车票管理系统

一车站每天有n个发车班次，每个班次都有一班此号（1,2,3，....n）,固定的发车时间，固定的路线（起始站，终点站），大致的行车时间，固定的额定载客量。

（1）班次信息的增删改查，支持导入班次信息（格式自定义）

（2）浏览班次信息可显示所有班次当前状况，如果当前系时间超过了耨班次的发车时间，则显示“此班已发出”的提示信息。

（3）查询路线：可按班次号查询，可按终点站查询

（4）统计售票情况

（5）售票和退票功能

（6）支持用户导出购票情况。

# 目 录

1. **需求分析………………………………………………………………1**

1、程序的功能……………………………………………………………………………1

2、输入输出的要求……………………………………………………………………1

1. **概要设计………………………………………………………………3**

1、各模块的功能…………………………………………………………………………3

2、系统总体设计…………………………………………………………………………4

1. **详细设计………………………………………………………………5**

1、各模块的类算法。……………………………………………………………………5

2、画出函数（前后端）的调用关系图。………………………………………………10

1. **程序测试及运行结果……………………………………………………11**

1、测试数据与程序运行的效果截图及说明………………………………………………11

1. **心得体会…………………………………………………………………16**

**一、需求分析**

**1、程序的功能。**

车票管理系统

一车站每天有n个发车班次，每个班次都有一班此号（1,2,3，....n）,固定的发车时间，固定的路线（起始站，终点站），大致的行车时间，固定的额定载客量。

（1）班次信息的增删改查，支持导入班次信息（格式自定义）

（2）浏览班次信息可显示所有班次当前状况，如果当前系时间超过了耨班次的发车时间，则显示“此班已发出”的提示信息。

（3）查询路线：可按班次号查询，可按终点站查询

（4）统计售票情况

（5）售票和退票功能

（6）支持用户导出购票情况。

**2、输入输出的要求**

2.1输入要求

班次信息的增删改查

输入:

班次信息，包括班号、起始站、终点站、发车时间、行车时间、额定载客量、已售票数、票价。

操作类型：增加、删除、修改、查询。

形式: 通过前端表单提交班次信息，使用HTML表单、Servlet接收处理。

浏览班次信息

输入:

查询条件：班次号或终点站。

形式: 用户在前端表单中输入查询条件，提交查询请求。

售票和退票

输入:

售票信息：班次号、购票数量。

退票信息：班次号、退票数量。

形式: 用户在前端表单中输入售票或退票信息，提交操作请求。

导入班次信息

输入:

.csv文件，包含班次信息。

形式: 用户通过文件上传控件选择并上传.csv文件。

2.2输出要求

班次信息的增删改查

输出:

操作结果提示，如“添加成功”、“删除成功”、“修改成功”、“查询结果”。

查询结果显示在前端页面。

形式: 通过页面提示或弹窗反馈操作结果。

浏览班次信息

输出:

所有班次的详细信息，包括当前状态（是否已发车）。

形式: 以表格形式展示在前端页面。

查询路线

输出:

查询结果：符合条件的班次信息。

形式: 以表格形式展示在前端页面。

统计售票情况

输出:

各班次的售票情况统计结果。

形式: 以表格或图表形式展示在前端页面。

售票和退票

输出:

操作结果提示，如“售票成功”、“退票成功”、“操作失败”等。

形式: 通过页面提示或弹窗反馈操作结果。

导入班次信息

输出:

导入结果提示，如“导入成功”或“导入失败”。

形式: 通过页面提示或弹窗反馈操作结果。

导出班次信息

输出:

.csv文件，包含当前数据库中所有班次信息。

导出结果提示，如“导出成功”。

形式: 提供文件下载链接，下载完成后弹窗提示导出成功。

**二、概要设计**

**1、各模块的功能**

1.1 模块组成

（1）数据库连接模块：连接数据库

（2）登录模块：登录系统

（3）管理班次模块：对班次信息进行增删改查

（4）浏览班次模块：在网页对班次信息进行浏览

（5）查询路线模块：对到某地的所有班次进行查询

（6）统计售票情况：对所有班次的售票情况进行统计

（7）售票模块：实现售票功能

（8）退票模块：实现退票功能

（9）导出班次信息模块

（10）导入班次信息模块

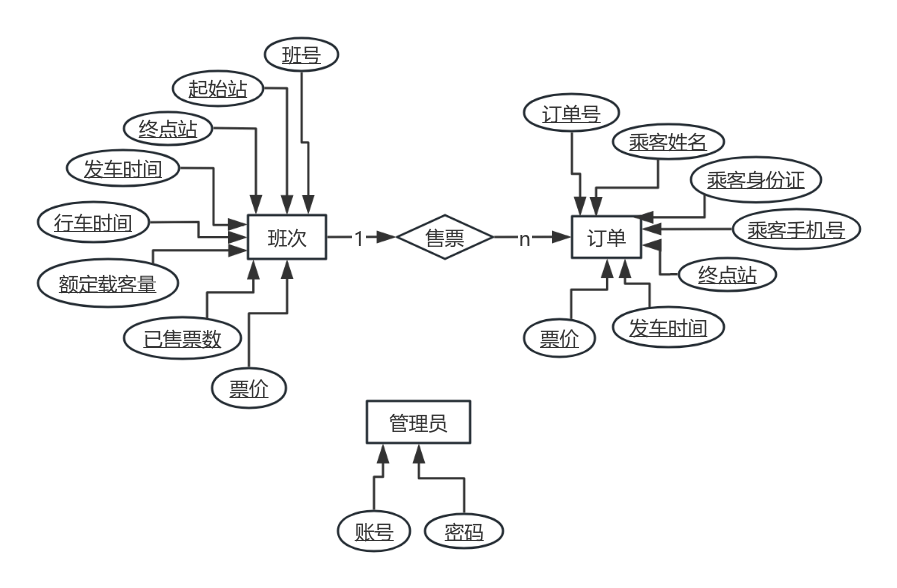
1.2 模块间的调用关系

（1）数据库连接模块是每个模块都需要调用的模块

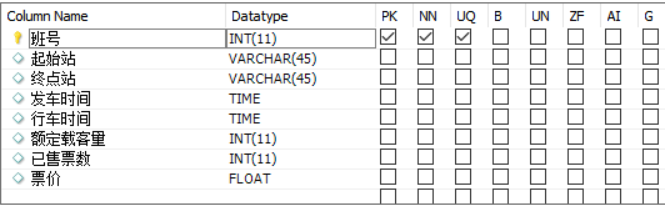
（2）其余模块间相对独立，并无调用关系

**2、系统总体设计**

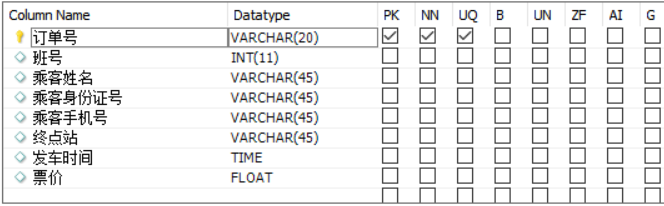
2.1 系统的总体数据结构

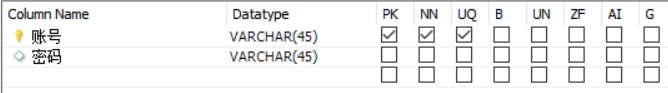


2.2 数据库结构

（1）班次

（2）订单

（3）管理员



**三、详细设计**

**1、各模块的类算法。**

1.1数据库连接模块

JDBC.java

该类负责管理与数据库的连接。主要功能如下：

定义数据库连接所需的参数（用户名、密码、URL、驱动名称）。

提供一个静态方法 getConn，用于获取数据库连接。

在 getConn 方法中，首先加载数据库驱动，然后使用 DriverManager 获取连接。

如果加载驱动或获取连接失败，将打印错误堆栈信息。

1.2登录模块

LoginServlet.java

该类是一个处理登录请求的Servlet。主要功能如下：

在 doPost 方法中获取请求中的用户名和密码参数。

使用 AdminDAO 类验证用户名和密码。

如果验证成功，重定向到首页 index.html；否则，将错误信息设置为请求属性，并

转发到登录页面login.jsp。

1.3管理班次模块

AddScheduleServlet.java

该类是一个处理添加班次请求的Servlet。主要功能如下：

在 doPost 方法中获取请求中的班次信息参数。

通过 JDBC.getConn 方法获取数据库连接。

使用 PreparedStatement 构建并执行插入班次信息的SQL语句。

如果插入成功，重定向到首页 index.html；否则，向响应中写入错误信息。

最后，关闭 PreparedStatement 和数据库连接。

DeleteScheduleServlet.java

该类是一个处理删除班次请求的Servlet。主要功能如下：

在 doPost 方法中获取请求中的班号和终点站参数。

通过 JDBC.getConn 方法获取数据库连接。

使用 Statement 构建并执行查询班次信息的SQL语句。

构建一个HTML表格，显示查询到的班次信息，每行提供一个删除按钮。

将构建的HTML表格设置为请求属性，并转发到 deleteSchedule.jsp。

最后，关闭 ResultSet、Statement 和数据库连接。

UpdateScheduleServlet.java

该类是一个处理更新班次请求的Servlet。主要功能如下：

在 doGet 方法中获取请求中的班号和终点站参数。

通过 JDBC.getConn 方法获取数据库连接。

使用 PreparedStatement 构建并执行查询班次信息的SQL语句。

构建一个HTML表格，显示查询到的班次信息，每行提供一个更新表单。

将构建的HTML表格设置为请求属性，并转发到 updateSchedule.jsp。

在 doPost 方法中获取请求中的更新后的班次信息参数。

使用 PreparedStatement 构建并执行更新班次信息的SQL语句。

如果更新成功，向响应中写入成功信息并重定向到首页 index.html。

最后，关闭 PreparedStatement 和数据库连接。

QueryScheduleServlet.java

该类是一个处理查询班次请求的Servlet。主要功能如下：

在 doGet 方法中获取请求中的多个查询参数（如班号、终点站、发车时间范围、票价范围等）。

通过 JDBC.getConn 方法获取数据库连接。

使用 PreparedStatement 构建并执行查询班次信息的SQL语句。

构建一个HTML表格，显示查询到的班次信息。

将构建的HTML表格设置为请求属性，并转发到 querySchedule.jsp。

最后，关闭 ResultSet、PreparedStatement 和数据库连接。

1.4浏览班次模块

ViewSchedulesServlet.java

该类是一个处理查看所有班次请求的Servlet。主要功能如下：

在 doGet 方法中通过 JDBC.getConn 方法获取数据库连接。

使用 Statement 构建并执行查询所有班次信息的SQL语句。

构建一个HTML表格，显示所有班次信息，包括班号、起始站、终点站、发车时间、行车时间和状态（已发车或未发车）。

将构建的HTML表格设置为请求属性，并转发到 viewSchedules.jsp。

最后，关闭 ResultSet、Statement 和数据库连接。

1.5查询路线模块

QueryRouteServlet.java

该类是一个处理查询路线请求的Servlet。主要功能如下：

在 doGet 方法中获取请求中的路线编号和终点站参数。

通过 JDBC.getConn 方法获取数据库连接。

使用 Statement 构建并执行查询班次信息的SQL语句。

根据请求参数构建SQL查询条件。

构建一个HTML表格，显示查询到的班次信息。

将构建的HTML表格设置为请求属性，并转发到 queryRoute.jsp。

最后，关闭 ResultSet、Statement 和数据库连接。

1.6统计售票情况

TicketStatisticsServlet.java

该类是一个处理票务统计请求的Servlet。主要功能如下：

在 doGet 方法中通过 JDBC.getConn 方法获取数据库连接。

使用 PreparedStatement 构建并执行查询班次信息的SQL语句。

构建一个HTML表格，显示班次信息和统计数据（如已售票数和剩余票数）。

将构建的HTML表格设置为请求属性，并转发到 ticketStatistics.jsp。

最后，关闭 ResultSet、PreparedStatement 和数据库连接。

1.7售票模块

BuyTicketServlet.java

该类是一个处理购票请求的Servlet。主要功能如下：

在 doPost 方法中获取请求中的购票信息（班次、票价、乘客信息等）。

通过 JDBC.getConn 方法获取数据库连接。

更新班次表中的已售票数。

获取最新的订单号，并插入新的订单信息到订单表中。

将购票成功信息设置为请求属性，并转发到 buyTicket.jsp。

最后，关闭 PreparedStatement、ResultSet 和数据库连接。

1.8退票模块

RefundTicketServlet.java

该类是一个处理退票请求的Servlet。主要功能如下：

在 doPost 方法中获取请求中的乘客手机号。

通过 JDBC.getConn 方法获取数据库连接。

使用 PreparedStatement 构建并执行查询订单信息的SQL语句。

构建一个HTML表格，显示查询到的订单信息，每行提供一个退票按钮。

将构建的HTML表格设置为请求属性，并转发到 refundTicket.jsp。

最后，关闭 ResultSet、PreparedStatement 和数据库连接。

1.9导出班次信息模块

ExportSchedulesServlet.java

该类是一个处理导出班次请求的Servlet。主要功能如下：

在 doGet 方法中设置响应的内容类型和文件头，用于下载CSV文件。

通过 JDBC.getConn 方法获取数据库连接。

使用 Statement 构建并执行查询所有班次信息的SQL语句。

将查询结果写入CSV文件，并通过响应输出流返回给客户端。

最后，关闭 ResultSet、Statement 和数据库连接。

1.10导入班次信息模块

ImportSchedulesServlet.java

该类是一个处理导入班次请求的Servlet。主要功能如下：

在 doPost 方法中处理上传的CSV文件。

解析CSV文件并将班次信息插入到班次表中。

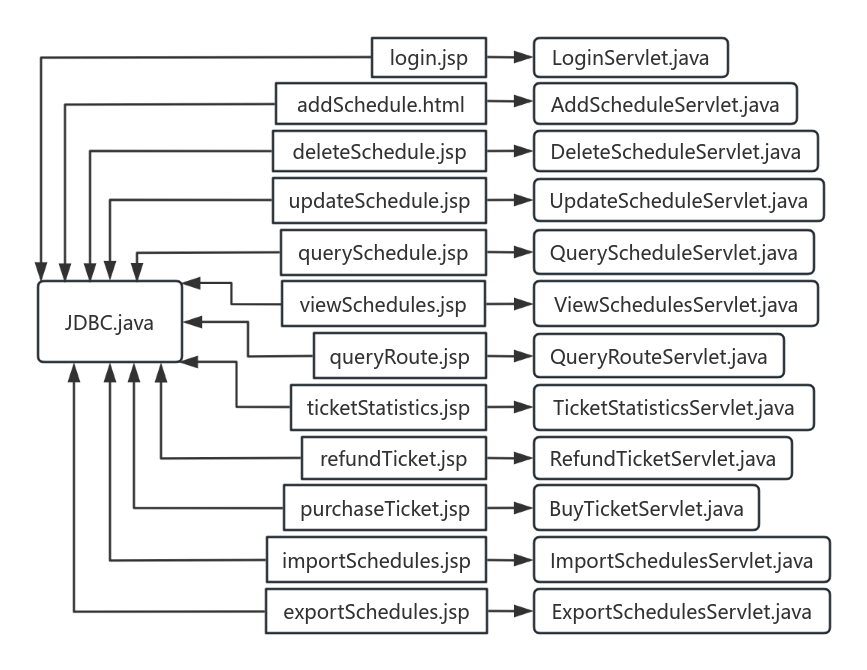
通过 JDBC.getConn 方法获取数据库连接。

使用 PreparedStatement 构建并执行插入班次信息的SQL语句。

将操作结果反馈给客户端，并删除临时上传的文件。

最后，关闭 PreparedStatement 和数据库连接。

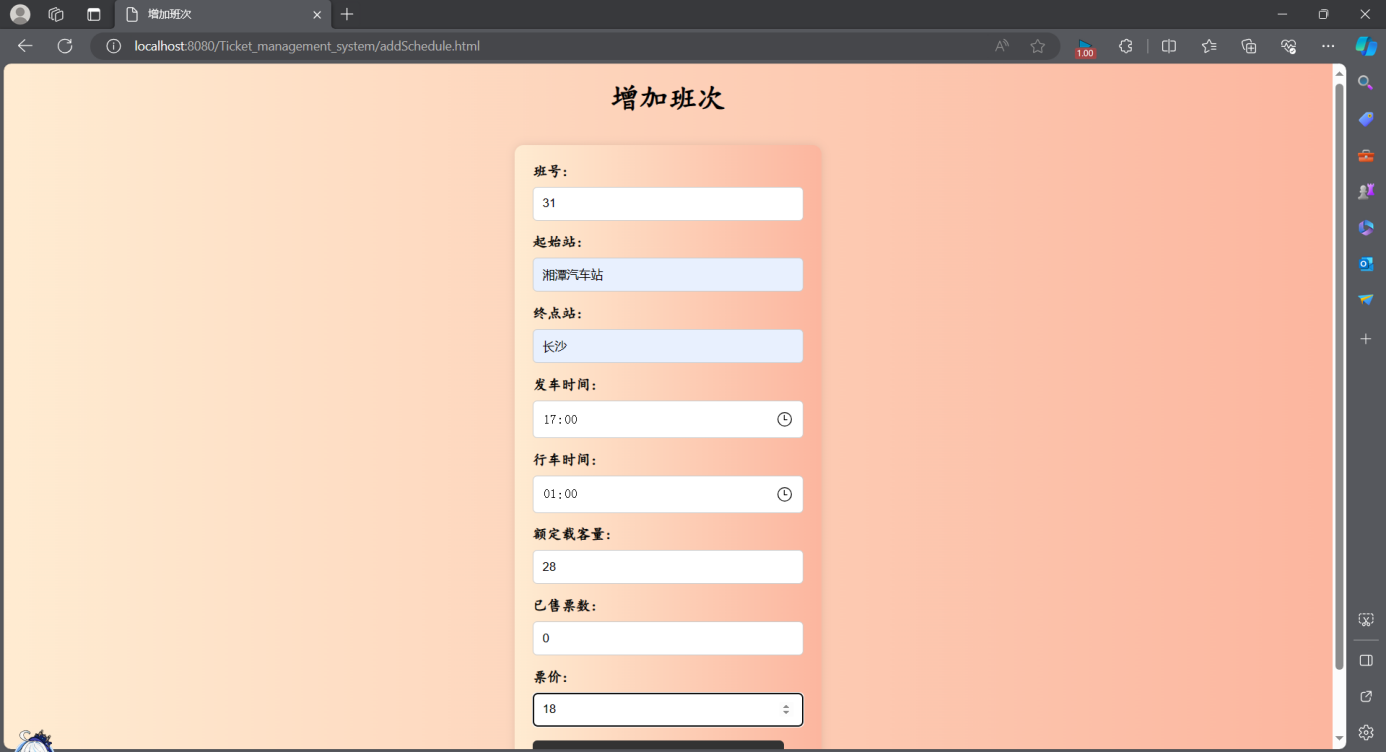
**2、画出函数（前后端）的调用关系图。**

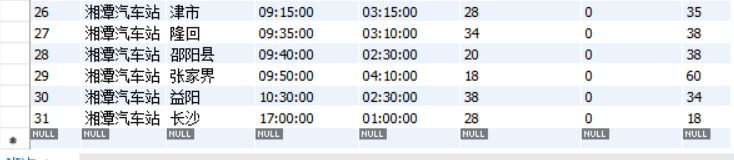
****

**四、程序测试及运行效果**

**1、测试数据与程序运行的效果截图及说明**

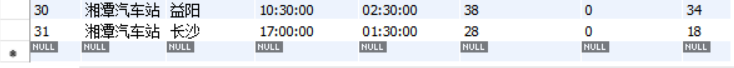
（1）增加一个班次：“31,湘潭汽车站,长沙,17:00,1:00,28,0,18”





（2）修改31号班次的行车时间为1:30



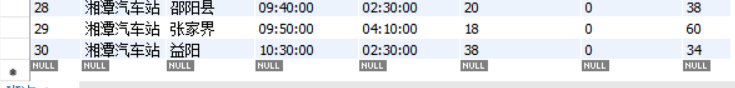


（3）查询发车时间在9:00到14:00之间的班次



（4）删除31号班次





（5）买一张16号班次到湘乡的车票

售票

（6）退掉刚刚买的票

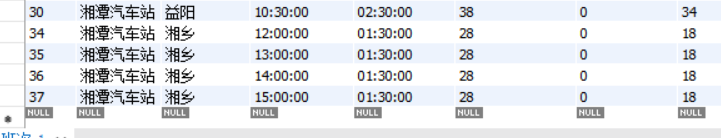


退票

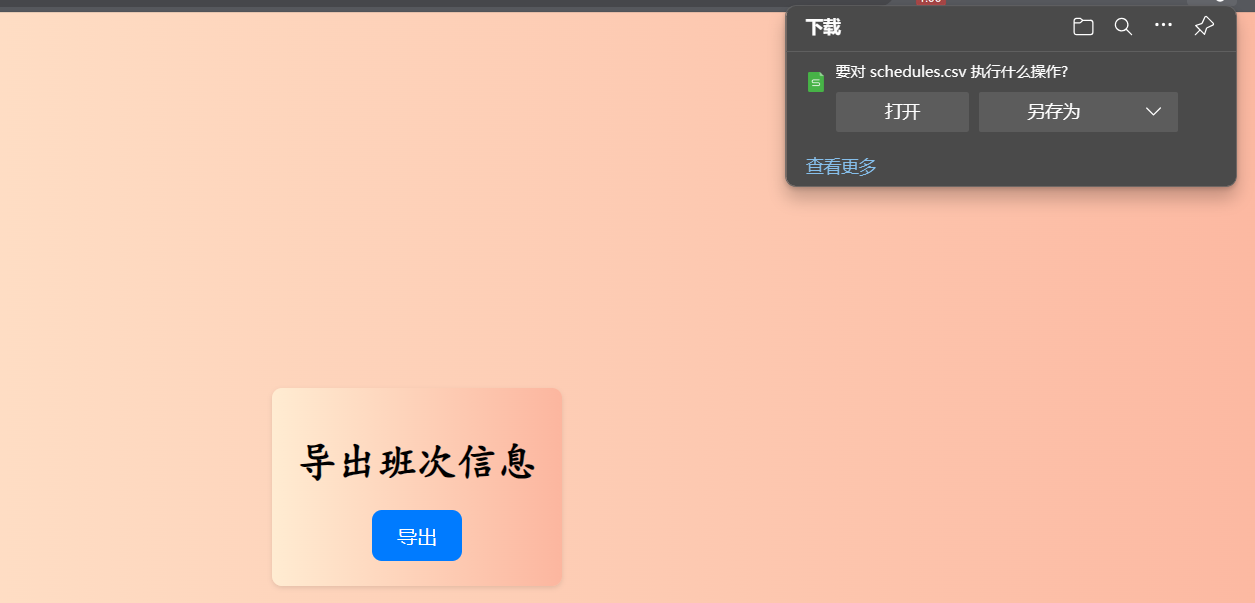
（7）导入“班次信息.csv”文件

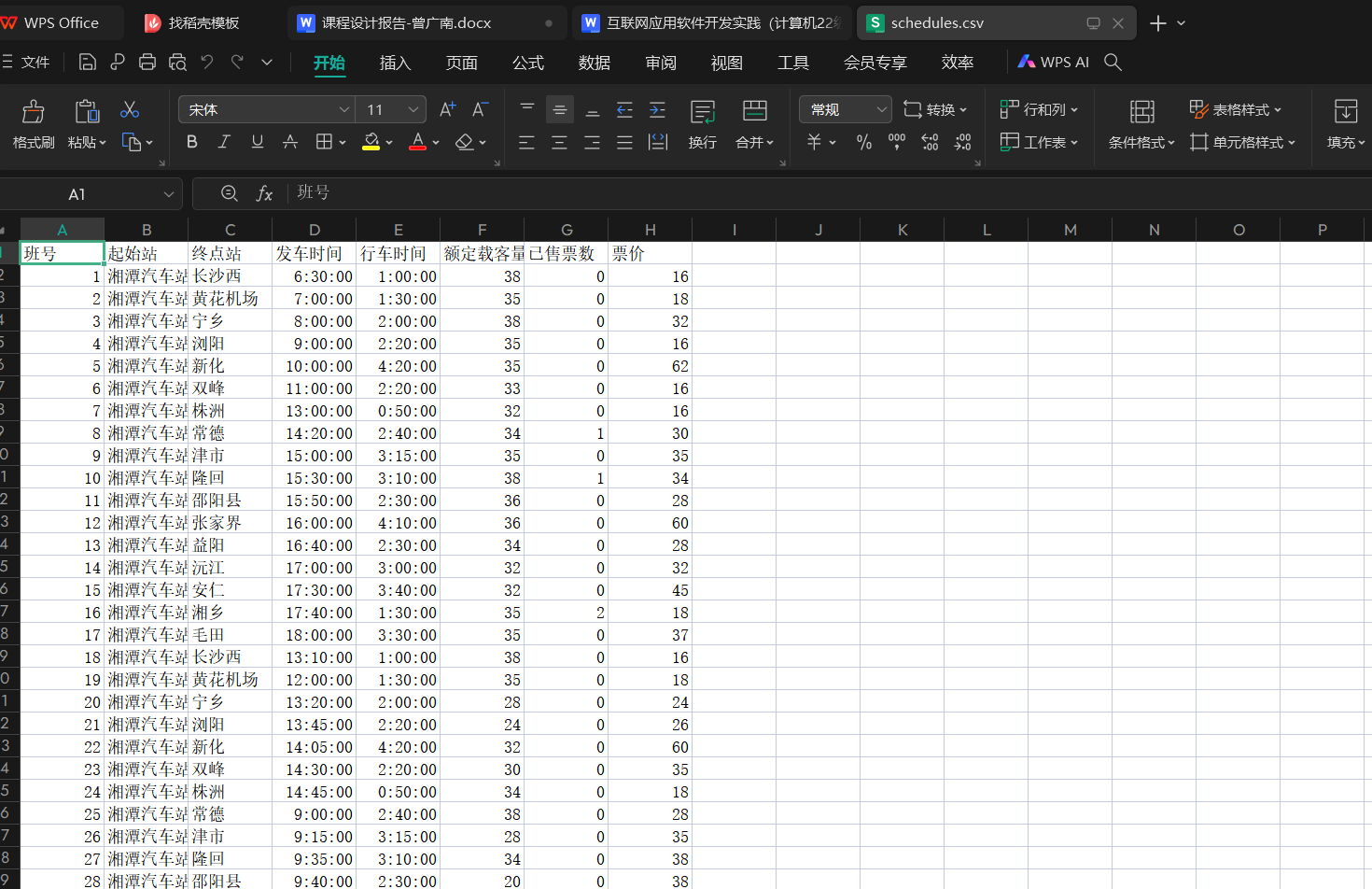






（8）导出班次信息





**五、心得体会**

在这个车票管理系统的开发过程中，我经历了从需求分析、设计、编码到测试和部署的全过程。这个项目不仅提高了我的编程技能，还让我学会了如何解决问题和应对挑战。以下是我在开发过程中遇到的问题、解决的方法以及收获的总结。

遇到的问题及解决方法

数据库设计与连接问题

问题: 在设计数据库表结构时，如何确保表与表之间的关系合理，数据存取高效，以及如何实现数据库的连接和操作。

解决方法: 我通过反复思考和设计，最终确定了班次信息表的结构。为了实现数据库的连接和操作，我学习并使用了JDBC技术，并通过编写SQL语句来实现数据的增删改查操作。

页面跳转与数据传递

问题: 在实现页面之间的跳转时，如何在不同页面之间传递数据，特别是在增删改查操作时，如何将数据从前端传递到后端，并在后端处理后再返回到前端。

解决方法: 我通过使用表单和Servlet技术，实现了页面之间的数据传递。在需要传递数据的地方使用表单提交，Servlet接收请求并处理数据，然后通过重定向或转发将结果返回前端。

文件上传与下载功能

问题: 如何实现文件的上传和下载，特别是在导入班次信息和导出班次信息时，如何处理文件的读写操作。

解决方法: 在文件上传方面，我使用了Servlet的MultipartConfig注解来处理多部分数据。在导出班次信息时，我通过Servlet生成CSV文件并将其写入响应流，实现了文件的下载功能。

数据的编码问题

问题: 在处理文件读写和数据库操作时，如何处理数据的编码问题，特别是中文字符的处理，避免乱码问题。

解决方法: 在处理文件时，我指定了文件的编码格式为UTF-8，确保读取和写入时都使用相同的编码格式。在数据库操作时，也确保使用UTF-8编码。

用户界面与用户体验

问题: 如何设计用户友好的界面，并确保用户操作简单、方便，提供良好的用户体验。

解决方法: 我使用了HTML、CSS和JavaScript来设计和美化界面，确保界面简洁、美观，操作方便。同时，在关键操作如导入导出时，提供了提示信息，让用户了解操作的结果。

收获与体会

编程技能的提升

通过这个项目，我在Java编程方面得到了很大的提升，特别是在Servlet技术、JDBC数据库操作以及文件处理方面，积累了丰富的经验。

问题解决能力的提高

在开发过程中遇到的各种问题，迫使我不断学习和查找资料，通过实践来解决问题，这大大提高了我的问题解决能力。

用户体验的重要性

在设计和实现用户界面时，我深刻体会到用户体验的重要性。一个好的用户界面可以大大提高用户的满意度和使用效率。

总的来说，这个车票管理系统的开发让我在技术和软技能方面都得到了极大的提升。通过这个项目，我不仅学会了如何开发一个完整的系统，还学会了如何面对和解决开发过程中遇到的各种问题。这些收获将对我未来的学习和工作产生深远的影响。