Principle of Database Management – Final Project Report

**Topic: 23 – Restaurant Management System**

**Team Members:**

Le Do Huy Du – ITITIU18027

Dang Ngoc Minh Huy – ITITIU18052

Phan Tien Tai – ITITIU18269

**About database system and application used:**

NetBeans IDE 8.0.2 for Java codes

MySQL for creating database

**About submission file:**

* 1 file report
* 2 files SQL for creating database, table, data
* 1 file java code including library folder and project folder

**About Github:** https://github.com/JBenjaminn/Hotel-Management-System.git

**INTRODUCTION:**

In this report, we will show all the step and progress of developing a Hotel Management System with SQL and Java language, from the planning to the program itself.

The Hotel Management System will help improve functionality and efficiency of works and transaction in any Hotel system.

**ANALYSIS:**

- Customers that want to make a reservation in the hotel can access to book via online website or at the hotel’s reception desk

- Customers must provide basic information for security reasons.

- a reservation must have the booking room, the temporary ID for the customer, date and time of stay, number of people and the total cost.

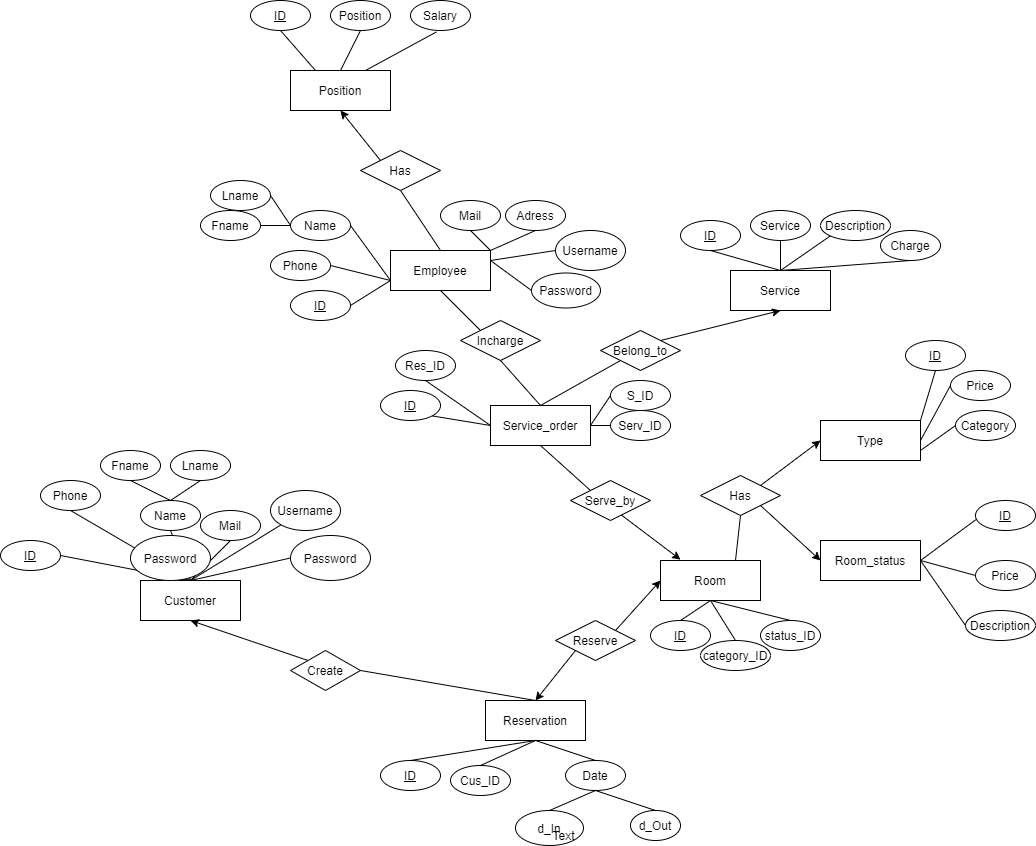
- Rooms of hotel consisted of many categories with various prices for the customers to choose.

- While staying at the hotel, customers can pay for extra services provided by the hotel facilities and employees, the cost will be added into the total cost in the reservation form on computer.

- Each services will be recorded as a service order with the service type, the employee providing the service and the customer demanding the service

- Each employee personal information and position also stored in the system for security, service management and salary payment

**ENTITY RELATIONSHIP DIAGRAM:**



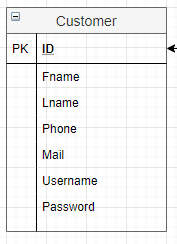
**Relational Diagram:**

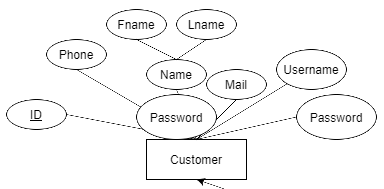
In this diagram, we can see there are total 9 strong entities:

Customer, Reservation, Room, Room\_status, Category, Service\_order, Service, Staff and Position

For these 9 entities we will have 9 tables with the same name

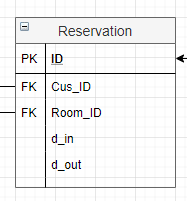
Let us convert all the entities into tables, one by one:

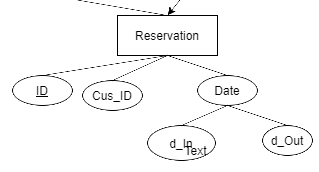
**1.Customer**



Schema: Customer (ID, Fname, Lname, Phone, Mail, Username, Password)

This table contain the needed information of the customers

**2.Reservation**



Schema: Reservatiom (ID, Cus\_ID, Room\_ID, d\_in, d\_out)

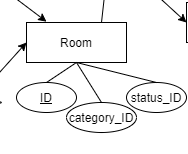
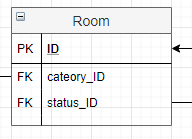
Cus\_ID is a foreign key with references to Customer (ID)

Room\_ID is a foreign key with references to Room (ID)

As a reservation must have the information of a customer and the room he/she has reserved

\*each customer can make many reservations

**3.Room**



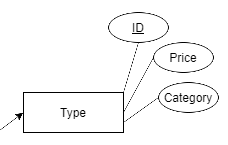
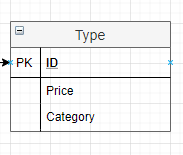
Schema: Room (ID, category\_ID, Status\_ID)

Category\_ID is a foreign key with references to Category (ID)

Status\_ID is a foreign key with references to Room\_status (ID)

As there are many rooms belong to different categories and each room has their own status

**4.Type**

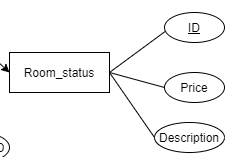
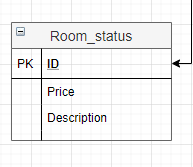


Schema: Type (ID, Price, Category)

This table contain the prices of various room categories

\*each category can have many rooms and each room can only belong to one category

**5.Room\_status**

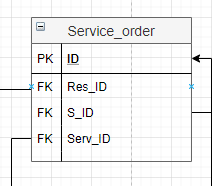


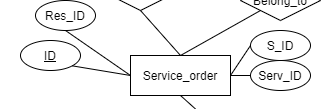
Schema: Room\_status(ID, Price, Description)

This table contain the description of room statuses and the prices that the room status can affect beside the room category

\*each room can only have one status at a time, the status will be frequently updated

**6.Service\_order**





Schema: Service\_order (ID, Res\_ID, S\_ID, Serv\_ID)

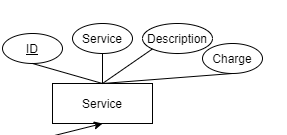
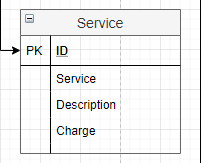
Res\_ID is a foreign key with references to Reservation (ID) which indicate the customer or room of the reservation that demanded the service

S\_ID is a foreign key with references to Staff (ID) that will specify the staff who will do the service

Serv\_ID is a foreign key with references to Service (ID) that indicate the type of service that need to be done

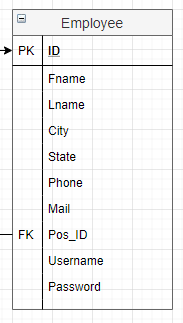
\*each service order can have one service each and one staff do the servicing

**7.Service**

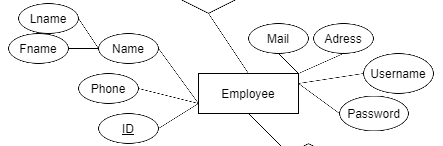


Schema: Service (ID, Service, Description, Charge)

This table contain information about types of service, their description and the charge



**8.Employee**

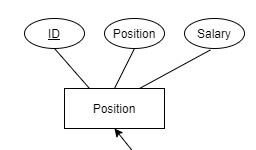
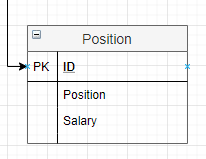


Schema: Employee (ID, Fname, Lname, City, State, Phone, Mail, Pos\_ID,Username, Password)

Pos\_ID is a foreign key with references to Position (ID) which indicate the job and position of the employee in the Hotel system

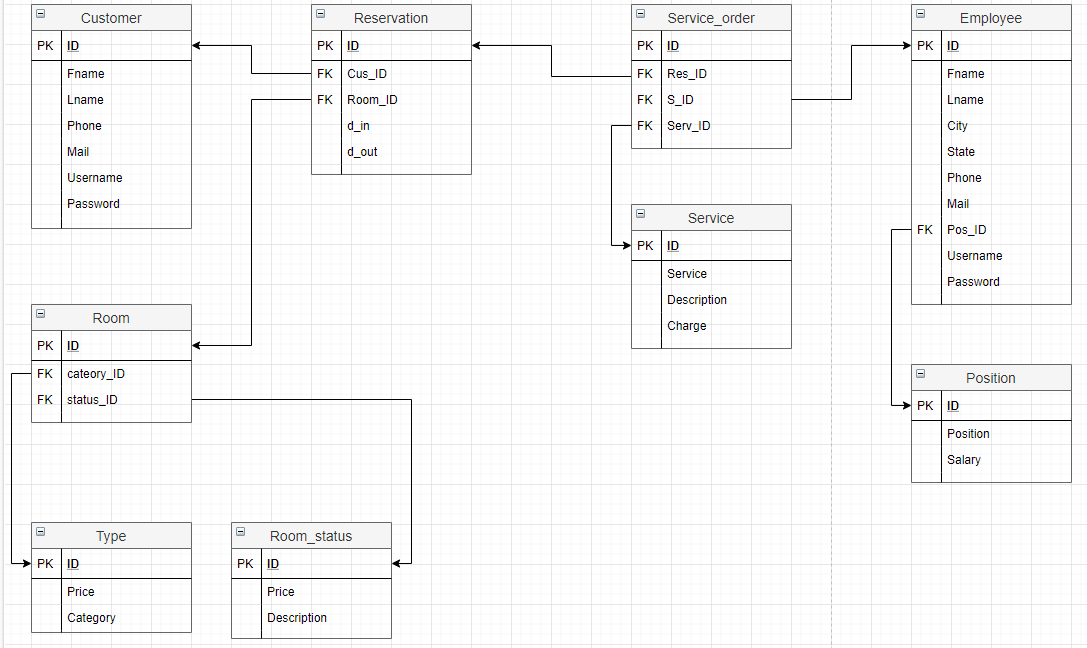
Each staff member can only have one position, this can be update

**9.Position**



Schema: Position (ID, Position, Salary)

This table contain the position and the Salary of each job

**The Relational model:**

FINAL RESULT:

We can redefine the Model above as tuples:

**Customer** (ID, Fname, Lname, Phone, Mail, Username, Password)

**Reservation** (ID, Cus\_ID, Room\_ID, d\_in, d\_out)

**Room** (ID, category\_ID, status\_ID)

**Type** (ID, Price, Category)

**Room\_status** (ID, Price, Description)

**Service**\_order (ID, Res\_ID, S\_ID, Serv\_ID)

**Service** (ID, Description, Charge)

**Staff** (ID, Fname, Lname, City, State, Phone, Mail, Pos\_ID)

**Position** (ID, Position, Salary)

**Instruction:**

**Database:**

Because our group use MySQL, so we don’t have back up file. So, we can just File -> Run SQL Script in MySQL.

Then, we run the HMS\_CreateTable.sql first, next is the HMS\_Data.sql.

After you run 2 files sql, you will get the database and tables like this:

A screenshot of a cell phone

Description automatically generated

**JAVA CODE:**

First, if you use Netbeans, maybe you can open it right away. If not, you can create with Existing code.A screenshot of a cell phone screen with text

Description automatically generated

Then you check the library folder in the project. If it doesn’t have any jar file, choose add JAR/Folder, open 4 files:

A screenshot of a computer

Description automatically generated

Next, open the ConnectDb class, you have to change the username and password to fit with your MySQL. Replace “root”,”123” by yours

A screenshot of a cell phone

Description automatically generated

Finally, you run project.

The Java codes are long so we will only name the Classes and the functionality of each class

…

…

…

And this is just of our project with some of tables to get familiar with database so we haven’t have full features like the ERD have.

We apply query in database to the project to get, insert, delete, update information from the database.

Here is some of our code:

As an admin(manager), you can manage all categories like room, reservation, employee, account, customer.

**1. Edit, Update Customer**

We do it in the ManageCustomerFrame, we use the UPDATE … SET … query to edit the information.

A screenshot of a cell phone

Description automatically generated

We will show you a demo, but just for an example because you must see the implementation in class to get the whole project.

A customer before we edited:

A screenshot of a computer

Description automatically generated

Editing:

A screenshot of a cell phone

Description automatically generated

After editing:

A screenshot of a computer

Description automatically generated

We use DELETE query to delete a record in the database.

A screenshot of a cell phone

Description automatically generated

So, you can see that the code maybe similar, but the query is different.

**2. Adding Room**

In order to adding room, we use INSERT INTO … VALUES … query

A screen shot of a computer

Description automatically generated

Similar like that, you can apply for adding room, reservation, customer, employee…

**3.checking room status**

When admin delete or add a reservation, we will need to check if the room is reserved or not.

**A screen shot of a smart phone

Description automatically generated**

**4. Manage Account**

The admin manage employee account in the Accountmanager Frame, we use a more specific query :

A screenshot of a cell phone

Description automatically generated

Next, we will see some of our project GUI:

First, we will login

A screenshot of a cell phone

Description automatically generated

If we login as a Customer, we will have some features like

A screenshot of a cell phone

Description automatically generated

Choose Booking Room:

A screenshot of a cell phone

Description automatically generated

Choose Your reservation:

A screenshot of a cell phone

Description automatically generated

If we login as a Manager, we will have some features like

MainFrame:

A screenshot of a cell phone

Description automatically generated

Choose Manage Customers:

A screenshot of a cell phone

Description automatically generated

Choose Manage Rooms:

A screenshot of a social media post

Description automatically generated

Choose Manage Reservation:

A screenshot of a computer screen

Description automatically generated

Choose Manage Employees:

A screenshot of a computer

Description automatically generated

Choose Manage Account:

A screenshot of a computer

Description automatically generated

**Features in the future:**

* Let customer book room for many times in different periods of time in a same room.
* Add service order
* Take the bill base on reservation and service order
* Improve GUI
* Try to make an app

**CONCLUSION**

Hotel management system is a system that have various properties and has a complex database if the creator doesn’t get used to it. Through this project, we learned a lot about working with database and saw how database work. Although there are many mistakes and the code aren’t clean enough, we still try to improve our skill and our knowledge about database and coding in the future to gain more achievement.