# TopView 2019考核

#### 考核题目：酒店管理系统

**通过JavaSE，在本地线下的环境下开发设计一个酒店管理系统。在该系统中能够实现酒店信息以及酒店预定信息的管理功能，系统面向的用户是酒店管理员以及顾客。要求在该系统中实现对顾客预定信息的基本管理以及对酒店信息的基本管理功能**

**数据设计参考（可以自由发挥）**

**提升部分的数据设计不给予提供**

## 项目基本需求

#### 用户（User）：

主键（id），用户名，密码

#### 顾客（Customer)：

主键（id），身份证号，联系电话，用户id

#### 酒店管理员(Hotel Administrator)：

主键（id），身份证号，联系电话，用户id

#### 房间(Room)：

主键（id），房间类型，占地面积，床宽等房间信息，价格

* **用户可以在系统中注册账号**
* **用户可以通过账号和密码登录系统**
* **酒店管理员登录进入管理系统后，可以管理酒店的房间信息，包括增加房间信息，删除房间信息，修改房间信息，对酒店信息进行查询**
* **顾客登录进入管理系统后，可以查看酒店的房间信息，然后预约**

## 项目进阶需求

#### 酒店(Hotel)：

主键（id），星级，评分，酒店类型

#### 订单(Order)：

主键（id），日期，住宿时段，房间类型，酒店id，房间id，顾客id

#### 超级管理员(Administrator)

主键（id），用户名，密码，联系电话，身份证号

* **使用JavaFX或Swing来进行图形化界面的展示**
* **记住登录状态，第二次不用打密码或者不用登陆直接进入软件**
* **用户可以管理自己的用户信息，如：修改密码**
* **顾客可以选择自己预约时间，时间不能冲突**
* **添加订单功能：顾客可以查看自己的预约订单，顾客可以申请取消预约等；酒店管理员可以管理订单信息，如：查看订单、取消订单等**
* **超级管理员可以对酒店进行增删查改**

## 项目升华需求

#### 评价(Comment)：

主键（id），评价内容，评价分数，订单id

#### 服务(Service)

主键（id），服务内容，服务房间，服务时间段，价格，酒店id

* **顾客可以查看酒店的评价以及评分，并且可以发表对酒店的评价以及评分**
* **酒店管理员可以查看顾客的评价以及评分（ps：可考虑添加回复评论的功能）**
* **通过条件筛选酒店（如酒店名，价钱）**
* **顾客可以通过系统查看酒店服务并可申请酒店服务，酒店管理员可以在系统中上架可提供的酒店服务并进行管理**

## 超级无敌螺旋升天需求

#### 账单(Account)：

主键（id）,酒店id，顾客id，日期，金额，房间id

#### 日志（Log）:

主键（id）,日志等级，日志内容，记录时间

* **在前面所提及到的需求之上添加交易系统，如：付款，退款等(账单中记录了资金进出的信息)**
* **日志记录敏感操作，并且高级权限可以查看日志记录**
* **定时删除30天以前的日志**
* **添加VIP功能：实现VIP顾客折扣优惠等**
* **记录酒店账单情况，使酒店管理员可以管理账目信息(ps:可以添加图表展示形式)**

#### 以上拓展功能可选择性完成，且项目不仅限于此，有任何新奇的想法都可以加入到项目中，完成可加分

## 要求技术

#### Java基本语法->面向对象->常用类->异常处理->容器io->Jdbc->MySQL数据库（安卓可以不用Jdbc和数据库）

#### 必须采用Git进行项目版本管理

#### 不能使用任何框架

## 工具环境选择

* **编译工具：IDEA 或者 Eclipse**
* **数据库版本要求：5.7.17以上**
* **数据库图形化工具：SQLyog 或者Navicat**
* **JDK的开发版本应当为1.8**

## 评分标准

* **代码质量高者优先考虑**
* **实现功能多者优先考虑**

## 考核提交

提交内容：源代码，数据库的SQL文件（包括库和表的结构以及数据），项目介绍文档（内容包括测试数据以及测试情况的截图，还可以）

提交方式：强制采用Git进行提交，具体方式留意考核群内通知。

考核过程中只能在master分支push 3次代码，前2次会有反馈，最后一次没有，多交的按照第3次提交的而为准。每次提交需通知对应方向师兄师姐。

**自己的代码可以提交到dev分支，需要发布正式版时再合并到master分支**

Git 提交的粒度要足够小，比如每完成一个功能或修复一个bug尽量都进行提交

|  |
| --- |
| 时间 |
| 3月24日 开始 |
| 4月26日 23点 截止提交 |

**具体 Git 提交规范**

***<type>(<scope>): <subject>***

**type包括：**

feat：新功能（feature）

fix：修补bug

docs：文档（documentation）

style： 格式（不影响代码运行的变动）

refactor：重构（即不是新增功能，也不是修改bug的代码变动）

**scope指影响范围，具体到包即可**

**subject指具体修改内容**

**示例：**



其中， detail应为同学们项目中对应的包名

## 分包

* **com.yourname.www.po（与数据库表一一对应的实体类）**
* **com.yourname.www.util（工具类，与业务逻辑无关，通常为各个项目都可以通用的代码）**
* **com.yourname.www.dao（放置数据库或文件读写相关的代码）**
* **com.yourname.www.service（放置处理业务逻辑的代码，隔离dao层与view层）**
* **com.yourname.www.view（放置与页面相关的代码，图形化界面相关代码，或者文字界面相关代码）**

## 代码规范（很重要）

* **使用驼峰命名法（不懂的话可以百度理解一下），变量名和方法名中不要出现下划线。**
  + 正确例子：userName，getUserName
  + 错误的例子：username*，*getuser\_name
* **不要出现没有意义的变量命名，不要图方便（a ，b，c），达到见名知义的效果，例如需要定义一个用户名字。**
  + 正确例子：String userName
  + 错误的例子：String a
* **代码缩进。**
* **尽量不要写重复代码，要运用封装的思想。**
* **尽量不要在代码中直接使用常量，常量应该放到一个常量类里面。**
* **代码是死的，尽量把它写活。**
* **项目中只能导入jdbc的jar包**
* **项目提交到Git上的私有仓库，将师兄师姐拉入私有仓库中**
* **独立完成项目，不能直接抄别人的代码**
* **其他详情请见：《TopView编码规范》（见群文件）**