

文档编号：平行志愿管理系统 - SDS - 2.1

平行志愿管理系统 软件设计规格说明书

日期：2020 年 12 月 27 日

文档变更历史记录

序号	变更日期	变更人员	变更内容详情描述	版本
1				
2				

目录

1、引言.....	4
1.1 编写目的.....	4
1.2 读者对象.....	4
1.3 软件项目概述.....	4
1.4 文档概述.....	5
1.5 参考资料.....	5
2、软件设计约束.....	5
2.1 软件设计目标和原则.....	5
2.2 软件设计的约束和限制.....	6
3、软件设计.....	6
3.1 软件体系结构设计.....	6
3.2 用户界面设计.....	7
3.3 用例设计.....	11
3.4 类设计.....	14
3.5 数据设计.....	28
3.6 部署设计.....	31

1、引言

1.1 编写目的

写此文档的目的是进一步定制软件开发的细节问题，希望能使本软件开发工作更具体。为了使用户、软件开发者及分析和测试人员对该软件的初始规定有一个共同的认识和理解，它说明了本软件的各项功能需求、性能需求 and 数据需求，明确标识各项功能的具体含义，阐述实用背景及范围，提供客户解决问题或达到目标所需要的条件或权能，提供一个度量和遵循的基准。

具体而言，编写软件需求规格说明的目的是为开发的软件提出：

- a) 软件设计总体要求，作为软件开发人员、软件测试人员相互了解的基础。
- b) 功能、性能要求，数据结构和采集要求，重要的接口要求，作为软件设计人员进行概要设计的依据。
- c) 软件确认测试的依据。

1.2 读者对象

用户、软件设计人员、开发人员、需求分析和测试人员

1.3 软件项目概述

- 项目名称：平行志愿录取系统
- 用户单位：各大高校
- 开发单位：广东外语外贸大学数据库工程小组
- 软件项目的背景和大致功能：

随着高考参考人数的增多，现如今的高考志愿早已采用网上填志愿的方式。志愿填报与每个高考考生今后的发展息息相关，如果考生的高考分数达到自己的理想水平，在加上志愿填报的好就可以锦上添花，进入理想的大学，与之相反，如果考生在填报志愿时盲目的填报

高考志愿，轻则无法进入较为满意的学校、学习喜欢的专业，重则直接会导致退档。因此，高校正确录取学生的志愿显得尤为重要，一套软件系统能够在录取学生的时候处理海量的学生数去，能在录取中减少录取失误的可能性。并且能够分析生源数据，为学校分析学生水平，来源地等都有了很大的帮助。

1.4 文档概述

1) 软件的设计约束部分。它包括软件设计目标和原则、软件设计受到的约束和限制。

2) 软件的设计部分。它主要分为软件体系结构设计、用户界面设计、用例设计、类设计、数据设计以及部署设计。

1.5 参考资料

[1]. 《软件工程（第三版）》. 钱秋乐，赵文耘，牛军钰. 北京:清华大学出版社，2016

2、软件设计约束

2.1 软件设计目标和原则

目标:

- 实现用户的登录，注册功能
- 实现考生的模拟投档功能
- 实现学校和专业的预测推荐功能
- 实现各大院校专业的招生信息查看功能
- 可选择查看全校/学院/专业查看最高分、最低分、学院专业成绩区间分布、生源地分布等信息
- 录取结果可查询功能

- 软件应能快捷地提供查询服务，软件正常启动的响应时间应该小于 1s，各个功能模块的点击，键盘输入响应时间也应在用户能够容忍的范围之内，数据更新的时间应小于 0.2s

原则：

- (1) 抽象化原则
- (2) 模块化原则
- (3) 可拓展性原则
- (4) 软件重用的原则
- (5) 迭代设计的原则
- (6) 可追踪性的原则

2.2 软件设计的约束和限制

- 运行环境要求：安卓操作系统 Android、windows 操作系统、Macos 操作系统、ubuntu 等支持 Internet 的系统
- 开发语言：Java、html
- 标准规范：Java 编码风格（规范）
- 开发工具：Idea 2020

3、软件设计

3.1 软件体系结构设计

该系统主要分为前端界面层，业务逻辑层，数据交互层。界面层主要用于用户对系统进行交互，能够使用户使用系统。其中“loginUi”用于用户登录，验证用户的账号密码。“StudentUI”

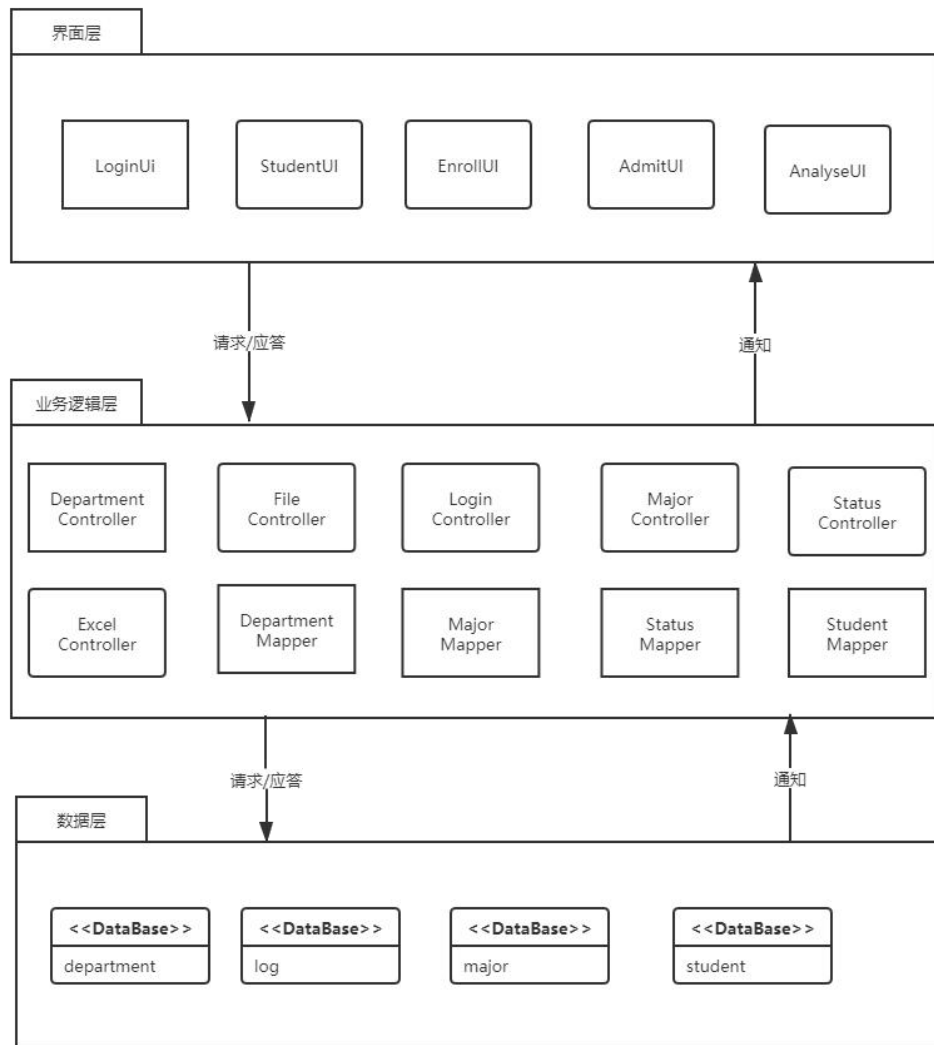


图 3.1.2: “平行志愿录取系统”的总体设计架构

3.2 用户界面设计

根据“平行志愿录取系统”的用例描述以及每个用例的交互图，可以发现该软件系统在网页版上需要有以下一组界面以支持用户的操作。

- 登录界面“LoginUI”，其职责是帮助用户输入用户信息以登录到系统之中。
- 概况界面“OverviewUI”，其职责是显示志愿录取执行的各个步骤。
- 历史操作界面“HistoryUI”，其职责是帮助用户查看录取执行操作历史。
- 招生计划界面“EnrollmentUI”，其职责是显示学校的招生计划。
- 学生信息界面“StudentUI”，其职责是显示所有学生信息。

- 录取结果界面“ResultUI”，其职责是帮助用户查看并搜索录取结果。
- 数据分析界面“AnalyzeUI”，其职责是帮助用户查看数据分析结果。

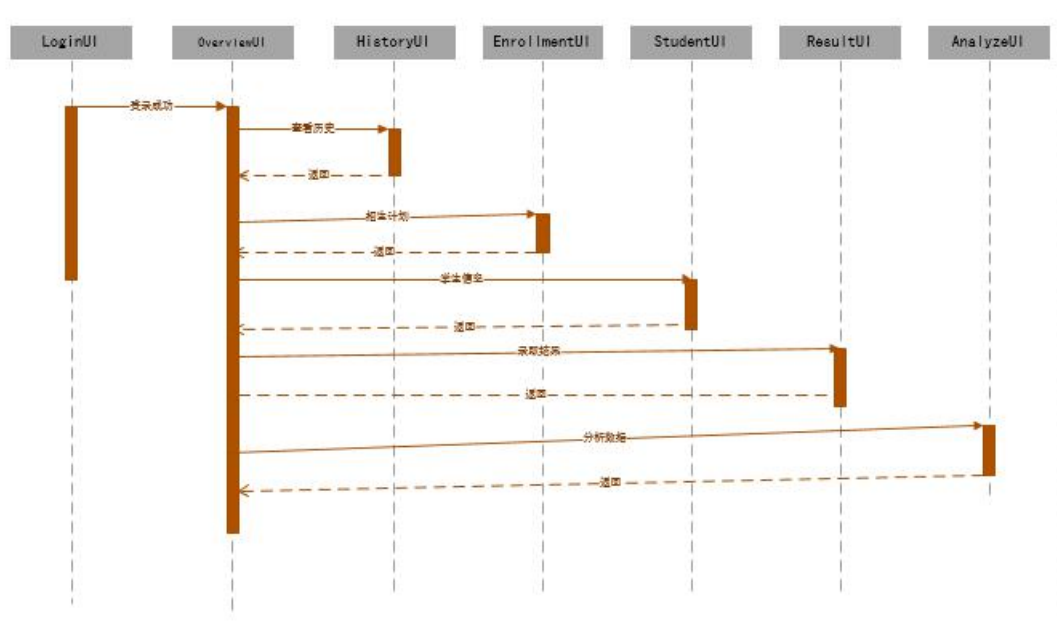


图 3.2 描述“平行志愿录取系统”用户界面跳转关系的顺序图

- 1) 本界面为用户进行登录的界面



图 3.2.1 系统用户登录界面

- 2) 下图为软件的全体概况界面

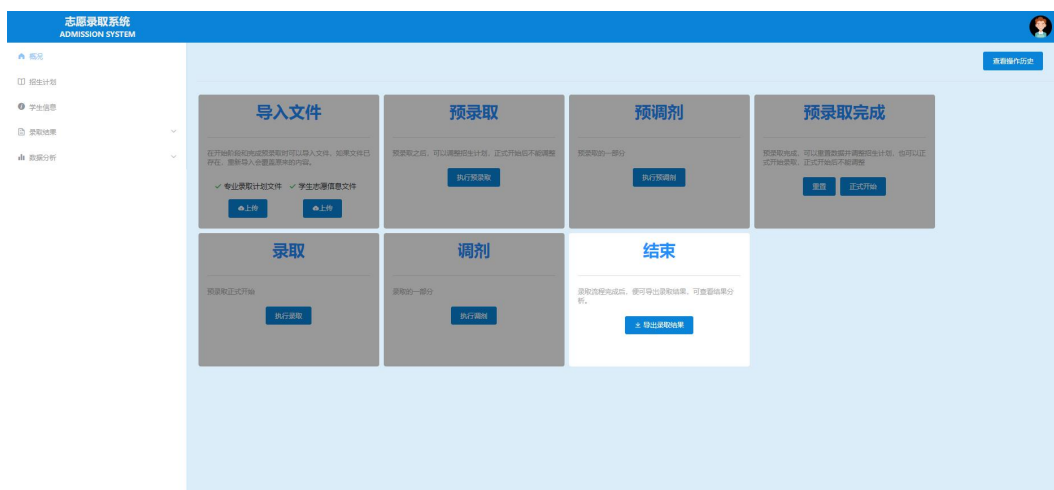


图 3.2.2 系统全体概况

3) 下图为操作历史界面

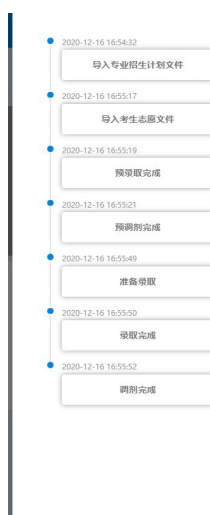


图 3.2.3 操作历史界面

4) 下图为招生计划界面

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									
专业代码	专业名称	学院	专业名称	备注	学制年限	计划招生	招生数		
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	轻工化学院	化学工程与工艺	广东理科	4	177	177		
009	070302	轻工化学院	应用化学	广东理科	4	159	159		
010	082701	轻工化学院	食品科学与工程	广东理科	4	72	72		
011	0807	信电学院	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	080905	计算机学院	数据科学与大数据技术	广东理科	4	970	970		

图 3.2.4 招生计划界面

5) 下图为学生信息界面

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	肇庆市	理工
8816590176	罗*9171	547	31108	006	007	016	011	019	018	广东	揭阳市	理工
8816590179	杜*6076	536	38839	019	011	023	028	029	022	广东	江门市	理工
8816590182	江*5541	535	39772	046	011	038	035	045	023	广东	揭阳市	理工
8816590185	刘*6436	541	35463	006	018	005	012	029	023	广东	惠州市	理工
8816590188	李*6703	524	48631	023	028	029	011	008	007	广东	深圳市	理工
8816590191	郑*3485	525	47477	029	023	012	020	035	021	广东	深圳市	理工
8816590194	何*6254	521	51315	047	018	045	049	048	033	广东	潮州市	理工
8816590197	龙*2410	532	41854	018	003	015	043	046	024	广东	揭阳市	理工
8816590200	廖*4585	524	47821	034	035	042	043	039	033	广东	中山市	理工
8816590203	廖*2852	521	51083	004	005	006	023	010	049	广东	潮州市	理工
8816590206	黄*1612	553	27500	004	001	005	011	023	022	广东	江门市	理工
8816590209	黎*5840	552	28313	017	016	018	006	019	033	广东	惠州市	理工
8816590212	张*7739	537	38158	017	019	007	004	029	023	广东	韶关市	理工
8816590215	廖*2691	555	26208	004	001	016	007	005	018	广东	东莞市	理工
8816590218	徐*1304	538	37064	004	001	005	011	023	028	广东	揭阳市	理工
8816590221	黄*6381	535	39705	005	022	001	003	020	024	广东	江门市	理工
8816590224	彭*5143	525	47123	033	043	032	035	034	040	广东	东莞市	理工
8816590227	陈*1346	557	25451	007	004	017	016	001	039	广东	中山市	理工
8816590230	廖*4634	535	47367	011	036	033	030	031	010	广东	中山市	理工

图 3.2.5 学生信息界面

6) 下图为录取结果界面

“导入文件”的控制器为“FileController”；当用户发出上传文件请求时，该请求将会被转发至控制器“FileController”，再由控制器调用“ExcelServiceImpl”服务类，对“导入文件”该业务进行处理。在“ExcelServiceImpl”服务类中，一共封装了三类处理Excel文件方法，分别是“读取专业录取计划”、“读取学生志愿信息”及“导出结果”，在处理“导入文件”请求时，主要用到前两类方法。该服务类处理“导入文件”请求时，以导入“专业录取计划”为例，其主要业务逻辑为：

- ① 获取当前所处流程，检查流程合法性；若当前录取流程处于“缺失专业录取计划”，则允许导入
- ② 调用数据访问层“majorMapper”，对数据库中“t_department”表内容进行清空
- ③ 读取“专业录取计划”Excel文件，调用“ReadMajorListener”类，首先实例化“Department”及“ExcelMajor”实体类；而后调用数据访问层“DepartmentMapper”，对数据库中“t_department”表进行插入数据操作，将招生计划中所有“系信息”录入数据库；然后再使用数据访问层“departmentMapper”类，对“招生计划”数据导入
- ④ 针对不同情况，调用数据访问层“statusMapper”，对操作历史（流程状态）写入相应内容，如：导入专业招生计划文件、重新导入专业招生计划文件

(3) “录取”用例实现的设计方案

“录取”的控制器为“StudentController”；当用户发出录取请求时，该请求将会被转发至控制器“StudentController”，再由控制器调用“StudentServiceImpl”服务类中的“doEnroll”方法，对“录取”该业务进行处理，该类中“doEnroll”方法使用了“重载”特性，即根据传入的参数不同，调用不同方法；第一种方法不需要传入参数，其用途为获取“招生计划”及“学生志愿信息”，不返回结果；第二种方法需要传入“专业”参数，其用途为检查该学生能否被学校录取，返回“Boolean”类型结果。在“StudentServiceImpl”服务类中，对于“录取”的业务逻辑如下：

- ① 获取当前所处流程，检查流程合法性；若当前录取流程处于“Ready”状态，则允许进行“录取”操作
- ② 获取招生计划，该步“StudentServiceImpl”服务类会调用“majorMapper”完成。
- ③ 进行录取操作，该操作单次对两百名学生进行操作，直至录取完成。在该操作中，按学生排位获取学生信息，而后交至同类下同名“doEnroll”方法进行处理，检查该生能否被志愿专业录取，而后将操作返回无参“doEnroll”方法，对录取信息进行写入数据库操作
- ④ 针对不同情况，调用数据访问层“statusMapper”，对操作历史（流程状态）写入相应内容，例如：录取完成

(4) “调剂”用例实现的设计方案

“调剂”的控制器为“StudentController”；当用户发出录取请求时，该请求将会被转发至控制器“StudentController”，再由控制器调用“StudentServiceImpl”服务类中的“doAdjust”方法，对“调剂”该业务进行处理。在“StudentServiceImpl”服务类中，对于“调剂”的业务逻辑如下：

- ① 获取当前所处流程，检查流程合法性；若当前录取流程处于“ENROLLED”录取完成状态，则允许进行“调剂”操作
- ② 获取专业调剂计划，该步“StudentServiceImpl”服务类会调用“majorMapper”完成，其 SQL 语句逻辑为检索招生计划“t_major”与学生志愿信息“t_student”，查找实际招生人数大于计划招生人数的专业以及其最低排位“bottom_rank”
- ③ 进行调剂操作，该操作单次对一百名学生进行操作，直至调剂完成。在该操作中，对“AcceptedTyp”为 0，即状态为可调剂的学生，按排位获取学生信息，而后顺序检索专业调剂计划，检查该生能否被未录满的专业录取，然后对录取信息进行写入数据库操作
- ④ 针对不同情况，调用数据访问层“statusMapper”，对操作历史（流程状态）写入相应内容，例如：调剂完成

(5) “导出结果”用例实现的设计方案

“导出结果”的控制器为“FileController”；当用户发出上传文件请求时，该请求将会被转发至控制器“FileController”，再由控制器调用“ExcelServiceImpl”服务类，对“导出结果”该业务进行处理。该服务类处理“导出结果”请求时，其主要业务逻辑为：

- ① 获取当前所处流程，检查流程合法性；若当前录取流程处于“调剂已完成”，则允许导出结果
- ② 调用数据访问层“StudentMapper”，查询录取结果，其 SQL 语句逻辑为检索“t_student”，“t_major”及“t_department”三表，对上述三表进行自然连接，筛选“accepted_type”状态码为 1-7 的学生，展示特定字段，完成查询
- ③ 将查询结果写出至 Excel 文件，每次写出 200 行，直至全部写出完成

3.4 类设计

(1) 配置类

① EnrollConfig

完成 SpringBoot 框架中，自定义拦截器与跨域问题配置

其包含两种方法，分别为“addInterceptors(InterceptorRegistry registry)”与“addCorsMappings(CorsRegistry registry)”方法，分别对应拦截器配置与 Cors 配置。

“addInterceptors(InterceptorRegistry registry)”会对登录状态进行验证，只有登录后才能访问相应的控制器，否则跳转到登录页面。

“addCorsMappings(CorsRegistry registry)”则是对跨域访问进行配置，其允许所有路径、源、请求方法、头部设置等进行跨域访问

② LoginProperties 注入配置文件的管理员账户名及密码至该类

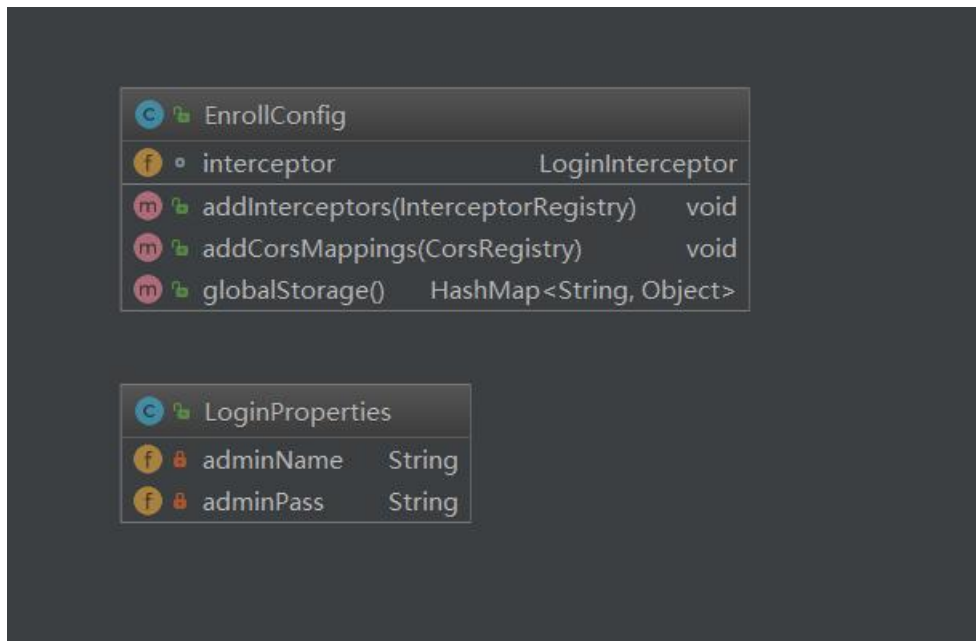


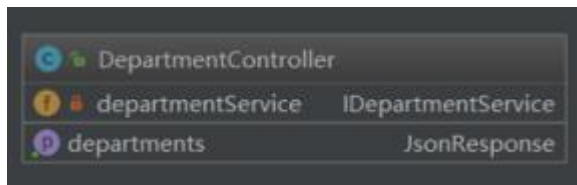
图 3.4.1 EnrollConfig、LoginProperties 类图

(2) 控制器

① DepartmentController （学院信息控制器）

方法：getDepartments 获取学院信息

可调用“DepartmentService”服务类，完成系信息相关业务处理



② FileController （文件控制器）

方法：

“uploadMajorExcel” 上传招生计划

“uploadStudentExcel” 上传学生志愿信息

“export” 导出结果

可调用“ExcelService”服务类，完成文件相关业务处理

FileController	
excelService	IExcelService
uploadMajorExcel(MultipartFile)	JsonResponse
uploadStudentExcel(MultipartFile)	JsonResponse
export(HttpServletResponse)	void

③ LoginController （登录控制器）

成员变量: LoginProperties (管理员账户及密码配置)、Map<String, Object> (目前登录状态)

方法:

“doLogin” (登录) 验证管理员用户名及密码, 若验证成功向登录状态 storage 写入 session 对象, 值为 “authSession”

“checkLogin” (检查登录状态) 验证登录状态 storage 中是否有名为 “authSession” 的 session 对象, 完成对登录状态的检查

“logout” (退出登录); 移除在登录状态 storage 中值为 “authSession” 的 session 对象

LoginController	
properties	LoginProperties
storage	Map<String, Object>
doLogin(String, String, HttpSession)	JsonResponse
checkLogin(HttpSession)	JsonResponse
logout()	JsonResponse

④ MajorController （招生计划控制器）

方法:

“getMajorPlan” 获取详细招生计划

“updateMajorPlan” 更新招生计划

“getMajors” 获取招生专业 id 与专业名

“getMajorsByDepartment” 根据学院 id 获取学院下专业
可调用 “MajorService” 服务类, 完成专业相关业务处理

MajorController	
majorService	IMajorService
updateMajorPlan(String, int)	JsonResponse
getMajorsByDepartment(int)	JsonResponse
majors	JsonResponse
majorPlan	JsonResponse

⑤ StatusController（操作历史<流程>控制器）

方法：

“getStatus” 获取当前所处流程

“getLogList” 获取操作历史

可调用“StatusService”服务类，完成操作历史（流程）相关业务处理

⑥ StudentController（学生志愿信息控制器）

方法：

“getStudentRaw” 获取学生志愿信息

“getAdjustStudentRaw” 获取处于调剂队列学生信息

“getExitStudentRaw” 获取被退档学生信息

“doEnroll” 执行录取操作

“doAdjust” 执行调剂操作

“getResult” 获取录取结果

“getResultByDepartment” 根据学院查询录取结果

“getResultByMajor” 根据专业查询录取结果

“searchStudent” 搜索学生信息

“searchStudentByCandidate” 根据准考证号查询学生信息

“getStudentBeforeRank” 根据排名查询学生信息

“getStatisticsResult” 获取录取结果统计信息

“getStatisticsResultInDepartment” 根据学院获取录取结果统计信息

“getStatisticsResultInMajor”根据专业获取录取结果统计信息

“getDistribute”获取录取学生地域分布信息

“getDistributeInProvince”根据省份获取录取学生地域分布信息

“getGradeDistribute”获取录取学生分数段分布信息

“getGradeDistributeByDepartment”根据学院获取录取学生分数段分布信息

“getGradeDistributeByMajor”根据专业获取录取学生分数段分布信息

“getCountDistributeInDepartment”根据学院查询该学院录取人数

“getCountDistributeInMajor”根据专业查询该学院录取人数

“getCountDistributeInMajorByDepartment”根据学院查询该学院下专业录取人数

“reset”清空学生志愿信息表

“formallyReady”检查录取前流程合法性

可调用“StudentService”服务类，完成学生志愿信息、录取信息相关业务处理

StudentController	
studentService	IStudentService
getStudentRaw(Integer)	JsonResponse
getAdjustStudentRaw(int)	JsonResponse
getExitStudentRaw(int)	JsonResponse
doEnroll()	JsonResponse
doAdjust()	JsonResponse
getResult(int, boolean, QueryResultOption)	JsonResponse
getResultByDepartment(int, int, boolean)	JsonResponse
getResultByMajor(String, int, boolean)	JsonResponse
searchStudent(int, String)	JsonResponse
searchStudentByCandidate(int, String)	JsonResponse
getStudentBeforeRank(int, int)	JsonResponse
getDistributeInProvince(String)	JsonResponse
getGradeDistributeByDepartment(int)	JsonResponse
getGradeDistributeByMajor(String)	JsonResponse
getCountDistributeInMajorByDepartment(int)	JsonResponse
reset()	JsonResponse
formalReady()	JsonResponse
statisticsResultInDepartment	JsonResponse
countDistributeInMajor	JsonResponse
countDistributeInDepartment	JsonResponse
statisticsResult	JsonResponse
distribute	JsonResponse
statisticsResultInMajor	JsonResponse
gradeDistribute	JsonResponse

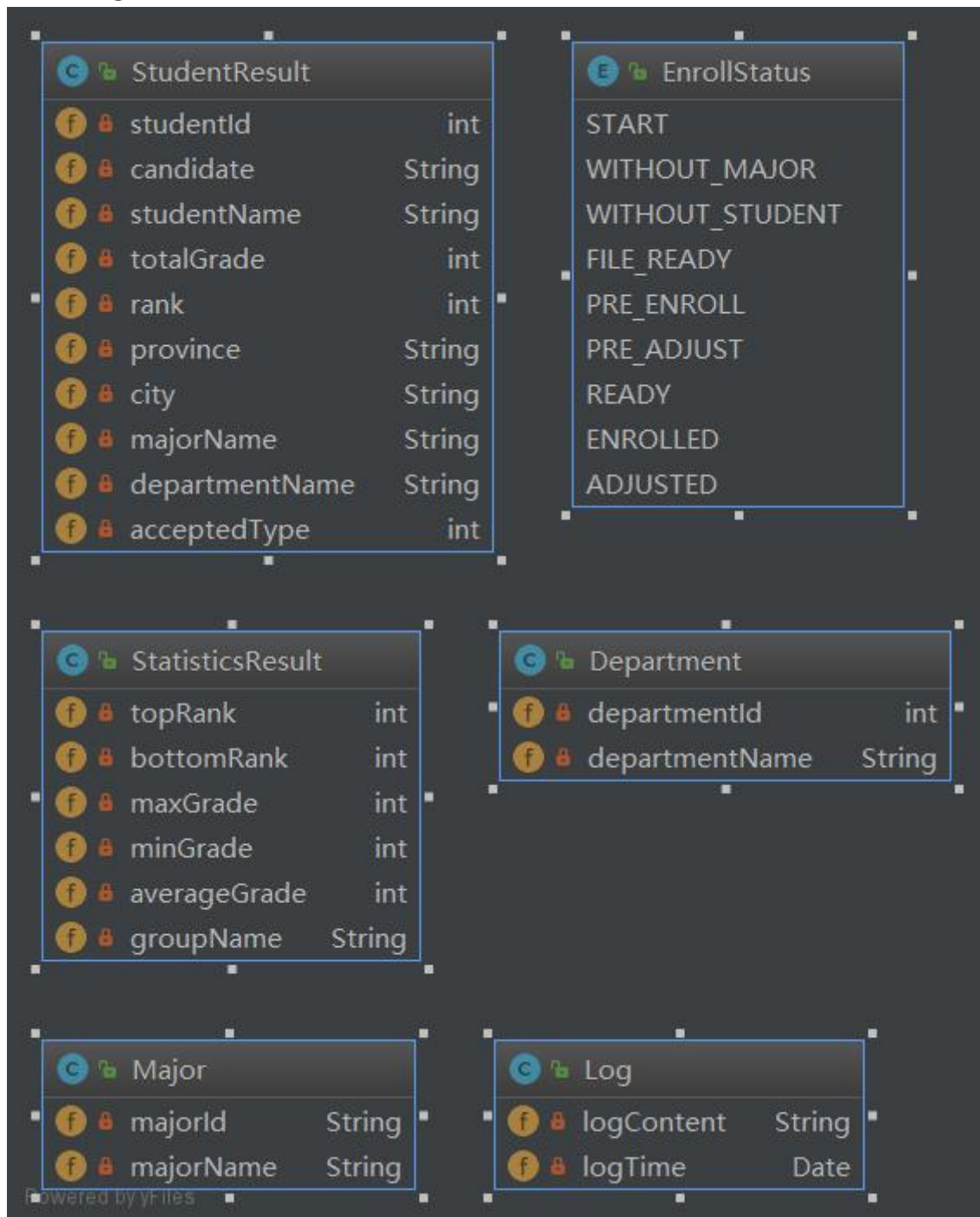
⑦ IndexViewConfig（默认首页控制器）

方法：addViewControllers 对 UriPath 为 “/” 进行转发，跳转至首页 “index.html”

(3) 实体类

- ① Department 学院
- ② EnrollStatus 流程状态
- ③ Log 操作历史内容
- ④ Major 招生专业 ID 及专业名（用于 SQL 语句查询返回类型）
- ⑤ StatisticsResult 统计结果（用于 SQL 语句查询返回类型）
- ⑥ StudentResult 录取结果（用于 SQL 语句查询返回类型）
- ⑦ ExcelMajor 招生计划

⑧ ExcelStudent 学生志愿信息



(4) 服务类

① ReadMajorListener (招生计划服务类)

成员变量:

List<ExcelMajor> list 招生计划列表 (私有全局变量)

int BATCH_COUNT 每批导入数据库招生计划数量 (私有全局)

变量)

MajorMapper majorMapper 招生计划 Mapper 类 (私有变量)

DepartmentMapper departmentMapper 学院信息 Mapper 类
(私有变量)

Map<String, Integer> departmentIds 学院信息列表 (私有变量)

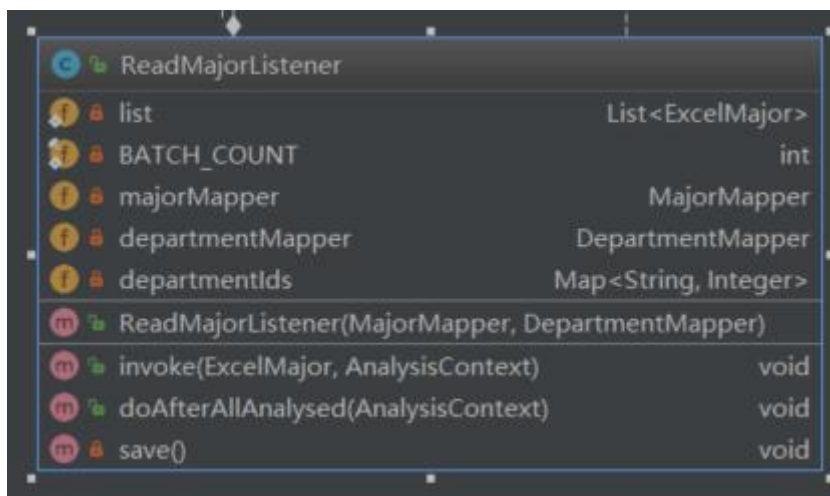
方法:

“ReadMajorListener” 构造方法

“Invoke” 自动保存, 当 List 容器中信息条数达指定值, 将其写入数据库

“doAfterAllAnalysed” 当 Excel 文件中所有数据被读取完成, 执行数据写入数据库及清空 list 操作

“save” 将 list、departmentIds 信息写入数据库



② ReadStudentListener (学生志愿信息服务类)

成员变量:

List<ExcelStudent> list 学生志愿信息列表 (私有全局变量)

int BATCH_COUNT 每批导入数据库志愿信息数量 (私有全局变量)

StudentMapper studentMapper 学生志愿信息 Mapper 类(私有变量)

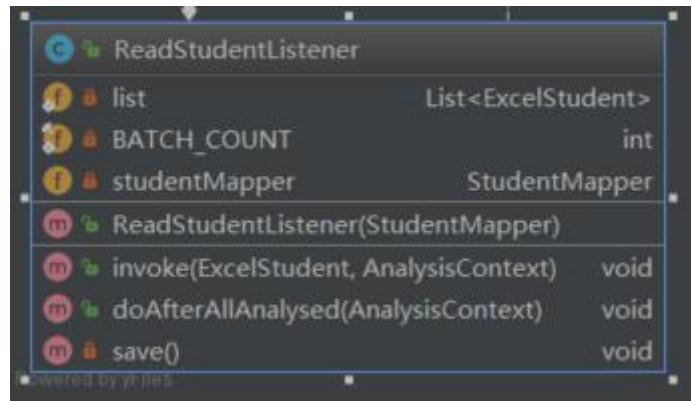
方法:

“ReadStudentListener” 构造方法

“Invoke” 自动保存, 当 List 容器中信息条数达指定值, 将其写入数据库

“doAfterAllAnalysed” 当 Excel 文件中所有数据被读取完成, 执行数据写入数据库及清空 list 操作

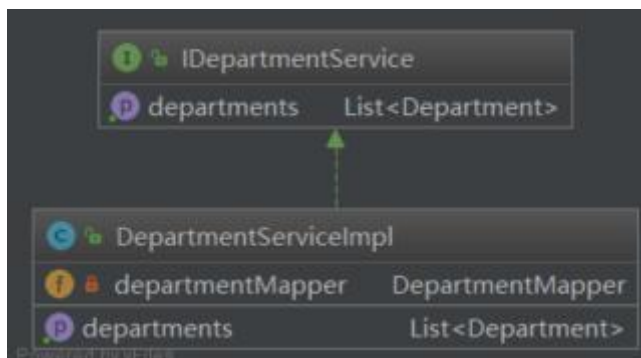
“save” 将志愿信息写入数据库



③ DepartmentServiceImpl (系信息服务类)

成员变量: DepartmentMapper Mapper 层接口

方法: “getDepartments” 调用 Mapper 层相关类, 获取学院信息



④ ExcelServiceImpl (Excel 文件服务类)

成员变量:

MajorMapper Mapper 层接口

DepartmentMapper Mapper 层接口

StudentMapper Mapper 层接口

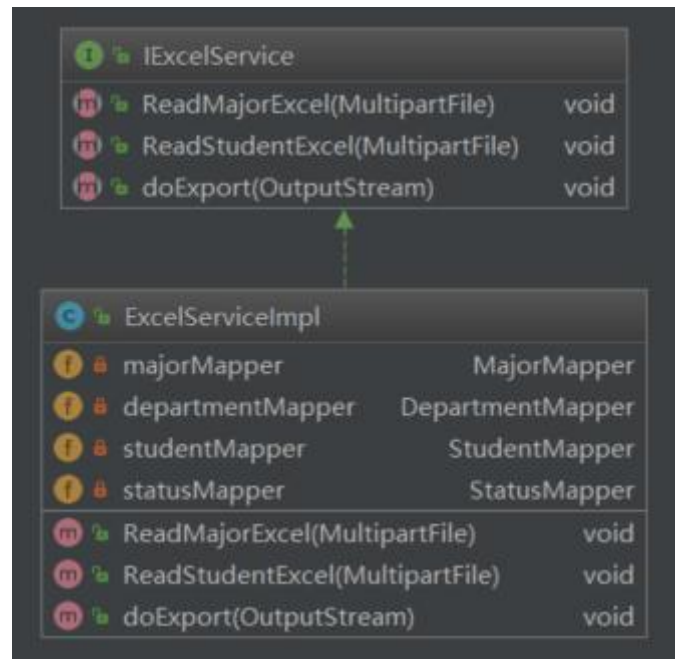
StatusMapper Mapper 层接口

方法:

“ReadMajorExcel” 导入招生计划

“ReadStudentExcel” 导入学生志愿信息

“doExport” 导出结果



⑤ MajorServiceImpl (招生计划服务类)

成员变量:

MajorMapper Mapper 层接口

StatusMapper Mapper 层接口

方法:

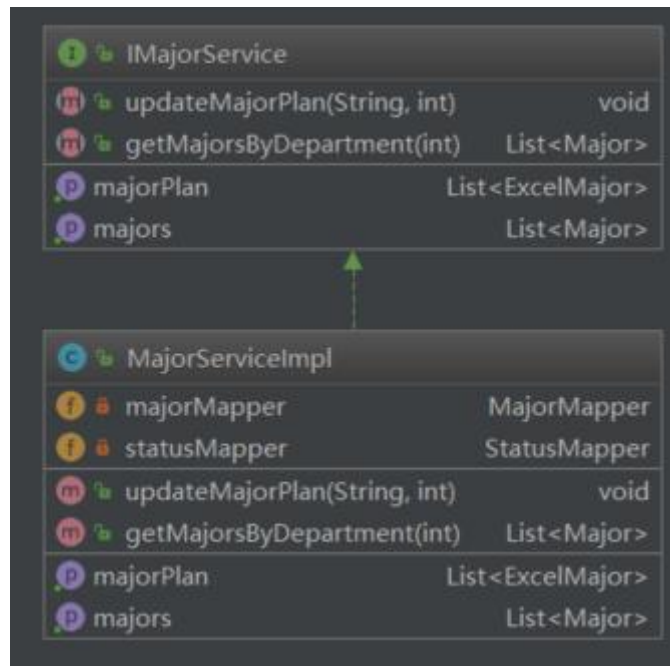
“updateMajorPlan” 调用 Mapper 层相关类, 更新招生计划

“getMajorPlan” 调用 Mapper 层相关类, 获取详细招生计划

“getMajors” 调用 Mapper 层相关类, 获取招生专业 id 与专业名

“getMajorsByDepartment” 调用 Mapper 层相关类, 根据学院

id 获取学院下专业



⑥ StatusServiceImpl （操作历史<流程>服务类）

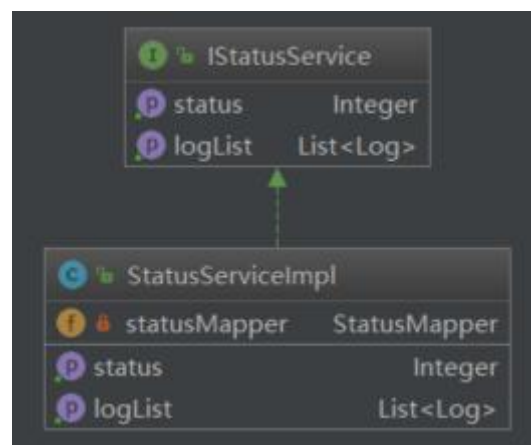
成员变量：

StatusMapper Mapper 层接口

方法：

“getStatus” 调用 Mapper 层相关类，获取当前所处流程

“getLogList” 调用 Mapper 层相关类，获取操作历史



⑦ StudentServiceImpl （学生志愿信息服务类）

成员变量：

MajorMapper Mapper 层接口

StudentMapper Mapper 层接口

StatusMapper Mapper 层接口

方法:

“getStudentRaw” 调用 Mapper 层相关类, 获取学生志愿信息

“getAdjustStudentRaw” 调用 Mapper 层相关类, 获取处于调剂
队列学生信息

“getExitStudentRaw” 调用 Mapper 层相关类, 获取被退档学生
信息

“doEnroll” 调用 Mapper 层相关类, 执行录取操作

“doAdjust” 调用 Mapper 层相关类, 执行调剂操作

“getResult” 调用 Mapper 层相关类, 获取录取结果

“getResultByDepartment” 调用 Mapper 层相关类, 根据学院查
询录取结果

“getResultByMajor” 调用 Mapper 层相关类, 根据专业查询录取
结果

“searchStudent” 调用 Mapper 层相关类, 搜索学生信息

“searchStudentByCandidate” 调用 Mapper 层相关类, 根据准考证号
查询学生信息

“getStudentBeforeRank” 调用 Mapper 层相关类, 根据排名查询
学生信息

“getStatisticsResult” 调用 Mapper 层相关类, 获取录取结果
统计信息

“getStatisticsResultInDepartment” 调用 Mapper 层相关类,
根据学院获取录取结果统计信息

“getStatisticsResultInMajor” 调用 Mapper 层相关类, 根据专
业获取录取结果统计信息

“getDistribute” 调用 Mapper 层相关类，获取录取学生地域分布信息

“getDistributeInProvince” 调用 Mapper 层相关类，根据省份获取录取学生地域分布信息

“getGradeDistribute” 调用 Mapper 层相关类，获取录取学生分数段分布信息

“getGradeDistributeByDepartment” 调用 Mapper 层相关类，根据学院获取录取学生分数段分布信息

“getGradeDistributeByMajor” 调用 Mapper 层相关类，根据专业获取录取学生分数段分布信息

“getCountDistributeInDepartment” 调用 Mapper 层相关类，根据学院查询该学院录取人数

“getCountDistributeInMajor” 调用 Mapper 层相关类，根据专业查询该学院录取人数

“getCountDistributeInMajorByDepartment” 调用 Mapper 层相关类，根据学院查询该学院下专业录取人数

“reset” 调用 Mapper 层相关类，清空学生志愿信息表

“formallyReady” 调用 Mapper 层相关类，检查录取前流程合法性

IStudentService		
getStudentRaw(int)		PageInfo
getAdjustStudentRaw(int)		PageInfo
getExitStudentRaw(int)		PageInfo
doEnroll()		void
doAdjust()		void
getResult(int, boolean, QueryResultOption)		PageInfo
getResultByDepartment(int, int, boolean)		PageInfo
getResultByMajor(String, int, boolean)		PageInfo
searchStudent(int, String)		PageInfo
searchStudentByCandidate(int, String)		PageInfo
getStudentBeforeRank(int, int)		PageInfo
getDistributeInProvince(String)		List<Map<String, Integer>>
getGradeDistributeByDepartment(int)		List<Map<String, Integer>>
getGradeDistributeByMajor(String)		List<Map<String, Integer>>
getCountDistributeInMajorByDepartment(int)		List<Map<String, Integer>>
reset()		void
formallyReady()		void
distribute		List<Map<String, Integer>>
countDistributeInDepartment		List<Map<String, Integer>>
statisticsResult		List<StatisticsResult>
statisticsResultInDepartment		List<StatisticsResult>
gradeDistribute		List<Map<String, Integer>>
statisticsResultInMajor		List<StatisticsResult>
countDistributeInMajor		List<Map<String, Integer>>
StudentServiceImpl		
studentMapper		StudentMapper
majorMapper		MajorMapper
statusMapper		StatusMapper
getStudentRaw(int)		PageInfo
getAdjustStudentRaw(int)		PageInfo
getExitStudentRaw(int)		PageInfo
doEnroll()		void
doEnroll(ExcelMajor)		boolean
updateSingleStudent(List<ExcelStudent>)		void
doAdjust()		void
getResult(int, boolean, QueryResultOption)		PageInfo
getResultByDepartment(int, int, boolean)		PageInfo
getResultByMajor(String, int, boolean)		PageInfo
searchStudent(int, String)		PageInfo
searchStudentByCandidate(int, String)		PageInfo
getStudentBeforeRank(int, int)		PageInfo
getDistributeInProvince(String)		List<Map<String, Integer>>
getGradeDistributeByDepartment(int)		List<Map<String, Integer>>
getGradeDistributeByMajor(String)		List<Map<String, Integer>>
getCountDistributeInMajorByDepartment(int)		List<Map<String, Integer>>
reset()		void
formallyReady()		void
distribute		List<Map<String, Integer>>
countDistributeInDepartment		List<Map<String, Integer>>
statisticsResult		List<StatisticsResult>
statisticsResultInDepartment		List<StatisticsResult>
gradeDistribute		List<Map<String, Integer>>
statisticsResultInMajor		List<StatisticsResult>
countDistributeInMajor		List<Map<String, Integer>>

(5) Mapper 层接口

① DepartmentMapper

- ② MajorMapper
- ③ StatusMapper
- ④ StudentMapper

(6) Mapper 层配置文件

- ① DepartmentMapper.xml
- ② MajorMapper.xml
- ③ StatusMapper.xml
- ④ StudentMapper.xml

(7) Filter 过滤器

CharacterEncodingFilter 拦截用户向服务器的请求以及服务器向用户的回应，对不符合编码要求进行过滤

3.5 数据设计

(1) 设计永久保存数据的数据库表及字段

<<table>>	
T_department	
<<key>>	department_id
	string[11]
	department_name
	string[20]

为了保存学院的信息，针对“平行志愿录取系统”中的 Department 类，需要为其设计表“T_department”，该表有两个字段：长度为 11 的字符串“department_id”表示学院 id，长度为 20 的字符串“department_name”表示学院名信息。

<<table>> t_log
<<key>>log_id int log_content varchar[100] log_time datetime status tinyint

为了保存操作历史的信息，针对“平行志愿录取系统”中的 Log 类,需要为其设计表“t_log”，该表有四个字段：类型为整数的“log_id”表示 log 的 id, 长度为 100 的字符串“log_content”表示操作内容, 类型为“datetime”的“log_time”表示操作时间，类型为“tinyint”的 status 表示操作状态。

<<table>> t_major
<<key>>major_id varchar[5] major_code varchar[10] department_id int major_name varchar[20] comment varchar[255] period varchar[255] plan_student_count int realistic_student_count int

为了保存招生计划的信息，针对“平行志愿录取系统”中的 ExcelMajor 类,需要为其设计表“t_major”，该表有八个字段，major_id, major_code , department_id, major_name, comment, period, plan_student_count , realistic_student_count 分别表示专业的 id, 专业编码，学院 id, 专业名称，专业类型，学年，计划招生人数，真实招生人数。

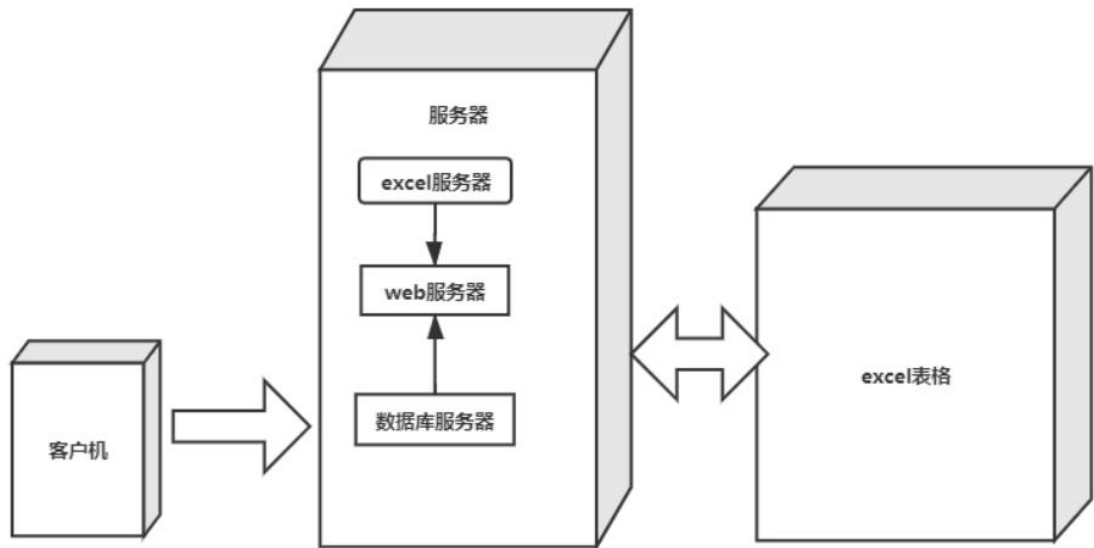
<<table>> t_student	
<<key>>student_id int candidate varchar[12] student_name varchar[255] total_grade int will1 varchar[5] will2 varchar[5] will3 varchar[5] will4 varchar[5] will5 varchar[5] will6 varchar[5] adjust tinyint[4] rank int province varchar[20] city varchar[20] subject_type varchar[20] accepted_major_id varchar[5] accepted_type tinyint[4]	

为了保存学生信息，针对“平行志愿录取系统”中的 ExcelStudent 类，需要为其设计表“t_student”，该表有十七个字段，按顺序分别表示学生 id，准考证号，姓名，总分，志愿 1，志愿 2，志愿 3，志愿 4，志愿 5，志愿 6，调剂，排位，省份，城市，科类，最终接受专业的 id，接受的方式。

(2) 设计永久数据的操作

为了支持对“T_department”、“t_log”、“t_major”、“t_student”四个数据库表的操作，设计模型中有相应四个关键设计接口“DepartmentMapper”，“StatusMapper”，“MajorMapper”，“StudentMapper”，它们提供了一组方法以实现对数据的操作。

3.6 部署设计



本软件工程从部署和运行方面主要分为客户端、服务器以及 Excel 表格。

首先，客户端为 web 客户端。用户可以在网站上登陆账户，并且通过简介的界面一步步导入数据，处理数据，自动实现不同的功能。用户只需要进行几步简单的操作，就能够获得处理好的数据。Web 客户端利用 Vue.js 来进行网页的 html 语言的编程，生成对应的网页，界面风格简洁，用户操作简单。。

其次，服务器方面主要分为应用服务器，web 服务器、excel 服务器以及数据库服务器。web 客户端通过 web 服务器与应用服务器相连接，应用客户端直接与应用服务器相连接。数据库服务器与应用服务器通过 mybatis 相连接，excel 服务器通过 excel 服务器与应用服务器相连接。其中应用服务器是连接客户端和外部设备的桥梁，在整个软件开发中起到至关重要的作用。

最后，软件的 Excel 表格分为大学的招生计划，以及平行志愿考生的数据两部分。需要用户自行提供。其中表格样式需要与模板一致，拥有相同的列名以及格式。