Hotel Management System

4 Points Hotel

Software Engineering 2

Team Members

“Leader :Bayader Saad AlHarbi ..”

Name	ID
Bayader Saad Al Harbi	438009056
Areej Jameel Alsalmi	436000562
Shatha Muslih Alharbi	438007605
Safia Mrzoog Alfahmi	438007175
Atheer Waleed AlHazmi	438006113
Afrah Bakheet Allhyani	438009788
Sara Ahmed Almajnuni	438009482

*All members in group number 2

Team Report

Report Components

1. System Overview
2. Team Diagrams

   2.1 Use Case Diagram

   2.2 Class Diagram

3. Implementation Of Class Diagram

System Overview

System Overview

The hotel system is designed to facilitate work for hotel employees and make the reservation process easier for the clients.

The hotel employees have the ability to access a lot of information through the system, including customer information, reservation information, room status, and terms and conditions of the reservation service and cancellation. Using this information, the employees can easily perform their jobs like confirm reservations, confirm cancelation, process the reservation, check-in, and check-out for the client.

The client can reserve rooms in the hotel easily over the system and also the client can cancel the reservation.

The employees targeted to work on the hotel system are : the reservation clerk, the cancellation clerk, and the receptionist.

System Overview

The following are more details about how the hotel system works:

The client places a reservation order and provides the reservation details to the receptionist, then the receptionist response to the client to confirm the reservation or reject the reservation request.

When a reservation order arrives, the receptionist receives the client order and insert the client information and booking information in the system. Then the receptionist makes a transaction in the system which includes all the reservation request information, and after the transaction became updated in the system, the receptionist responds to the client with confirm the reservation and room details.

If there are any issues regarding the reservation, the transaction will be updated with rejecting, the receptionist informs the client by rejecting the reservation request.

System Overview

When a client wants to cancel the reservation, the receptionist receives the client order and retrieves the client information from the system, and makes a transaction in the system, after the transaction is updated the receptionist informs the client by accepting the cancelation request and return the amount paid to the client.

If there are any problems the transaction will be updated with rejecting, the receptionist informs the client by rejecting the cancelation request.

The reservation clerk is responsible for processing the transactions. He checks the availability of rooms through the system if there is an available room, he reserves the rooms for the client then he updated the booking information and updated the transactions state. In case there is no room is available, The reservation clerk updated the transaction's state with reject.

System Overview

The cancellation clerk receives the cancelation transactions, he checks the terms and conditions for the hotel if the request has compatible with the conditions he updates booking information and room information in the system, then he updated the transaction in the system.

If any conditions are not compatible with the cancellation request, the cancellation clerk updated the transaction with reject.

When the client wants to check-in, the receptionist has to check the validation of booking ID and Client details, then the receptionist allows the customer to occupy a room.

When the client wants to check-out, the receptionist gets the client order and gets the booking information and client details, he issues the invoice and takes the amount from the client, then the receptionist updated the room's state.

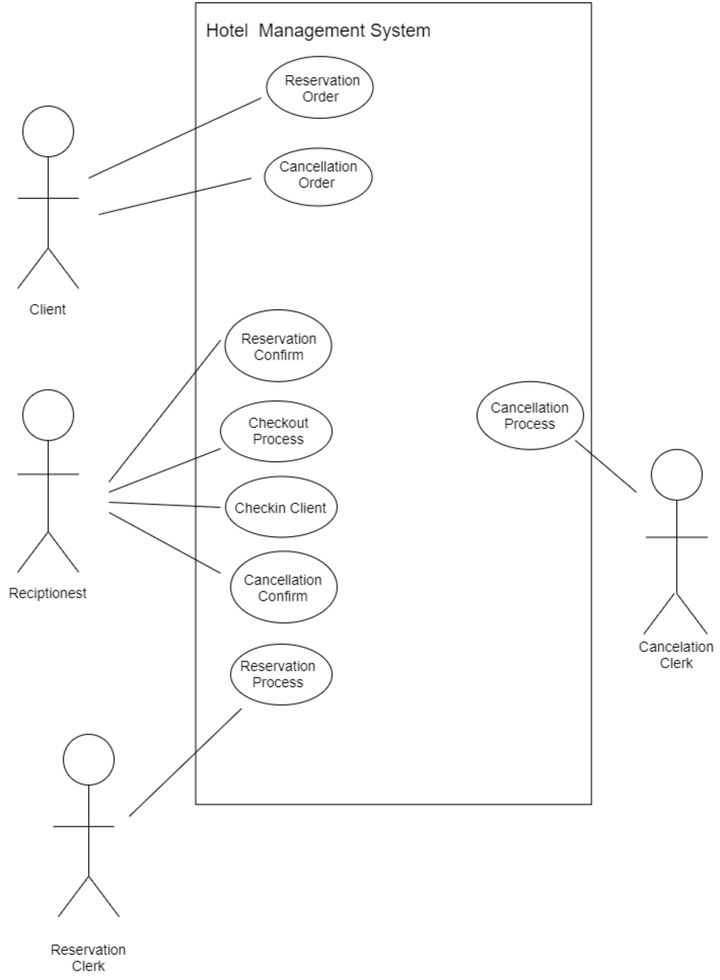
Team Diagrams

Link for Team Diagrams ☺ ....

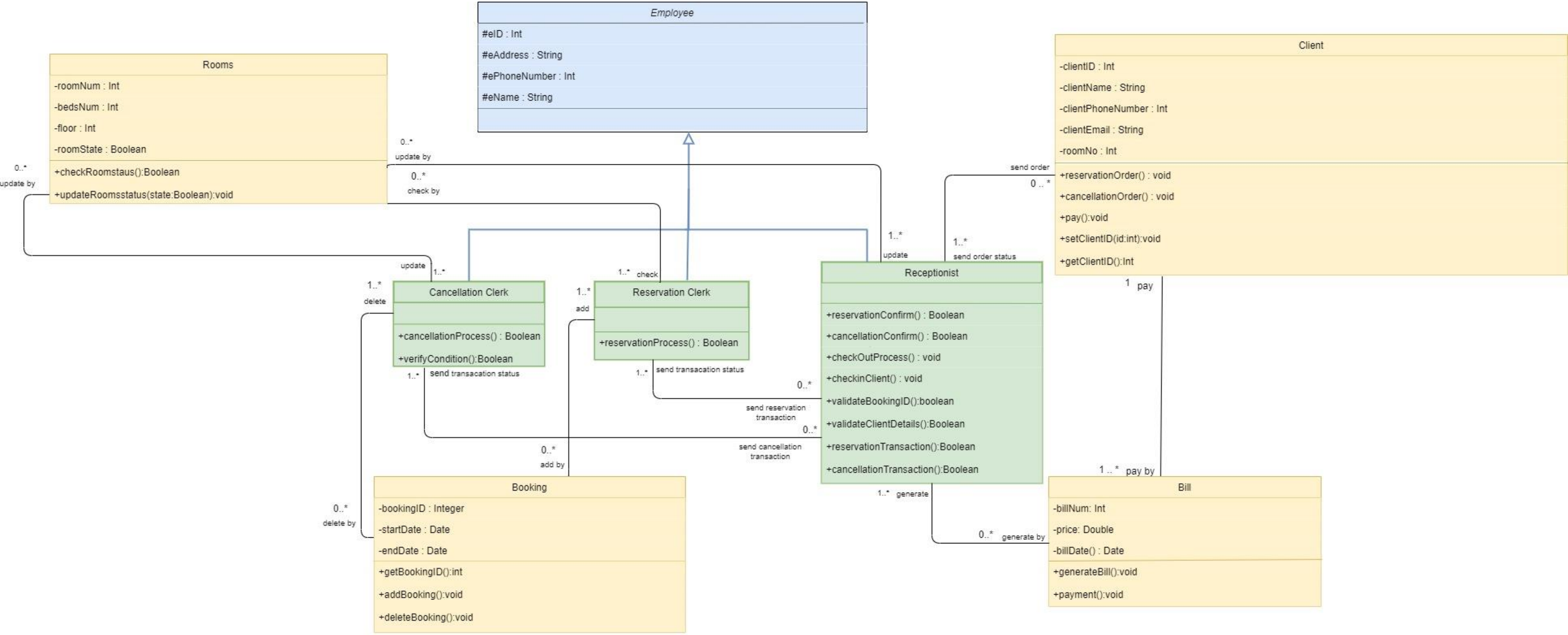
Note :-
To illustrate the drawings more, you will access them through the link
* If the link does not work, please contact the team leader

Team Diagrams - diagrams.net

Use Case Diagram



Class Diagram



Employee

#eID : Int
#eAddress : String
#ePhoneNumber : Int
#eName : String

Rooms

-roomNum : Int
-bedsNum : Int
-floor : Int
-roomState : Boolean
+checkRoomstaus():Boolean
+updateRoomsstatus(state:Boolean):void

Client

-clientID : Int
-clientName : String
-clientPhoneNumber : Int
-clientEmail : String
-roomNo : Int
+reservationOrder() : void
+cancellationOrder() : void
+pay():void
+setClientID(id:int):void
+getClientID():Int

Cancellation Clerk

+cancellationProcess() : Boolean
+verifyCondition():Boolean

Reservation Clerk

+reservationProcess() : Boolean

Receptionist

+reservationConfirm() : Boolean
+cancellationConfirm() : Boolean
+checkOutProcess() : void
+checkinClient() : void
+validateBookingID():boolean
+validateClientDetails():Boolean
+reservationTransaction():Boolean
+cancellationTransaction():Boolean

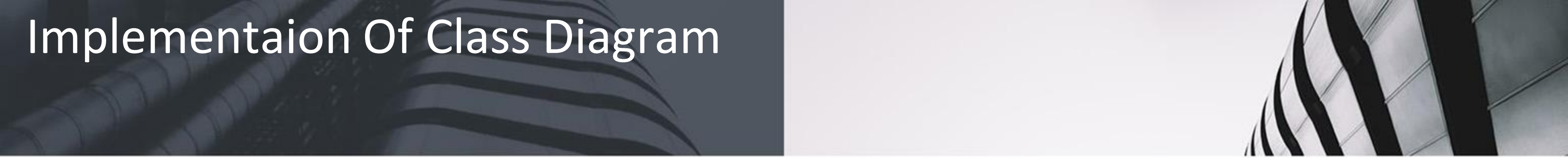
Booking

-bookingID : Integer
-startDate : Date
-endDate : Date
+getBookingID():int
+addBooking():void
+deleteBooking():void

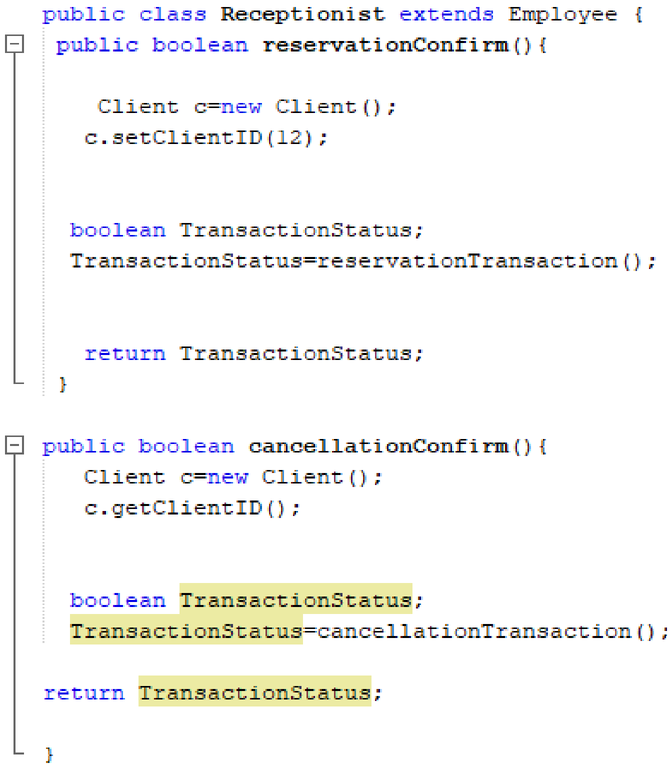
Bill

-billNum: Int
-price: Double
-billDate() : Date
+generateBill():void
+payment():void

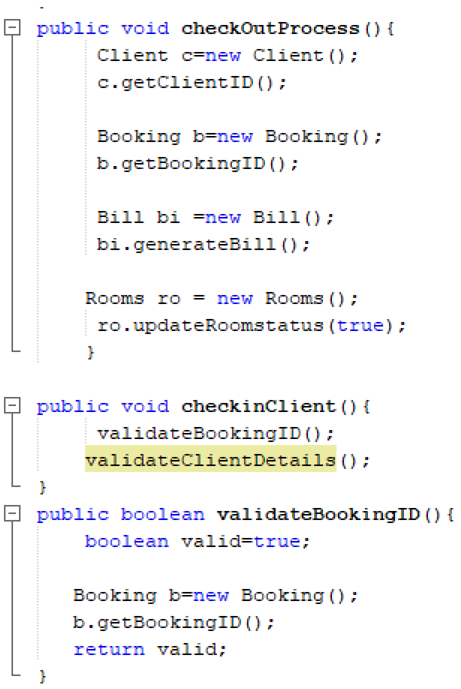
Implementaion Of Class Diagram



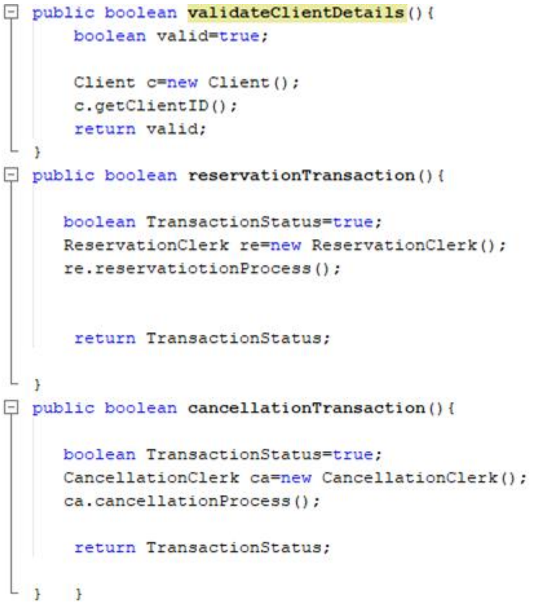
Receptionist class



```
public class Receptionist extends Employee {
 public boolean reservationConfirm(){

    Client c=new Client();
   c.setClientID(12);


  boolean TransactionStatus;
  TransactionStatus=reservationTransaction();


   return TransactionStatus;
 }

public boolean cancellationConfirm(){
   Client c=new Client();
   c.getClientID();


  boolean TransactionStatus;
  TransactionStatus=cancellationTransaction();

return TransactionStatus;

}
```



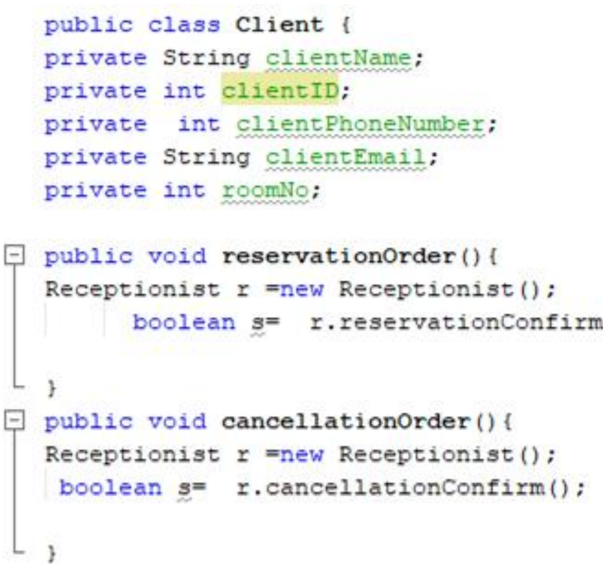
```
public void checkOutProcess(){
     Client c=new Client();
     c.getClientID();

     Booking b=new Booking();
     b.getBookingID();

     Bill bi =new Bill();
     bi.generateBill();

    Rooms ro = new Rooms();
     ro.updateRoomstatus(true);
    }

public void checkinClient(){
     validateBookingID();
    validateClientDetails();
}
public boolean validateBookingID(){
    boolean valid=true;

   Booking b=new Booking();
   b.getBookingID();
   return valid;
}
```



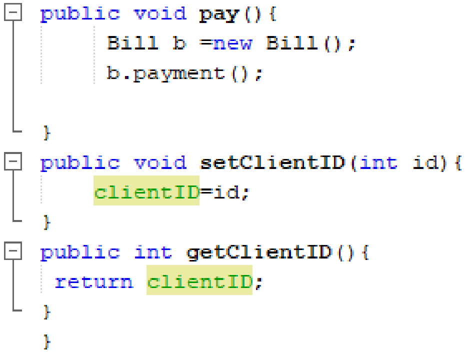
```
public boolean validateClientDetails(){
    boolean valid=true;

    Client c=new Client();
    c.getClientID();
    return valid;
}
public boolean reservationTransaction(){

   boolean TransactionStatus=true;
   ReservationClerk re=new ReservationClerk();
   re.reservatiotionProcess();


    return TransactionStatus;

}
public boolean cancellationTransaction(){

   boolean TransactionStatus=true;
   CancellationClerk ca=new CancellationClerk();
   ca.cancellationProcess();

    return TransactionStatus;

}   }
```

Cont

Client class

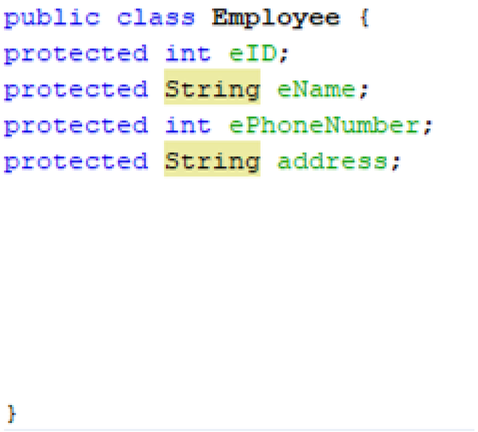


```
public class Client {
private String clientName;
private int clientID;
private  int clientPhoneNumber;
private String clientEmail;
private int roomNo;

public void reservationOrder(){
Receptionist r =new Receptionist();
      boolean s=  r.reservationConfirm

}
public void cancellationOrder(){
Receptionist r =new Receptionist();
 boolean s=  r.cancellationConfirm();

}
```



```
public void pay(){
     Bill b =new Bill();
     b.payment();

}
public void setClientID(int id){
    clientID=id;
}
public int getClientID(){
 return clientID;
}
}
```

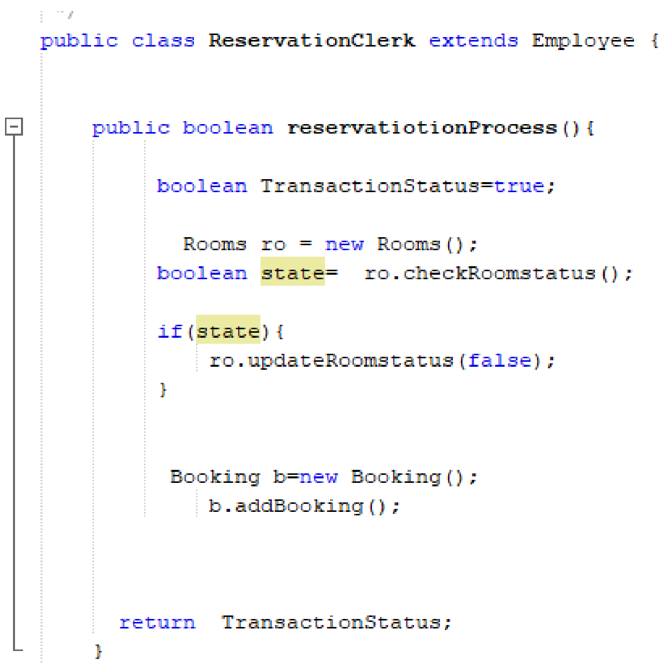
Employee class



```
public class Employee {
protected int eID;
protected String eName;
protected int ePhoneNumber;
protected String address;





}
```

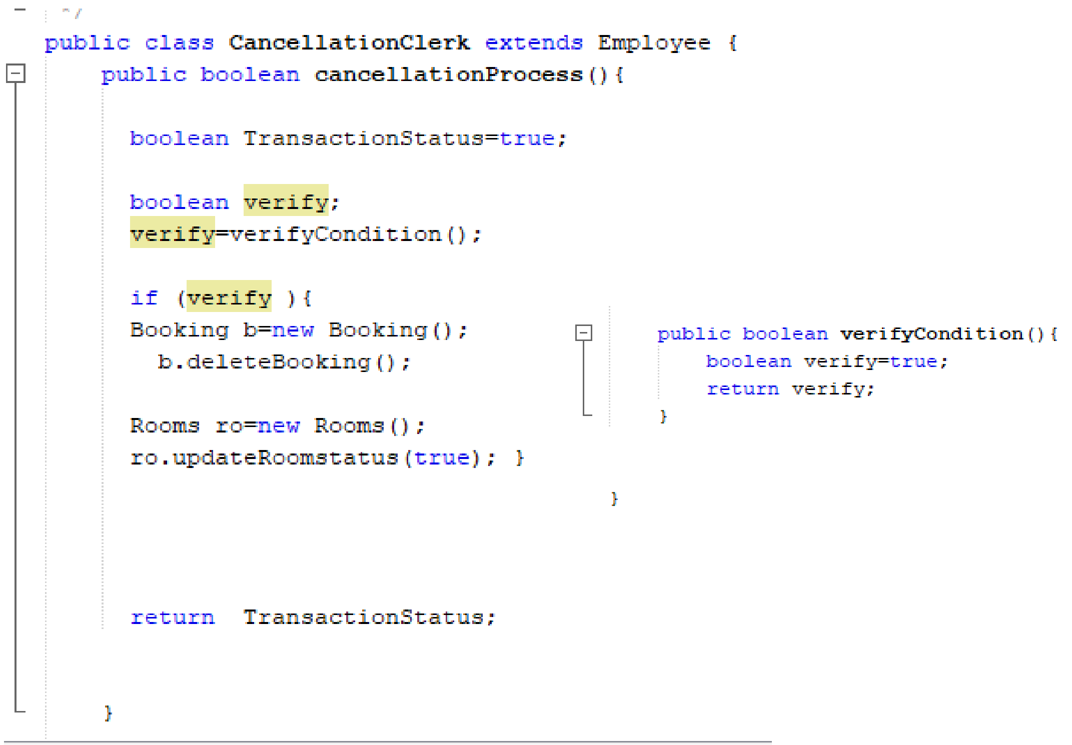
Cont

ReservationClerk class



```
*/
public class ReservationClerk extends Employee {


    public boolean reservatiotionProcess(){

        boolean TransactionStatus=true;

          Rooms ro = new Rooms();
        boolean state=  ro.checkRoomstatus();

        if(state){
            ro.updateRoomstatus(false);
        }


         Booking b=new Booking();
            b.addBooking();



     return  TransactionStatus;
    }
```

CancellationClerk class

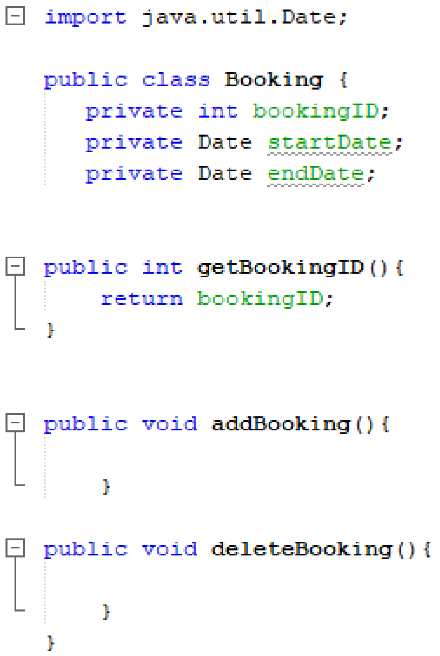


```
*/
public class CancellationClerk extends Employee {
    public boolean cancellationProcess(){

        boolean TransactionStatus=true;

        boolean verify;
        verify=verifyCondition();

        if (verify ){
        Booking b=new Booking();
          b.deleteBooking();

        Rooms ro=new Rooms();
        ro.updateRoomstatus(true); }

                                }



        return  TransactionStatus;


    }
```

```
    public boolean verifyCondition(){
        boolean verify=true;
        return verify;
    }
```

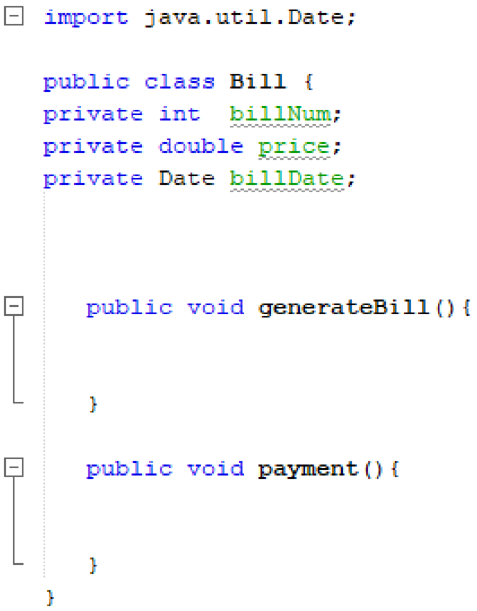
Cont

Booking class



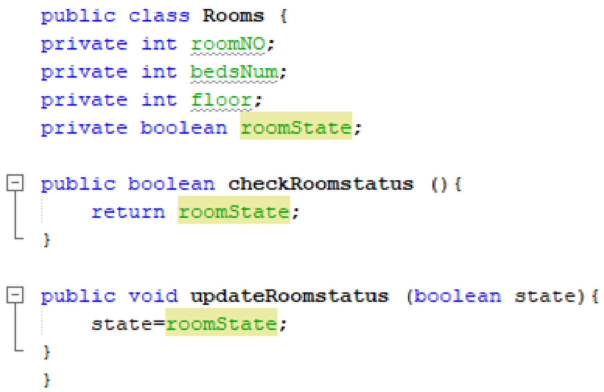
```
import java.util.Date;

public class Booking {
    private int bookingID;
    private Date startDate;
    private Date endDate;


public int getBookingID(){
      return bookingID;
}


public void addBooking(){

      }

public void deleteBooking(){

      }
}
```

Bill class



```
import java.util.Date;

public class Bill {
private int  billNum;
private double price;
private Date billDate;



    public void generateBill(){


    }

    public void payment(){


    }
}
```

Rooms class



```
public class Rooms {
private int roomNO;
private int bedsNum;
private int floor;
private boolean roomState;

public boolean checkRoomstatus (){
     return roomState;
}

public void updateRoomstatus (boolean state){
     state=roomState;
}
}
```

Individual Report

Reports Components

Section for each member and contains on :

- Report about her Individual work
- Use Case Description
- UML Activity Diagram
- UML State Machine Diagram
- UML Sequence Diagram

Bayader Saad AlHarbi

Cancellation Confirm :

I **chosen cancellation confirm use case** to apply all the individual tasks required in stage 1,2, 4 and 5 which include: use case description, UML activity diagram, UML state machine diagram and UML sequence diagram.

Of course, the lectures were not enough to cover all the concepts of the diagrams, I went back to many sites to see more and to know the difference between the diagrams in general, also had problems knowing the difference between my use case and the rest of the system's use cases, but after several discussions with the project team, the differences were clarified for everyone and emphasize it.

As for group work, I often participated in all group work, for example the UML class diagram, we needed to modify it a lot to suit our system and to make sure of the methods we needed as well as the implementation phase of the class diagram, it was difficult to clarify the relationships between classes, I modified them a lot with the help of other team members.

Some of the sites viewed :-

- https://www.geeksforgeeks.org/unified-modeling-language-uml-sequence-diagrams/
- https://www.smartdraw.com/state-diagram/

Use Case Description

Use Case : Cancellation Confirm

Purpose : Confirm cancellation reservation for client.

Require : 1- Client already initiated an order .

Event Flow :

1- Receive client order.

2- Retrieval client information from the system .

3- Create a transaction on the system to process the reservation cancellation request .

4- The status of the transaction is received after it is available on the system .

5- Respond to the client by accepting the request to cancel the reservation and returning the amount paid. END USE CASE .

Success:The client Cancellation process is successfully completed and the amount paid is received.

Extension : Reject Cancellation Order

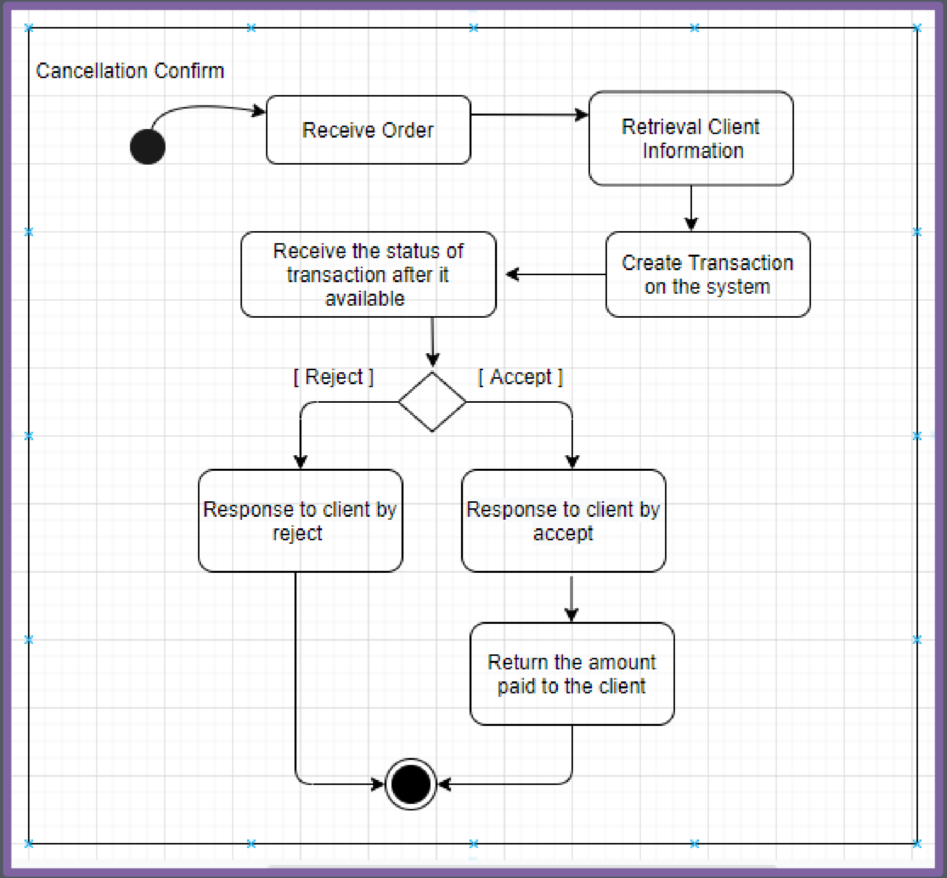
Point : Cancellation Confirm ,after step 3.

Event Flow :

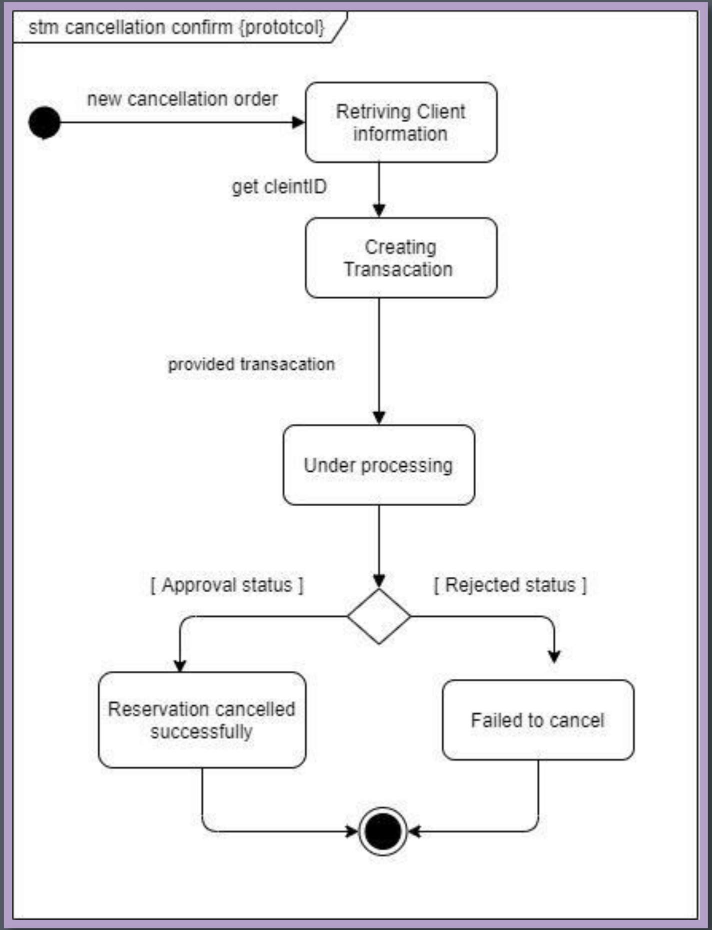
1-- Respond to the client by rejecting the request to cancel the reservation. END USE CASE .

Failure : The client don't receive the paid .

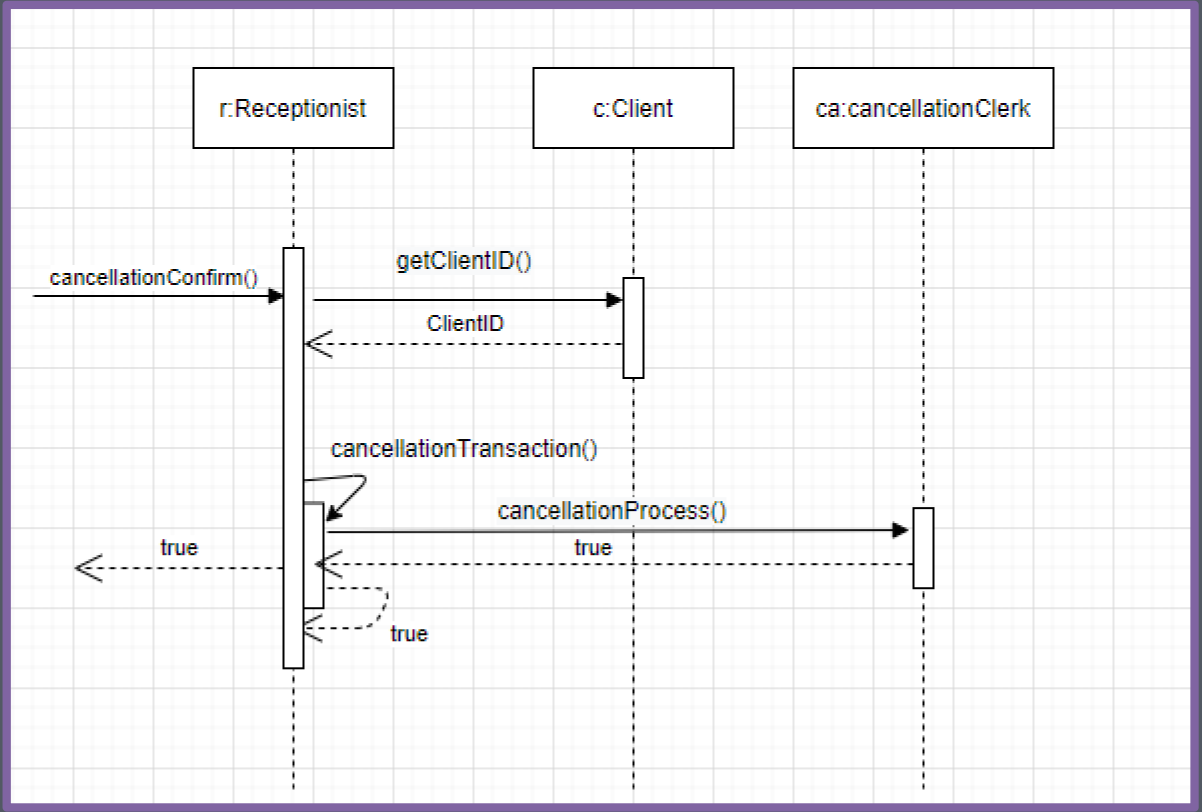
UML Activity Diagram


Cancellation Confirm
Receive Order
Retrieval Client Information
Create Transaction on the system
Receive the status of transaction after it available
[Reject]
[Accept]
Response to client by reject
Response to client by accept
Return the amount paid to the client

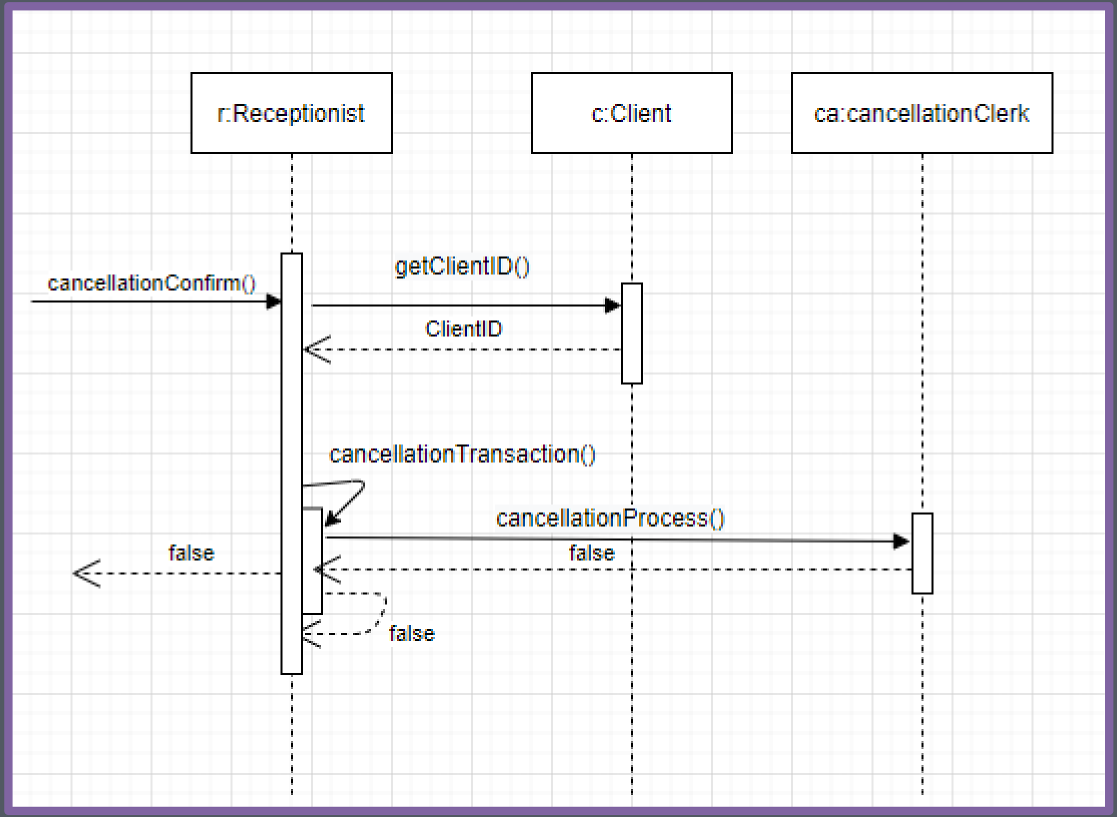
UML State Machine Diagram



UML Sequence Diagram



UML Sequence Diagram



If transaction status was rejected !

Areej jameel Alsalmi

Cancellation process use case:

I started Stage 1 use case Description in the eighth week I choose the **Cancellation process** use case . I faced several difficulties ,but I listen to the lectures and I saw some of sites and videos , And asked help from my team .

Then I completed the stage 2, 4 and 5 from week 9 to week 15 which is included: UML activity diagram, UML state machine diagram and UML sequence diagram. And I tried to Implement parts of the code with the rest of the team. And I sent the UML activity diagram to the performance supervisor to see it and make notes and I modified it more than once.

Some of sites and videos helped me :

https://creately.com
https://www.tutorialspoint.com/uml/index.htm
https://www.youtube.com/watch?v=pCK6prSq8aw
https://youtu.be/o2jDxiwM1tw

Use Case Description

Use Case: Cancellation process

Purpose: Allow a customer to cancel his reservation.

Require: 1- Receive transaction of canceling a reservation.
2- the transaction matches the terms and condition.

Event flow:

1. Receive client cancellation transaction.
2. ensure that the terms and condition are compatible with the cancellation transaction.
3. update booking information.
4. update Room information.
5. produce the transaction status in the system. END USE CASE .

Success : produce the transaction status in the system.

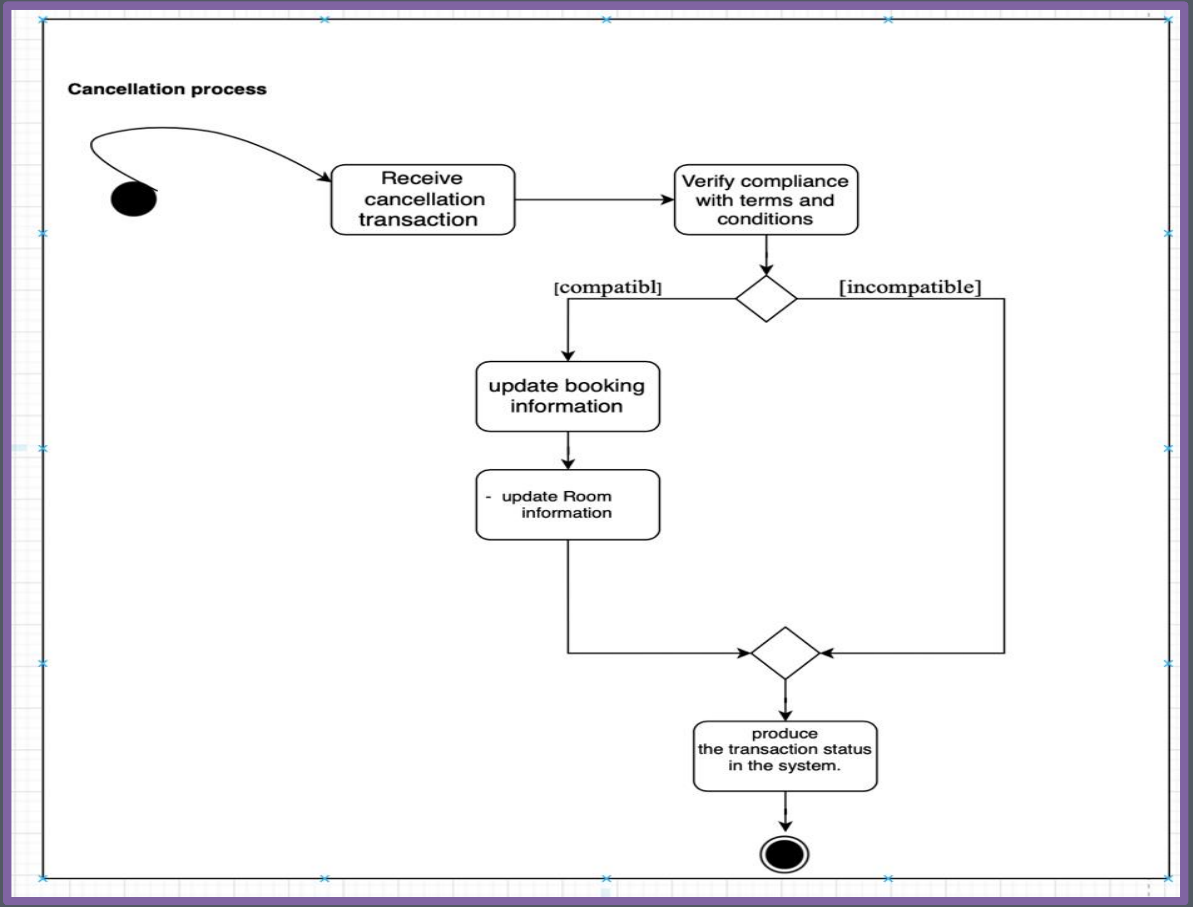
Extension: The cancellation transaction incompatible with the
terms and conditions

point: Cancellation process ,after step 2.

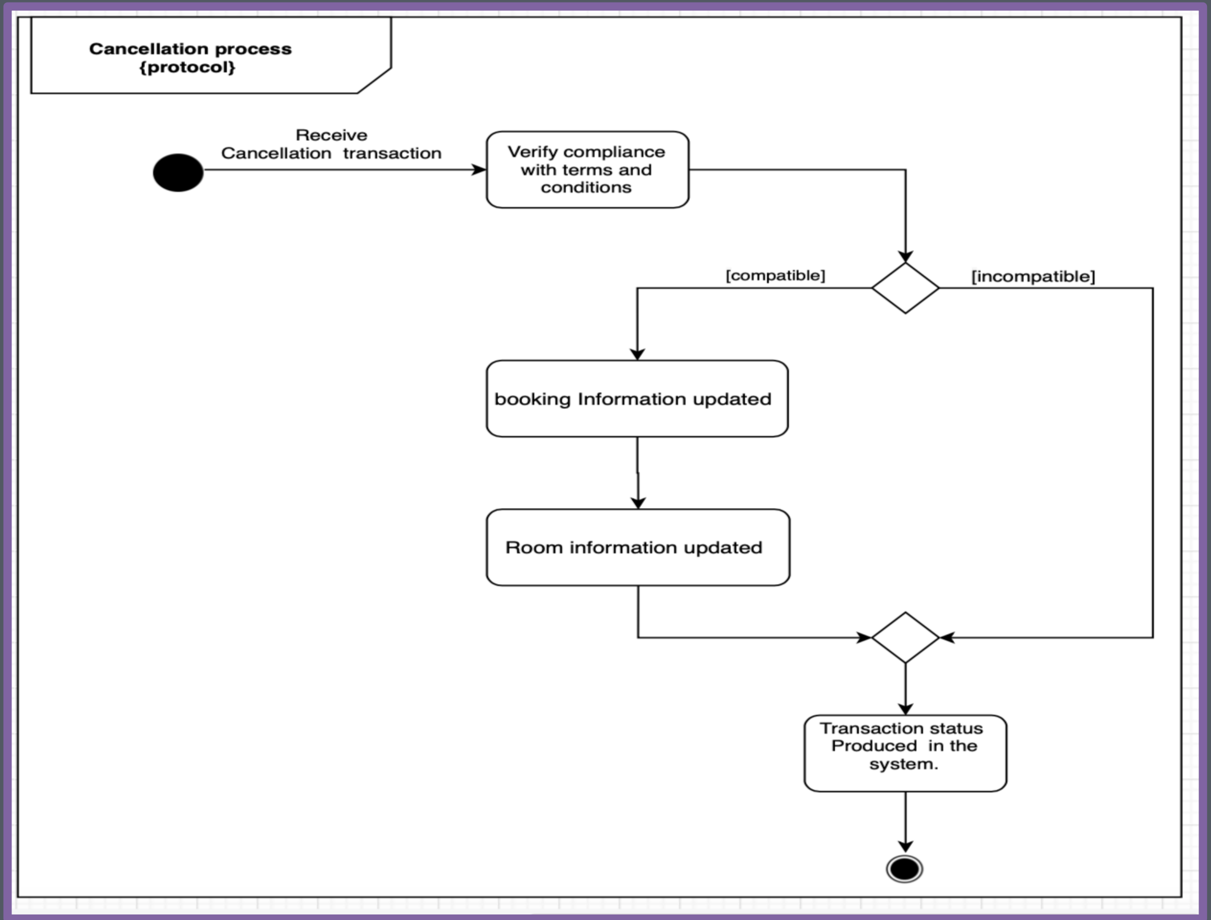
Event flow:

1-Do not update booking information.
2-Do not update Room information. RESUME USE CASE At STEP 5 .

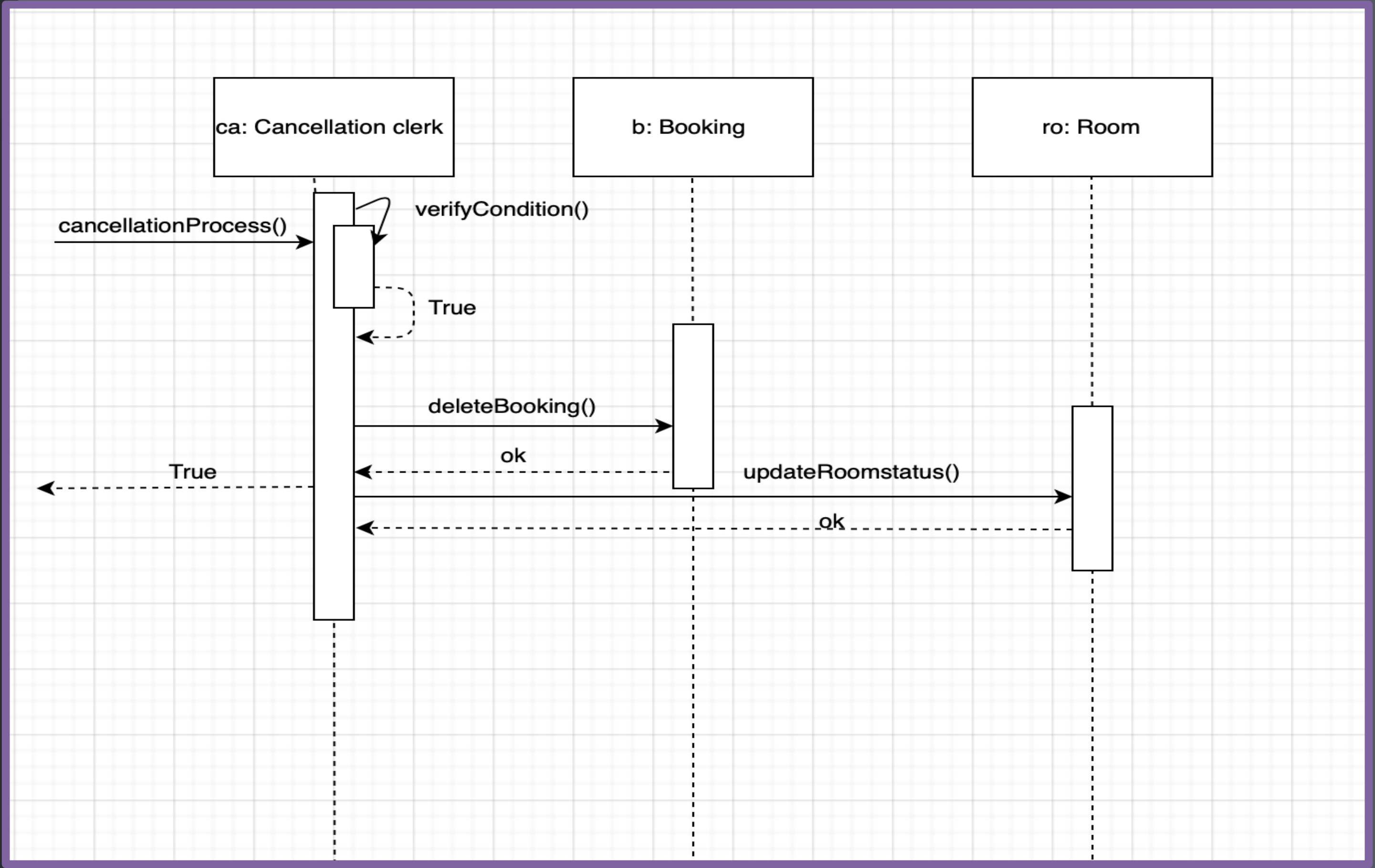
UML Activity Diagram


Cancellation process
Receive cancellation transaction
Verify compliance with terms and conditions
[compatibl]
[incompatible]
update booking information
- update Room information
produce the transaction status in the system.

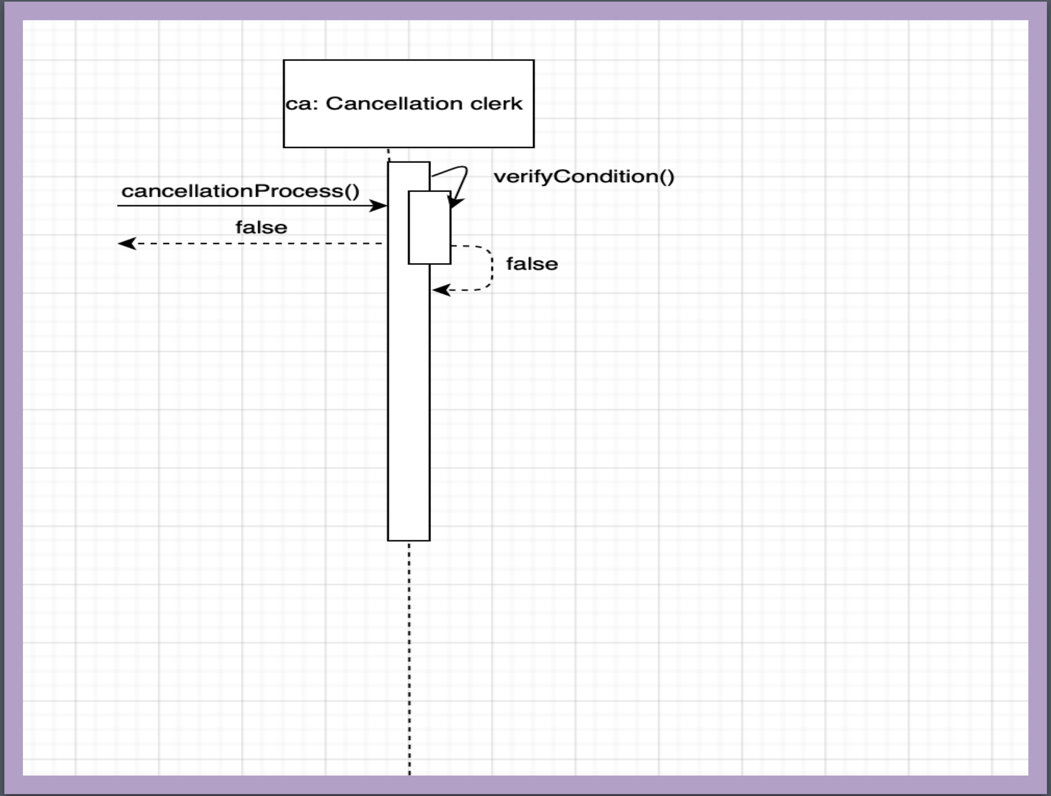
UML State Machine Diagram



UML Sequence Diagram



UML Sequence Diagram



Shatha Muslih Alharbi

I worked on **a reservation confirm** use case which is a job done by the receptionist. I started with writing the description, then the UML Activity diagram, UML State Machine diagram, and UML Sequence diagram.

I faced some issues when drawing the Activity diagram and State Machine diagram especially The part that relates to the state of rejecting reservation.

I was able to complete and finish the diagrams successfully with the help of the members of the team and read more about UML diagrams from different sources.

I also wrote the **system overview**, which I had to rewrite it many time to corresponds to our system.

Use Case Description

Use Case: Reservation Confirm

Purpose: confirm a reservation for a client.

Require: the client place a reservation request.

Event Flow:

1. Received the reservation request.
2. Add client information to the client's records.
3. Create a transaction in the system for reservation request.
4. The status of the transaction is received after the transaction updated in the system.
5. Informs the client by confirming the reservation, details of the room . **End use case.**

Success: The client request has confirm .The client has the room key and knows the room details.

Extension: Reject Reservation

Point: Reservation Confirm, after step 3.

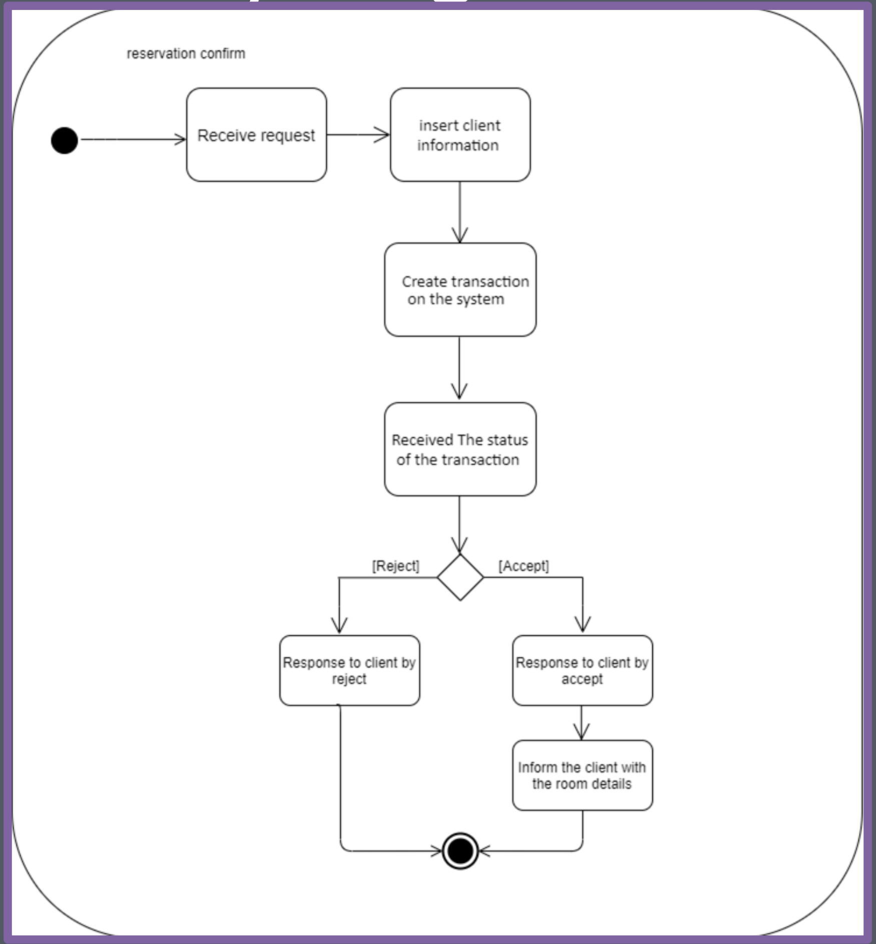
Require: The transaction has been updated to reject.

Event flow:

1. Transaction has been updated to the rejection in the system.
2. Informs the client by rejecting the reservation. **End use case.**

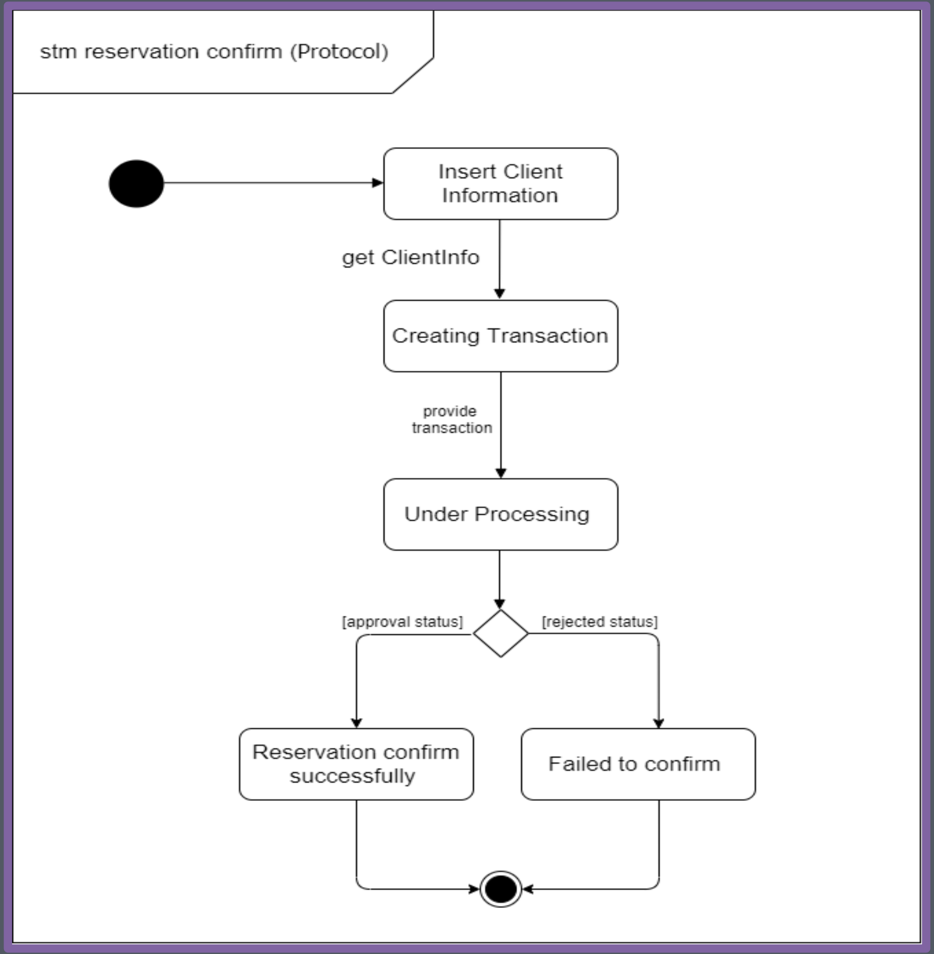
Failure: The client has not booked the room.

UML Activity Diagram

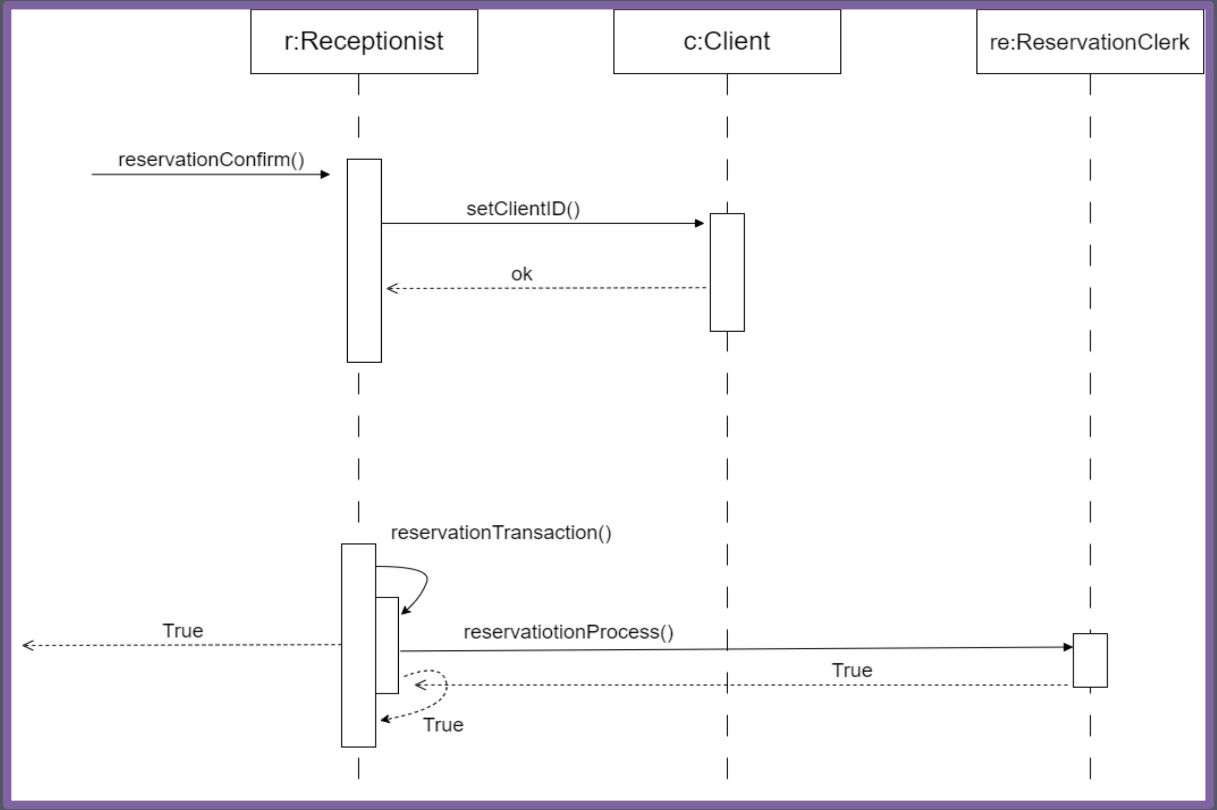


reservation confirm
Receive request
insert client information
Create transaction on the system
Received The status of the transaction
[Reject]
[Accept]
Response to client by reject
Response to client by accept
Inform the client with the room details

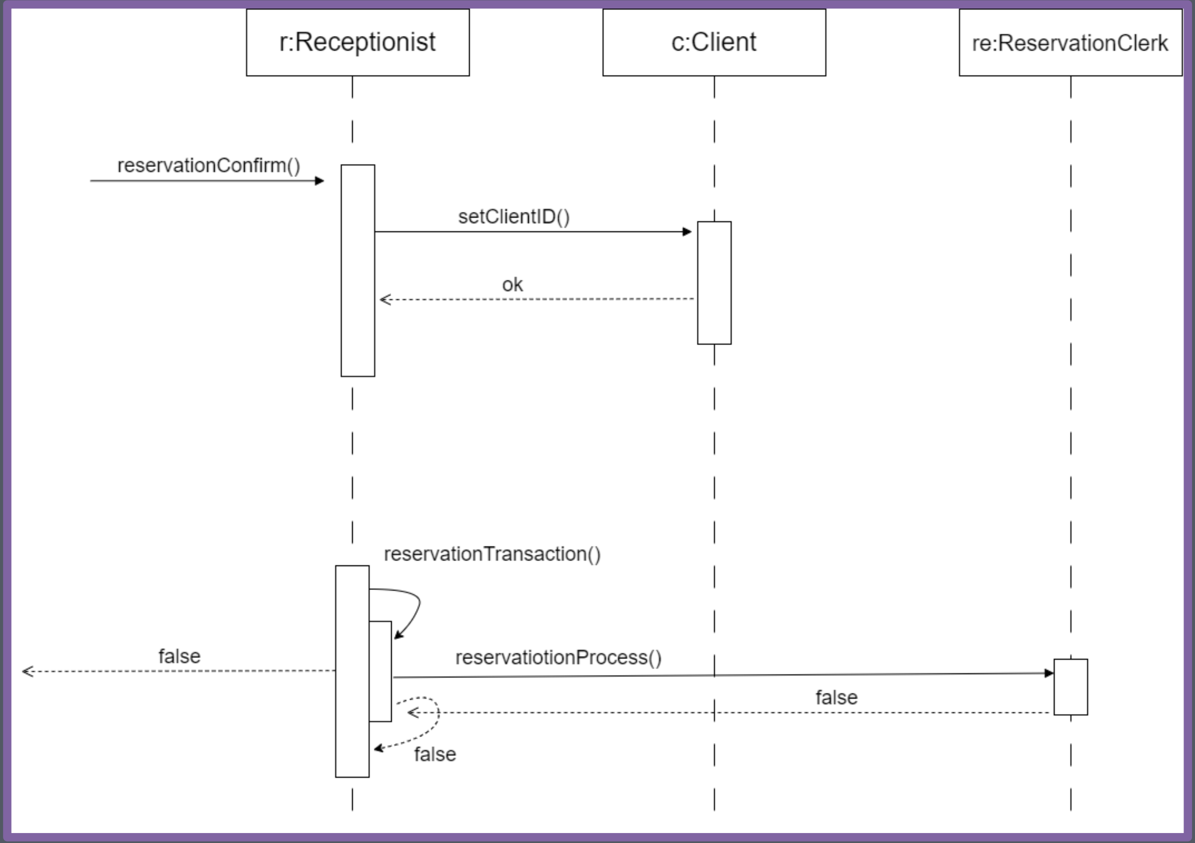
UML State Machine Diagram



UML Sequence Diagram



UML Sequence Diagram



Safia Marzoog Alfahmi

I have chosen the checkout process

To implement stages 1, 2, 4 and 5

In stage 1 I created a description for this process, in this stage I used physical DFD

In phase 2

I created a UML activity diagram

In stage 3

She discussed classroom, attribute and relationships with team members

In stage 4

I chose Protocol to execute the drawing

In stage 5

Implement parts of the code with the rest of the members

One of the problems I encountered during implementation was "modification".

Because any modification I make to one of the previous drawings will affect all graphics related to the checkout process

Use Case Description

Use case : checkout process

Purpose : allowing the client to leave he hotel.

Require: the client submit a order to leave to the recetionst .

Event flow *:*

1- Receive client request.

2-takes the reservation information from system .

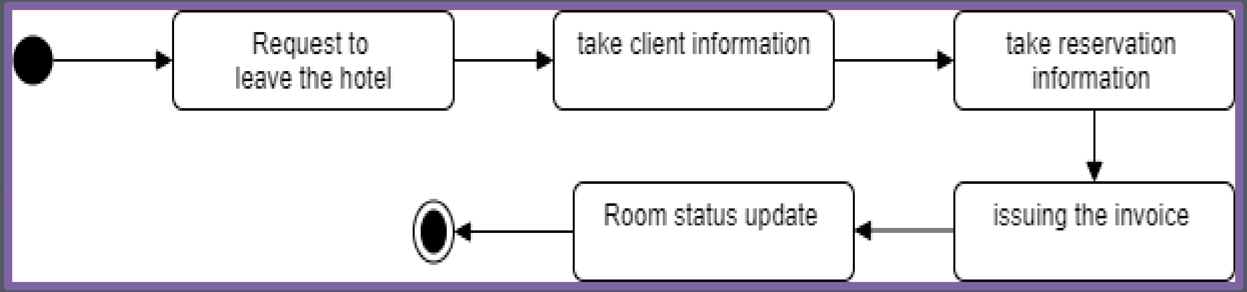
3-takes client information from system .

4-After the client pays the amount, an invoice will be issued for him.

5-Room status update to avilable . END USE CASE.

Success : If an invoice is issued to the client and if the condition of the room changes and the room becomes available for rent.

UML Activity Diagram



Request to leave the hotel

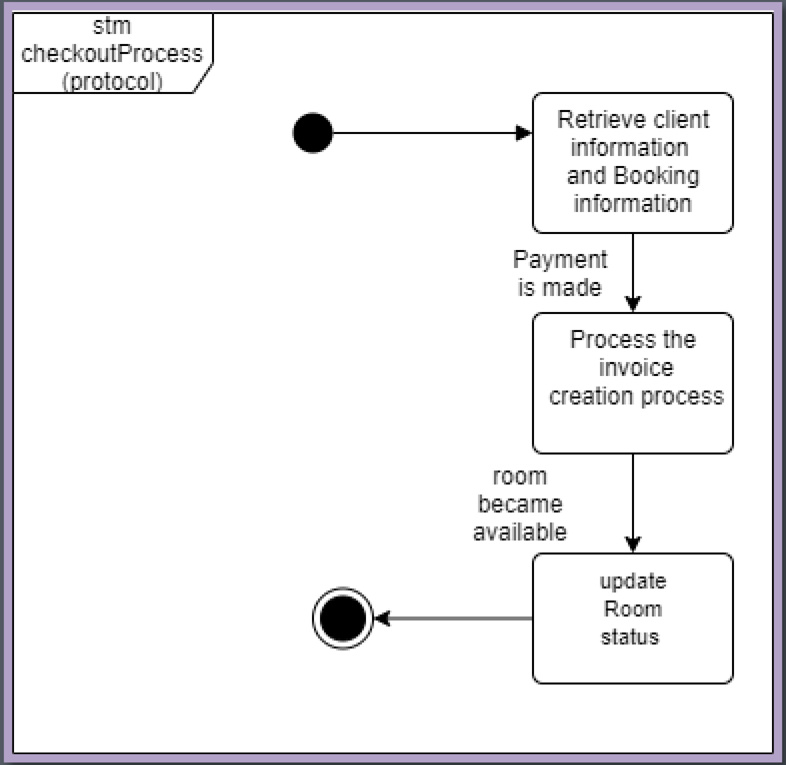
take client information

take reservation information

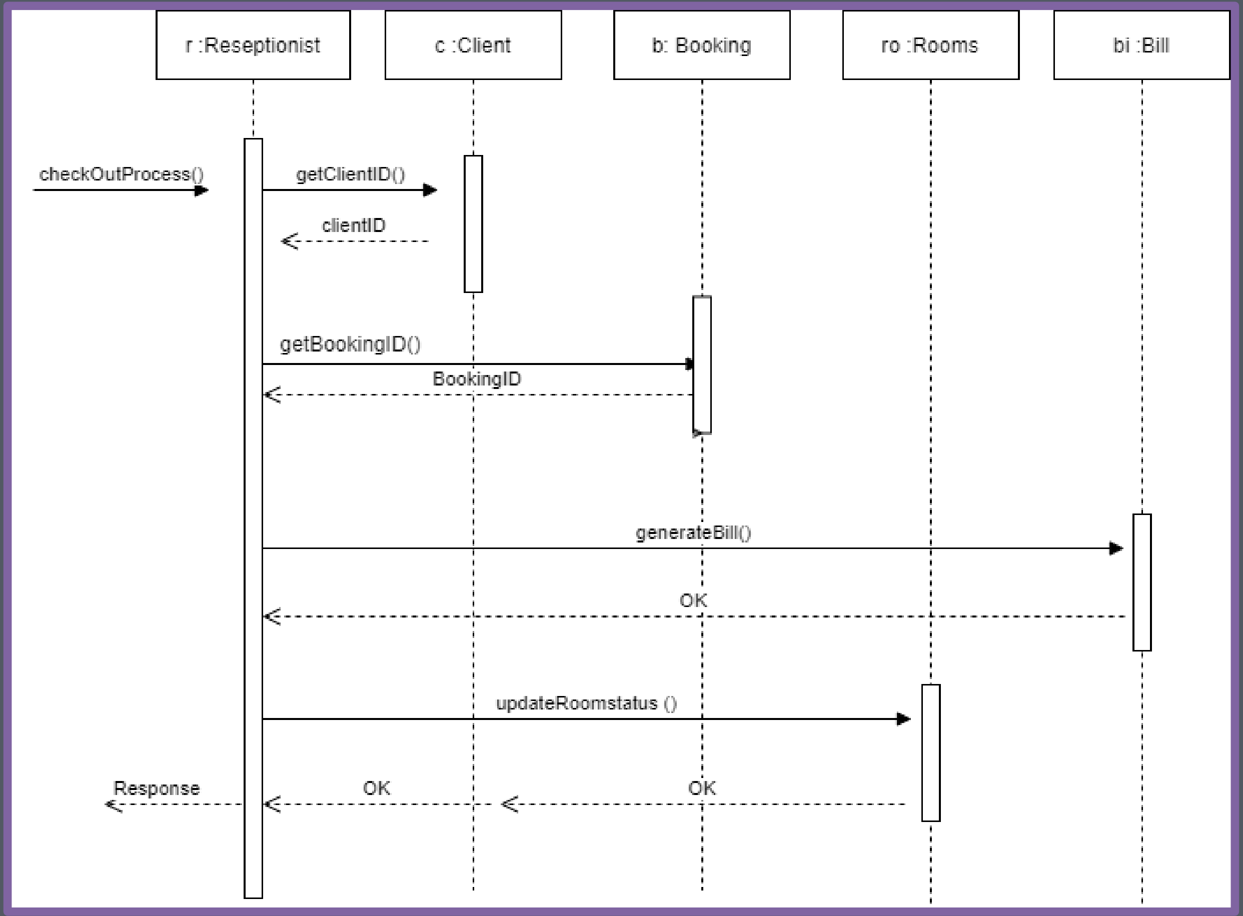
issuing the invoice

Room status update

UML State Machine Diagram


stm checkoutProcess (protocol)
Retrieve client information and Booking information
Payment is made
Process the invoice creation process
room became available
update Room status

UML Sequence Diagram



Atheer Waleed Al Hazmi

reservation process use case :

My individual part in the project is to work on the reservation process use case, I started with :

1. writing the description for the use case.
2. Drawing the UML Activity diagram.
3. Drawing UML State Machine diagram.
4. Drawing UML Sequence diagram.

It was difficult to differentiate between the diagram, mostly the alternative part in the activity diagram and the state machine diagram I had to redraw the diagrams many time to ensure it is correct , but by referring to the team members as well as reviewing the educational material I was able to finish it, however, I modified a lot until I came to the best diagram.

Use Case Description

Use case : Reservation Process

Purpose : Allow the client to reservation.

Require :. 1-Receive transaction about client reservation.

2-there are available room match client order.

Event Flow :

1- Receive client reservation transaction.

2- Ensure the availability of the required room.

3- Reserve the room for the client.

4- Update booking information.

5- Produce the transaction status in the system .END USE CASE.

Success : Provide the transaction status in the system successfully.

Extension : Unavailability of the room

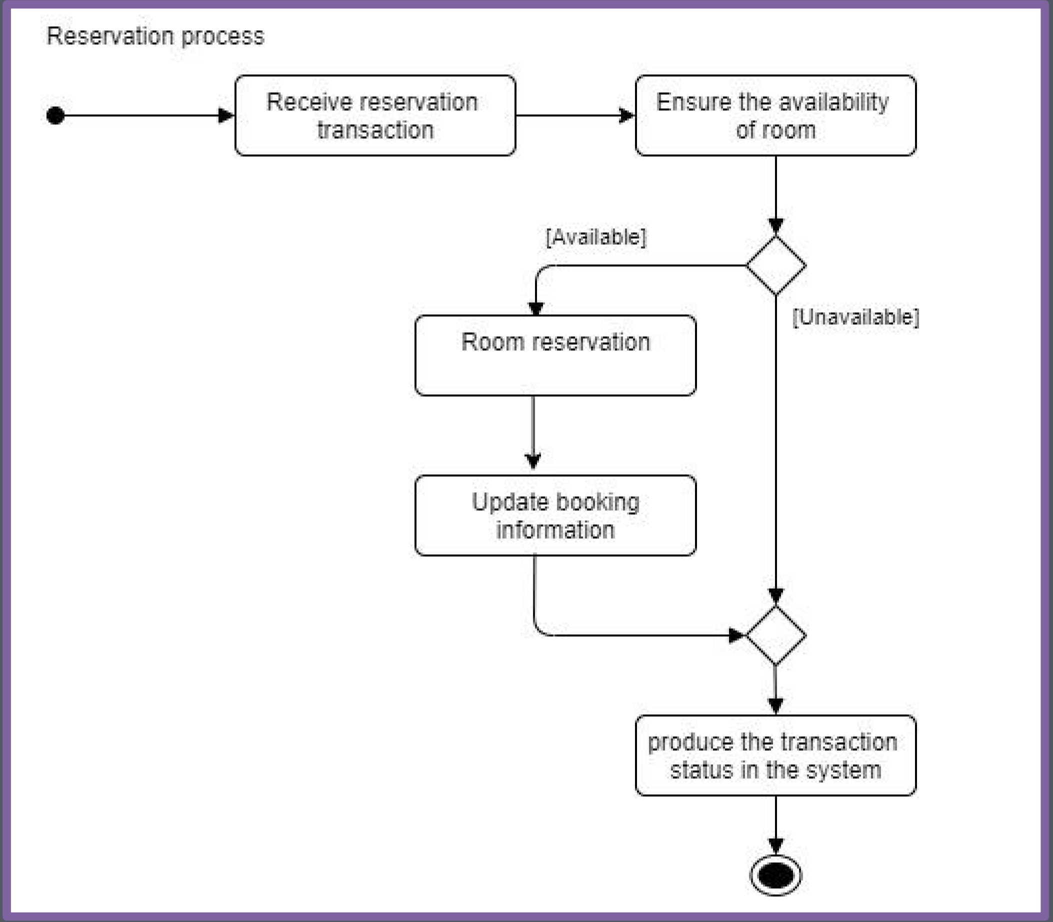
Point : Reservation Process , after step 2.

Event Flow :

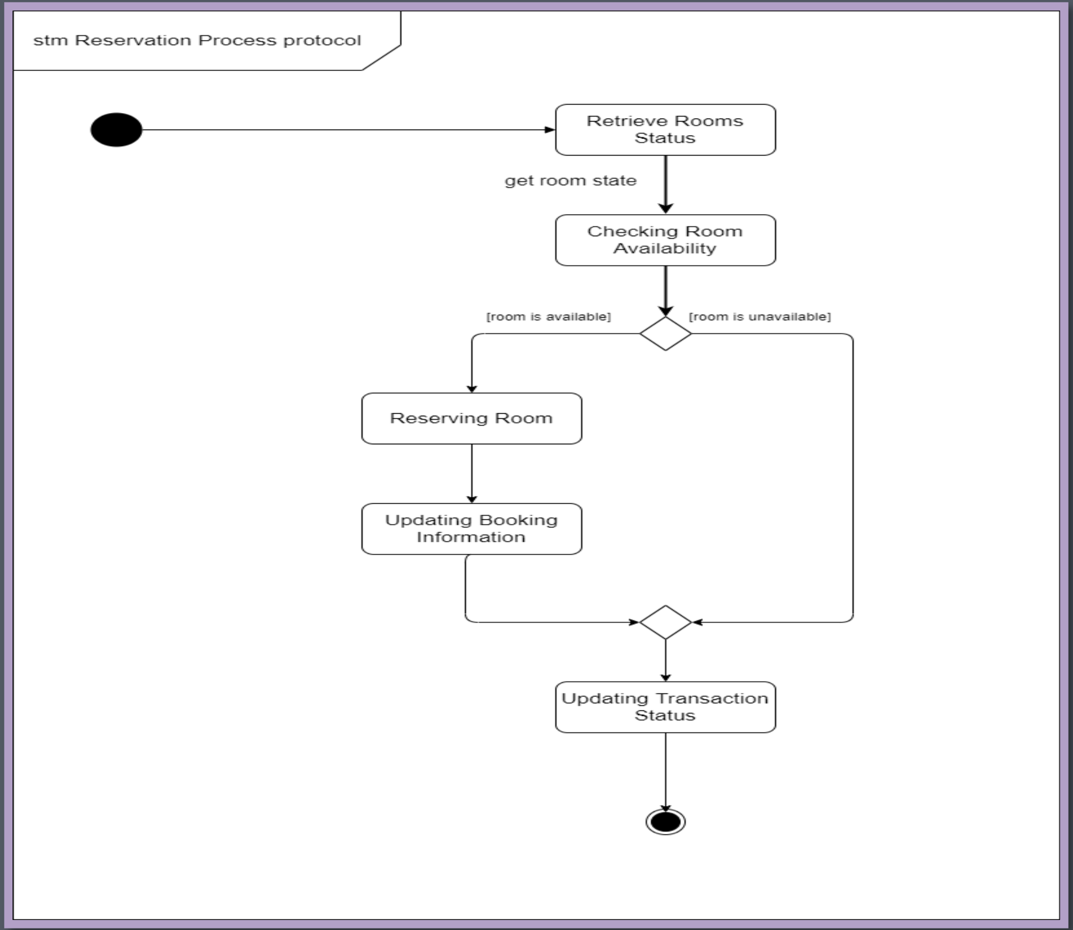
1- Don't reserve the room for the client.

2- Don't update booking information. RESUME USE CASE AT STEP 5.

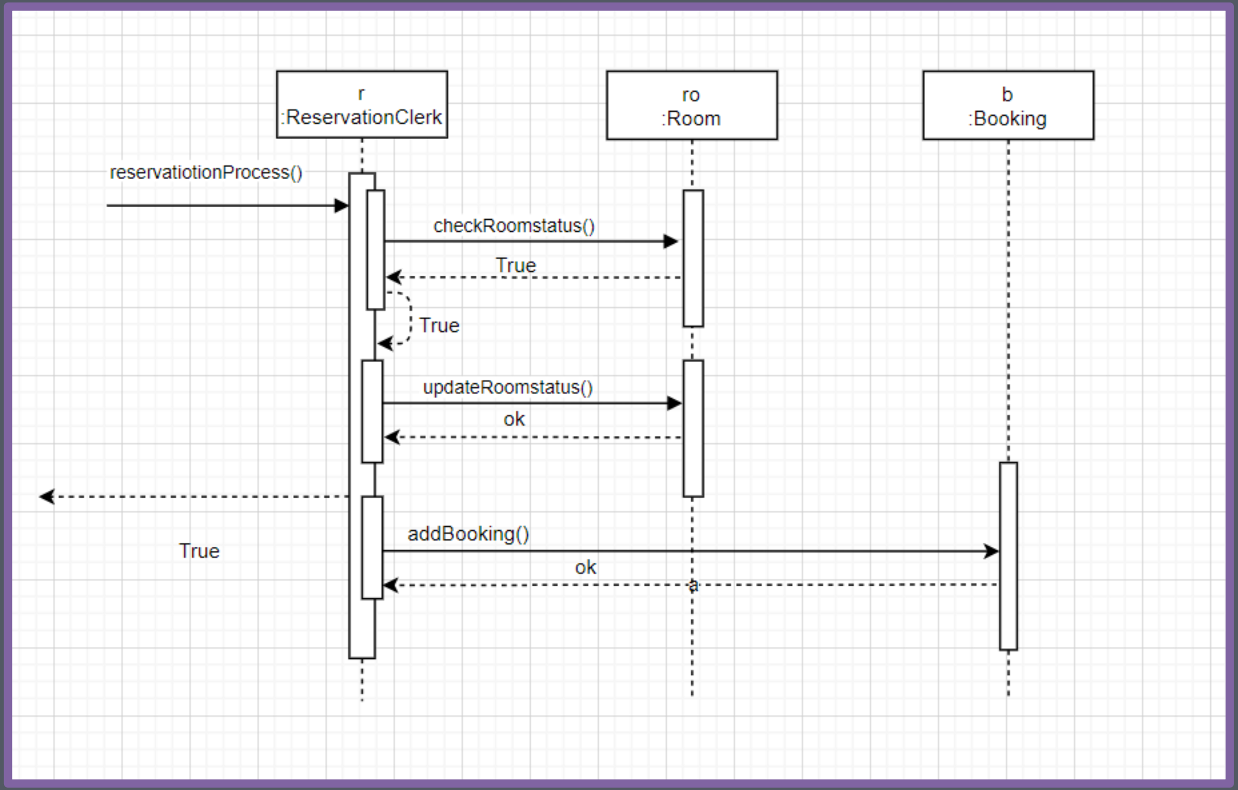
UML Activity Diagram


Reservation process
Receive reservation transaction
Ensure the availability of room
[Available]
[Unavailable]
Room reservation
Update booking information
produce the transaction status in the system

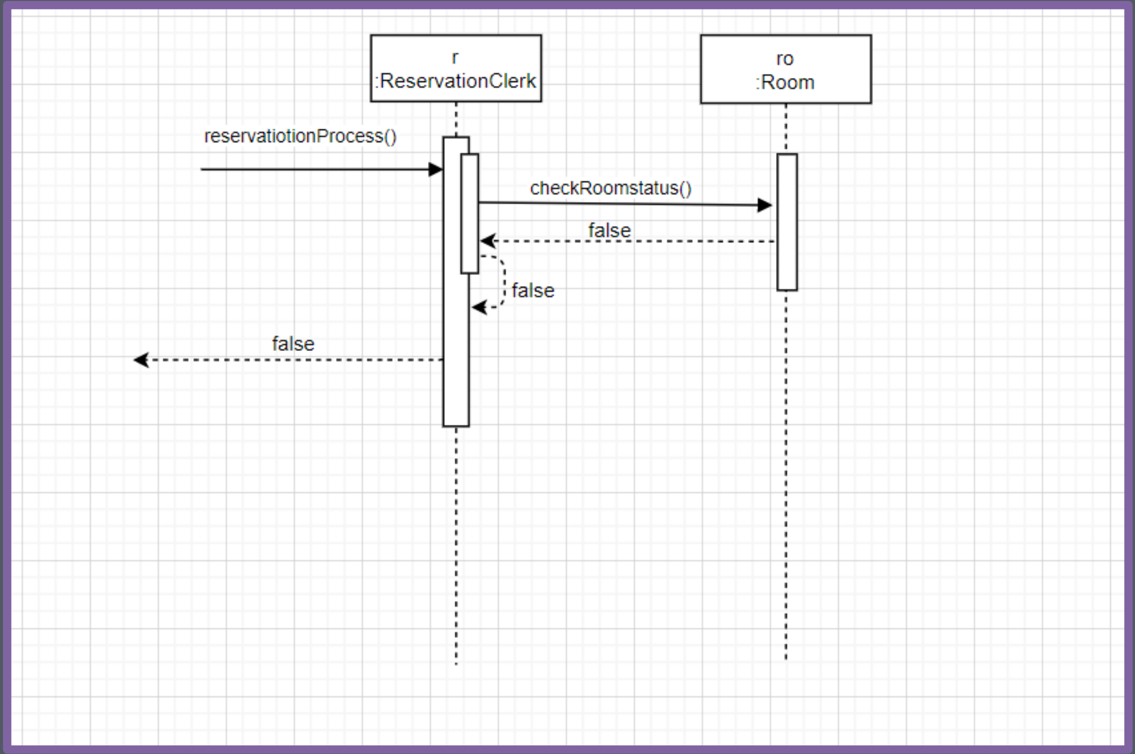
UML State Machine Diagram


stm Reservation Process protocol
Retrieve Rooms Status
get room state
Checking Room Availability
[room is available]
[room is unavailable]
Reserving Room
Updating Booking Information
Updating Transaction Status

UML Sequence Diagram



UML Sequence Diagram



Afrah Bakheet Allhyani

Check-in client process
Individual report

This process is done by the receptionist where the receptionist allow the client to occupy a room but after validation of booking ID and Client details .

I wrote the use case description, After that, I drew the UML Activity diagram, UML State Machine diagram, and UML Sequence diagram for this process .

I faced some difficulties while drawing the diagrams but by doing some research, I managed to complete it .

I also did discussions with the team members to complete the teamwork.

I've participated in Implementing the UML Class diagram in java where we discovered some problems in the UML Class diagram, but we were able to fixe it and complete the work

Use Case Description

Use case : check-in client

Purpose : allow client to occupies a room at the hotel.

Require :   there is a reservation for the client

Event flow:

1- the receptionist inquiries the client for booking ID.

2- the receptionist validates the booking ID.

3- the receptionist validates client details.

4- the receptionist confirms client check-in. End use case.

Success: the client has successfully occupied a room.

Extension: cancel client check-in

Point: check-in client, after step 2 or step 3.

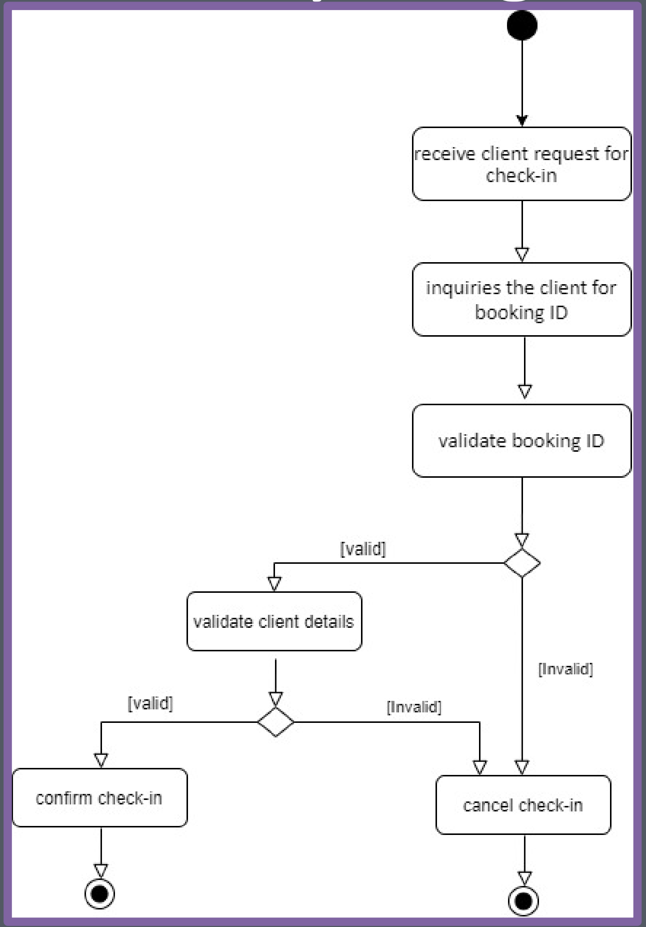
Require: the client provided invalid booking ID or invalid personal details.

Event flow:

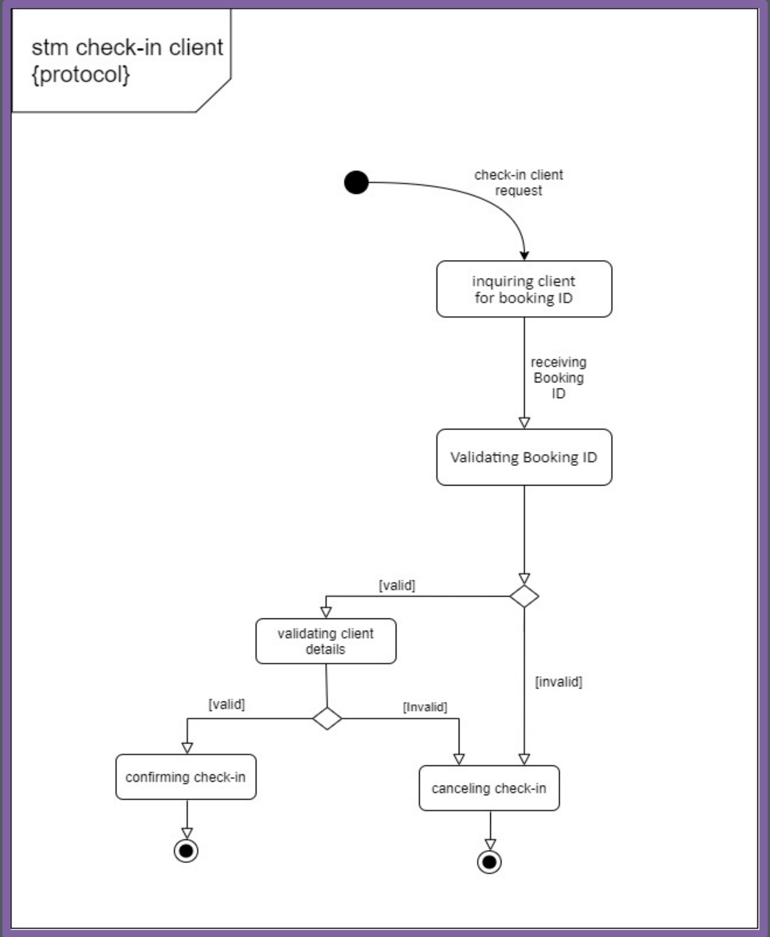
1-cancel client check in .End use case.

Failure: the client cannot occupy a room.

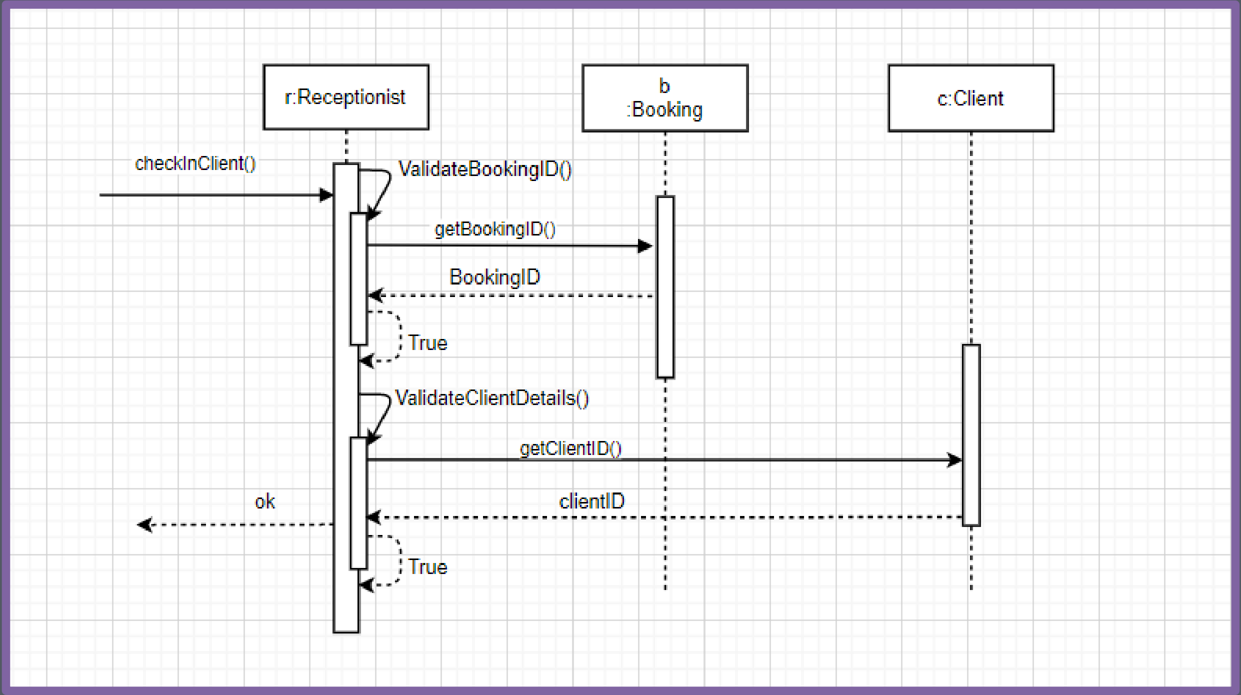
UML Activity Diagram



UML State Machine Diagram

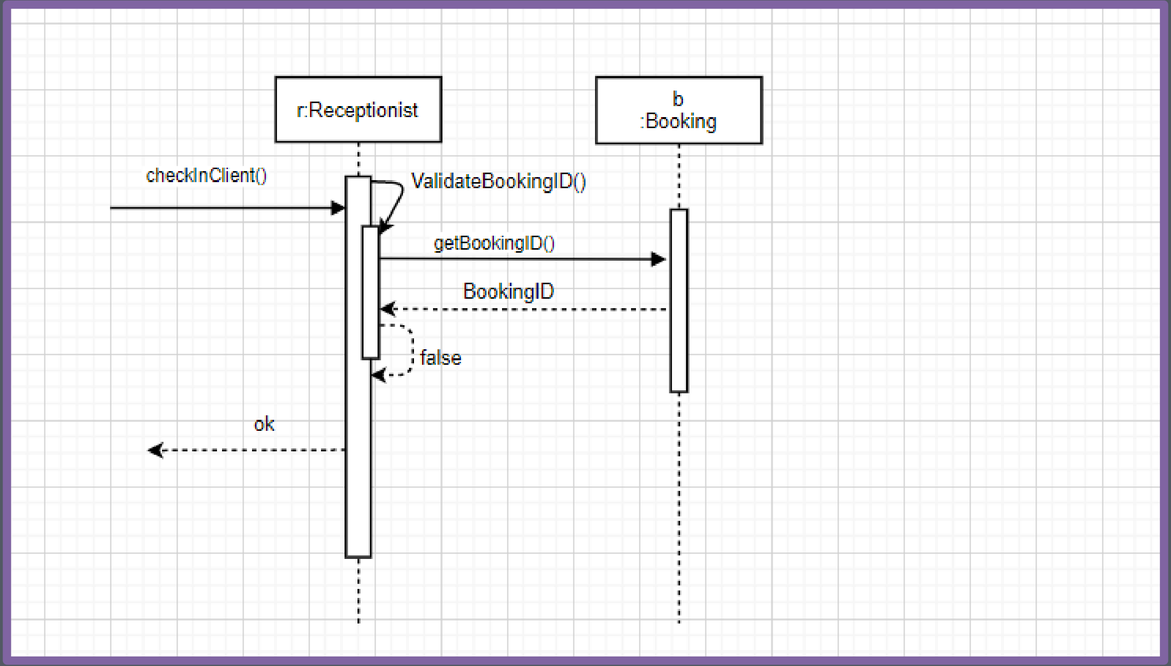


UML Sequence Diagram



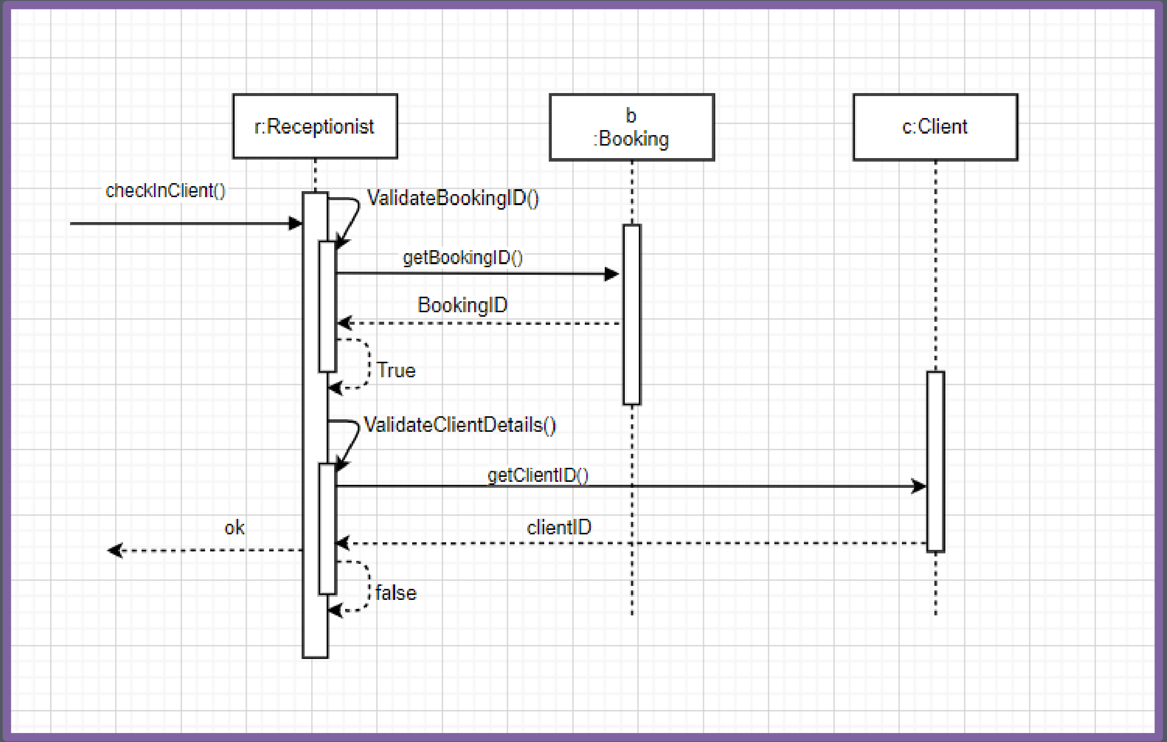
Scenario if the process done successfully

UML Sequence Diagram



Scenario if the Booking ID is invalid

UML Sequence Diagram



Scenario if the client ID is invalid

Sara Ahmed Almajnuni

My work:

Writing use case description for reservation order use case, and drawing activity diagram, state machine diagram,

and sequence diagram for this use case ,and participated in class diagram.

Difficulties and problem:

Redraw according to the following stages and the

difficulty dealing with the program used in drawing.

Solutions:

- To seek help from those who have experience.

- Searching for learning resources to collect information to complete the drawing.

- Using a program is easier to draw.

- Team members cooperated to solve the problem.

Sources and references:

-The course.

- YouTube.

-Drawn by "Diagrams.net"

Use Case Description

Use Case :
Reservation Order

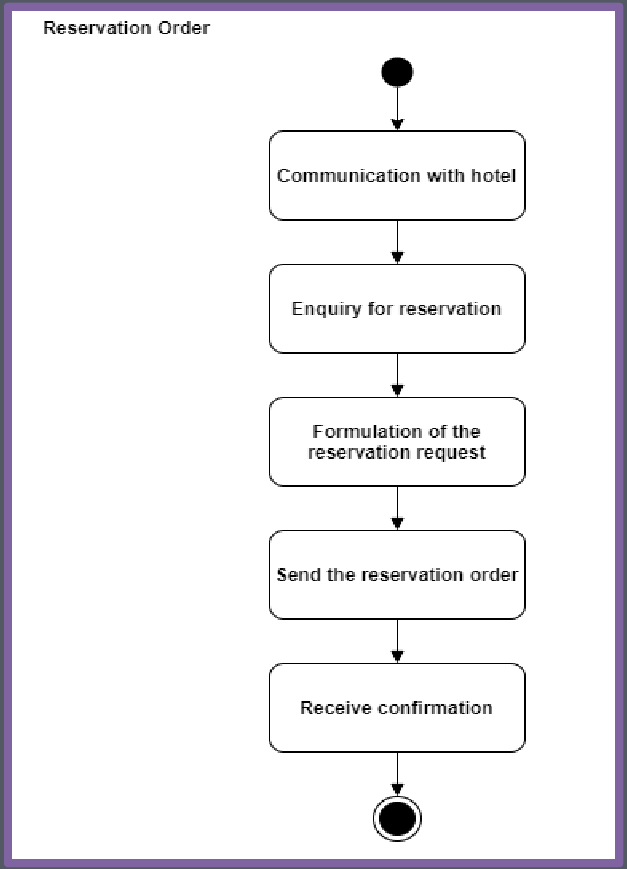
Purpose:
This use case enable client to make reservation order in hotel.

Require:
Provide all client details to receptionist .

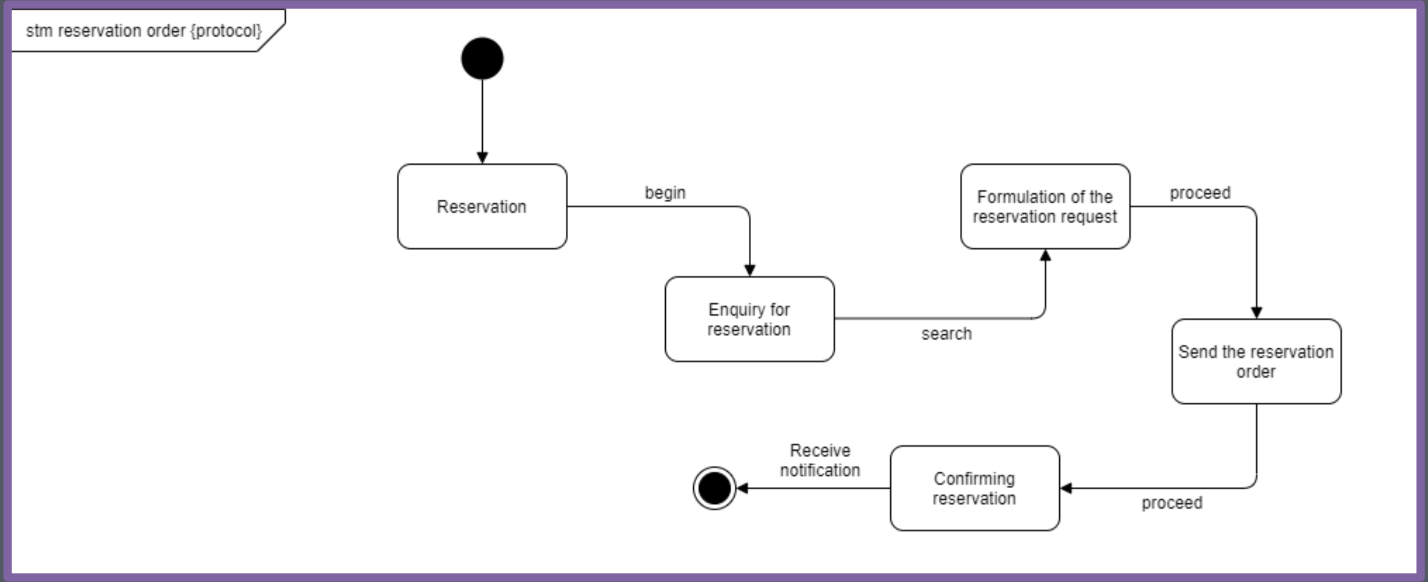
Event Flow:
1.Communication with hotel.
2. Enquiry for reservation.
3. Formulation of the reservation request.
4. send the reservation order.
5. receive confirmation .END USE CASE.

Success :
The whole process done successfully .

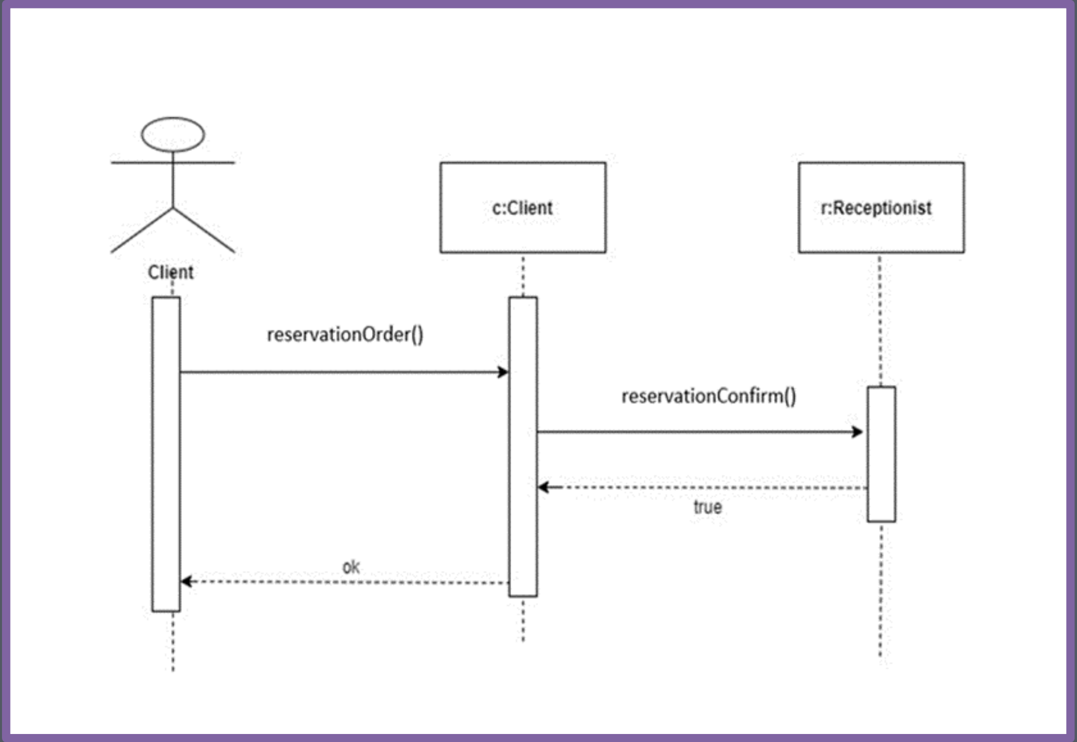
UML Activity Diagram



UML State Machine Diagram



UML Sequence Diagram



Client
c:Client
r:Receptionist
reservationOrder()
reservationConfirm()
true
ok