



平行志愿录取系统

小组成员

目录

1. 项目背景与需求.....	2
1.1 项目背景.....	2
1.2 项目需求.....	3
2. 项目主要代码展示.....	3
2.1 代码组织结构.....	3
3. 项目部署与运行.....	5
3.1 系统登录界面.....	5
3.2 系统主界面.....	6
3.3 系统招生界面.....	7
3.4 学生信息界面.....	8
3.5 录取结果界面.....	9
3.6 调剂队列界面.....	10
3.7 录取结果界面.....	11
3.8 成绩分析界面.....	12
3.9 人数分布界面.....	13
3.10 生源地分布界面.....	15
4. 所遇问题与解决方案.....	16
4.1Mapper 层配置文件.....	16
4.2Pojo 实体类.....	17
4.3 Excel.....	17
5.项目中完成要求的相关代码截图.....	18
错误页面处理.....	19
二级访问的虚拟路径.....	19
解决中文乱码.....	20
设置默认首页.....	20
后台数据校验.....	21
事务管理.....	22
视图解析器.....	22
6.感受与体会.....	23
6.1 小组成员蔡诚杰感受与体会.....	23
6.2 小组成员蔡梓锋感受与体会.....	24
6.3 小组成员高子雄感受与体会.....	24
6.4 小组成员陈柏霖感受与体会.....	24

“平行志愿系统”项目总结报告

1. 项目背景与需求

1.1 项目背景

“十年树木，百年树人”，“科教兴国,教育为本”……，在祖国大江南北,我们都以看到这样的标语。改革开放二十几年来，我国的教育事业蓬勃发展,为我国的社会主义现代化建设培养了无数人才,为中华民族的伟大复兴做出了巨大贡献。有一句俗语：学得好，不如考得好；考的好，不如报得好。虽然观点有失偏颇，但却充分体现了高考志愿填报的重要性。随着普通高校网上录取工作的不断推进和深入，高考改革工作的不断进行，对于各院校的高考录取系统的要求也进一步细化，比如平行志愿录取，平行志愿录取的规则是“分数优先、遵循志愿、一次投档、不再补档”。平行志愿投档具有三个特点：

一是每一批次每一阶段院校录取时，高考总分高的考生先于总分较低的考生投档；二是每个考生的平行院校志愿是有先后顺序的，计算机投档时按考生填报的院校志愿顺序依次进行检索；三是平行志愿实行“多次模拟，一次投档”，不补充投档。这些不断深入的需求，对高考录取系统提出了很高的要求，特别是投档、录取工作。在投档时不但要保证程序运行的效率，还要综合比较考生的各种成绩、各种特征，根据考生的志愿由高到低进行；而对于客户端，则需要根据不同的需求采用不同的算法来满足系统的需要。

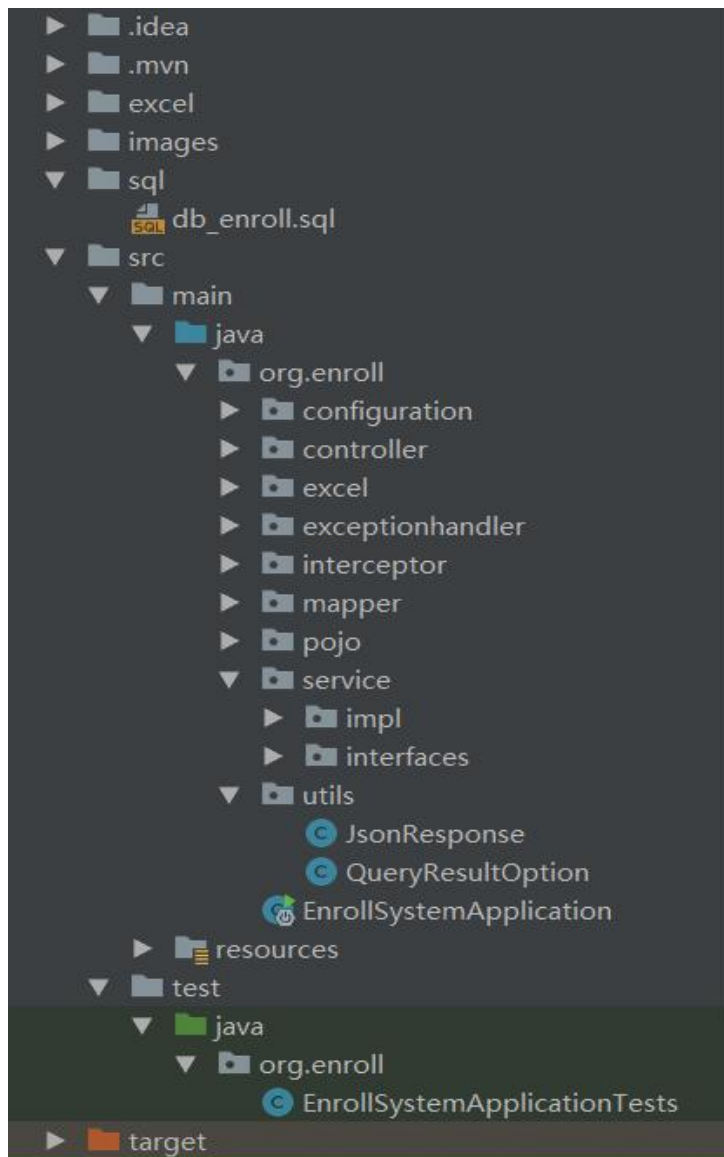
1.2 项目需求

考生进行投档时，系统除了需要保证程序的运行效率，还要在正式录取之前实现预录取、预调剂功能，从而降低录取错误发生的概率；除此之外，系统也应能对录取结果进行相应的分析，例如成绩分析、录取人数分析、成绩区间分布、生源地分布等等，因此客户端则需要根据不同的需求采用不同的算法来满足系统的需要。

2. 项目主要代码展示

2.1 代码组织结构

展示代码组织结构如图所示，独立编写的代码量约 2000，开源代码量约 1500，其主要是对数据的读取和部分功能的实现。



代码符合编写规范（命令、换行等符合要求）

通过单元测试、集成测试检查代码逻辑等方面错误

合并分支时对代码进行审查，查找缺陷

```

@RestController
@RequestMapping("/login")
public class LoginController {

    @Autowired
    LoginProperties properties;

    @Resource(name = "globalStorage")
    Map<String, Object> storage;

    @RequestMapping("/doLogin")
    public JsonResponse doLogin(String name, String pass, HttpSession session) {
        if(properties.getAdminName().equals(name) && properties.getAdminPass().equals(pass)) {
            storage.put("authSession", session);
            return new JsonResponse(JsonResponse.OK, null, null);
        } else {
            return new JsonResponse(JsonResponse.AUTH_ERR, null, "登陆失败");
        }
    }

    @RequestMapping("/checkLogin")
    public JsonResponse checkLogin(HttpSession session) {
        if (session.equals(storage.get("authSession"))) {
            return new JsonResponse(JsonResponse.OK, null, "已登录");
        } else {
            return new JsonResponse(JsonResponse.AUTH_ERR, null, "未登录");
        }
    }
}

```

3. 项目部署与运行

3.1 系统登录界面

登录界面，用户在此界面输入账号密码，后台进行验证。正确的话能够进入系统。



图 3.1 系统登录界面

3.2 系统主界面

主界面，用户在此界面进行重要操作。导入专业录取计划和学生志愿信息，能够进行预录取、预调剂等操作。最后完成整个录取，导出结果。

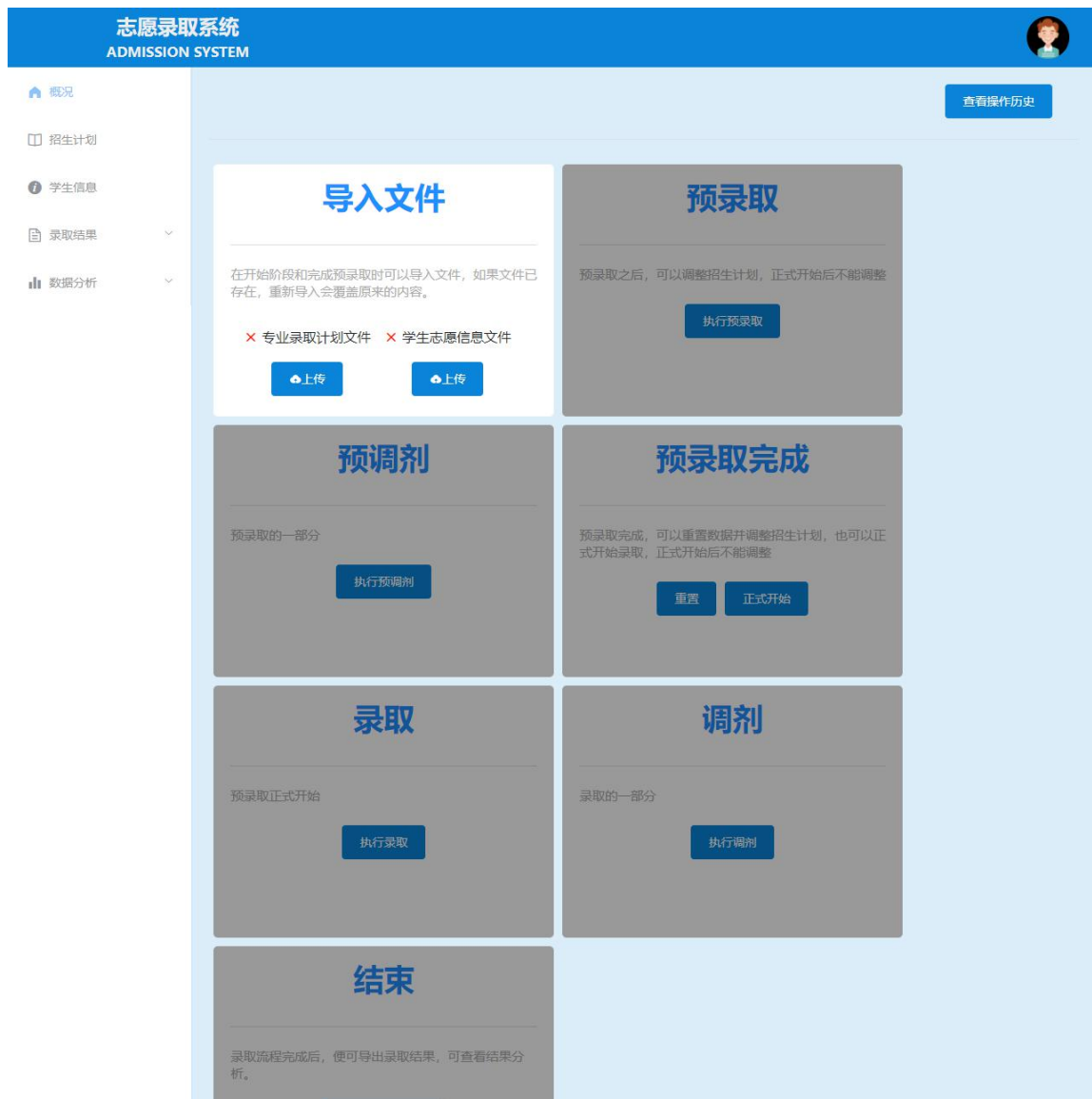



图 3.2 系统主界面

3.3 系统招生界面

招生计划界面，用户在此可以查看招生计划，并且对招生计划的详细信息进行修改。

志愿录取系统
ADMISSION SYSTEM



概况

招生计划

学生信息

录取结果

数据分析

共49条数据

专业代号	专业代码	学院	专业名称	备注	学制年限	计划招生	招生数目
001	0802	机电学院	机械类	广东理科	4	468	468
002	080301	机电学院	测控技术与仪器	广东理科	4	84	84
003	120701	机电学院	工业工程	广东理科	4	73	73
004	080801	自动化学院	自动化	广东理科	4	293	293
005	080601	自动化学院	电气工程及其自动化	广东理科	4	223	223
006	080905	自动化学院	物联网工程	广东理科	4	86	86
007	080910	自动化学院	数据科学与大数据技术	广东理科	4	90	90
008	081301	轻工化工学院	化学工程与工艺	广东理科	4	177	170
009	070302	轻工化工学院	应用化学	广东理科	4	159	132
010	082701	轻工化工学院	食品科学与工程	广东理科	4	72	60
011	0807	信息学院	电子信息类	广东理科	4	519	519
012	0810	土木与交通学院	土木类	广东理科	4	434	434
013	081201	土木与交通学院	测绘工程	广东理科	4	30	30
014	081801	土木与交通学院	交通运输	广东理科	4	35	35
015	120103	土木与交通学院	工程管理	广东理科	4	80	80
016	080901	计算机学院	计算机科学与技术	广东理科	4	203	203
017	080902	计算机学院	软件工程	广东理科	4	202	202
018	080903	计算机学院	网络工程	广东理科	4	167	167
019	080904	计算机学院	信息安全	广东理科	4	92	92
020	0804	材料与能源学院	材料类	广东理科	4	279	279
021	080203	材料与能源学院	材料成型及控制工程	广东理科	4	124	124

图 3.3 系统招生界面

3.4 学生信息界面

学生信息界面，用户在此可以查看学生的具体信息。采用分页处理，每一页显示一定的数量。

志愿录取系统
ADMISSION SYSTEM

概况

招生计划

学生信息

录取结果

全部结果

搜索查询

调剂队列

退档队列

数据分析

共6862条数据

准考证号	姓名	总分	排名	志愿1	志愿2	志愿3	志愿4	志愿5	志愿6	省份	城市	科类
8816590173	黄*5446	520	51906	035	009	021	040			广东	汕尾市	理工类
8816590176	罗*9171	547	31108	006	007	016	011	019	018	广东	潮州市	理工类
8816590179	杜*6076	536	38839	019	011	023	028	029	022	广东	梅州市	理工类
8816590182	江*5541	535	39772	046	011	038	035	045	023	广东	茂名市	理工类
8816590185	刘*6436	541	35463	006	018	005	012	029	023	广东	韶关市	理工类
8816590188	李*6703	524	48631	023	028	029	011	008	007	广东	清远市	理工类
8816590191	邓*3485	525	47477	029	023	012	020	035	021	广东	湛江市	理工类
8816590194	何*6254	521	51315	047	018	045	049	048	033	广东	肇庆市	理工类
8816590197	龙*2410	532	41854	018	003	015	043	046	024	广东	韶关市	理工类
8816590200	潘*4585	524	47821	034	035	042	043	039	033	广东	湛江市	理工类
8816590203	潘*2852	521	51083	004	005	006	023	010	049	广东	惠州市	理工类
8816590206	吴*1612	553	27500	004	001	005	011	023	022	广东	潮州市	理工类
8816590209	邹*5840	552	28313	017	016	018	006	019	033	广东	梅州市	理工类
8816590212	张*7739	537	38158	017	019	007	004	029	023	广东	茂名市	理工类
8816590215	梁*2691	555	26208	004	001	016	007	005	018	广东	清远市	理工类
8816590218	徐*1304	538	37064	004	001	005	011	023	028	广东	云浮市	理工类
8816590221	黄*6381	535	39705	005	022	001	003	020	024	广东	惠州市	理工类
8816590224	彭*5143	525	47123	033	043	032	035	034	040	广东	潮州市	理工类
8816590227	陈*1346	557	25451	007	004	017	016	001	039	广东	东莞市	理工类
8816590230	蔡*6574	525	47297	011	028	023	020	021	010	广东	肇庆市	理工类
8816590233	谭*9008	554	26906	004	047	005	011	022	001	广东	珠海市	理工类
8816590236	詹*6635	524	48493	033	035	032	034	036	037	广东	肇庆市	理工类
8816590239	孙*9240	536	38678	001	005	047	045	049		广东	珠海市	理工类

图 3.4 学生信息界面

3.5 录取结果界面

录取结果界面，用户在此界面可以查询所有的录取信息，也可以根据学院，专业，名次等条件查询。

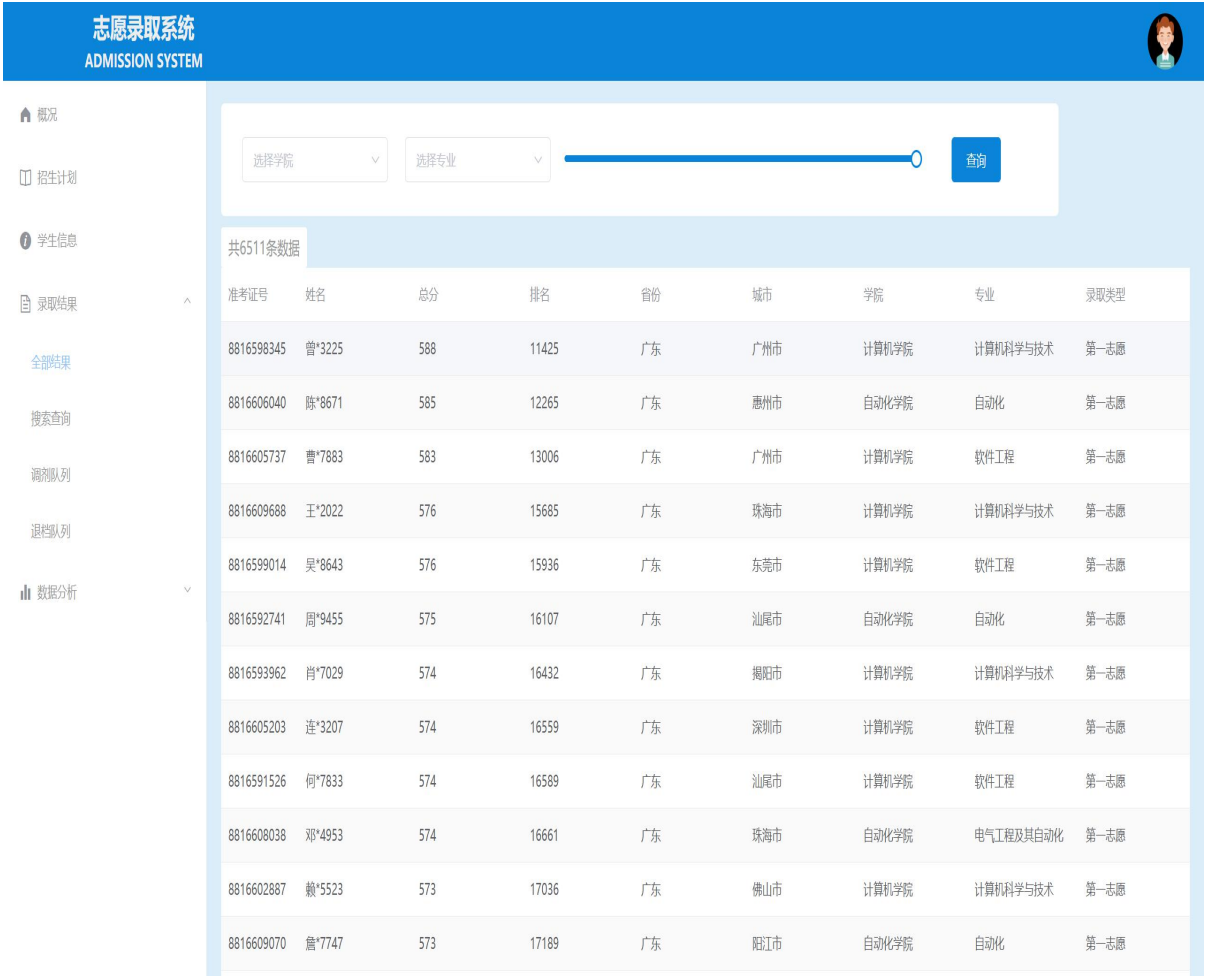


图 3.5 录取结果界面

3.6 调剂队列界面

调剂队列界面，用户在此界面可以查询学生对滴调剂信息，能够清楚哪个学生被调剂。

志愿录取系统
ADMISSION SYSTEM

概况

招生计划

学生信息

录取结果

全部结果

搜索查询

调剂队列

退档队列

数据分析

共350条数据

准考证号	姓名	总分	排名	省份	城市
8816590278	何*5505	547	31393	广东	潮州市
8816590557	朱*5215	531	42553	广东	云浮市
8816590596	何*4513	523	48651	广东	佛山市
8816590671	陈*5193	521	50461	广东	肇庆市
8816590776	朱*4139	522	50313	广东	湛江市
8816590920	郭*2702	520	51728	广东	佛山市
8816590989	吴*9861	530	43306	广东	潮州市
8816591175	朱*8246	523	48644	广东	肇庆市
8816591250	姚*6456	549	29777	广东	清远市
8816591286	李*2434	533	41396	广东	梅州市
8816591451	李*1908	520	51470	广东	深圳市
8816591601	杨*7478	526	46392	广东	惠州市
8816591688	黄*8476	521	51130	广东	云浮市
8816591703	王*7466	547	31429	广东	广州市

图 3.6 调剂队列界面

3.7 录取结果界面

录取结果界面，用户在此界面可以查询所有的录取信息，也可以根据学院，专业，名次等条件查询。

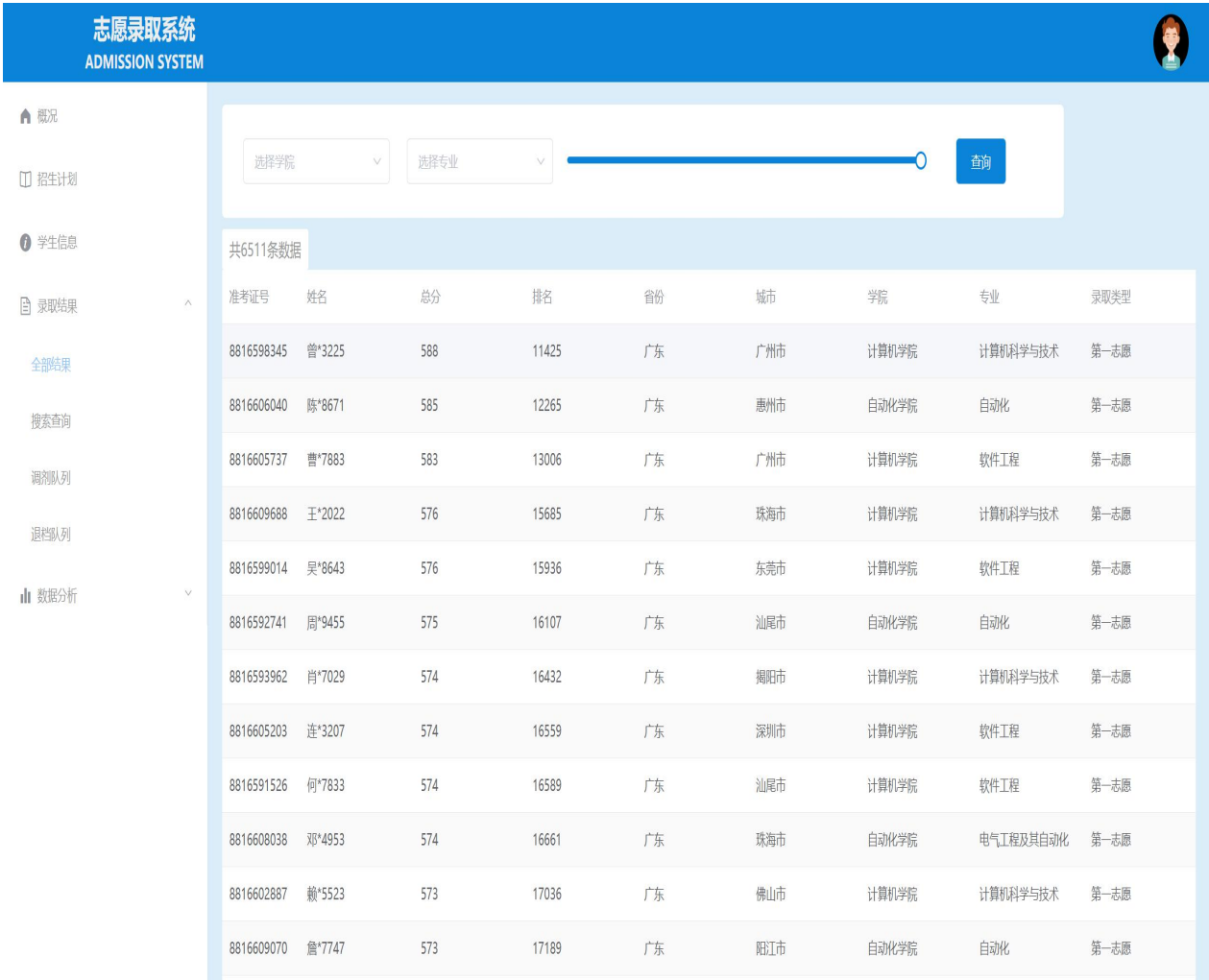
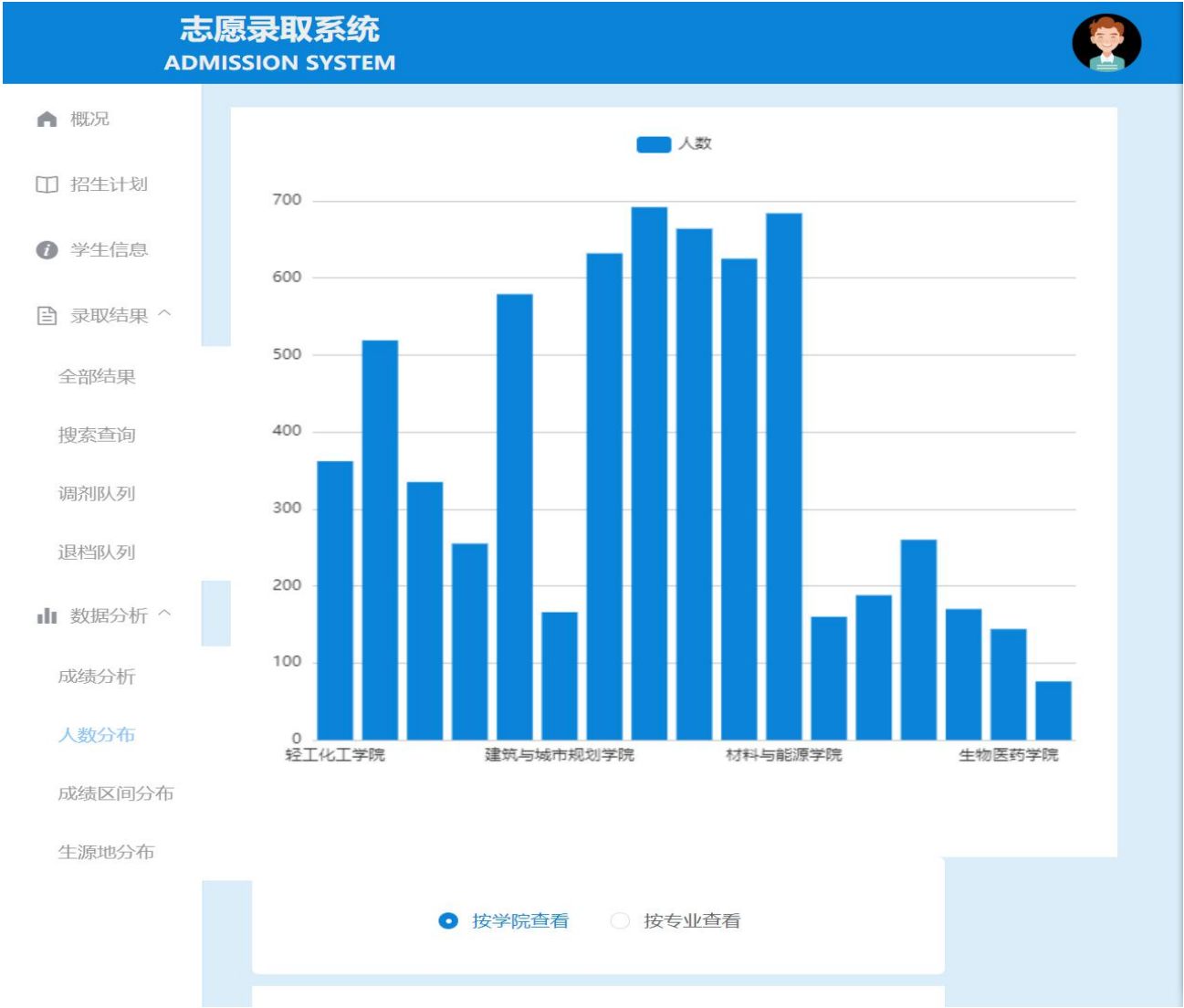




图 3.8 成绩分析界面

3.9 人数分布界面

人数分布界面，用户在此界面可以通过柱状图直观的看见不同学院的人数分布，可以通过此图进行分析学生录取信息。



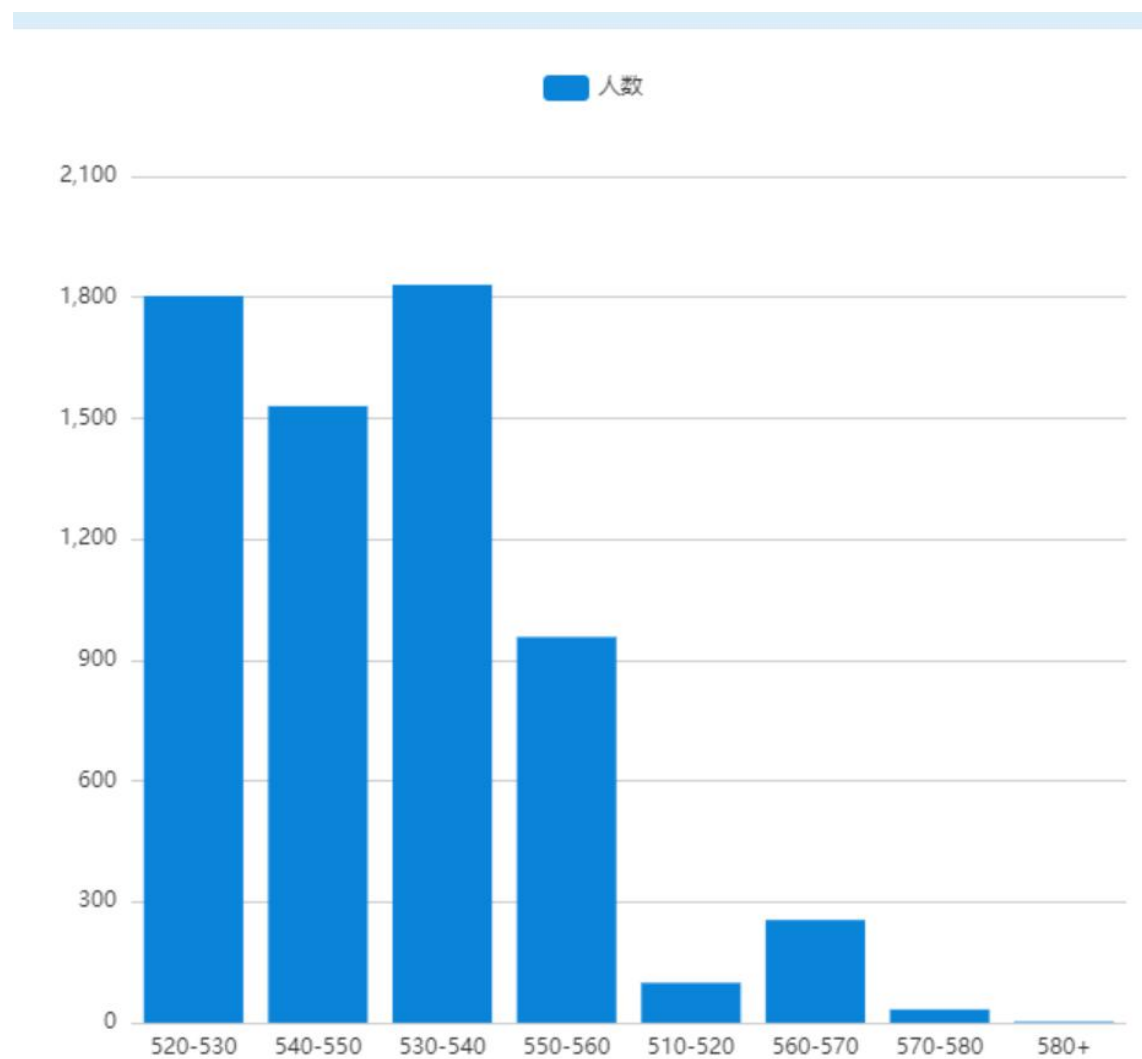


图 3.9.2 人数分布界面

3.10 生源地分布界面

生源地分布界面，此界面详细的显示了生源的全部过分，省内分布，以及通过柱状图显示分布的数量。

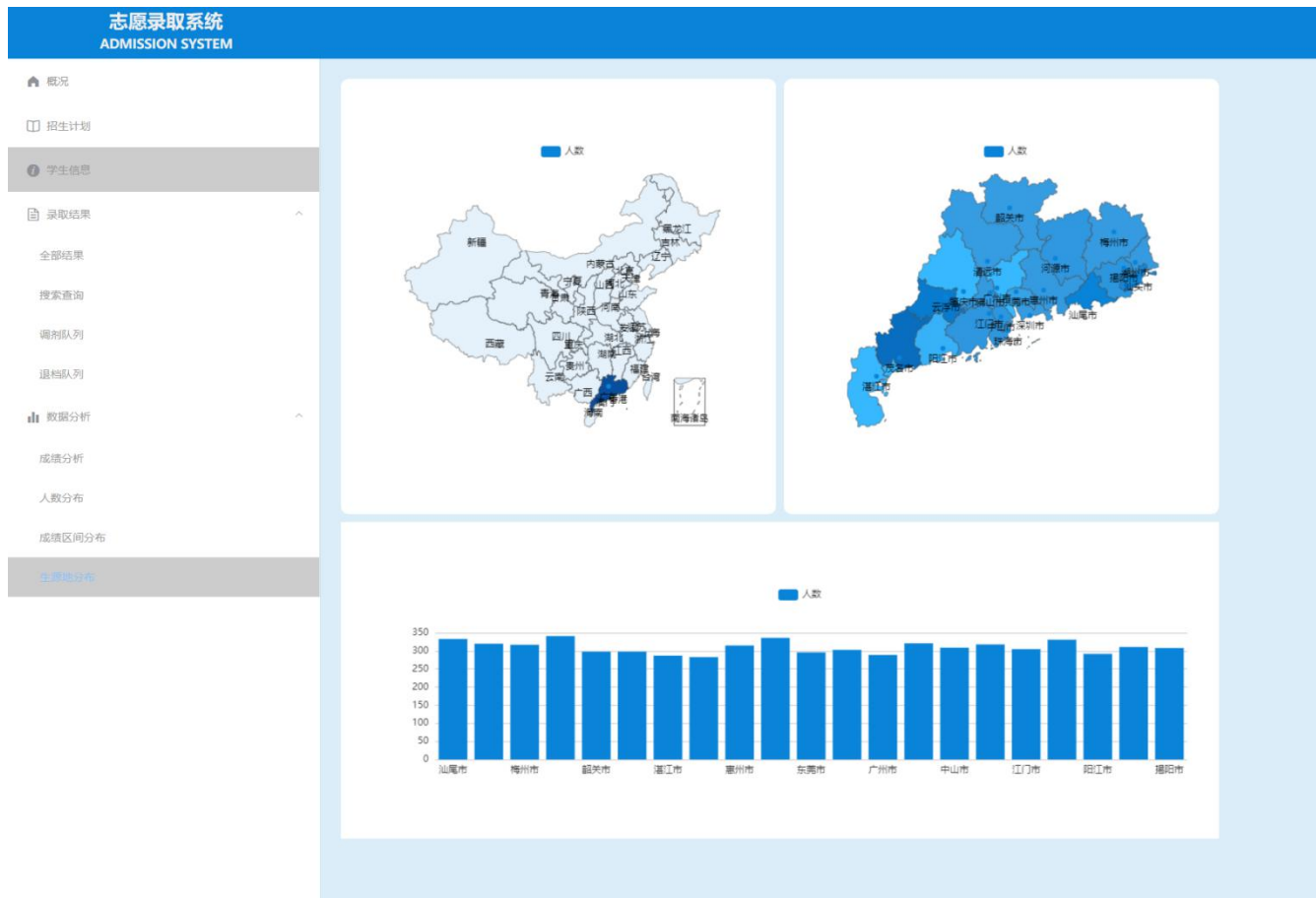


图 3.10 生源地分布界面

4. 所遇问题与解决方案

4.1 Mapper 层配置文件

问题：在编写 Mapper 层配置文件时，由于涉及很多 SQL 语句，且 Idea 无法检查 SQL 语句语法错误，如 Group By 语句所以遇到很多问题

解决方法：将编写好的 SQL 放入“Navicat for MySQL”中运行，检查无误后，再移植至配置文件

问题：在配置文件中，有很多查询是基于同一查询结果进行排序的
如：

```
SELECT t1.candidate, t1.student_id, t1.student_name, t1.total_grade,
t1.`rank`, t1.province, t1.city, t2.major_name, t3.department_name,
t1.accepted_type
FROM t_student t1, t_major t2, t_department t3
WHERE t1.accepted_major_id = t2.major_id
AND t2.department_id = t3.department_id
AND accepted_type BETWEEN 1 AND 7
AND `rank` <= #{rank}
```

```
ORDER BY `rank`
```

重复编写同一段代码不仅使代码可读性差，且查询效率低

解决方法：使用<include refid/>标签，引用重复使用率较高的查询语句，优化后代码如下：

```
<sql id="studentResultModel">
    SELECT t1.candidate, t1.student_id, t1.student_name, t1.total_grade,
    t1.`rank`, t1.province, t1.city, t2.major_name, t3.department_name,
    t1.accepted_type
    FROM t_student t1, t_major t2, t_department t3
    WHERE t1.accepted_major_id = t2.major_id
    AND t2.department_id = t3.department_id
    AND accepted_type BETWEEN 1 AND 7
</sql>
```

```
<select id="getStudentBeforeRank" resultType="studentResult">
    <include refid="studentResultModel"/>
    AND `rank` <= #{rank}
    ORDER BY `rank`
</select>
```

4.2 Pojo 实体类

问题：在其他小组成员负责代码部分，需要大量涉及查询当前流程，验证流程合法性，但是在数据库中，流程状态以数值表示，数字不能直观表示流程意义

解决方法：设计枚举类型实体类，表示每个数值的意义

该枚举类如下：

```
public enum EnrollStatus
{
    START,
    WITHOUT_MAJOR,
    WITHOUT_STUDENT,
    FILE_READY,
    PRE_ENROLL,
    PRE_ADJUST,
    READY,
    ENROLLED,
    ADJUSTED
}
```

4.3 Excel

问题：执行“录取”、“调剂”等操作时，对于不符合标准的数据条目，会抛出

错误

解决方法：在 Excel.pojo 下，添加 “@NotNull” 等标注，对数据进行检验

问题：当导入数据量较大时，ArrayList 无法缓存所有数据实体类，出现溢出错误

解决方法：设置静态变量，当数据条目操作该数值时，自动调用 Mapper 层相关语句，执行插入数据库操作，并清空 ArrayList，防止 ArrayList 溢出，然后继续往 ArrayList 中添加数据。

```
}
```

5.项目中完成要求的相关代码截图

将错误信息写入 log4j 日志文件中；

```
<configuration monitorInterval="5">
    <!--日志级别以及优先级排序: OFF > FATAL > ERROR > WARN > INFO > DEBUG > TRACE > ALL -->
    <!--变量配置-->
    <Properties>
        <!-- 格式化输出: %date表示日期, %thread表示线程名, %-5level: 级别从左显示5个字符宽度 %msg: 日志消息, %n是换行符-->
        <!-- %logger{36} 表示 Logger 名字最长36个字符 -->
        <property name="LOG_PATTERN" value="%date{HH:mm:ss.SSS} [%thread] %-5level %logger{36} - %msg%n" />
        <!-- 定义日志存储的路径 -->
        <property name="FILE_PATH" value="D:\idea_project\admissionsystem_backend-master\src\main\log" />
        <property name="FILE_NAME" value="admissionsystem_backend-master" />
    </Properties>

    <appenders>

        <console name="Console" target="SYSTEM_OUT">
            <!--输出日志的格式-->
            <PatternLayout pattern="%{LOG_PATTERN}" />
            <!--控制台只输出level及其以上级别的信息 (onMatch)，其他的直接拒绝 (onMismatch) -->
```

错误页面处理

```
Logger logger = LoggerFactory.getLogger(MainsiteErrorController.class);

private final String ERROR_PATH = "/error";

/**
 * 出现错误，跳转到如下映射中
 * @return
 */
@Override
public String getErrorPath() { return ERROR_PATH; }

/**
 * Web页面错误处理
 */
@RequestMapping(value = ERROR_PATH)
public ModelAndView errorHtml(HttpServletRequest request, HttpServletResponse response) {
    int code = response.getStatus();
    if (404 == code) {
        return new ModelAndView( "viewName: \"404.html\"");
    } else {
        return new ModelAndView( "viewName: \"505.html\"");
    }
}
```

二级访问的虚拟路径

```
@RestController
@RequestMapping("/department")
public class DepartmentController {

    @Autowired
    private IDepartmentService departmentService;

    @RequestMapping("/getDepartments")
    public JsonResponse getDepartments() {
        return new JsonResponse(JsonResponse.OK, departmentService.getDepartments(), null);
    }
}
```

解决中文乱码

```
@WebFilter(urlPatterns = "/*",filterName = "CharacterEncodingFilter")

public class CharacterEncodingFilter implements Filter {

    @Override

    public void init(FilterConfig filterConfig) throws ServletException {

    }

    @Override

    public void doFilter(ServletRequest servletRequest, ServletResponse servletResponse, FilterChain filterChain) throws IOException, ServletException {

        HttpServletRequest request = (HttpServletRequest) servletRequest;

        HttpServletResponse response = (HttpServletResponse) servletResponse;

        request.setCharacterEncoding("UTF-8");

        response.setCharacterEncoding("UTF-8");

        filterChain.doFilter(request, response);

    }

}
```

设置默认首页

```
package org.enroll.controller;

import ...

@Configuration
public class IndexViewConfig implements WebMvcConfigurer {

    @Override

    public void addViewControllers(ViewControllerRegistry registry) {

        registry.addViewController( urlPath: "/").setViewName("forward:/index.html");

        registry.setOrder(Ordered.HIGHEST_PRECEDENCE);

    }

}
```

后台数据校验

```
public class ExcelMajor {

    @NotNull
    @ExcelProperty("专业代号")
    private String majorId;

    @NotNull
    @ExcelProperty("专业代码")
    private String majorCode;

    @NotNull
    private int departmentId;

    @NotNull
    @ExcelProperty("学院")
    private String departmentName;

    @NotNull
    @ExcelProperty("专业名称")
    private String majorName;

    @NotNull
    private String majorCodeName;
}
```

事务管理

```
@Service
@Transactional(isolation = Isolation.DEFAULT, propagation = Propagation.REQUIRED)
public class ExcelServiceImpl implements IExcelService {

    @Autowired
    private MajorMapper majorMapper;

    @Autowired
    private DepartmentMapper departmentMapper;

    @Autowired
    private StudentMapper studentMapper;

    @Autowired
    private StatusMapper statusMapper;

    public void ReadMajorExcel(MultipartFile file) throws IOException {
        Integer status = statusMapper.getStatus();
    }
}
```

视图解析器

本项目中所有视图解析器封装在前端代码中，如登录页面视图解析器，当用户发送登录请求后，后端 LoginController 会执行数据库查询操作，验证用户名及密码，而后返回 Json 类型状态码及需要展示给用户的信息，前端解析 Json 数据后，返回相应视图

其解析代码如下：

```
methods: {
  doLogin() {
    this.loading = true;
    request({
      url: 'login/doLogin',
      method: 'POST',
      data: {
        name: this.username,
        pass: this.password
      }
    }).then ( res => {
      if(res.code === '000') {
        this.$router.push('/backStage/situation')
        this.$message.success('登陆成功');
      }
    })
  }
}
```

```

        } else {
            this.$message.error(res.message);
        }
        this.loading = false;
    }).catch( err => {
        this.$message.error('系统错误');
        this.loading = false;
    })
}
},

```

控制器代码:

```

public JsonResponse doLogin(String name, String pass, HttpSession
session) {
    if(properties.getAdminName().equals(name) &&
properties.getAdminPass().equals(pass)) {
        storage.put("authSession", session);
        return new JsonResponse(JsonResponse.OK, null, null);
    } else {
        return new JsonResponse(JsonResponse.AUTH_ERR, null, "登陆失败
");
    }
}

```

6.感受与体会

6.1 小组成员xxx感受与体会

蔡诚杰总结:

在本次大作业里,我对控制器的理解更加深刻了。首先通过注解@Controller 注解,表示是表现层。这个注解没有代替。在 SpringMVC 中,控制器 Controller 负责处理由 DispatcherServlet 分发的请求,它把用户请求的数据经过业务处理层处理之后封装成一个 Model,然后再把该 Model 返回给对应的 View 进行展示。在 SpringMVC 中提供了一个非常简便的定义 Controller 的方法,你无需继承特定的类或实现特定的接口,只需使用 @Controller 标记一个类是 Controller,然后使用 @RequestMapping 和 @RequestParam 等一些注解用以定义 URL 请求和 Controller 方法之间的映射,这样的 Controller 就能被外界访问到。此外 Controller 不会直接依赖于 HttpServletRequest 和 HttpServletResponse 等 HttpServletRequest 对象,

它们可以通过 **Controller** 的方法参数灵活的获取到。在这次大作业中一共定义了七个控制器，每个控制器都有对应的功能。在设计控制器的时候花费了很多精力，并且要去与别的层联系起来。是一次很大的挑战。

6.2 小组成员xxx感受与体会

总结：

在本次大作业中，我主要负责前端那一部分，最开始解除是学校开设的网页设计课，用的是 **dw**，**dw** 的好处就是你可以在写代码的时候就预览效果，虽然有时会有一些偏差，但是大部分都是可以显示出来的，不用打开浏览器来查看啦。深入学习之后使用的是 **subline** 和 **webstrom** 两个代码编辑器自己用下来是没有什么太大的区别的，可能是学得比较基础，但是足以应付基础的代码学习，这两个代码编辑器的好处就是方便快捷，在使用软件之前很有必要了解一下快捷键，这样可以很大程度上的减少写代码的时间和数量，提高效率。另外也要学会如何去查找资料，拓宽学习知识面

6.3 小组成员xxx感受与体会

总结：

在这次大作业中，我对一个项目的建立流程理解得更深刻了。对于整个项目而言，使用 **MVC** 架构模式有利于开发过程中的分工，同时代码可以做到高内聚低耦合，便于后期的整理和维护。项目的 **Service** 部分是负责业务逻辑处理，和开发项目的需求息息相关，接受相应的请求数据，并返回最终的处理结果，对应的注解是 **@Service**。并且由于其在进行业务处理时一般会依赖于对数据的访问，所以有时候还涉及事务管理。在项目进行的过程中，注解为我带来了很大的便利，熟练使用注解可以使我们的开发提升效率。这次大作业让我更深入了解了各层之间的联系，收获颇丰。

6.4 小组成员xxx感受与体会

总结：在本次大作业中，我对 **Mapper** 层配置文件的编写更加熟悉了，也根据项目需要，查找了很多相关资料，学习很多新的标签语句，及对已学过的标签语句，了解更加深入了，运用起来也更加得心应手，例如 **for each** 语句等；同时也深刻感受到了，使用 **MyBatis** 框架的便捷性，巩固了课堂上所学到的相关内容，并成功运用在项目中。对于 **Excel** 文件的导入，可以说是从零开始，之前完全没有这方面的知识，但是通过查阅资料、浏览博客、别人成功的案例等……最终完成了导入 **Excel** 文件里的数据这项工作，相信这在以后的开发中也能派的上用场。本次大作业的感受是，一定要明确分工，在项目开始前讨论好整个项目的框架，明确自己所需要负责的工作，并在开发过程中不断沟通，以了解其他成员的需求、合理优化项目、减少重复工作量；还有一定要善于利用互联网资源，学习别人成功的案例，根据自己的需求改进，站在巨人的肩膀上学习，使自己不断前进。

