1. **Uses Cases description**

Description of Use Case “new visit”:

1. Actors:
   * Front-Desk Clerk
   * Customer
   * System
2. Main success scenario:
   * Customer ask for room reservation service at given period
   * Clerk checks in system if customer have no active visit and room is available at given time and adding new reservation
   * If customer need to avail various services during a new visit, clerk adds the service and requests a resource (adding new service to visit use case)
   * System verifies data
   * System saves data
3. Alternate scenario:
   * 1. System informs that customer have active visit at given period:

* Clerk informs customer that he/she can not make a reservation at given period and ask if he/she is interested reservation in another period.

1. Time dependencies:
   * Frequency of Occurrence: ~5-10 times/day
   * Anticipated accumulation: during the holidays
   * Typical realization time: 15 min.
   * Maximal realization time: 1 hour
2. Values obtained by the actors after the end of the use case:
   * Information for customer about success or failure of the room reservation,
   * New record in the reservation system related to the currently entered reservation.

Description of Use Case “adding new service to visit”:

1. Actors:
   * Front-Desk Clerk
   * Customer
   * System
2. Main success scenario:
   * Customer ask for uses new service,
   * Clerk makes request for a resource related to the service in system,
   * System checks for that availability of the resource and adds that resource to the list of availed services during that visit,
   * Clerk informs that new service was added to visit.
3. Alternate scenario:
   * 1. System informs that resource is unavailable:

* Clerk informs customer that particular service is is unavailable and offers her/him other service.

1. Time dependencies:
   * Frequency of Occurrence: ~30 times/day
   * Anticipated accumulation: mornings and evenings
   * Typical realization time: 2 min.
   * Maximal realization time: 20 min
2. Values obtained by the actors after the end of the use case:
   * Information for customer about success or failure of adding new service
   * New service added to that visit in system.
3. **Class diagram description**

* Visit class beside group of fields like startDate, endDate, isActive describing entity consists Customer object and list of HotelServices which are using during visit.
* HotelService is an abstract, basic class for concrete service classes like RoomReservationService, RestaurantService, SaunaService, PoolService and CasinoService. Each of them has specific own fields and methods as well as common.
* HotelService consists of SingleResources.
* SingleResources is an abstract, basic class for specific classes like Pool or Room.
* Two classes- FrontDeskClerk and Manager- heritate from abstract, basic Employee class.

1. **Communication diagrams contracts**

* CheckOut operation:

|  |  |
| --- | --- |
| **Contract : Checkout Customer** |  |
| Operation: | -CheckOutCustomer( customer: customerId ) |
| Cross References: | -Use Cases: CheckOut Customer, Pay, Send Receipt |
| Preconditions: | -There is one active customer’s visit. |
| Postconditions: | - Customer has no active visit. |

* NewVisit operation:

|  |  |
| --- | --- |
| **Contract : createNewVisit** |  |
| Operation: | -createNewVisit(customerId ) |
| Cross References: | -Use Cases: new visit, check if any customer’s visit is active |
| Preconditions: | -Customer is registered in database and doesn’t have any active visit |
| Postconditions: | - New Visit item saved in database  - Visit was associated with the customer  - Customer has new active visit |