**Pedro Manuel Moreira Teixeira**

**António José Ribeiro Mendes**

**Hotel Management Project**

**Description of the work**

As the final project of SIBD we decided to do a database that manages the room booking of a hotel, both for guest, or third-party agents, the payment system, and the managing of the cleaning crew.

This project should implement a simplified version of the daily needs of a small hotel.

**Applications**

* The reservations of the rooms can be made by the client himself or by an agency. For each client booking we need to store, his first and last name, the address, phone number, SS number, whether it has disability, and a unique id.
* If the reservation is made by an agent, we should store the agent first and last name, the address, phone number, SS number of the agency, the agency name, a unique id and a reference to the client.
* It’s important to say that each entry on the booking table must always have a client associated, even if the reservation is made by an agent. The only job of the agent is to book and pay in the name of the client.
* Each booking must have an id and keep track of the id of the agent, the payment id, the guest id, and the room id that is referencing. The number of occupants, the room number, the reservation date, the total price, and the period of time the room is occupied should be stored.
* It is only possible to book one room on each booking.
* Each room should have a unique id, a number, occupation state variable, and a reference to: the floor; the rating of the room; and the room type.
* The cleaning of each floor is made by at least one specialized team.
* The cleaning service is not available in all rooms
* Each team is made up of cleaning staff, that can integrate multiple teams, and each team is only responsible for cleaning one floor, and each floor can be cleaned by many teams.
* Each staff member should have an id, a staff number, the shift that they are working on, the salary, and basic info such as name, address, email, phone number and sex.
* The date and the time spent on each cleaning service should be stored in memory.
* Each floor should keep track of the number of rooms it has, a unique id, a floor number, if it is ready to receive disabled people, and a reference to the team that is cleaning it, and each room.
* The hotel divides the rooms into room types, that have a unique id, a type name, a small description and a price per occupant associated.
* The system provides a rating system, that allows each user to rate each room, with a given number of stars, a small description of the experience (optional), and a rating date.
* For each booking the hotel system provides a payment service, that has a specific id, the id of the agent or the id of the guest it payed it, the type of payment (MB, Money …), the value, and the date of payment.
* Each booking can have multiples payments, but each payment can only be relative to one booking.
* The cleaners and the agents can also be costumers of the hotel

**Class and Tables**

This part of the report will include the tables and classes we will include in our database.

Classes:

* Client
* Agent
* Cleaner
* Cleaner\_team
* Room
* Room\_type
* Booking
* Floor
* Payment
* Rating

Tables:

* Client
* Agent
* Cleaner
* Cleaner\_team
* Room
* Room\_type
* Booking
* Floor
* Payment
* Rating
* Cleaning\_status

**UML Diagram**

**Relation Model**

Person (id, firstName, lastName, adress , email, phoneNr, sex)

NOT NULL (id, firstName, lastName, adress , email, phoneNr, sex)

Cleaner (cleanerID, staffNumber, dayNightShift, salary)

NOT NULL (cleanerID, staffNumber, dayNightShift, salary)

Cleaner Team (cleanerTeamID, cleanerTeamNum, dayNightShift, floor->floor)

NOT NULL (cleaner\_team\_id, cleaner\_team\_num,day\_night\_shift,floor)

Agent (agentID, clientID->Client, agentSsn, agencyName)

NOT NULL (agentID, clientID, agentSsn, agencyName)

Client (clientID, ssn, disabled)

NOT NULL (clientID, ssn, disabled)

Rating (ratingID, description, nrStar, date)

NOT NULL (nrStars, date)

Room (roomID, roomTypeID->RoomType, roomNR, ocupationStatusID, nrBeds, floorID->Floor)

NOT NULL (roomID, roomType, roomNR, ocupationStatusID, nrBeds, floor)

RatingRoomClient(roomID->Room, ratingID->Rating, clientID->Client)

Booking (bookingID, agentID->Agent, guestID, roomID->Room, payment\_id->payment, reservationDate, fromDate, toDate, nrOcupants, price)

NOT NULL (bookingID, agentID, guestID, roomID, payment\_id, reservationDate, fromDate, toDate, nrOcupants, price)

Payment (paymentID, agentID->agent, guestID, type, value, paymentDate)

NOT NULL (paymentID, agentID, guestID, type, value, paymentDate)

RoomType (roomTypeID, roomTypeName, description, pricePerOcupant)

NOT NULL roomTypeID, roomTypeName, description, pricePerOcupant)

Floor (floorID, floorID->floor, nrRooms, disabledAdapted)

NOT NULL (floorID, floorNumber, nrRooms, disabledAdapted)