name: Assign Status
Use Case Author: Marcus Domingo
Summary: The system will assign a "Guaranteed" status when the credit card is validated at time of reservation. If the card is not validated then a "Non-Guaranteed" status will be assigned.
Dependency: Validate Credit Card
Actor(s): N/A
Precondition(s): The customer has made a reservation and has either entered a credit card or has not.
Description of Main Sequence:

1. The system gets a valid or invalid response from the bank server.
2. If the card is valid then assign the reservation with a "Guaranteed" status. If the user doesn't enter a credit card at that time or the card is invalid then assign the reservation with a "Non-Guaranteed" status.
3. Update the reservation in the system.

Description of Alternative Sequence(s):

1. N/A

Non-functional Condition(s): Assigning a status should almost be instantaneous after receiving the valid or invalid card information.
Postcondition(s): Room reservation is either Guaranteed or Non-Guaranteed.

Use Case Name: Cancel Non-Guaranteed Reservations
Use Case Author: Marcus Domingo
Summary: At a specified time the Non-Guaranteed reservations will be canceled.
Dependency: N/A
Actor(s): System Clock
Precondition(s): There is a list of Non-Guaranteed Reservations.
Description of Main Sequence:

1. The system clock triggers the system to cancel all Non-Guaranteed reservations at a specific time.
2. The system cancels all the Non-Guaranteed Reservations.
3. The System marks Guaranteed reservations as ''must pay"
4. The system updates the Non-Guaranteed List.

Description of Alternative Sequence(s):

1. N/A

Non-functional Condition(s): Canceling the Non-Guaranteed reservations should only take as long as the list itself.

Postcondition(s): All Non-Guaranteed Reservations are canceled and the Non-Guaranteed List is empty.

Use Case Name: Bill Guaranteed No-Shows

Use Case Author: Marcus Domingo

Summary: After a specified time, the system will charge the cards provided by all Guaranteed no-shows.

Dependency: Validate Credit Card, Print Report

Actor(s): System Clock, Bank Server

Precondition(s): The customers credit card must be validated and they must also have a Guaranteed reservation.

Description of Main Sequence:

1. The system clock triggers the system to bill all Guaranteed no-show reservations at a specific time.
2. A report is generated of all Guaranteed no-show Reservations
3. The system creates a billing record for each Guaranteed no-show reservation
4. The system bills the card assigned to each reservation through the bank server.
5. The system removes all reservations that have been billed.
6. The system updates the Guaranteed no-show list.

Description of Alternative Sequence(s):

1. N/A

Non-functional Condition(s): Contacting the bank server and billing should take no longer than 10 seconds. All information will be secure.

Postcondition(s): All Guaranteed no-shows will have been billed and removed from the system.

Requirement Satisfiability Table

Requirement	Satisfying Use Case(s)
1	Create Room Reservation
2	Check-in
3	Modify Reservation, Cancel Reservation
4	Assign Status
5	Cancel Non-Guaranteed Reservations
6	Validate Credit Cards
7	Assign Room
8	Print Report
9	Check-out, Late Check-out Charge
10	Bill Guaranteed No-Shows
11	Validate Credit Cards
12	Validate Credit Cards