****

**公共管理学院**

《信息系统分析与设计》实验报告

题 目： 实验7-8 计划系统订单车导入

姓 名： 张岳平 学 号：2022315220314

专业班级：信管2201班 任课教师： 李优柱

学年学期： 2023-2024-2

|  |
| --- |
| 评语：  评分： 评阅教师： |

目录

[1.项目简介 3](#_Toc18102)

[1.1.总体简介 3](#_Toc16719)

[1.2.实现环境 3](#_Toc14471)

[1.3.实验工具 3](#_Toc4357)

[1.4.程序界面简介 3](#_Toc13191)

[1.5.程序功能简介 4](#_Toc21697)

[2.代码详解 4](#_Toc28873)

[2.1前端源代码 4](#_Toc4027)

[2.1.1订单车录入 4](#_Toc17203)

[2.2数据库设计 15](#_Toc31634)

[2.2.1.结构解析 15](#_Toc3854)

[2.3后端代码设计 15](#_Toc16390)

[2.3.1订单车接口 15](#_Toc18247)

[3. 功能演示 22](#_Toc3509)

[4. 个人心得 23](#_Toc26061)

**1.项目简介**

1.1.总体简介

数据存放通过数据库创建一个数据表，存放订单车导入基础信息以及另一个单独数据表存放不可操作的用户操作日志记录；目前支持的数据库存放的表是shopping\_car

1.2.实现环境

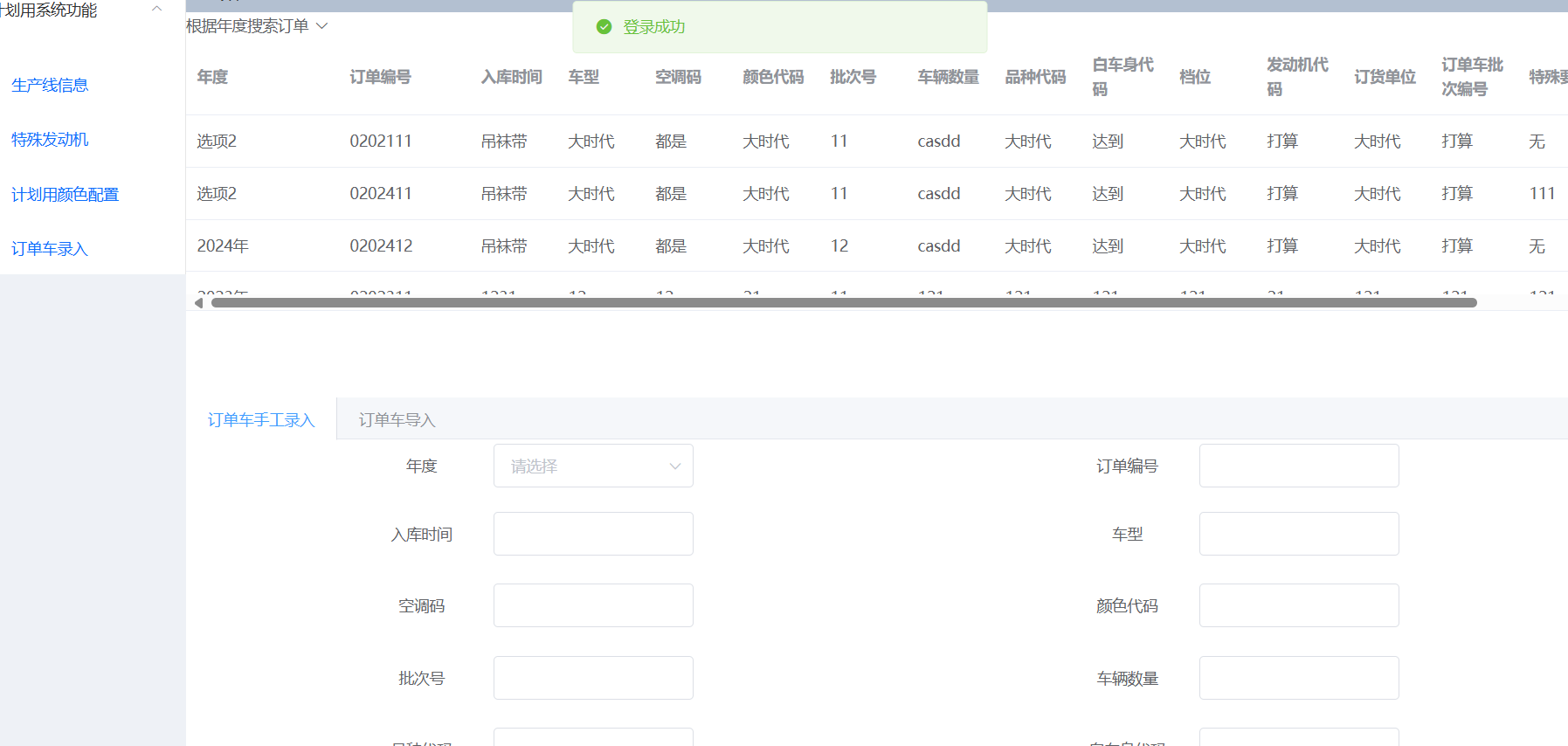
本次实验的实现总体基于Window 11系统，

前端界面主要通过vue框架，后端通过springboot框架完成。

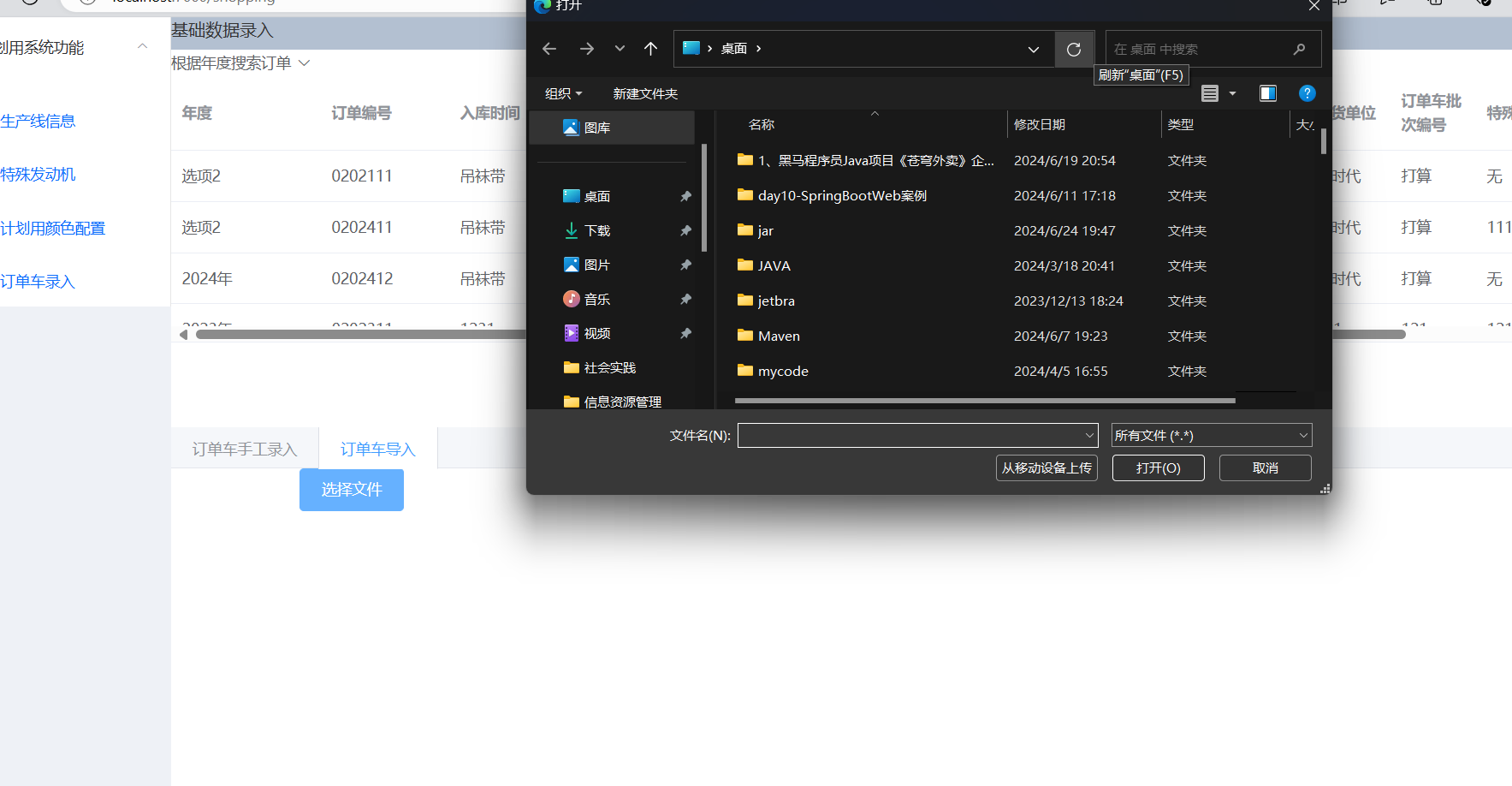
1.3.实验工具

后端代码主要通过IDEA进行开发，前端由vscode开发，数据库的设计由IDEA完成

1.4.程序界面简介



\



Vue-project中的UserLogin.vue用于用户登录，ShoppingCar是订单车的录入功能。

1.5.程序功能简介

程序UserLogin.vue主要功能分为用户登录LoginController.java用于用户日志记录和用户数据的录入；

ShoppingCar.vue用于订单车的录入，ShoppingController.java是ShoppingCar.vue的后端处理程序

**2.代码详解**

2.1前端源代码

2.1.1订单车录入

<template>

  <el-container style="height: 500px; border: 1px transparent #eee">

    <el-aside width="200px" style="background-color: rgb(238, 241, 246)">

      <el-menu :default-openeds="['1', '3']">

        <el-submenu index="1">

          <template slot="title">计划用系统功能</template>

          <el-menu-item-group>

            <el-menu-item index="1-1">

              <router-link to="/public">生产线信息</router-link>

            </el-menu-item>

            <el-menu-item index="1-2">

              <router-link to="/specialEngine">特殊发动机</router-link>

            </el-menu-item>

            <el-menu-item index="1-3">

              <router-link to="/color">计划用颜色配置</router-link>

            </el-menu-item>

            <el-menu-item index="1-4">

              <router-link to="/shopping">订单车录入</router-link>

            </el-menu-item>

          </el-menu-item-group>

        </el-submenu>

      </el-menu>

    </el-aside>

    <el-container>

      <el-header style="padding: 0; height: 30px">

        <p>基础数据录入</p>

      </el-header>

      <el-main style="padding: 0; flex: none; height: 383px">

        <el-dropdown @command="handleCommand">

          <span class="el-dropdown-link">

            根据年度搜索订单<i class="el-icon-arrow-down el-icon--right"></i>

          </span>

          <el-dropdown-menu slot="dropdown">

            <el-dropdown-item command="2020年">2020</el-dropdown-item>

            <el-dropdown-item command="2021年">2021</el-dropdown-item>

            <el-dropdown-item command="2022年">2022</el-dropdown-item>

            <el-dropdown-item command="2023年">2023</el-dropdown-item>

            <el-dropdown-item command="2024年">2024</el-dropdown-item>

          </el-dropdown-menu>

        </el-dropdown>

        <el-table :data="tableData" height="250" @row-click="clickTr">

          <el-table-column prop="dateYear" label="年度" width="140">

          </el-table-column>

          <el-table-column prop="shoppingNumber" label="订单编号" width="120">

          </el-table-column>

          <el-table-column prop="inputDate" label="入库时间"> </el-table-column>

          <el-table-column prop="typeCar" label="车型"> </el-table-column>

          <el-table-column prop="kongTiaoMa" label="空调码"> </el-table-column>

          <el-table-column prop="colorCode" label="颜色代码"> </el-table-column>

          <el-table-column prop="piCiHao" label="批次号"> </el-table-column>

          <el-table-column prop="carNumber" label="车辆数量"> </el-table-column>

          <el-table-column prop="typeNumber" label="品种代码">

          </el-table-column>

          <el-table-column prop="whiteCode" label="白车身代码">

          </el-table-column>

          <el-table-column prop="tapPosition" label="档位"> </el-table-column>

          <el-table-column prop="specialCode" label="发动机代码">

          </el-table-column>

          <el-table-column prop="orderingUnit" label="订货单位">

          </el-table-column>

          <el-table-column prop="orderCarLotNumber" label="订单车批次编号">

          </el-table-column>

          <el-table-column prop="specialRequirements" label="特殊要求">

          </el-table-column>

          <el-table-column prop="remark" label="备注"> </el-table-column>

        </el-table>

      </el-main>

      <el-footer style="padding: 0; margin-top: -30px">

        <el-tabs type="border-card">

          <el-tab-pane

            label="订单车手工录入"

            style="overflow: scroll; height: 300px"

          >

            <el-form ref="form" :model="form" label-width="120px">

              <el-row :gutter="20" style="margin-bottom: 0; height: 50px">

                <el-col :span="6" :offset="3"

                  ><div class="grid-content bg-purple" style="margin-bottom: 0">

                    <el-form-item label="年度">

                      <template>

                        <el-select v-model="value" placeholder="请选择">

                          <el-option

                            v-for="item in options"

                            :key="item.value"

                            :label="item.label"

                            :value="item.value"

                          >

                          </el-option>

                        </el-select>

                      </template>

                    </el-form-item></div

                ></el-col>

                <el-col :span="6" :offset="6"

                  ><div class="grid-content bg-purple" style="margin-bottom: 0">

                    <el-form-item label="订单编号">

                      <el-input

                        v-model="form.shoppingNumber"

                        v-bind:disabled="disabled"

                      ></el-input>

                    </el-form-item></div

                ></el-col>

              </el-row>

              <el-row :gutter="20" style="margin-bottom: 0">

                <el-col :span="6" :offset="3"

                  ><div class="grid-content bg-purple" style="margin-bottom: 0">

                    <el-form-item label="入库时间">

                      <el-input v-model="form.inputDate"></el-input>

                    </el-form-item></div

                ></el-col>

                <el-col :span="6" :offset="6"

                  ><div class="grid-content bg-purple" style="margin-bottom: 0">

                    <el-form-item label="车型" class="box1">

                      <el-input v-model="form.typeCar"></el-input>

                    </el-form-item></div

                ></el-col>

              </el-row>

              <el-row :gutter="20" style="margin-bottom: 0">

                <el-col :span="6" :offset="3"

                  ><div class="grid-content bg-purple" style="margin-bottom: 0">

                    <el-form-item label="空调码">

                      <el-input v-model="form.kongTiaoMa"></el-input>

                    </el-form-item></div

                ></el-col>

                <el-col :span="6" :offset="6"

                  ><div class="grid-content bg-purple" style="margin-bottom: 0">

                    <el-form-item label="颜色代码">

                      <el-input v-model="form.colorCode"></el-input>

                    </el-form-item></div

                ></el-col>

              </el-row>

              <el-row :gutter="20" style="margin-bottom: 0">

                <el-col :span="6" :offset="3"

                  ><div class="grid-content bg-purple" style="margin-bottom: 0">

                    <el-form-item label="批次号">

                      <el-input v-model="form.piCiHao"></el-input>

                    </el-form-item></div

                ></el-col>

                <el-col :span="6" :offset="6"

                  ><div class="grid-content bg-purple" style="margin-bottom: 0">

                    <el-form-item label="车辆数量">

                      <el-input v-model="form.carNumber"></el-input>

                    </el-form-item></div

                ></el-col>

              </el-row>

              <el-row :gutter="20" style="margin-bottom: 0">

                <el-col :span="6" :offset="3"

                  ><div class="grid-content bg-purple" style="margin-bottom: 0">

                    <el-form-item label="品种代码">

                      <el-input v-model="form.typeNumber"></el-input>

                    </el-form-item></div

                ></el-col>

                <el-col :span="6" :offset="6"

                  ><div class="grid-content bg-purple" style="margin-bottom: 0">

                    <el-form-item label="白车身代码">

                      <el-input v-model="form.whiteCode"></el-input>

                    </el-form-item></div

                ></el-col>

              </el-row>

              <el-row :gutter="20" style="margin-bottom: 0">

                <el-col :span="6" :offset="3"

                  ><div class="grid-content bg-purple" style="margin-bottom: 0">

                    <el-form-item label="档位">

                      <el-input v-model="form.tapPosition"></el-input>

                    </el-form-item></div

                ></el-col>

                <el-col :span="6" :offset="6"

                  ><div class="grid-content bg-purple" style="margin-bottom: 0">

                    <el-form-item label="发动机代码">

                      <el-input v-model="form.specialCode"></el-input>

                    </el-form-item></div

                ></el-col>

              </el-row>

              <el-row :gutter="20" style="margin-bottom: 0">

                <el-col :span="6" :offset="3"

                  ><div class="grid-content bg-purple" style="margin-bottom: 0">

                    <el-form-item label="订货单位">

                      <el-input v-model="form.orderingUnit"></el-input>

                    </el-form-item></div

                ></el-col>

                <el-col :span="6" :offset="6"

                  ><div class="grid-content bg-purple" style="margin-bottom: 0">

                    <el-form-item label="订单车批次编号">

                      <el-input v-model="form.orderCarLotNumber"></el-input>

                    </el-form-item></div

                ></el-col>

              </el-row>

              <el-row :gutter="20" style="margin-bottom: 0">

                <el-col :span="6" :offset="3"

                  ><div class="grid-content bg-purple" style="margin-bottom: 0">

                    <el-form-item label="特殊要求">

                      <el-input v-model="form.specialRequirements"></el-input>

                    </el-form-item></div

                ></el-col>

                <el-col :span="6" :offset="6"

                  ><div class="grid-content bg-purple" style="margin-bottom: 0">

                    <el-form-item label="备注">

                      <el-input v-model="form.remark"></el-input>

                    </el-form-item></div

                ></el-col>

              </el-row>

              <el-form-item>

                <el-button

                  type="primary"

                  @click="onSubmit"

                  style="margin-bottom: 0"

                  class="box19"

                  >添加</el-button

                >

                <el-button

                  type="primary"

                  @click="onUpdateList"

                  style="margin-bottom: 0"

                  >确定修改</el-button

                >

                <el-button @click="cancel">取消修改</el-button>

              </el-form-item>

            </el-form>

          </el-tab-pane>

          <el-tab-pane label="订单车导入" class="box3">

            <el-upload>

              <el-button type="primary">选择文件</el-button>

            </el-upload>

          </el-tab-pane>

        </el-tabs>

      </el-footer>

    </el-container>

  </el-container>

</template>

<script>

import axios from "axios";

export default {

  data() {

    return {

      tableData: [],

      form: {

        dateYear: "",

        shoppingNumber: "",

        inputDate: "",

        typeCar: "",

        kongTiaoMa: "",

        colorCode: "",

        piCiHao: "",

        carNumber: "",

        typeNumber: "",

        whiteCode: "",

        tapPosition: "",

        specialCode: "",

        orderingUnit: "",

        orderCarLotNumber: "",

        specialRequirements: "",

        remark: "",

      },

      row: {},

      id: "",

      input: "",

      upLinenumber: "",

      options: [

        {

          value: "2020年",

          label: "2020年",

        },

        {

          value: "2021年",

          label: "2021年",

        },

        {

          value: "2022年",

          label: "2022年",

        },

        {

          value: "2023年",

          label: "2023年",

        },

        {

          value: "2024年",

          label: "2024年",

        },

      ],

      value: "",

      disabled: false,

      command: "",

    };

  },

  mounted() {

    this.queryUserList();

  },

  methods: {

    queryUserList() {

      //发送异步请求

      axios.get("api/shopping").then((response) => {

        for (const list of response.data.data) {

          this.tableData.push(list);

        }

      });

    },

    //添加

    addList() {

      this.form.dateYear = this.value;

      axios

        .post("api/shopping", {

          dateYear: this.form.dateYear,

          shoppingNumber: this.form.shoppingNumber,

          inputDate: this.form.inputDate,

          typeCar: this.form.typeCar,

          kongTiaoMa: this.form.kongTiaoMa,

          colorCode: this.form.colorCode,

          piCiHao: this.form.piCiHao,

          carNumber: this.form.carNumber,

          typeNumber: this.form.typeNumber,

          whiteCode: this.form.whiteCode,

          tapPosition: this.form.tapPosition,

          specialCode: this.form.specialCode,

          orderingUnit: this.form.orderingUnit,

          orderCarLotNumber: this.form.orderCarLotNumber,

          specialRequirements: this.form.specialRequirements,

          remark: this.form.remark,

        })

        .then(

          this.$nextTick(() => {

            /\*  this.$nextTick用来改变数据后更新DOM元素\*/

            if (this.$refs.form) {

              this.queryUserList();

            }

            alert("添加成功");

          })

        )

        .catch((err) => console.log(err));

    },

    //单击表格行

    clickTr(row) {

      this.id = row.id;

      axios.get("api/shopping/" + this.id).then((res) => {

        this.form = res.data.data;

        this.value = this.form.dateYear;

        this.disabled = true;

      });

    },

    //添加

    onSubmit() {

      this.addList();

      this.form = {};

      this.value = "";

      window.location.reload();

    },

    onUpdateList() {

      this.updateList();

      this.form = {};

      window.location.reload();

    },

    //更新

    updateList() {

      axios

        .put("api/shopping", {

          id: this.id,

          dateYear: this.form.dateYear,

          shoppingNumber: this.form.shoppingNumber,

          inputDate: this.form.inputDate,

          typeCar: this.form.typeCar,

          kongTiaoMa: this.form.kongTiaoMa,

          colorCode: this.form.colorCode,

          piCiHao: this.form.piCiHao,

          carNumber: this.form.carNumber,

          typeNumber: this.form.typeNumber,

          whiteCode: this.form.whiteCode,

          tapPosition: this.form.tapPosition,

          specialCode: this.form.specialCode,

          orderingUnit: this.form.orderingUnit,

          orderCarLotNumber: this.form.orderCarLotNumber,

          specialRequirements: this.form.specialRequirements,

          remark: this.form.remark,

        })

        .then(

          this.$nextTick(() => {

            this.queryUserList();

            alert("修改成功");

          })

        );

    },

    cancel() {

      this.form = {};

    },

    handleCommand(command) {

      this.command = command;

      //发送异步请求

      axios.get("api/shopping/check/" + this.command).then((response) => {

        this.tableData = [];

        console.log(response.data.data);

        if (response.data.data.length == 0) {

          alert("没有该订单车");

        } else if (response.data.data.length > 1) {

          for (const list of response.data.data) {

            this.tableData.push(list);

          }

        } else {

          this.tableData.push(response.data.data[0]);

        }

      });

    },

  },

};

</script>

<style>

.box3 {

  display: flex;

  justify-content: space-between;

}

.box4 {

  width: 87vh;

}

.box5 p {

  background-color: #ece9d8;

  width: 181px;

  margin: 0;

}

.box2 {

  background-color: #ece9d8;

  height: 43vh;

  width: 87vh;

}

.box6 {

  margin-top: 10vh;

}

.el-header {

  background-color: #b3c0d1;

  color: #333;

}

.el-aside {

  color: #333;

  height: 100vh;

}

p {

  text-align: left;

}

.el-col {

  border-radius: 4px;

}

.el-row {

  margin-top: 4px;

}

.grid-content {

  border-radius: 4px;

  min-height: 36px;

}

.el-tabs--border-card > .el-tabs\_\_content {

  padding: 0 !important;

}

.el-tabs--border-card {

  border: 0 !important;

  box-shadow: none !important;

}

.el-form-item\_\_label {

  text-align: center !important;

}

.el-button {

  margin-left: 120px !important;

}

a {

  text-decoration: none;

}

.box19 {

  margin-left: 220px !important;

}

.box20 {

  margin-left: 600px !important;

  margin-top: 200px !important;

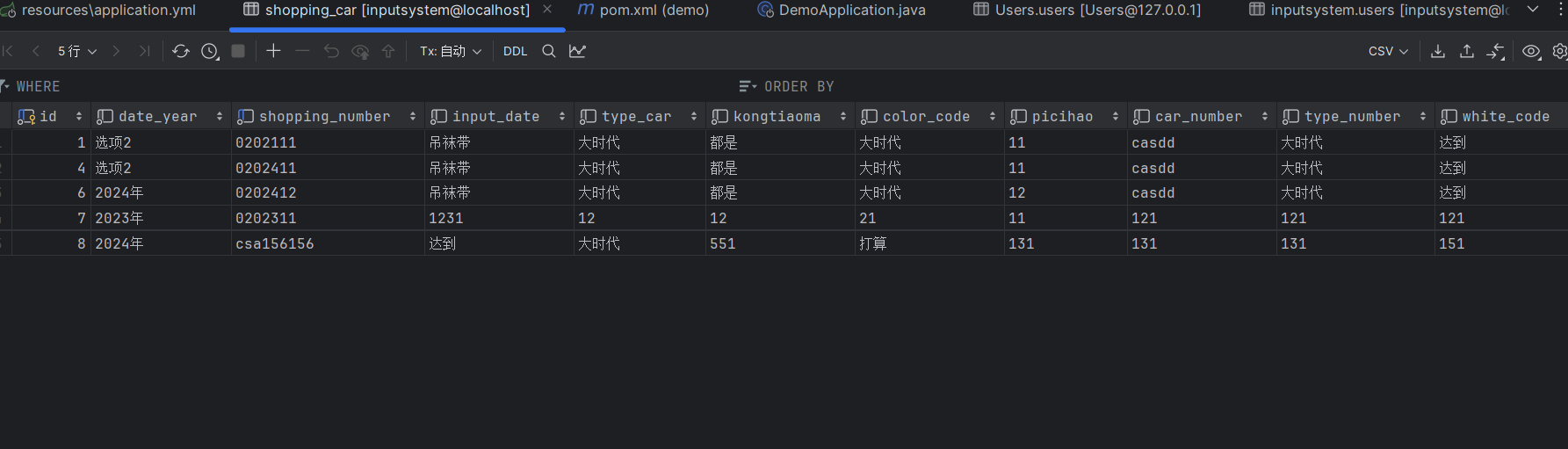
}

</style>

这段代码承接了订单车信息从前端页面到后端的全过程，制作了前端与后端的接口，通过响应事件，如点击、点击表格行与后端数据完成交互。

2.2数据库设计

2.2.1.结构解析



数据库以localhost作为连接主体，设计了数据库inputstream用于存放实验7-8的相关数据表；其中shopping\_car为实验7-8中的数据表

2.3后端代码设计

2.3.1订单车接口

package com.example.demo.controller;  
  
import com.example.demo.pojo.PublicInformation;  
import com.example.demo.pojo.Result;  
import com.example.demo.pojo.ShoppingCar;  
import com.example.demo.service.ShoppingService;  
import lombok.extern.slf4j.Slf4j;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.\*;  
  
import java.io.UnsupportedEncodingException;  
import java.net.URLDecoder;  
import java.util.List;  
  
@Slf4j  
@RestController  
@RequestMapping("/shopping")  
public class ShoppingController {  
 @Autowired  
 ShoppingService shoppingService;  
 //查询数据库中所有数据  
 @GetMapping  
 public Result list(){  
 *log*.info("查询所有数据");  
 //调用service查询部门数据  
 List<ShoppingCar> publicList=shoppingService.list();  
 return Result.*success*(publicList);  
 }  
 @GetMapping("/check/{year}")  
 public Result list2(@PathVariable String year) {  
 List<ShoppingCar> publicInformation=shoppingService.getByYear(year);  
 return Result.*success*(publicInformation);  
 }  
 //增加  
 @PostMapping()  
 public Result init(@RequestBody ShoppingCar shoppingCar){  
 *log*.info("公共信息注入：{}",shoppingCar);  
 shoppingService.add(shoppingCar);  
 return Result.*success*();  
 }  
 //根据ID查询  
 @GetMapping("/{id}")  
 public Result getById(@PathVariable Integer id){  
 *log*.info("根据ID查询信息 id:{}",id);  
 ShoppingCar publicInformation=shoppingService.getById(id);  
 return Result.*success*(publicInformation);  
 }  
 //修改  
 @PutMapping  
 public Result update(@RequestBody ShoppingCar shoppingCar){  
 *log*.info("修改数据");  
 shoppingService.update(shoppingCar);  
 return Result.*success*();  
 }  
}

这是后端用来接收前端订单车信息的控制层

package com.example.demo.mapper;  
  
import com.example.demo.pojo.ShoppingCar;  
import org.apache.ibatis.annotations.Insert;  
import org.apache.ibatis.annotations.Mapper;  
import org.apache.ibatis.annotations.Select;  
  
import java.util.List;  
  
@Mapper  
public interface ShoppingMapper {  
 @Select("select \* from shopping\_car")  
 List<ShoppingCar> list();  
 @Insert("insert into shopping\_car(date\_year, shopping\_number, input\_date, type\_car, kongtiaoma, color\_code, picihao, car\_number, type\_number, white\_code, tap\_position, special\_code, ordering\_unit, order\_car\_lot\_number, special\_requirements, remark)"+  
 "values (#{dateYear},#{shoppingNumber},#{inputDate},#{typeCar},#{kongTiaoMa},#{colorCode},#{piCiHao},#{carNumber},#{typeNumber},#{whiteCode},#{tapPosition},#{specialCode},#{orderingUnit},#{orderCarLotNumber},#{specialRequirements},#{remark})"  
 )  
 void add(ShoppingCar shoppingCar);  
 @Select("select \* from shopping\_car where id=#{id}")  
 ShoppingCar getById(Integer id);  
  
 void update(ShoppingCar shoppingCar);  
 @Select("select \* from shopping\_car where date\_year=#{year}")  
 List<ShoppingCar> getByYear(String year);  
}

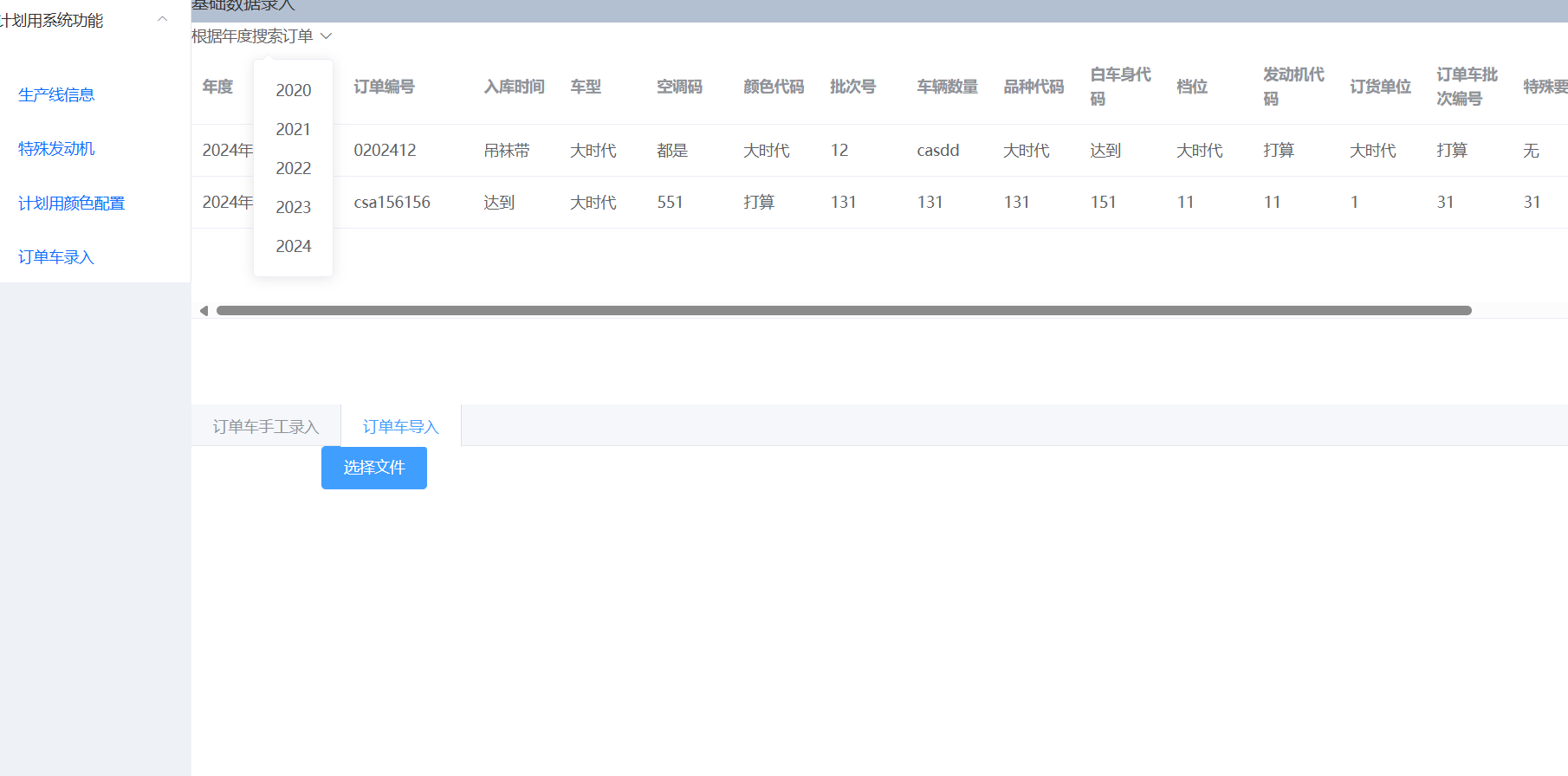
这是后端链接数据库的mapper层

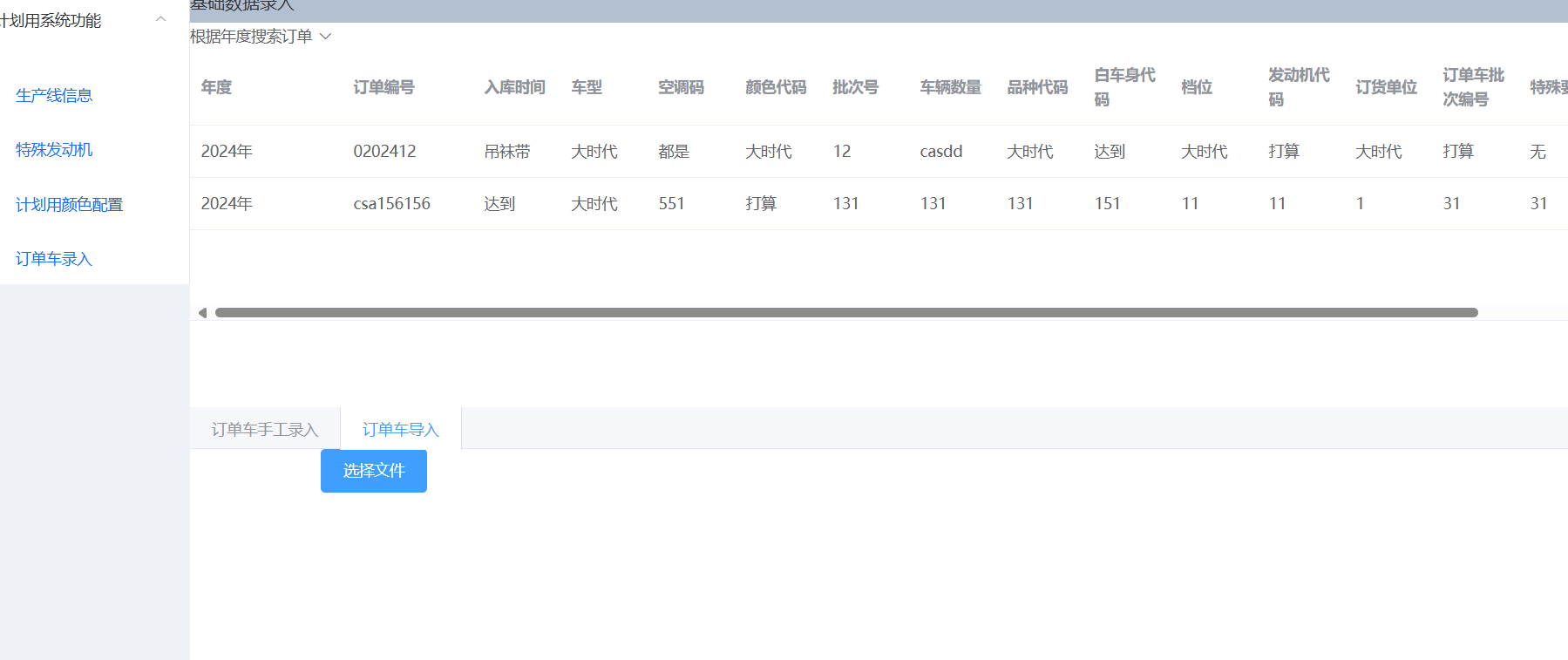
package com.example.demo.service;  
  
import com.example.demo.pojo.ShoppingCar;  
  
import java.util.List;  
  
public interface ShoppingService {  
 List<ShoppingCar> list();  
  
 void add(ShoppingCar shoppingCar);  
  
  
 ShoppingCar getById(Integer id);  
  
 void update(ShoppingCar shoppingCar);  
  
  
 List<ShoppingCar> getByYear(String year);  
}

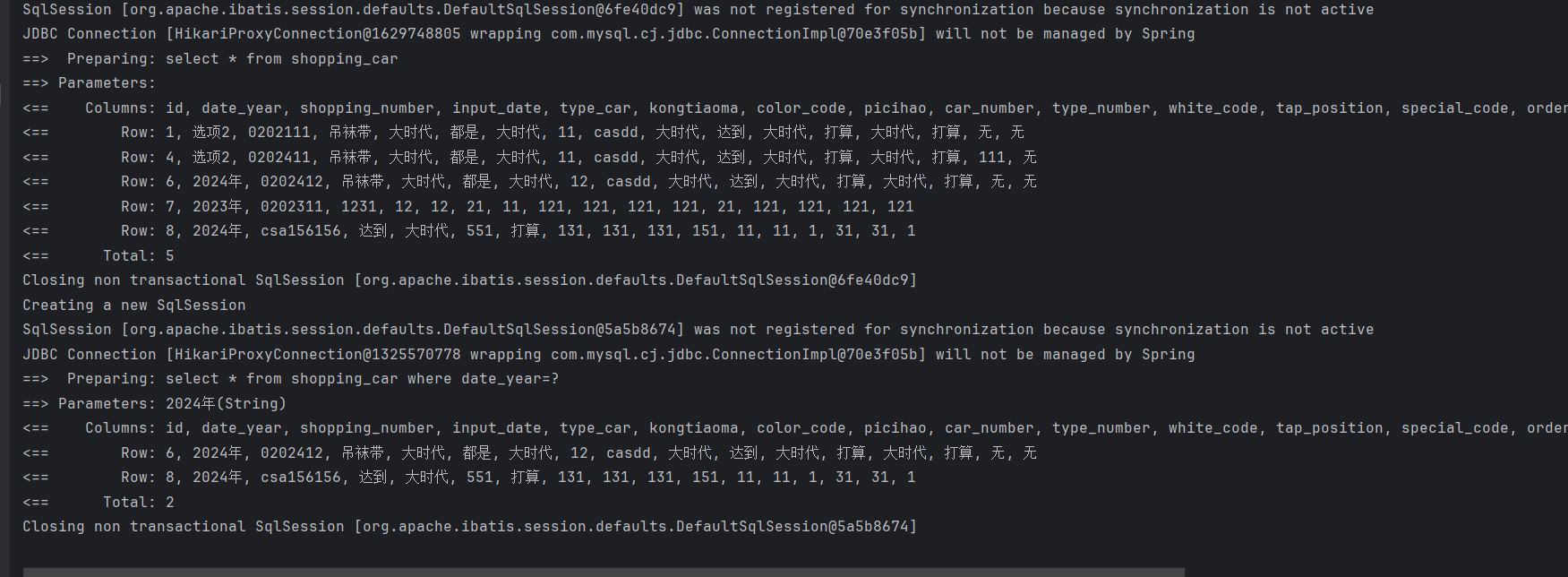
package com.example.demo.service.impl;  
  
import com.example.demo.mapper.ShoppingMapper;  
import com.example.demo.pojo.ShoppingCar;  
import com.example.demo.service.ShoppingService;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.stereotype.Service;  
  
import java.util.List;  
  
@Service  
public class ShoppingImpl implements ShoppingService {  
 @Autowired  
 ShoppingMapper shoppingMapper;  
 @Override  
 public List<ShoppingCar> list() {  
 return shoppingMapper.list();  
 }  
  
 @Override  
 public void add(ShoppingCar shoppingCar) {  
 shoppingMapper.add(shoppingCar);  
 }  
  
 @Override  
 public ShoppingCar getById(Integer id) {  
 return shoppingMapper.getById(id);  
 }  
  
 @Override  
 public void update(ShoppingCar shoppingCar) {  
 shoppingMapper.update(shoppingCar);  
 }  
  
 @Override  
 public List<ShoppingCar> getByYear(String year) {  
 return shoppingMapper.getByYear(year);  
 }  
  
}

这是后端接收数据的逻辑层

1. **功能演示**







1. **个人心得**

在实验7-8中，我使用springboot框架和MySQL数据库创建了一个Java应用程序，前端使用vue框架用于实现生产线公共信息、特殊信息、特殊发动机、计划用颜色配置在数据库中的录入和其他相关操作。通过这次综合实验，我学到了以下几点。

首先，我了解了springboot的基本概念和用法。springboot框架是java集成用来制作后端程序，它结合了许多技术如maven,mybatis，在前人已经制作框架的基础上，我们更容易、快速的制作界面

其次，我学习了如何连接MySQL数据库并执行SQL查询。在应用程序中，我使用IDEA工具与MySQL数据库进行交互。通过设置数据库连接信息和执行SQL语句，我能够获取结果集并处理其中的数据。这为我提供了与数据库进行数据交互的基本能力。

另外，我还学会了如何使用vue脚手架的，vue脚手架集成了html,css,js的相关技术，能够通过各种前人已经制作好的组件进行制作，elementui就是一个最好的例子，在本次的实验中我熟练的使用了他们，并且在实验中处处用到了。并且知道了如何创建导入并解析Excel文件。

总的来说，这次实验我通过实际动手编写代码，掌握了vue,springboot和MySQL数据库的基本用法，并且理解了如何将它们结合起来构建一个功能完整的应用程序。很大程度提升了我的编程能力，还增加了我对Java应用开发的信心。我相信这些所学将对我未来的工作和学习产生积极的影响。