

# QG后台组最终考核

## 前言

## 时间安排

时间	安排
2023-04-14	发布最终考核题目
2023-04-23	中期进度检查、答疑
2023-04-30	考核项目提交
2023-05-04	最终考核预答辩

## 评分细则

1. 功能完成程度
2. 拓展与创新
3. 代码质量
4. 代码风格
5. 项目报告文档
6. 项目答辩汇报

## 提交内容

1. **源代码**（注意包含数据库SQL脚本）
2. **项目打包**（war文件，如果修改了Tomcat配置请一并附上，并进行说明）
3. **项目报告文档**
4. **项目汇报PPT**
5. 建议附加**进度文档**，写明开始的计划与过程中某个时间点解决了什么问题

**注意：请重视项目报告文档和项目汇报PPT这两部分内容！**

## 备注

1. 考核着重于后台部分，不要求前端做的精美。
2. 建议先做好需求分析与规划，再开始写代码。
3. 不建议在有限的时间内实现需求外的功能，在需求框架内不断精进更有意义。当然有意义有价值的扩展也会酌情加分

## 打印机

现在有一个企业，其中有非常多的3D打印机，它们会源源不断的产生**设备信息**并写入**txt**文件中，为了便于**管理与实时监控**设备的运行状态，现需要设计一个系统，读取txt文件中的消息，并实时展示给前端，以供管理人员时刻观察设备状态，并能提供一个**实时预警**效果。

[打印机](#)

## 需求

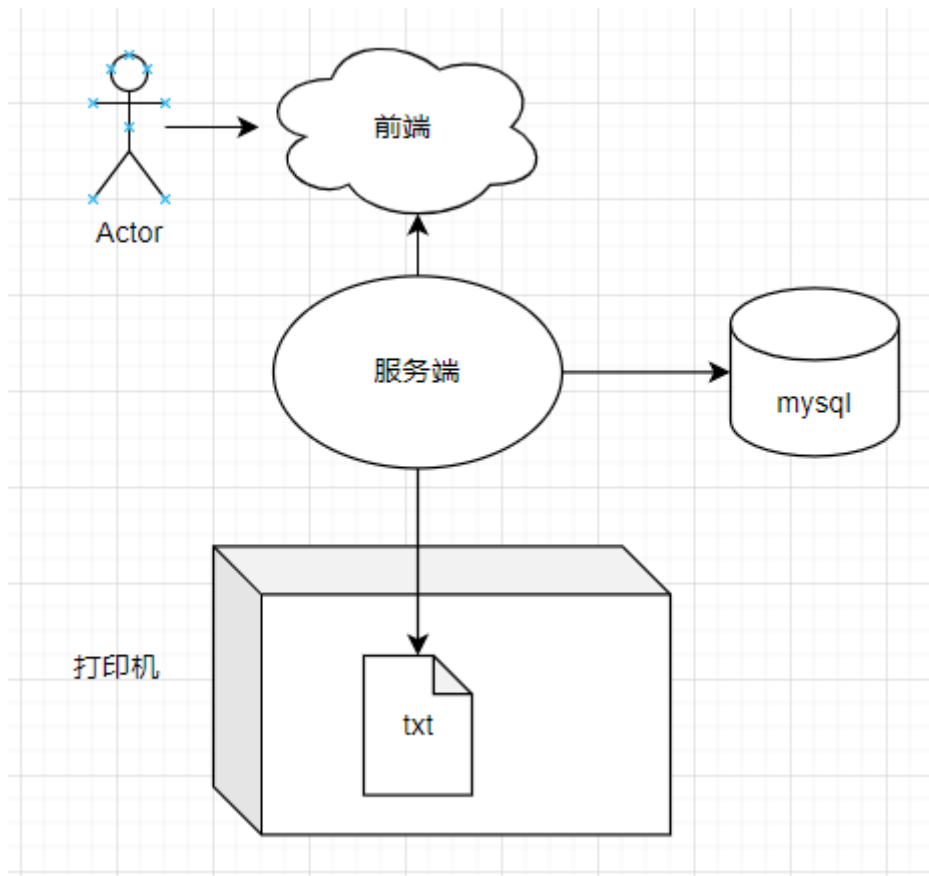
### 基本需求

#### 用户模块

1. 用户分为**管理员、企业用户、普通用户**三个角色，登录、注册、修改查看用户信息是基本的功能。
2. 管理员可以查看所有的用户和打印机情况；企业用户可以看到其下所有的普通用户和打印机情况，普通用户可以绑定父账户（即企业用户），然后只能查看企业其下的打印机情况。

#### 文件模块

该3D打印机的特点是，它会源源不断的生成设备的当前信息，并写入txt文件中（文件路径位于C:\TPM\printer.txt），你需要做到的就是**监听**文件的变动，并读出最新的消息，对数据进行解析处理后入库并展示于前端页面。



txt文件中，每一行的数据都是一条最新的消息，格式为：**t: s (: p...)**；

```
1677979705:1
1677979706:2
1677979707:80:8:TEST.ejw
1677979708:40:1677979708
1677979709:41:529
1677979710:41:528
1677979711:41:509
1677979712:41:499
1677979713:41:490
1677979714:41:247
1677979715:41:213
1677979716:41:179
1677979717:41:147
1677979718:41:114
```

t为机器产生的消息的时间戳，返回的是1970年1月1日至当前的秒数。s是表示机器工作作态改变或一些监控的值发生改变或者发生异常值，如果有额外参数，依次用p1、p2、p3。。。。。来表示。

#### **s的定义如下：**

当s为1，作业未开始，无额外参数，读到这个值可视为用户打开了打印机硬件控制系统并成功登陆。

当s为2，正在作业，无额外参数，表示用户点击了开始作业按钮，从而进入正在作业的状态（预热，s为40表示真正开始打印）

当s为3，作业由用户暂停，无额外参数，

当s为4，作业被动暂停，p1是32位的无符号整形，表示被动暂停的原因，

p1为1表示“当前的作业零件数为零，作业已暂停，请先加件！”

p1为2表示“刮板复位发生错误，已暂停作业！”

p1为3~18为“氮气压力异常”

当s为5，作业由用户停止，无额外参数，

当s为6，作业被动停止，p1为32位的无符号整形，代表被动停止的原因，

p1为1表示“工作台或成型桶温度过高”

p1为2表示“急停开关压下”

p1为3~4为“氮气压力异常”

当s为7，作业正常结束，无额外参数

当s为21，表示工作台的当前温度，p1为单精度浮点数类型

当s为40，表示作业进入打印阶段，p1为无符号整数类型，表示进入打印的时间，可以利用该时间，计算打印作业开始了多长时间。（这个时间减去当前时间，动态更新前端数据）

当s为41，表示正在进行的作业距离完成还需要的秒数，p1为无符号整数类型

当s为46，用户正常关闭控制软件，无额外参数

当s为80，输出作业名称字符串，两个额外参数，p1是32位无符号整形，表示作业名称字符串的长度（字节数），p2为作业名称字符串（GBK编码）

## **打印机模块**

## 1. 前端需要展示的数据有

- 打印机的**当前状态**（s1~s7、s46）和异常或者闲置原因
- 任务进度（利用s为41和打印持续的时间进行计算）
- 当前温度信息
- 用户信息
- 打印机信息

## 2. 数据统计

- **今日累积开机时间**：当接收s1开始统计，直到接收到s46（可能实际情况中，不会有关机的状态）
- **今日累积打印时间**：当接收s2开始统计，闲置、异常时暂停统计
- **今日累积闲置时间**：当接收到s3、s4、s5、s7开始统计，打印、异常时暂停统计
- **今日累积异常时间**：当接收s6开始统计，打印、闲置时暂停统计

**注意**：统计字段都是今日累积，表明需要**跨天清零**！

3. 对于统计的时间数据，和温度数据，以及操作信息，需要**存入数据库**中，便于我们复原用户操作和打印机状态信息。
4. 温度统计，可以自定义时间范围查看某个温度的**温度曲线**。

## 进阶需求

1. 使用**websorcket**，对接受的打印机消息，实时地展示在前端页面中。
2. 对于一些状态信息，可以自定义临界值，当达到某个临界值，需要进行一个**实时预警**的处理，如邮箱、短信、网页中的实时预警通知。
3. 在实际生产过程中，我们发现，我们把太多的数据处理放在了打印机客户端中，影响了它一定的性能，所以请你考虑把架构从**单体架构**（即数据接收和处理全部放在打印机客户端上）拆分为**web服务端（处理数据）和驻留端（接收数据）**的架构，驻留端接收消息后，只需要把消息发送到web服务端即可，并考虑同时存在多台打印机的情况下，如何去存储每台打印机的数据，包括**软件层面**的设计，**数据库表**的设计等（一台机器每天产生的数据量为几百万条，需要考虑一次性几十台机器的情况）。



## 高级需求

1. 请你使用**本地缓存**进行数据的统计和存储，以减轻数据库的压力。
2. 正常情况下，web服务端每秒接受的消息量是稳定的，但是在某些场景下，会出现波动的情况（例如驻留端停机后的同步恢复操作，这时候，设备的txt文件中将会存在大量未读取的数据），考虑使用**队列或者一些限流算法**对web服务端接受驻留端消息部分和数据库操作部分（考虑是否存在一些数据不需要每次接受消息后都进行入库）进行优化。
3. 对一些非绝对重要的信息的数据库入库部分使用**异步**操作。

# QG购物网

该项目为开发一个购物页面，以满足多用户的购物需求，提供安全、高效、方便的购物体验。

## 背景

- 线上购物成为越来越多人的购物方式选择，一个安全、高效、便捷的购物网站能够更好的满足用户的需求
- 在购物页面上首先会有许多商家的商品展示，用户可以通过浏览、搜索的方式发现自己心爱的商品，进入商品浏览界面了解商品信息，还可以联系客服进行咨询

# 需求

---

## 基本需求

### 系统模块

#### 1. 店铺

1. 普通用户可以申请开店铺
2. 店铺拥有自己的简介以及相关信息，如拥有粉丝数量，月均销量等信息
3. 店铺中拥有多个商品信息
4. 店铺可以发布最新商品，发布动态以获取更高关注度
5. 店铺拥有聊天室，用户可以上来进行咨询

#### 2. 商品

1. 商品拥有自己的文字介绍，图片展示以及月销量等信息
2. 每个商品拥有独自的评论信息
3. 可以通过商品进入到该店铺中，查看更多商品信息
4. 当商品被成功收货后，该商品的月销量也会同步更新

#### 3. 购物车

1. 用户对心仪商品进行加入购物车的操作
2. 购物车中的商品可以进行批量删除，批量购买等操作

#### 4. 用户中心

1. 用户设置查看、修改个人信息，包括用户名、头像、收货地址、手机号等个人信息
2. 可以查看的已购买商品的状态（未发货、已发货、已收货、售后）
3. 可以对已购商品进行退货操作

### 用户模块

#### 1. 普通用户

- 账号注册（录入用户名、昵称、密码、地址等其他基本信息，可自行扩展）
- 实现登录、退出登录功能（未登录则为游客权限）
- 支持查看、修改个人信息功能（昵称、密码等，每个用户注册后都有一个默认头像，用户可以自行修改头像）
- 进行商品浏览、搜索，查看商品的基本信息以及其他用户的评论信息，对心爱的商品加入购物车、购买。
- 对商品进行咨询以及评论。
- 对店铺进行关注，当店铺发布动态后有消息提醒

#### 2. 游侠：

- 未注册账号的用户为游侠，游侠仅可浏览商品，查看商品评论

#### 3. 店铺管理员

- 店员是普通用户注册，向平台申请注册店铺
- 注册店铺成功后，上传自己店铺中的商品（文字、图片展示，所剩库存、月销量等信息）
- 聊天室中，回复用户的咨询

#### 4. 网站管理员

- 监管店铺，审核申请（包括新店铺的注册，以及店铺发布的商品）
- 对用户的举报信息进行审核，对不良商品进行下架，进行下架操作后提醒店铺进行改进。
- 可以对不良评论进行删除

### 前端

1. 前端页面展示，与后台有一个良好的交互

## 加分需求

### 用户模块

- 支持**验证码**登录，对信息敏感数据进行加密传输
- 数据库的**敏感数据**不可明文存储
- 支持忘记密码，找回密码
- 支持用户私聊，可以关注其他用户
- 支持关注店铺，当店铺发布新的商品时进行**产品推送**提示
- 管理员下架商品时，提醒店铺。删除评论后提醒该用户

### 系统模块

- 前端数据实时展示
- 考虑到**高并发**，**资源抢占问题**，保证服务器**安全**
- 禁止刷单的行为

## 高级需求

- 使用本地缓存进行数据的统计和存储，并考虑如何更高效的存储进数据库
- **对数据库进行保护**，有抵抗恶意攻击的能力
- 前后端数据传输使用**加密传输**。
- 可支持高并发的访问，并且保证平台的**稳定性**

# QG-教务系统

## 需求

### 基本需求

#### 系统模块

- 结构，对象包含学院、专业、年级、班级、学生、教师、学期、课程、周次、节次、教学场地等
- 培养计划
  - 不强制要求区分专业课、基础课、必修课、选修课等概念，仅考虑必修和选修即可
  - 培养计划是实际课程的**抽象**
- 考试安排
- 成绩、绩点、学分，并且学期末要计算总绩点，折算方式不要求与实际相同，但要**与学分相关**
- 选课
  - **区分公选课和体育课**，两者性质不同，体育课是必修的，公选课是选修的
  - 要能够定时开放/关闭抢课
  - **分两种模式**，一种是**不限抢课人数**，截至时随机分配(类似抽签)；一种是**先到先得抢课**
  - 退选
- **学生毕业**（需满足必修及学分要求），毕业后保留数据、档案，但不再参与排课等活动

### 学生端

- 学号登录

- 学籍卡片：查看学籍信息，不要求过于详细
- 课表查询：按班级显示、按个人显示（包含选修）
- 查询成绩：分课程成绩、考级成绩和体测成绩
- 选课：公选课、体育课
- 修改密码
- 教师评价
- 考试安排

## 教师端（大家不熟悉，简单设计即可）

- 查看个人信息及上课安排
- 录入学生课程分数

## 管理端

- 排课
  - 要求**场地，时间不能重复**
  - **排课是按照培养计划自动安排？还是手动排课？**
- 录入课程分数
- 查看学生信息，成绩等，需要筛选与排名功能
- 注册新生/教师入职

## 进阶需求（上述标蓝同为进阶需求）

- 验证码
- 支持忘记密码找回
- 数据库**密码加密**
- 抢课涉及**多线程**，**数据库事务**和**隔离级别**等，要求解决“多人抢最后一名额”的问题并保证效率
- 使用**java.util.logging**包记录日志，**操作记录**等

## 通用加分需求

- 手机短信验证码
- 安全性
  - 后端接口要求权限校验，不可随意响应
  - 前后端都做好**正则表达式**校验
  - 防止**SQL注入**，**XSS攻击**
- 扩展性：代码结构，数据库结构清晰，方便扩展功能
- 健壮性
  - 无严重bug
  - 使用JUnit进行单元测试
  - 异常处理

# 技术要求及规范

## 基本技术要求

1. JDK 版本推荐为 1.8，也可以使用 jdk11、jdk15 或 jdk17
2. MySQL 推荐为 5.7 版本，也可以使用 8.0
3. Java、Mysql、Javaweb、Git

4. 可以使用前端框架 JQuery、Ajax、Bootstrap、Layui 等开发前端
5. 可以使用Maven，但是只能引入如下的依赖
  - Mysql连接驱动
  - Servlet
  - JSON序列化工具
  - Junit
  - logging等日志框架
  - 如果有其他需要使用的maven依赖，请在群里问一下师兄。

**注意：不允许使用一些第三方的框架！**

## 进阶技术要求

---

1. 使用**JAVA8**的新特性优化你的代码，如时间类（LocalDateTime）、Stream、Optional等
2. 使用**Apifox**、**postman**、**eoLink**、**swagger**等工具（任选其一）记录接口文档（这是开发的良好习惯，也方便测试接口）
3. 使用小组作业要求的**数据库操作工具类，CRUD操作工具类**
4. 使用小组作业选做要求的**数据库连接池**
5. 使用**本地缓存**，优化你的代码，减轻数据库压力
6. 不要使用**联表查询**，并了解为什么
7. 使用**常量类和配置文件**
8. 实现**全局异常处理**
9. 对多角色多权限使用**RBAC模型和JWT进行认证鉴权**
10. 拥有良好的**日志记录**和**注释习惯**
11. 需要有足够的健壮性和防御，前后端使用正则表达式校验，**抵御Sql注入、Xss攻击，并对用户数据进行脱敏**
12. 学习像Spring一样实现**依赖注入(DI:Dependency Injection)**、进行**控制反转(IOC:Inverse Of Control)**、**面向切面编程(AOP:Aspect Oriented Programming)**
13. 使用**自定义注解**，对12点的要求进行优化

## 规范

---

1. 前后端通信使用**json格式**
2. 使用小组作业要求**BaseServlet**
3. 使用小组作业要求的**MVC分包**
4. 使用小组作业要求的**统一结果集**
5. Git**提交粒度**尽可能小，一个功能至少提交一次，修复一个bug也提交一次。
6. Git新建分支进行开发，master只能用于发布正式版本