# 汽车订票系统项目readme.md

所有页面的<title>标签显示部分都有ico汽车图标，配置在springboot默认的static/favicon.ico目录下。

数据库数据采用逻辑删除的方式，实际物理数据仍然保存在数据库的表中。所有表都增加了逻辑删除字段，使用mybatisplus的@tableLogic注解实现：

@TableLogic  
private Integer isDeleted;

## 登录模块

登录需填写字段用户名、密码、验证码以及选择身份。登录成功，跳转对应用户的主页面，登录校验失败，重定向回当前登录页面，且登录页面显示登录不成功的原因。

用户名的约束：

前端：用户名长度不能小于2以及用户名只能包含中文或英文字母

密码的约束：

前端：密码长度不能小于4

验证码使用Kaptcha，边框颜色等信息配置在application.properties文件，KaptchaConfig使用@Configuration和@Bean注解加入组件容器。

KaptchaController生产验证码字符串并保存到session中，以供登录验证使用，生成图片返回给登录页面，并在onclick函数增加时间参数，刷新验证码图片。不区分前台输入的大小写，因为在后台会全部被转换成小写。

身份选择：0表示用户，1表示管理员。选择用户身份，后台从用户表中查取数据，选择管理员后台从管理员表查取数据。

## 注册模块

注册按钮在登录页面表单，管理员不提供注册功能。注册须填写用户名，密码，确认密码，手机号，邮箱。

在前端校验：用户名长度不能小于2和用户名只能包含中文或英文字母、密码不能小于4、两次输入的密码是否一致、使用layui和正则表达式判断手机号和邮箱格式是否正确。

前后端交互使用Ajax提交用户输入的表单信息，在后台校验用户名是否已存在，若已存在则提示用户该用户名已存在，注册失败；在后台校验手机号是否已存在，若已存在则提示用户该手机号已存在，注册失败；在后台校验邮箱是否已存在，若已存在则提示用户该邮箱已存在，注册失败；保证账号中用户名、手机号和邮箱的唯一性。若不存在上述情况，则在数据库插入一条用户记录，使用layui消息框提示用户注册成功并跳登录页面。

## 用户修改密码模块

用户须输入旧密码、新密码以及确认密码，前端控制密码的长度不能小于4，前端校验新密码以及确认密码是否输入的一致。（后台也重复校验了新密码以及确认密码是否输入的一致，有点多余哈）通过Ajax异步后台校验输入的旧密码是否正确，如果正确使用layui消息提示框提示修改成功,并调用退出登录方法销毁session中保存的登录对象，重定向跳转登录页面。如果修改失败，使用layui消息提示框提示旧密码输入错误，页面保持不变。

## 退出登录功能

销毁session中保存的登录对象，重定向跳转登录页面。

## 用户管理乘客模块

调用findPassenger方法查询该用户下的所有乘客，并使用pagehelper和laypage完成分页。若该用户名下没有乘客，则引导用户点击添加乘客按钮进行添加乘客操作（监听lay-event = add)。点击添加乘客按钮弹出layui的模态框，添加乘客需填入姓名、身份证和性别。前端校验姓名姓名至少得2个字符以及姓名只能包含中文或英文字母，校验身份证为18为数字，性别为单选按钮，默认选中男。提交表单后，使用Ajax将数据发送给后台，根据用户id判断乘客表中是否已有该乘客身份证号，如果已有该身份证号码，则提示添加失败，否则，提示用户添加成功并调用findPassenger方法查询该用户下的所有乘客，并跳转乘客管理页面。

支持通过乘客姓名模糊查询乘客。

点击操作栏中的删除按钮，触发onclick中的deletePassenger方法，传入乘客id弹出layui的消息弹出框是否确实删除，选择取消则终止，点击确定则调用Ajax异步请求后台删除该乘客，弹出消息框提示删除成功，之后提交表单再次请求findPassenger方法查询该用户下的所有乘客，并跳转乘客管理页面。

点击点击操作栏中的编辑按钮（监听lay-event = edit)，通过自定义属性，将乘客的各个字段信息和当前页面的页码传给模态框的乘客表单，使用jQuery获取传来的属性后回显数据，同时通过前端约束姓名字段为长度不小于2的中文或英文字母。身份证为18位数字，通过单选框选择性别，默认选中男。点击提交修改表单后，使用Ajax将表单数据传入后台，如果身份证号码修改了，那么调用方法判断修改后的身份证号码是否已存在该用户下，如果已存在该身份证号码则提示用户修改失败，以及失败的原因。如果通过后台校验，则提示用户修改乘客信息成功，再次请求findPassenger方法查询该用户下的所有乘客，并跳转乘客管理页面。

## 管理员管理用户模块

点击会员管理，调用findAllUsers方法查询该系统下的所有会员，并使用pagehelper和laypage完成分页。并跳转会员管理页面。

提供用户姓名模糊查询并分页。

点击添加用户按钮，这里前后段的判断逻辑与用户注册一样，但是缺少确认密码输入框。同样使用的是Ajax交互数据，在后台判断保证账号中用户名、手机号和邮箱的唯一性。若不存在上述情况，则在数据库插入一条用户记录，调用findAllUsers方法查询该系统下的所有会员，并跳转会员管理页面。

点击操作栏中的删除按钮，触发onclick中的deleteUser方法，传入乘客id弹出layui的消息弹出框是否确认删除，选择取消则终止，点击确定则调用Ajax异步请求后台根据用户id删除该用户，弹出消息框提示删除成功，之后提交表单再次请求findAllUsers方法查询该系统下的所有会员，并跳转会员管理页面。

点击点击操作栏中的编辑按钮（监听lay-event = edit)，通过自定义属性，将用户的各个字段信息和当前页面的页码传给模态框的编辑用户表单，使用jQuery获取传来的属性后回显数据，修改数据的前端校验规则与添加用户相同。点击提交修改表单后，使用Ajax将表单数据传入后台，如果用户名、邮箱和手机号码做了修改，则判断系统下是否已有相同的用户名、邮箱和手机号码，没有则弹出消息框提示更新用户信息成功，之后再次请求findAllUsers方法查询该系统下的所有会员，并跳转会员管理页面。如果存在相同的用户名、邮箱和手机号码则弹出更新失败的原因。

## 管理员管理乘客模块

点击乘客管理，调用showPassenger方法查询该系统下的所有乘客，并使用pagehelper和laypage完成分页。并跳转乘客管理页面。

管理员只能根据乘客姓名和所属用户姓名模糊查询乘客信息。以及使用Ajax根据乘客id删除乘客操作。

## 管理员管理公告模块

点击公告管理，调用findAllNews方法查询该系统下的所有公告，并使用pagehelper和laypage完成分页。并跳转公告管理页面。使用css中的text-ellipsis 行内元素超出父元素宽度时，只显示省略号。就是说公告内柔太多时，多余的字用省略号代替，想查看完整的公告内容就点击对应内容onclick="showModal(this)"，弹出模态框并将完整公告内容填充至模态框显示出来。

提供模糊搜索标题查询公告功能，调用findAllNews方法查询该系统下的所有公告，并使用pagehelper和laypage完成分页。并跳转公告管理页面。

点击操作栏中的删除按钮，触发onclick中的deleteNews方法，传入公告id弹出layui的消息弹出框是否确认删除，选择取消则终止，点击确定则调用Ajax异步请求后台根据公告id删除该公告，弹出消息框提示删除成功，之后提交表单再次请求findAllNews方法查询该系统下的所有公告，并使用pagehelper和laypage完成分页。并跳转公告管理页面。

点击添加公告按钮（监听lay-event = add)，需填入标题和内容，前端校验约束这两个字段不能为空以及textarea给了10行25列的大小，使用Ajax提交公告表单的数据至后台，发布公告时间直接在后台new一个，新增公告没有任何后台约束，直接插入一条公告记录，弹出消息框提示添加成功，之后提交表单再次请求findAllNews方法查询该系统下的所有公告，并使用pagehelper和laypage完成分页。并跳转公告管理页面。

点击点击操作栏中的编辑按钮（监听lay-event = edit)，通过自定义属性，将公告的各个字段信息和当前页面的页码传给模态框的编辑公告表单，使用jQuery获取传来的属性后回显数据，修改数据的前端校验规则与添加公告相同。点击提交修改表单后，使用Ajax将表单数据传入后台，重新new一个发布时间，将修改后的公告在数据库进行更新，弹出消息框提示更新公告信息成功，之后再次请求findAllNews方法查询该系统下的所有公告，并跳转公告管理页面。

## 管理员管理路线模块

点击路线管理，调用findAllRoutes方法查询该系统下的所有路线，并使用pagehelper和laypage完成分页。跳转路线管理页面。

提供通过出发地和目的地模糊搜索路线记录。调用findAllRoutes方法查询该系统下的所有路线，并使用pagehelper和laypage完成分页。跳转路线管理页面。

点击操作栏中的删除按钮，触发onclick中的deleteRoute方法，传入路线id弹出layui的消息弹出框是否确认删除，选择取消则终止，点击确定则调用Ajax异步请求后台根据路线id删除该车次，弹出消息框提示删除成功，之后提交表单再次请求findAllRoutes方法查询该系统下的所有公告，并使用pagehelper和laypage完成分页。并跳转路线管理页面。

点击添加公告按钮（监听lay-event = add)，需填入出发地和目的地，前端校验约束这两个字段为不少于2个字的中文以及输入的出发地与目的地不能相同，使用Ajax提交公告表单的数据至后台，后台校验新增的路线是否与数据库已有路线相同，如果不同则插入一条路线记录，弹出消息框提示添加成功，之后提交表单再次请求findAllRoutes方法查询该系统下的所有路线，并使用pagehelper和laypage完成分页。并跳转路线管理页面。如果已存在相同的路线，则弹出信息框提示新增路线失败的原因。

点击点击操作栏中的编辑按钮（监听lay-event = edit)，通过自定义属性，将路线的各个字段信息和当前页面的页码传给模态框的编辑路线表单，使用jQuery获取传来的属性值后回显数据，修改数据的前端校验规则与添加路线相同。点击提交修改表单后，使用Ajax将表单数据传入后台，后台先对比传过来的路线是否有过修改，没修改直接提示修改成功，如果传过来的路线有所修改则后台校验新增的路线是否与数据库已有路线相同，如果不同则更新一条路线记录，弹出消息框提示修改成功，之后提交表单再次请求findAllRoutes方法查询该系统下的所有路线，并使用pagehelper和laypage完成分页。

## 管理员管理车次模块

点击路线管理，调用showSchedule方法查询该系统下的所有车次，并使用pagehelper和laypage完成分页。跳转车次管理页面。

提供通过出发地和目的地以及出发时间模糊搜索车次记录，发车时间为null的时候不拼接时间查询条件，并且在sql语句使用Date方法只比较年月日的部分，不比较时分秒。之后调用showSchedule方法查询该系统下的所有车次，并使用pagehelper和laypage完成分页。跳转车次管理页面。

点击操作栏中的删除按钮，触发onclick中的deleteSchedule方法，传入车次id弹出layui的消息弹出框是否确认删除，选择取消则终止，点击确定则调用Ajax异步请求后台根据车次id删除该车次，弹出消息框提示删除成功，之后提交表单再次请求showSchedule方法查询该系统下的所有车次，并使用pagehelper和laypage完成分页。并跳转车次管理页面。

点击添加车次按钮，跳转添加车次页面，（由于需要用到layui的时间选择器试了很久会发生冲突，添加和修改功能都没有使用layui模态框，而是使用页面跳转的方式完成。以及由于为了显示路线，在车次实体加了路线属性，所以在车次请求方法上不能直接用schedule实体类来接受，而是使用schedule中的每一个属性来接受，不然就会出现实体类与数据库表字段不匹配的情况，因为数据库的schedule表是找不到route这个字段的），需选择路线和发车时间，填入车牌号、行驶时间和剩余座位。前端约束行车时间为如3.5h、4h、5.5h的格式（数字+h），车牌号码如粤A12345的格式（汉字+字母+5位数字），约束剩余座位必须是大于0或小于150的整数。点击提交按钮，提供Ajax将数据传入后台，如果存在发车时间、车牌号与路线同时与数据库已有车次相同，则弹出添加失败的原因。如果不存在同时相同的车次就再次请求showSchedule方法查询该系统下的所有车次，并使用pagehelper和laypage完成分页。并跳转车次管理页面。

点击修改车次按钮，跳转修改车次页面，通过路线id和车次id查询出车次记录传输给修改车次页面并回显数据，前端字段约束条件与添加车次相同。点击提交按钮，提供Ajax将表单数据传入后台，因为数据库直接取出来的datetime数据格式永远也不会equals java.util.Date,所以一定会进入发车时间做了修改的分支判断语句，这样的话如果完全不做车次修改也是无法修改成功的，这样误打误撞的提升了数据库的性能，因为不做修改的话为了数据库考虑就不应该进行更新操作。如果修改后的数据存在发车时间、车牌号与路线同时与数据库已有车次相同，则弹出修改失败的原因。如果不存在同时相同的车次则弹出修改成功，再次请求showSchedule方法查询该系统下的所有车次，并使用pagehelper和laypage完成分页。并跳转车次管理页面。