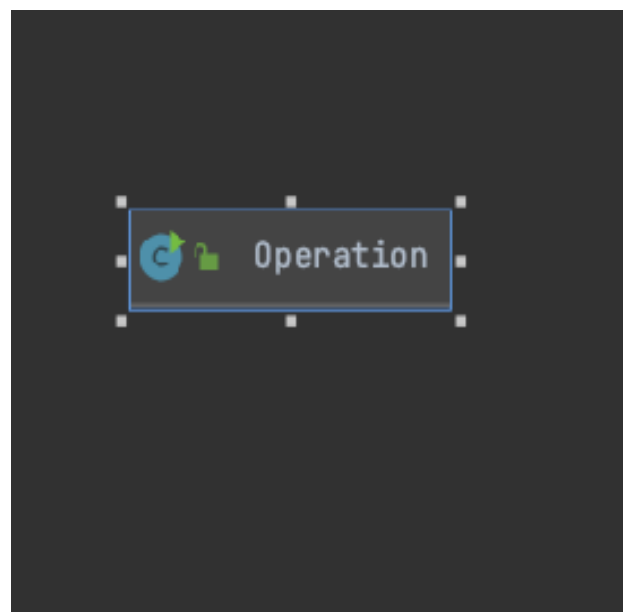
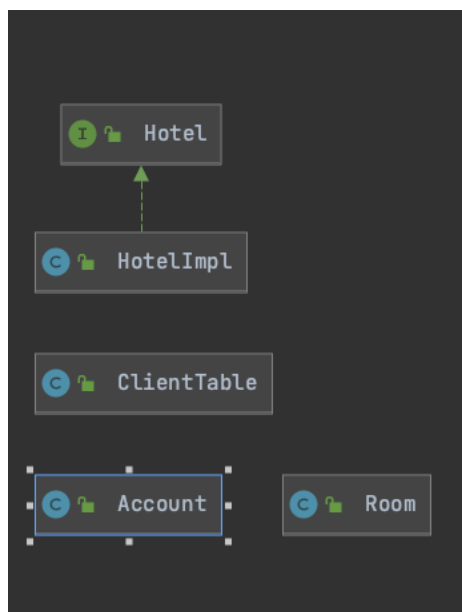
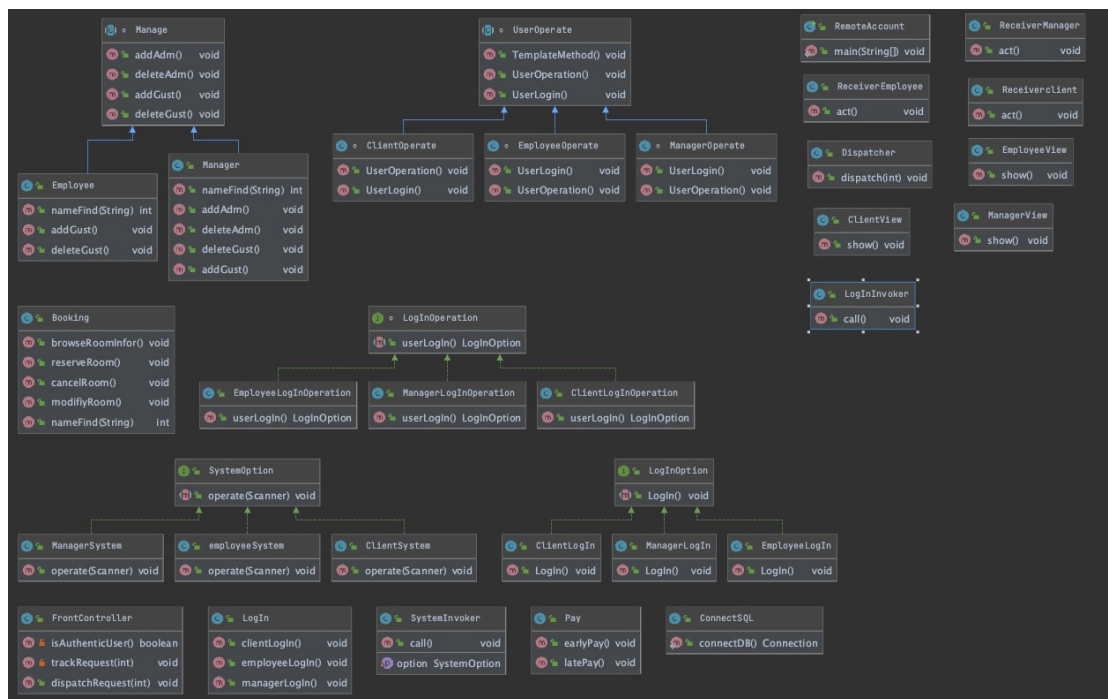


Introduction:

This project is an online hotel reservation management system, which can be used to book, withdraw and modify a room reservation order. Besides, this system allows the managers and employees to operate, and update the latest information of the rooms. In this report, I will first show the brief class diagram, and then I will show the screenshots of the example on server and on my machine respectively.

Updates: More than 10 classes, three design patterns, modification of previous design pattern, i.e., front controller pattern and authorization pattern.

Diagram:



Explanation:

1. Account.java contains the information of users' accounts and passwords.
2. Room.java includes the data of the room, such as bed's information, price.
3. ClientTable.java provide the methods to access the information of the orders.
4. Booking.java provides four functions, i.e., browse the room information, reserve a room, cancel an order and modify an order.
5. Pay.java provides two functions, i.e., pay early without additional fees and pay within 24 hours with additional fees. (This class hasn't completely finished)
6. SystemOpthin.java is an interface defines the operations based on different system operators. ClientSystem.java is for client, employeeSystem.java is for employees, and ManagerSystem.java is for the manager of the hotel.
7. Manage.java is an interface for employees and manager, which provides some methods for them to manage the system. Employee.java and Manager.java implements Manage.java to do the concrete things.
8. LogIn.java provides the log in methods, which operates the SQL, and only the registered users can use the system.
9. ConnectSQL.java is used to connect MySQL, account and password are required.
10. Hotel.sql is the scripts of the database which composed by 6 tables: client_acc(clients' account), manager_acc(manager's account), employee_account (employee's account), client_tbl (clients' basic information which will be recorded by the system), room_tbl (information about the rooms)
11. Operation.java provides the operation method for clients to use the online system.
12. ClientView.java, EmployeeView.java, ManagerView.java, FrontController.java, and Dispatcher.java are the classes corresponding to front controller patten.
13. UserOperate.java, ClientOperate.java, ManagerOperate.java and EmployeeOperate.java are the classes of template patten.
14. ClientLogIn.java, ManagerLogIn.java, EmployeeLogIn.java, LogInInvoker.java, LogInOption.java, Receiverclient.java, ReceiverManager.java, ReceiverEmployee.java are the classes of command patten.

15.LogInOperation.java, ClientLogInOperation.java, EmployeeLogInOperation.java and ManagerLogInOperation.java are the classes of factory patten.

Command Patten:

ClientLogIn.java, ManagerLogIn.java and EmployeeLogIn.java are the specific methods of login and implement the interface LogInOption.java which provide method for users to log in. Receiverclient.java, ReceiverManager.java, ReceiverEmployee.java are the receivers, and LogInInvoker.java is the invoker.

By this kind of design, different users can choose corresponding systems and log in via their accounts and passwords. The authorization is done during the process of login.

Template patten:

This patten is designed to operate the systems, there are two steps in general for users: login and operation, so I define the methods template in UserOperate.java, and the specific operates for different users based on their roles are defined in other classes: ClientOperate.java, ManagerOperate.java and EmployeeOperate.java.

Factory patten:

I use factory patten to manage the log in process. interface LogInOperation.java offers the login method for users, and ManagerLogInOperation.java, ClientLogInOperation.java, EmployeeLogInOperation.java implement interface LogInOperation.java and provide a object generation method which can call the interface LogInOption.java and return an object.

Remark: The above patterns are not totally separated, they are well coupled to work for the logging in and operation process.

Tables:

I created 5 tables below in MySQL at the beginning.

1. client_acc contains the information of clients' account

```
mysql> select * from client_acc;
+-----+-----+-----+-----+
| client_id | client_name | client_acc | client_pas |
+-----+-----+-----+-----+
|          1 | Jerry      |          1345 |          1234 |
+-----+-----+-----+-----+
```

2. client_tbl contains the information of orders

```
Database changed
mysql> select * from client_tbl;
+-----+-----+-----+-----+-----+-----+-----+
| client_id | client_name | client_address | client_credit | client_room | client_night | client_age | client_phone |
+-----+-----+-----+-----+-----+-----+-----+
| 1 | Jerry | LA | 123 | 1 | 2 | 34 | 123 |
| 2 | Tim | IN | 123 | 2 | 2 | 24 | 124 |
| 3 | Potter | NY | 1236 | 3 | 6 | 26 | 1567 |
+-----+-----+-----+-----+-----+-----+-----+
```

3. manager_acc

```
mysql> select * from manager_acc
-> ;
+-----+-----+-----+-----+
| manager_id | manager_name | manager_acc | manager_pas |
+-----+-----+-----+-----+
| 1 | Bob | 1234 | 1234 |
+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

4.employee_acc

```
mysql> select * from employee_acc;
+-----+-----+-----+-----+
| employee_id | employee_name | employee_acc | employee_pas |
+-----+-----+-----+-----+
| 1 | Marry | 1244 | 1234 |
| 2 | Lisa | 1245 | 1234 |
+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

5.room_tbl

```
mysql> select * from room_tbl;
+-----+-----+-----+-----+-----+-----+
| room_id | room_type | bed_type | bed_num | room_rate | room_smoking |
+-----+-----+-----+-----+-----+-----+
| 1 | medium | small | 1 | 150 | 0 |
| 2 | medium | small | 1 | 150 | 0 |
| 3 | small | small | 1 | 100 | 1 |
+-----+-----+-----+-----+-----+-----+
```

Workflow:

This system has mainly three subsystems, one is for clients, one is for manager, and one is for employees.

1. Choose system: At the beginning, user should choose his/her role, i.e., manager, employee, or client, before log in the system. Only the authorized users can do the next operation.
 2. Log in: Input account and password.
 3. Operation option: Choose what kind of business you want to do, if you are a client, you can choose to display the available rooms, book an room, or modify an order.
 4. Payment: Client should pay according to time and room rate.
- Besides, manager can add/delete the accounts of clients and employees.

Example:

The result of this assignment is similar to the result of the last assignment, I just modified the design pattern, and did not change the running logic of the system.

On local device:

1. At the beginning, user should choose his/her identity, and there are three systems for employee, client and manager.

```
2. java
Last login: Mon Apr 19 20:37:44 on ttys002
zhilin@zhilindeMacBook-Pro ~ % cd desktop/hw2
zhilin@zhilindeMacBook-Pro hw2 % java client/Operation
Welcome to our hotel!
Who are you? 1 client; 2 manager; 3 employee; 4 exit
1
Welcome to client system
```

2. Then log in the system via account and password:

```
-----
If you have an account already, you can log in directly
If you don't have an account, you should contact the employee to add an account
-----

Please input your account (1234)
1234
Please input your password (1234)
1234
Success
```

3. If you are a client, you can do the following operations:
Take booking a room as the example

```

Now you can do the following operations
1 book; 2 cancel; 3 modify; 4 display; 5 exit
1
Which room do you chose? 1 room 1; 2 room 2; 3 room 3; 4 exit
1
What's your name?
Jerry
Where are you from?
NY
What's your credit card number?
123
what's your id
1
How old are you?
23
What's your phone number?
123
How many nights will you stay?
3
you have successfully reserved a room
you should pay $ n

```

4. If you are a manager, you can do the following operations:
For example, you want to add a client' s account

```

Welcome to our hotel!
Who are you? 1 client; 2 manager; 3 employee; 4 exit
2
Welcome to manager system
-----
Please input your account (1236)
1236
Please input your password (1236)
1236
Success
-----
Now you can do the following operations
1 add manager's account; 2 remove manager's account; 3 add client's account 4 re
move client's account
3
You can add a client's account below
-----
client's id
1
What's the client's name?
Bob
What's the account?
1234
What's the password
1234
you have successfully add a client's account

```

5. If you are an employee, you want to add a client' s account:

```

Welcome to our hotel!
Who are you? 1 client; 2 manager; 3 employee; 4 exit
3
Welcome to employee system
-----
If you have an account already, you can log in directly
If you don't have an account, you should contact the employee to add an account
-----
Please input your account (1235)
1235
Please input your password (1235)
1235
Success
-----
Now you can do the following operations
1 add client's account; 2 delete client's account
1
You can add a client's account below
-----
client's id
1
What's the client's name?
Lisa
What's the account?
1234
What's the password
1234
you have successfully add a client's account

```

Operations on the server:

First, In in-csci-rrpc01.cs.iupui.edu, I run java hw2.hotel.RemoteAccount to prepare the hotel server

```

ERROR
zhilin@ubuntu:~/Desktop$ ssh wangzhil@tesla.cs.iupui.edu
wangzhil@tesla.cs.iupui.edu's password:
Duo two-factor login for wangzhil

Enter a passcode or select one of the following options:

1. Duo Push to +XX XXX XXXX 4575

Passcode or option (1-1): 1

Pushed a login request to your device...
Success. Logging you in...
[wangzhil@tesla ~]$ ssh in-csci-rrpc01.cs.iupui.edu
wangzhil@in-csci-rrpc01.cs.iupui.edu's password:
Last login: Mon Apr 12 08:22:14 2021 from tesla
[wangzhil@in-csci-rrpc01 ~]$ java hw2.hotel.RemoteAccount
Server is ready

```

Then I run java hw2.client.Operation in client server : in-csci-rrpc02.cs.iupui.edu

Since I use SQL in my code, there is a SQL connection problem when I run the code on Tesla, and I am not familiar with the SQL via java, so maybe in next assignment I will not use SQL.

Client:

```

Who are you? 1 client; 2 manager; 3 employee; 4 exit
1
Welcome to client system
-----
If you have an account already, you can log in directly
If you don't have an account, you should contact the employee to add an account
-----
-----
Please input your account
1345
Please input your password
1234
java.sql.SQLException: No suitable driver found for jdbc:mysql://localhost:3306/
hotel
Exception:java.lang.NullPointerException
[wangzhil@in-csci-rrpc02 ~]$ ls
example hw2 perl5 Python-3.5.2 Python-3.5.2.tgz shareTest test zhilin
[wangzhil@in-csci-rrpc02 ~]$

```

Manager:

```

Who are you? 1 client; 2 manager; 3 employee; 4 exit
2
Welcome to manager system
-----
Please input your account
1234
Please input your password
123
java.sql.SQLException: No suitable driver found for jdbc:mysql://localhost:3306/
hotel
Exception:java.lang.NullPointerException
[wangzhil@in-csci-rrpc02 ~]$

```

Employee:

```

Who are you? 1 client; 2 manager; 3 employee; 4 exit
3
Welcome to employee system
-----
If you have an account already, you can log in directly
If you don't have an account, you should contact the employee to add an account
-----
-----
Please input your account
1244
Please input your password
1244
java.sql.SQLException: No suitable driver found for jdbc:mysql://localhost:3306/
hotel
Exception:java.lang.NullPointerException
[wangzhil@in-csci-rrpc02 ~]$

```