项目简介及项目重现文档

标题页

目录页

# 简介

题目、用了什么工具、软件（包括版本号）、数据规模、什么语言、框架、技术，完成了什么功能，进行了哪几个方面的实验设计，达到了什么目的。

## 项目简介

题目：博影娱乐票务平台

开发工具

后端：IntelliJ IDEA 版本： 2020.1.3

前端：IntelliJ IDEA 版本： 2020.1.3

数据库：Oracle 12c/Timesten 11.2.2

数据库管理工具：Navicat

|  |  |  |
| --- | --- | --- |
| **工具** | **说明** | **官网** |
| IntelliJ IDEA | java开发工具、前端开发工具 | <https://www.jetbrains.com/idea/> |
| Navicat | 数据库设计工具 | <http://www.formysql.com/xiazai.html> |
| Typora | Markdown编辑器 | <https://typora.io/> |
| ProcessOn | 流程图绘制工具 | <https://www.processon.com/> |
| XFTP 6 | 上传文件至阿里云服务器 | <https://www.netsarang.com/en/xftp/> |
| Oracle | 关系型数据库Oracle | https://www.oracle.com/cn/downloads/ |
| Timesten | Oracle内存数据库 | https://www.oracle.com/cn/database/technology/timesten-overview.html |

数据规模

用户（条目数量：200000）

管理员（条目数量：1000）

订单（条目数量：2000000）

票（条目数量：11500000）

等级座位（条目数量：120000）

演出（条目数量：40000）

数据库实验环境

开发语言:

前端：HTML、CSS、Javascript

后端：Java

数据库：PL/SQL

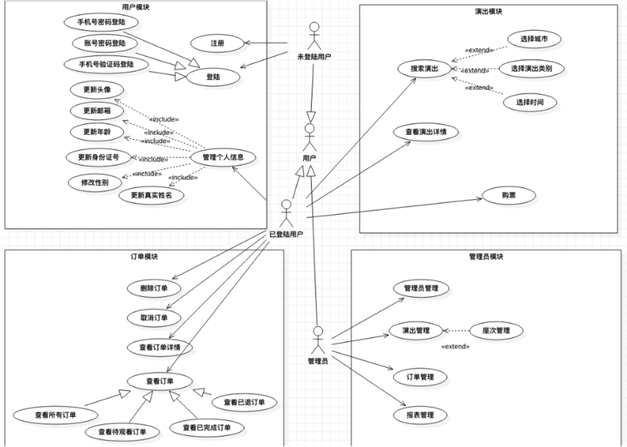
框架：

前端：Vue-CLI（Router、Axios、Element-UI）

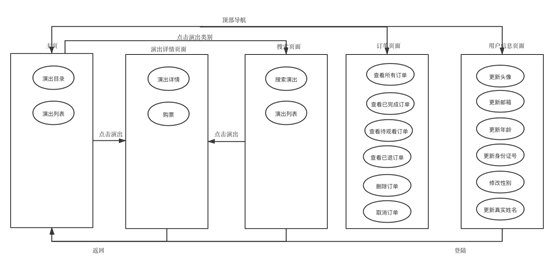
后端：SpringBoot(Mybatis、SpringSecurity、JWT、Swagger-UI)

概述

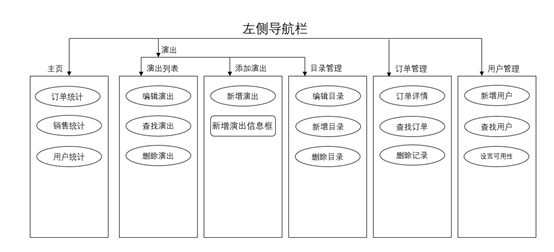
## 需求分析/主要功能



用例图



前台演出购票模块导航



后台管理模块导航

### 1.2.1 用户项目：

#### 用户模块

·注册：用户可以选择注册一个个人账号。

·登陆：用户可以用以下方式进行登陆，手机号验证码、手机号密码、账号密码。

·管理个人信息：已登陆用户可以修改如下个人信息，头像、邮箱、年龄、身份证号、性别、真实姓名。

·更新头像：登陆之后，用户进入我的信息界面可以修改头像。

·更新邮箱：登陆之后，用户进入我的信息界面可以修改邮箱。

·更新年龄：登陆之后，用户进入我的信息界面可以修改年龄。

·更新身份证号：登陆之后，用户进入我的信息界面可以修改身份证号。

·修改性别：登陆之后，用户进入我的信息界面可以修改性别。

·更新真实姓名：登陆之后，用户进入我的信息界面可以修改真实姓名。

#### 演出模块

·搜索演出：用户可以通过关键词、时间、城市、目录中的一个或多个条件搜索对应的演出进行浏览。

·查看演出详情：用户可以在演出页面找到自己感兴趣的演出，并点击进入演出详情页面，浏览演出全部内容。

·购票：已登陆用户可以选择某个演出进行相应购买。

#### 订单模块

·查看订单：用户可以查看以下类型的订单：查看所有订单、查看待观看订单、查看已完成订单、查看已退订单。

·查看订单详情：用户可以在订单页面找到想要查看的订单，并点击进入订单详情页面，浏览订单相关的详细内容。

·删除订单：用户可以删除已退订单。

·取消订单：用户可以取消待观看订单。

## 实验方案设计

# 系统设计与实现

## 项目特色

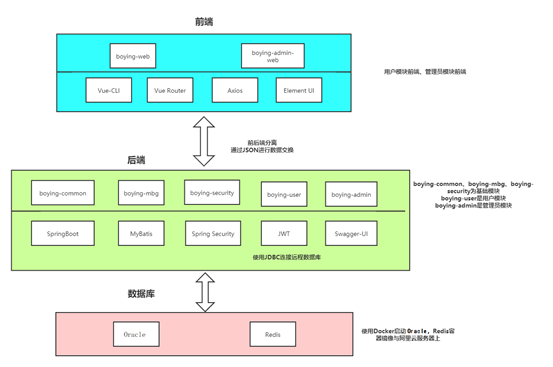
本项目是一款网上演出购票平台，用户可以登录后查看上线的各大演出，并下单购买自己喜欢的演出门票，也可以管理自己已经购买的票务详情。

## 2.2 系统总体架构设计

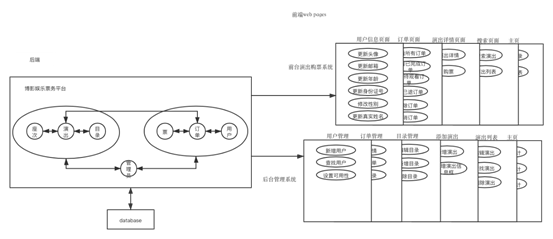
本项目是前后端分离项目，通过JSON进行交互。前端主要基于Vue开发，后端基于Springboot开发。后端连接远程Oracle数据库。

前端分为boying-web、boying-admin-web两个模块，即用户前端模块、管理员前端模块

后端分为五个模块,boying-common、boying-mbg、boying-security为其他两个模块提供基础的支持，包括权限认证、通用返回类型、全局日志等，boying-user、boying-admin即为后端的用户模块、管理员模块



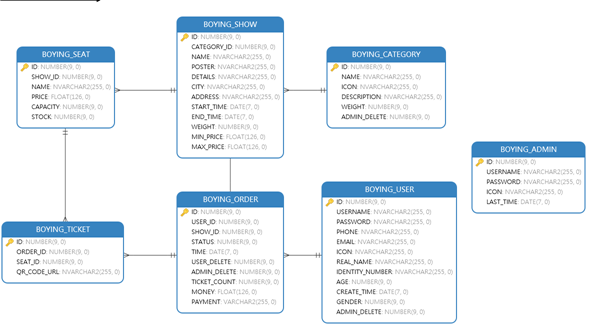
系统总体架构设计



系统软件结构设计图

|  |  |  |
| --- | --- | --- |
| **技术** | **版本** | **说明** |
| JDK | 1.8 | Java开发工具包JDK |
| Spring Boot | 2.3.0 | 容器+MVC框架 |
| Spring Security | 5.1.4 | 认证和授权框架 |
| MyBatis | 3.4.6 | ORM框架 |
| MyBatisGenerator | 1.3.3 | 数据层代码生成 |
| PageHelper | 5.1.8 | MyBatis物理分页插件 |
| Swagger-UI | 2.9.2 | 文档生产工具 |
| Redis | 5.0 | 分布式缓存 |
| Docker | 18.09.0 | 应用容器引擎 |
| Druid | 1.1.10 | 数据库连接池 |
| JWT | 0.9.0 | JWT登录支持 |
| Lombok | 1.18.6 | 简化对象封装工具 |
| Oracle | 12c | 关系型数据库Oracle |
| Timesten | 11.2.2 | Oracle内存数据库 |

## 2.3 数据库设计



**BOYING\_USER**

|  |  |  |  |
| --- | --- | --- | --- |
| **备注** | **字段名** | **数据类型** | **字段说明** |
| **PK** | ID | NUMBER(9) NOT NULL |  |
|  | USERNAME | NVARCHAR2(255) NOT NULL | 用户名 |
|  | PASSWORD | NVARCHAR2(255) NOT NULL | 密码 |
|  | PHONE | NVARCHAR2(255) NOT NULL | 联系电话 |
|  | EMAIL | NVARCHAR2(255) NOT NULL | 邮箱 |
|  | ICON | NVARCHAR2(255) NOT NULL | 个人头像 |
|  | REAL\_NAME | NVARCHAR2(255) NOT NULL | 用户真实姓名 |
|  | IDENTITY\_NUMBER | NVARCHAR2(255) NOT NULL | 用户身份证号 |
|  | AGE | NUMBER(9) NOT NULL | 年龄 |
|  | CREATE\_TIME | DATE NOT NULL | 创建时间 |
|  | GENDER | NUMBER(9) DEFAULT 0 NOT NULL | 用户性别 1->男,0->女 |
|  | ADMIN\_DELETE | NUMBER(9) DEFAULT 0 NOT NULL | 管理员是否删除了该用户,默认为0，即未删除状态 |

**BOYING\_ABOYING\_ADMIN**

|  |  |  |  |
| --- | --- | --- | --- |
| **备注** | **字段名** | **数据类型** | **字段说明** |
| **PK** | ID | NUMBER(9) NOT NULL |  |
|  | USERNAME | NVARCHAR2(255) NOT NULL | 用户名 |
|  | PASSWORD | NVARCHAR2(255) NOT NULL | 密码 |
|  | ICON | NVARCHAR2(255) NOT NULL | 头像 |
|  | LAST\_TIME | DATE NOT NULL | 最后登录时间 |

**BOYING\_CATEBOYING\_CATEGORYGORY**

|  |  |  |  |
| --- | --- | --- | --- |
| **备注** | **字段名** | **数据类型** | **字段说明** |
| **PK** | ID | NUMBER(9) NOT NULL |  |
|  | NAME | NVARCHAR2(255) NOT NULL | 目录名称 |
|  | ICON | NVARCHAR2(255) NOT NULL | 该目录的图标 |
|  | DESCRIPTION | NVARCHAR2(255) NOT NULL | 目录描述 |
|  | WEIGHT | NUMBER(9) NOT NULL | 用于排序,0则不显示 |
|  | ADMIN\_DELETE | NUMBER(9) DEFAULT 0 NOT NULL | 管理员是否删除了该目录,默认为0，即未删除状态 |

**BOYIBOYING\_SHOWNG\_SHOW**

|  |  |  |  |
| --- | --- | --- | --- |
| **备注** | **字段名** | **数据类型** | **字段说明** |
| **PK** | ID | NUMBER(9) NOT NULL |  |
| **FK** | CATEGORY\_ID | NUMBER(9) NOT NULL | 所属的目录 |
|  | NAME | NVARCHAR2(255) NOT NULL |  |
|  | POSTER | NVARCHAR2(255) NOT NULL | 该演唱会的海报信息 |
|  | DETAILS | NVARCHAR2(255) NOT NULL | 存储该演唱会等的图文信息 |
|  | CITY | NVARCHAR2(255) NOT NULL | 演出所在城市 |
|  | ADDRESS | NVARCHAR2(255) NOT NULL | 具体演出地址 |
|  | START\_TIME | DATE NOT NULL | 演出开始时间 |
|  | END\_TIME | DATE NOT NULL | 演出结束时间 |
|  | WEIGHT | NUMBER(9) NOT NULL | 该演出展示的优先基本,0为不展示 |
|  | MIN\_PRICE | FLOAT NOT NULL | 该演出的最低价 |
|  | MAX\_PRICE | FLOAT NOT NULL | 该演出的最高价 |

**BOYIBOYING\_SEATG\_SEAT**

|  |  |  |  |
| --- | --- | --- | --- |
| **备注** | **字段名** | **数据类型** | **字段说明** |
| **PK** | ID | NUMBER(9) NOT NULL |  |
| **FK** | SHOW\_ID | NUMBER(9) NOT NULL | 所属演出ID |
|  | NAME | NVARCHAR2(255) NOT NULL | 所属哪个级别,例如"A等票“，”B等票“ |
|  | PRICE | FLOAT NOT NULL | 该级别座位的定价 |
|  | CAPACITY | NUMBER(9) NOT NULL | 该级别座位的容量 |
|  | STOCK | NUMBER(9) NOT NULL | 库存 |

**BOYINBOYING\_ORDERORDER**

|  |  |  |  |
| --- | --- | --- | --- |
| **备注** | **字段名** | **数据类型** | **字段说明** |
| **PK** | ID | NUMBER(9) NOT NULL |  |
| **FK** | USER\_ID | NUMBER(9) NOT NULL | 所属用户ID |
| **FK** | SHOW\_ID | NUMBER(9) NOT NULL | 所属演出ID |
|  | STATUS | NUMBER(9) NOT NULL | 待观看，已完成，已取消(1，2，3) |
|  | TIME | DATE NOT NULL | 订单支付时间 |
|  | USER\_DELETE | NUMBER(9) DEFAULT 0 NOT NULL | 用户删除 |
|  | ADMIN\_DELETE | NUMBER(9) DEFAULT 0 NOT NULL | 管理员删除 |
|  | TICKET\_COUNT | NUMBER(9) NOT NULL | 票的总数 |
|  | MONEY | FLOAT NOT NULL | 订单总价格 |
|  | PAYMENT | NVARCHAR2(255) NOT NULL | 订单支付方式（支付宝，微信) |

**BOYING\_TBOYING\_TICKETKET**

|  |  |  |  |
| --- | --- | --- | --- |
| **备注** | **字段名** | **数据类型** | **字段说明** |
| **PK** | ID | NUMBER(9) NOT NULL |  |
| **FK** | ORDER\_ID | NUMBER(9) NOT NULL | 所属订单ID |
| **FK** | SEAT\_ID | NUMBER(9) NOT NULL | 所属座次ID |
|  | QR\_CODE\_URL | NVARCHAR2(255) NOT NULL | 二维码图片,供观影人验证入场 |

## 实验结论

### SQL语句优化实验

#### 2.2.1.1调整 WHERE 子句中的连接顺序

#### 实验结论：

实验结果表明，调整过滤能力强的句子先后顺序，仅可以起到略微的优化效果，与我们的预期不一致。条件A的过滤能力比较强，我们对比了条件 A在后和条件 B 在后时 oracle 的执行计划后。查阅资料后了解到，对于 where 子句中的多个条件连接，oracle 的优化器可以像调整表连接顺序一样自动进行优化，使得连接顺序自动调整为最优后执行，因此时间差仅仅来自于 oracle 优化器所使用的优化时间。

### 

# 项目重现步骤

## 项目结构

下述为源码的项目大体结构，boying-web、boying-admin-web、boying-back-user、boying-back-admin四个文件夹，分别对应着前端用户项目、前端管理员项目、后端用户项目、后端管理员项目，下面将分别介绍如何对四个项目进行搭建。

program

├─Back-end

│  ├─boying-back-admin

│  └─boying-back-user

├─Front-end

│  ├─boying-web

│  └─boying-admin-web

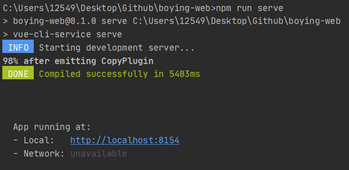
└─generateData

## 前端（boying-web、boying-admin-web）

1.注意，前端基于Vue-CLI 进行开发，如果没有相关环境，则需要先安装，可以参考以下网上教程：https://blog.csdn.net/qq\_45554909/article/details/111591308

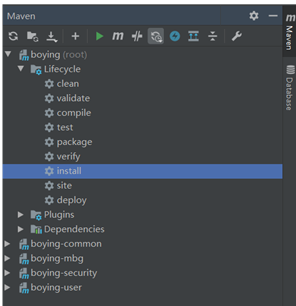
2.直接在项目根目录路径下运行npm install，耐心等待依赖包安装完成，随后启动npm run serve，即可在本地启动前端（地址会显示，例如：http://localhost:8154）

3.前端用户、管理员项目是分开的，要两个都启动，则两个项目都要npm install、npm run serve



## 后端（boying-back-user、boying-back-admin）

1. 后端基于SpringBoot开发，使用Java版本为JDK8，开发工具为IDEA，依赖管理为Maven（默认安装好了相关环境，如没装好，请参考相关网上教程）
2. 首先用Idea打开两个后端项目，进入后，在右边的Maven中，执行install操作，安装相关的依赖，并且点击左上角的刷新按钮，进行项目刷新。（如果项目标红，请查看是不是Java环境没配置好）



1. 随后要配置连接oracle数据库信息（我们用的远程服务器太贵，已关）：

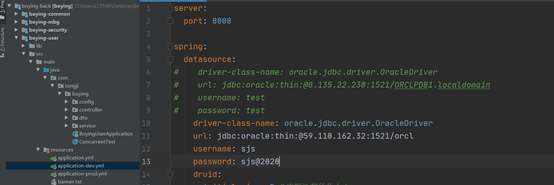
配置路径：

boying-back-web\boying-user\src\main\resources\application-dev.yml：

在该文件中修改自己的JDBC连接即可

同理，管理员后端项目也需要相同的配置步骤：

路径： boying-back-admin\boying-admin\src\main\resources\application-dev.yml



1. 随后可启动项目，在下面路径下启动main方法即可：

boying\boying-back-user\boying-user\src\main\java\com\tongji\boying\BoyingUserApplication.java

1. 管理员的项目启动路径:

boying-back-admin\boying-admin\src\main\java\com\tongji\boying\BoyingAdminApplication.java

## 数据库搭建

### Oracle数据库搭建

TODO: 1.22

### TimesTen数据库搭建

TODO: 1.22

### 数据导入

TODO: 1.22