**NJWU选课系统**

**(Course Selection System)**

**体系结构设计文档**



**南京大学软件学院Aurora小组**

2013-10-8

# 目录 目录有链接、还有导航的哟~

[更新历史](#更新历史)·······················································································································································

[1.引言](#引言)···························································································································································

[1.1编制目的](#编制目的)······································································································································

[1.2词汇表](#词汇表)···········································································································································

[1.3参考资料](#参考资料)······································································································································

1. [产品概述](#产品概述)···················································································································································
2. [逻辑视角](#逻辑视角)···················································································································································
3. [组合视角](#组合视角)···················································································································································

[4.1开发包图](#开发包图)······································································································································

[4.2运行时进程](#运行时进程)···································································································································

[4.3物理部署](#物理部署)······································································································································

[4.4构件图](#构件图)··········································································································································

1. [接口视角](#接口视角)···················································································································································

[5.1模块的职责](#模块的职责)··································································································································

[5.2用户界面层的分解](#用户界面层的分解)·····················································································································

[5.3业务逻辑层的分解](#业务逻辑层的分解)······················································································································

[5.4数据层的分解](#数据层的分解)·······························································································································

[5.4.1数据层模块的职责](#数据层模块的职责)··········································································································

[5.4.2 数据层模块的接口规范](#数据层模块的接口规范)································································································

5.5模块的关键类图···························································································································

[6.信息视角](#信息视角)····················································································································································

[6.1数据持久化对象](#数据持久化对象)··························································································································

[6.2Txt持久化格式](#Txt持久化格式)·····························································································································

# 更新历史

|  |  |  |  |
| --- | --- | --- | --- |
| 修改人员 | 日期 | 变更原因 | 版本号 |
| 陈姿丽 | 2013-10-8 | 最初草稿（创建文档模板） | V1.0 |
| 刘璟 | 2013-10-9 | 添加组合视角模块内容 | V1.0.1 |
| 陈姿丽 | 2013-10-9 | 添加接口视角模块内容（不包括业务逻辑层的分解、数据层的分解） | V1.0.2 |
| 刘璟 | 2013-10-10 | 添加userbl、framebl、coursebl、staticbl、planbl的模块接口规范 | V1.0.3 |
| 陈姿丽 | 2013-10-10 | 补充coursebl、planbl的模块接口规范，添加selection的模块接口规范 | V1.0.4 |
| 佘昀峰 | 2013-10-10 | 补充coursebl、staticbl、selection的模块接口规范，添加gradebl的接口规范 | V1.0.5 |
| 任天 | 2013-10-10 | 补充userbl、staticbl、gradebl的模块接口规范 | V1.0.6 |
| 陈姿丽 | 2013-10-20 | 评审会议后,增加staticbl与coursebl模块的关键类图,修改course.txt | V1.1 |

# 1.引言

## 1.1编制目的

本报告详细完成对NJWU选课系统CSS的概要设计，达到指导详细设计和开放的目的，同时实现和测试人员及用户的沟通

## 1.2词汇表

|  |  |  |
| --- | --- | --- |
| 词汇名称 | 词汇含义 | 备注 |
| CSS | Course selection system |  |
| \_ui | 表示某展示层 |  |
| \_bl | 表示某逻辑层 |  |
| \_data | 表示某数据层 |  |

## 1.3参考资料

1.IEEE std 1471-2000

2.AURORA小组，《NJWU选课系统CSS需求规格说明书V1.2》

3.丁二玉，刘钦.计算与软件工程（卷二）[M]机械工业出版社，2012：134—182

4.Frank Buschmann, Regine Meunier, Hans Rohnert, Peter Sommerlad, Micheael Stal.Pattern-Oriented Software Architecture Volume 1: A system of Patterns [M]机械工业出版社,2003

5.Christopher　Fox.软件工程设计导论：过程、原理与模式（UML2.0版）[M]清华大学出版社，2007

# 2.产品概述

NJWU选课系统是一个在线教务选课系统，旨在提供各种功能来简化教务员，老师，学生操作流程的应用程序。既极减少了教务员管理课程和师生时机械，重复的工作，也为老师和学生在处理发布课程和选择课程等任务时提供了更方便，友好的体验，从而全面提高了工作效率和用户的满意程度。

# 3.逻辑视角

NJWU选课系统CSS中，选择了分层体系结构风格，将系统分为3层（展示层、业务逻辑层、数据层）能够很好地示意整个高层抽象。展示层包含GUI页面的实现，业务逻辑层包含业务逻辑处理的实现，数据层负责数据的持久化和访问。分层体系结构的逻辑视角和逻辑设计方案如图3-1和如图3-2所示。



图3-1 参照体系结构风格的包图表达逻辑视角

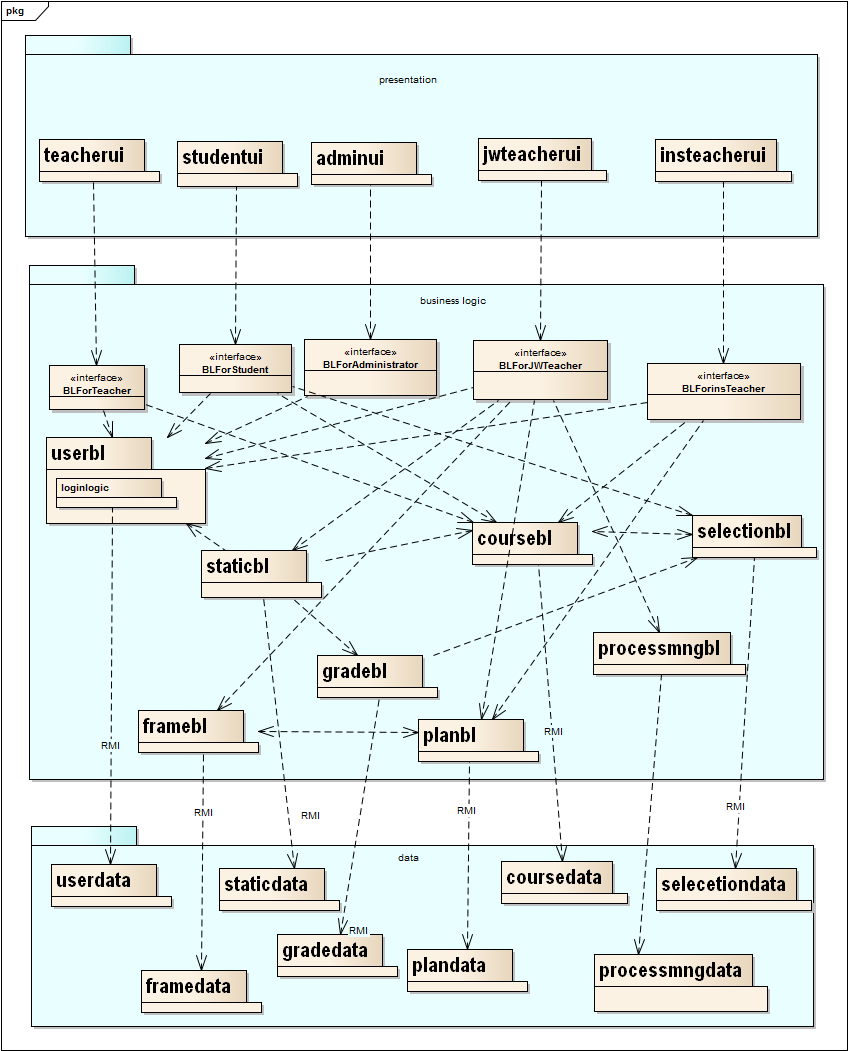


图3-2 软件体系结构逻辑设计方案

# 4.组合视角

## 4.1开发包图

NJWU选课系统CSS的最终开发包设计如表4.1-1

表4.1-1 NJWU选课系统CSS的最终开发包设计

|  |  |
| --- | --- |
| 开发包（物理） | 依赖的其他开发包 |
| mainui | Userui,frameui,sataicui,courseui,gradeui,planui  Selectionui |
| Userui | useBlservice,界面类库包，vo |
| Userblservice |  |
| Userbl | useBlservice, usedataservice,po |
| Userdataservice | Java RMI,po |
| Userdata | Txtutility,po, usedataservice |
| frameUi | useBlservice,界面类库包，vo |
| frameBlservice |  |
| frameBl | frameBlservice, framedataservice,po |
| frameDataservice | Java RMI,po |
| framedata | Txtutility,po, framedataservice |
| staticUi | frameBlservice,界面类库包，vo |
| staticBlservice |  |
| staticBl | staticBlservice, staticdataservice,po |
| staticDataservice | Java RMI,po |
| staticdata | Txtutility,po, staticdataservice |
| courseUi | courseBlservice,界面类库包，vo |
| courseBlservice |  |
| courseBl | courseBlservice, coursedataservice,po |
| courseDataservice | Java RMI,po |
| coursedata | Txtutility,po, gradedataservice |
| gradeUi | gradeBlservice,界面类库包，vo |
| gradeBlservice |  |
| gradeBl | gradeBlservice, gradedataservice,po |
| gradeDataservice | Java RMI,po |
| gradedata | Txtutility,po, gradedataservice |
| planUi | planBlservice,界面类库包，vo |
| planBlservice |  |
| planBl | planBlservice,dataservice,po |
| planDataservice | Java RMI,po |
| Plandata | Txtutility,po, plandataservice |
| selectionUi | selectionBlservice,界面类库包，vo |
| selectionBlservice |  |
| selectionBl | selectionBlservice, selectiondataservice,po |
| selectionDataservice | Java RMI,po |
| selectiondata | Txtutility,po, selectiondataservice |
| Vo |  |
| Po |  |
| Uyilitybl |  |
| 界面类库包 | Java.swing,java.awt,java.2D,java.3D |
| javaRMI |  |
| txtutility | Java.io |

NJWU选课系统客户端开发包图如图4.1-1所示,服务器端开发包图如图4.1-2所示

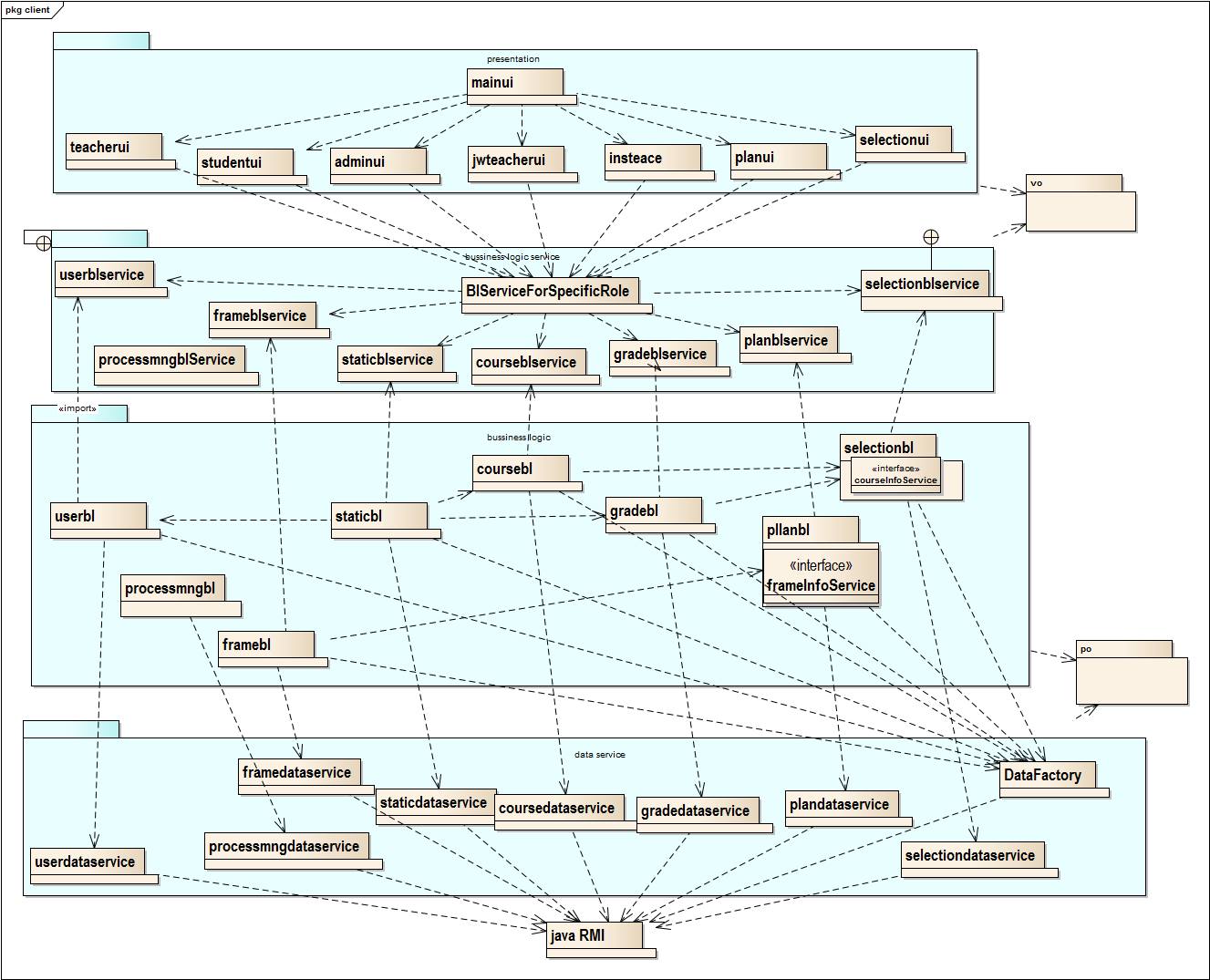


图4.1-1 NJWU选课系统CSS客户端开发包图

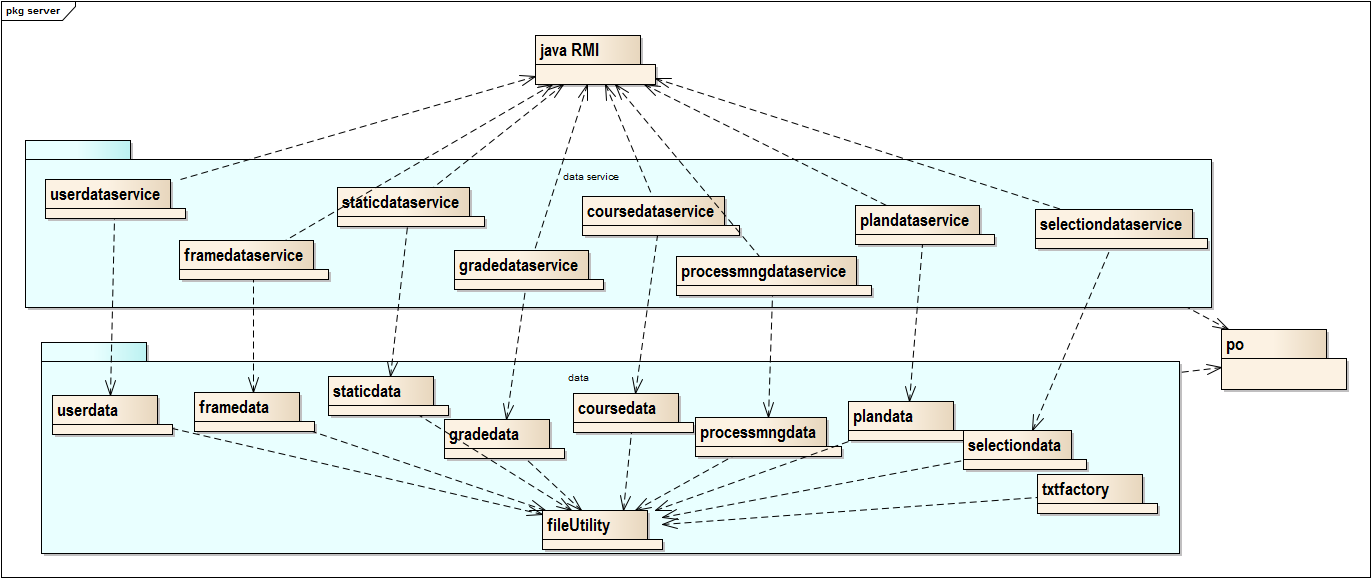


图4.1-2 NJWU选课系统CSS服务器端开发包图

## 4.2运行时进程

在NJWU选课系统CSS中,会有多个客户端进程和一个服务器端进程,其进程图如图4.2所示。结合部署图,客户端进程是在客户端机器上运行,服务器端进程在服务器端机器上运行。



图4.2 进程图

## 4.3物理部署

NJWU选课系统CSS中客户端构件是放在客户端机器上，服务器端构件是放在服务器端机器上。在客户端节点上，还要部署RMIStub构件。由于JavaRMI构件属于JDK6.0的一部分。所以，在系统JDK环境已经设置好的情况下，不需要再独立部署。部署图如图4.3所示。

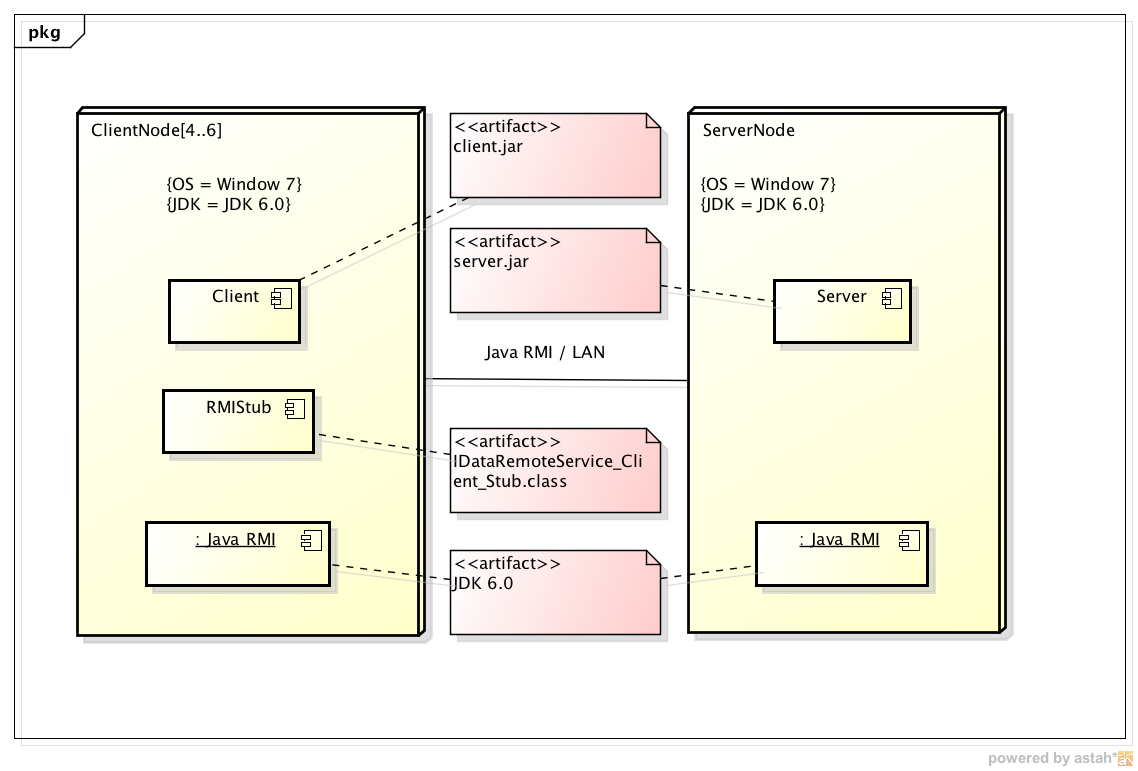


图4.3 部署图

## 4.4构件图

NJWU选课系统CSS有3个顶级构件：客户端、服务器端、以及负责连接两端的网络构件。从体系结构的角度，客户端和服务器端是构件（Component）；而网络构件是连接件（Connector）。每个顶层构件还可以进一步再细分出部分。

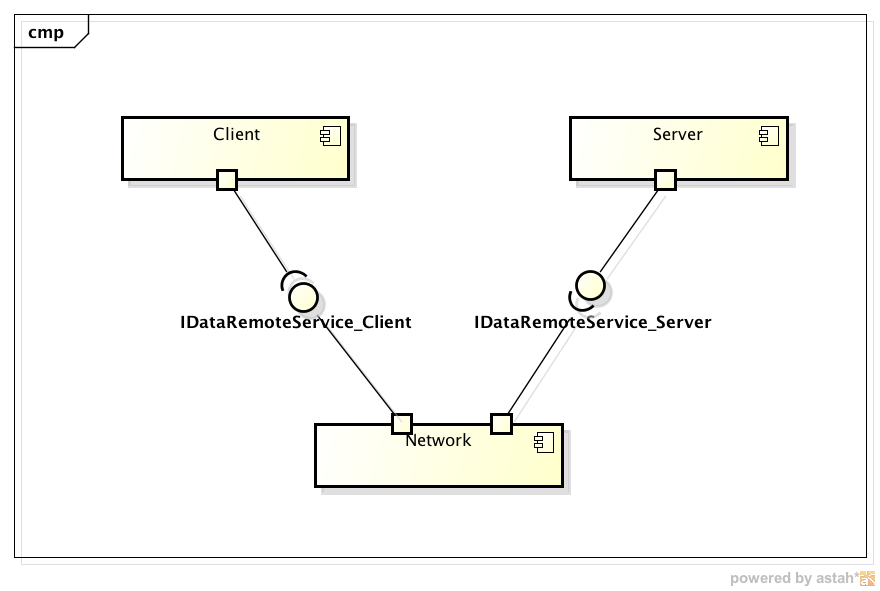


图4.4-1 0层构件图

1层构件图主要进一步细化了0层构件。这时候，需要考虑到逻辑视角的映射，通过1层构件图将逻辑视角的分层映射到0层构件图中的构件中去。逻辑视角的各层可能会分别部署在不同的物理机器上。

NJWU选课系统CSS，数据层是放在服务器端的。因为不同的用户（教务处老师、院系教务老师、任课老师、学生、管理员）会在客户端和服务器端分别完成相应的不同操作，所以客户端和服务器端都有各自的展示层和业务逻辑层。不同的是，客户端需要通过网络构件连接数据层的IDataRemoteService接口，而服务器端则是业务逻辑层直接访问IDataService接口。对于Network构建，我们也可以细化。由于使用了RMI技术，所以客户端需要有Stub类。Stub类和服务器端的需接口都共同依赖与Java RMI构件。

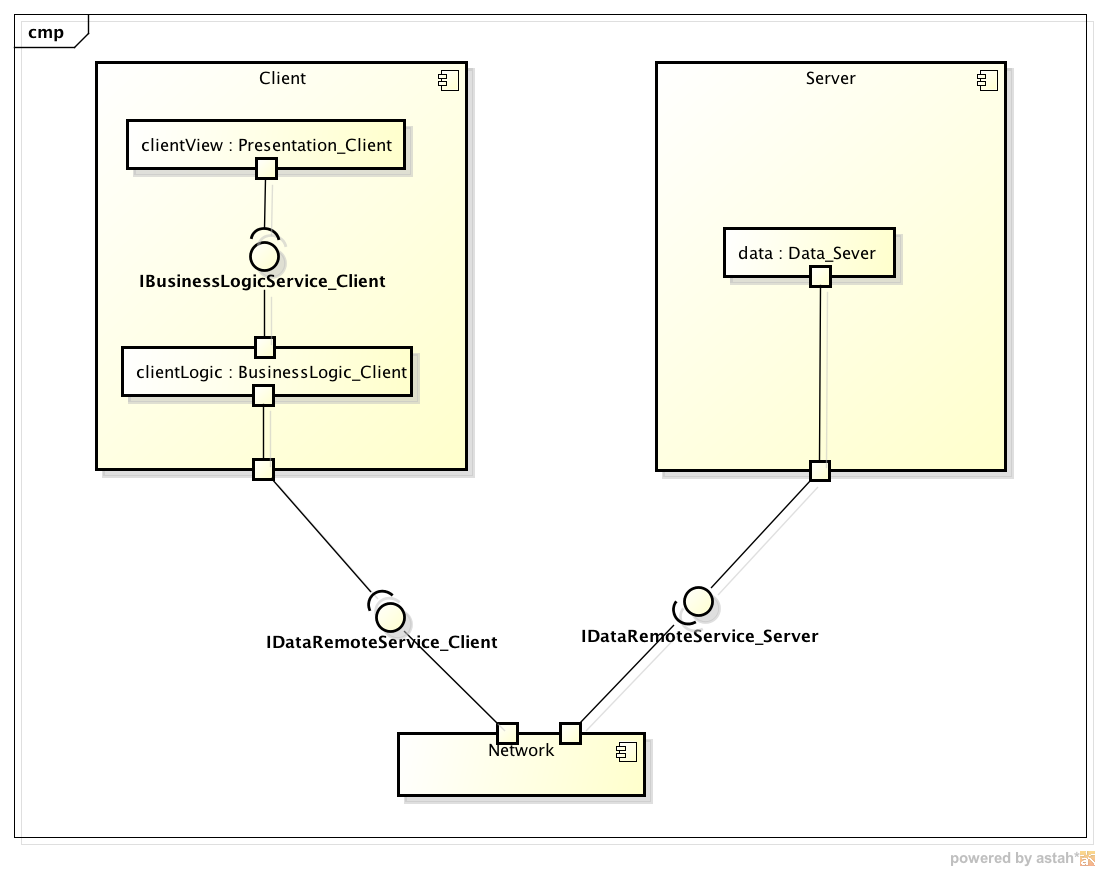


图4.4-2 1层构件图



图4.4-3 Network构件图

# 5.接口视角

## 5.1模块的职责

客户端模块和服务器端模块试图分别如图5.1-1和图5.1-2所示。客户端各层和服务器端各层的职责分别如表5.1-1和表5.1-2所示。



图5.1-1 客户端模块视图



图5.1-2 服务器端模块视图

表5.1-1 客户端各层的职责

|  |  |
| --- | --- |
| 层 | 职责 |
| 启动模块 | 负责初始化网络通信机制，启动用户界面。 |
| 用户界面层 | 基于窗口的NJWU选课系统CSS客户端用户界面。 |
| 业务逻辑层 | 对于用户界面的输入响应和业务处理逻辑。 |
| 客户端网络模块 | 利用Java RMI机制查找RMI服务 |

表5.1-2 服务器端各层的职责

|  |  |
| --- | --- |
| 层 | 职责 |
| 启动模块 | 负责初始化网络通信机制，启动用户界面。 |
| 用户界面层 | 基于窗口的NJWU选课系统CSS服务器用户界面。 |
| 业务逻辑层 | 对于用户界面的输入响应和业务处理逻辑。 |
| 数据层 | 负责数据的持久化及数据访问接口。 |
| 服务器端网络模块 | 利用Java RMI机制开启RMI服务，注册RMI服务 |

每一层只是使用下方直接接触的层。层与层之间仅仅是通过接口的调用来完成的。层之间调用的接口如表5.1-3所示。

表5.1-3 层之间调用的接口

|  |  |  |
| --- | --- | --- |
| 接口 | 服务调用方 | 服务提供方 |
| UserBLService  FrameBLService  StaticBLService  CourseBLService  GradeBLService  PlanBLService  SlectionBLService | 客户端展示层 | 客户端业务逻辑层 |
| UserDataService  FrameDataService  StaticDataService  CourseDataService  GradeDataLService  PlanDataService  SlectionDataService  DataFactory | 客户端业务逻辑层 | 服务器端数据层 |

借用发布整体框架策略用例来说明层之间的调用，如图5.1-3所示。每一层之间都是由上层依赖了一个接口（需接口），而下层实现这个接口（供接口）。FrameBLService提供了框架界面所需要的所有业务逻辑功能。FrameDataService提供了对数据库的增、删、改、查等操作。这样的实现就大大降低了层与层之间的耦合。

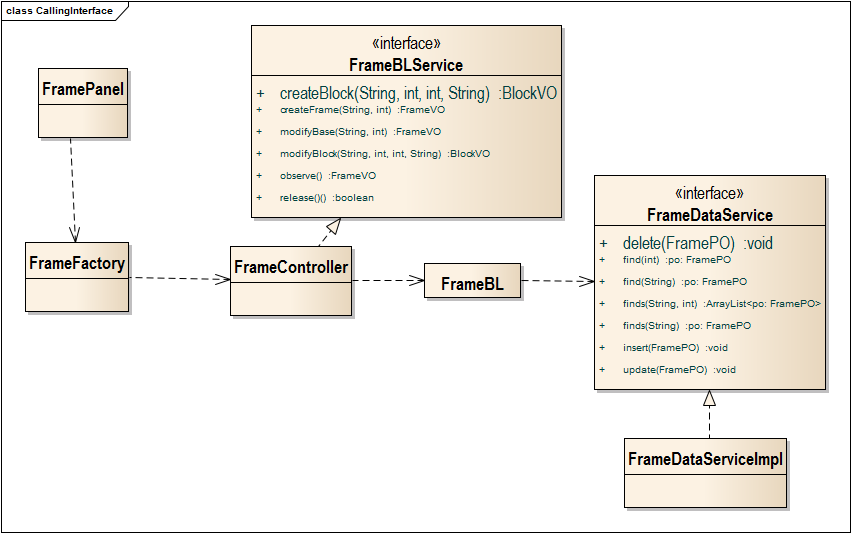


图5.1-3 发布整体框架用例层之间调用的接口

## 5.2用户界面层的分解

根据需求，系统存在24个用户界面：登陆界面、教务处老师主界面、院系教务老师主界面、任课老师主界面、学生主界面、管理员主界面、课程发布界面、通识课界面、通修课界面、统计信息界面、查看教学计划界面、发布整体框架界面、用户管理界面、查看学生界面、课程信息界面、录入成绩界面、选课界面、退选界面、补选界面、查看成绩界面、选择课程界面、录入计划界面、登记课程界面、查看课程界面。界面跳转如图5.2-1所示。

查看学生

课程信息

录入成绩

任课老师主界面

登陆界面

院系教务老师主界面

查看课程

录入计划

登记课程

通修课

统计信息

课程发布

通识课

发布框架

教务处老师主界面

查看计划

管理员主界面

用户管理

选择课程

退选

补选

学生主界面

选课

查看成绩

图5.2-1 用户界面跳转

服务器端和客户端的用户界面设计接口是一致的，只是具体的页面不一样。用户界面类如图5.2-2所示。

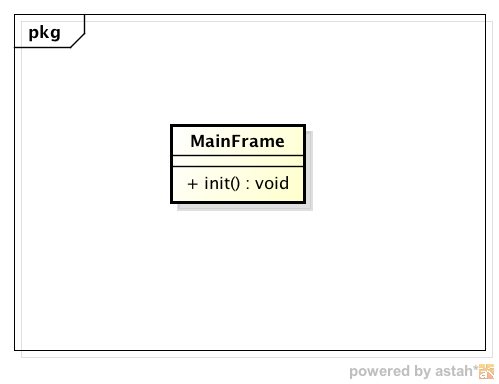


图5.2-2 用户界面类

### 5.2.1用户界面层模块的职责

如表5.2.1-1所示为用户界面层模块的职责。

表5.2.1-1 用户界面层模块的职责

|  |  |
| --- | --- |
| 模块 | 职责 |
| MainFrame | 界面Frame，负责界面的显示和界面的跳转。 |

### 5.2.2用户界面层模块的接口规范

用户界面层模块的接口规范如表5.2.2-1所示。

表5.2.2-1 用户界面层模块的接口规范

|  |  |  |
| --- | --- | --- |
| MainFrame | 语法 | Init(args[]) |
| 前置条件 | 无 |
| 后置条件 | 显示Frame以及LoginPanel |

用户界面层需要的服务接口如表5.2.2-2所示。

表5.2.2-2 用户界面层模块需要的服务接口

|  |  |
| --- | --- |
| 服务名 | 服务 |
| businesslogicservice.LoginBLService | 登陆界面的业务逻辑接口 |
| businesslogicservice.\*BLService | 每个界面都有一个相应的业务逻辑接口 |

### 5.2.3用户界面模块设计原理

用户界面利用Java的Swing和AWT库来实现。

## 5.3业务逻辑层的分解

业务逻辑层包括多个针对界面的业务逻辑处理对象。例如，User对象负责处理登陆界面的业务逻辑；Selection对象负责处理选择课程的业务逻辑。业务逻辑层的设计如图5.3-1所示。

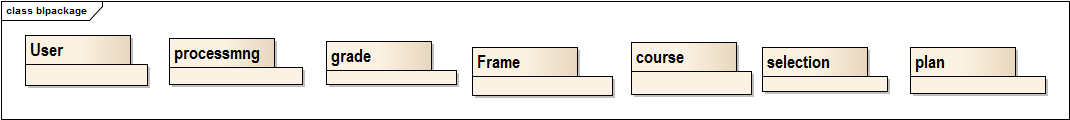


图5.3-1 业务逻辑层

### 5.3.1业务逻辑层模块的职责

业务逻辑层模块的职责如表5.3.1-1所示。

表5.3.1-1 业务逻辑层模块的职责

|  |  |
| --- | --- |
| 模块 | 职责 |
| userbl | 负责实现对应与登陆界面所需要的服务 |
| framebl | 负责实现发布框架界面所需要的服务 |
| staticbl | 负责实现统计信息界面所需要的服务 |
| coursebl | 负责实现课程发布界面与课程信息界面所需要的服务 |
| gradebl | 负责实现查看成绩界面和录入成绩界面所需要的服务 |
| planbl | 负责实现发布计划界面和查看计划界面所需要的服务 |
| selectionbl | 负责选择课程界面所需要的服务 |
| processmngbl | 负责处理与教学进度相关的服务 |

### 5.3.2业务逻辑层模块的接口规范

表5.3.2-1 userbl模块的接口规范

|  |  |  |  |
| --- | --- | --- | --- |
| 提供的服务（供接口） | | | |
| User.login | 语法 | | public RM login(long id,String password); |
| 前置条件 | | password符合输入规则 |
| 后置条件 | | 查找是否存在相应的User，根据输入的password返回登陆验证的结果 |
| User.addUser | 语法 | | public int addUsesr(String file) |
| 前置条件 | | 用户使用文件保存用户的信息 |
| 后置条件 | | 将文件的用户导入，创建相关用户 |
| User.ChangeUserInfo | 语法 | | public int changeUserInfo(String ID,UserVo vo) |
| 前置条件 | | 管理员输入了正确的修改的信息 |
| 后置条件 | | 修改相关用户信息 |
| User.deleteUser | 语法 | | public int deleteUser(String ID) |
| 前置条件 | | 管理员输入了合法的ＩＤ |
| 后置条件 | | 系统删除相关用户 |
| User.getUser | 语法 | | public ArrayList<UserVO> getUser(String type) |
| 前置条件 | | 用户使输入了合法的类型 |
| 后置条件 | | 用户查询相关类型的用户，并返回 |
| 需要的服务（需接口） | | | |
| 服务名 | | 服务 | |
| UserInfoDataService.find(String type) | | 根据用户类型进行查找单一持久化对象 | |
| UserInfoDataService.insert(ArrayList<UserInfoPO> pos) | | 插入单一持久化对象 | |
| UserInfoDataService.delete(UserInfoPO po) | | 删除单一持久化对象 | |
| UserInfoDataService.update(String ID,UserInfoPO po) | | 更新单一持久化对象 | |
| UserLoginDataService.find(UserLoginPO po) | | 根据登陆信息进行查找单一持久化对象 | |

表5.3.2-2 framebl模块的接口规范

|  |  |  |  |
| --- | --- | --- | --- |
| 提供的服务（供接口） | | | |
| Frame.createFrame | 语法 | public RM createFrame(int total,String description); | |
| 前置条件 | 启动发布框架 | |
| 后置条件 | 保存框架基本信息，返回框架信息 | |
| Frame.createBlock | 语法 | public RM createBlock(String name,int lower,int upper,String description) | |
| 前置条件 | 框架基本信息已输入 | |
| 后置条件 | 保存模块信息，返回模块信息 | |
| Frame.release | 语法 | public RM release(); | |
|  | 前置条件 | 框架信息已输入 | |
| 后置条件 | 框架信息设置为公开状态 | |
| Frame.modifyBase | 语法 | public RM modifyBase(FramePO fpo，int total,String description); | |
| 前置条件 | 框架已创建，启动修改框架 | |
| 后置条件 | 保存框架信息，返回新框架信息 | |
| Frame.modifyBlock | 语法 | public RM modifyBlock(FramePO fpo，String name,int lower,int upper,String description); | |
| 前置条件 | 模块存在，请求修改 | |
| 后置条件 | 保存模块信息，返回新模块信息 | |
| Frame.observe | 语法 | public RM observe(); | |
| 前置条件 | 框架已经设置为公开状态（非教务处老师）  框架已创建（教务员老师） | |
| 后置条件 | 无 | |
| 需要的服务（需接口） | | | |
| 服务名 | | | 服务 |
| FrameDataService.find() | | | 根据ID进行查找单一持久化对象 |
| FrameDataService.insert(FramePO fpo) | | | 插入单一持久化对象 |
| FrameDataService.delete(FramePO fpo) | | | 删除单一持久化对象 |
| FrameDataService.update(FramePO fpo) | | | 更新单一持久化对象 |

表5.3.2-3 staticsbl模块的接口规范

|  |  |  |  |
| --- | --- | --- | --- |
| 提供的服务（供接口） | | | |
| Statics.showTotal | 语法 | | Public StaticsVO show(); |
| 前置条件 | | 启动查看，选中查看主体 |
| 后置条件 | | 返回统计数据 |
| Statics.showIns | 语法 | | Public StaticsVO show(String department); |
| 前置条件 | | 启动查看院系统计，选中主体，院系 |
| 后置条件 | | 返回统计数据 |
| 需要的服务（需接口） | | | |
| 服务名 | | 服务 | |
| StaticsDataService.find(TeacherStaticPO tpo) | | 根据教师ID进行查找单一持久化对象 | |
| StaticsDataService.find(StudentStaticPO spo) | | 根据学生ID进行查找单一持久化对象 | |
| StaticsDataService.find(CourseStaticPO cpo) | | 根据课程ID进行查找单一持久化对象 | |

表5.3.2-4 coursebl模块的接口规范

|  |  |  |  |
| --- | --- | --- | --- |
| 提供的服务（供接口） | | | |
| Course.createCourse | 语法 | public ResultMessage createCourse(CourseVO cvo); | |
| 前置条件 | 启动发布课程 | |
| 后置条件 | 保存课程，返回课程信息 | |
| Course.release | 语法 | Public RM release(); | |
| 前置条件 | 课程信息已输入 | |
| 后置条件 | 课程信息设置为公开状态 | |
| Course.delete | 语法 | public ResultMessage delete(String courseID); | |
| 前置条件 | 启动删除课程，选择课程 | |
| 后置条件 | 保存删除信息，返回课程信息 | |
| Course.modify | 语法 | public ResultMessage modifyCourse(CourseVO cvo); | |
| 前置条件 | 启动课程信息修改 | |
| 后置条件 | 保存并返回课程信息 | |
| Course.observeList | 语法 | public ArrayList<CourseVO> observeList(String field,String value); | |
| 前置条件 | 启动课程列表查看 | |
| 后置条件 | 显示课程列表 | |
| Course.observeInfo | 语法 | public CourseVO observeInfo(String courseID); | |
| 前置条件 | 启动课程信息查看 | |
| 后置条件 | 无 | |
| Coruse.findCourseID | 语法 | public ArrayList<CourseVO> findCourseID(String text); | |
| 前置条件 |  | |
| 后置条件 | 无 | |
| Course.findCourseName | 语法 | publicArrayList<CourseVO> findCourseName(String text); | |
| 前置条件 |  | |
| 后置条件 | 无 | |
| Course.observeTeacherAllCoruse | 语法 | PublicArrayList<CourseVO>observeTeaAllCour(String teaID); | |
| 前置条件 |  | |
| 后置条件 | 无 | |
| Course.ObserverInsAllCourse | 语法 | publicArrayList<CourseVO>observeInsAllCour(String depart); | |
| 前置条件 |  | |
| 后置条件 | 无 | |
| Course.InsLeanCourse | 语法 | publicArrayList<CourseVO>observeInsLeanCour(String depart); | |
| 前置条件 |  | |
| 后置条件 | 无 | |
| Course.InsCreateCourse | 语法 | publicArrayList<CourseVO>observeInsCreateCour(String depart); | |
| 前置条件 |  | |
| 后置条件 | 无 | |
| Course.findModualCourse | 语法 | public ArrayList<CourseVO> findModualCour(String 通修); | |
| 前置条件 |  | |
| 后置条件 | 无 | |
| Course.getAllTerms | 语法 | public String[] getAllTerms(); | |
| 前置条件 |  | |
| 后置条件 | 无 | |
| Course.addCourses | 语法 | public CourseVO addCourses(ArrayList<CourseVO> vos) throws RemoteException; | |
| 前置条件 |  | |
| 后置条件 | 无 | |
|  | 语法 | public CourseVO addCourses(String text) throws RemoteException; | |
| Course.jaddCourse | 前置条件 |  | |
| 后置条件 | 无 | |
| Course.getAllTerms | 语法 | public String[] jgetAllTerms(); | |
| 前置条件 |  | |
| 后置条件 | 无 | |
| Course.jaddCourse | 语法 | public CourseVO jaddCourses(ArrayList<CourseVO> vos) throws RemoteException; | |
| 前置条件 |  | |
| 后置条件 | 无 | |
| Course.jaddCourses | 语法 | public CourseVO jaddCourses(String text) throws RemoteException; | |
| 前置条件 |  | |
| 后置条件 | 无 | |
| Course.jObserveAll | 语法 | public ArrayList<CourseVO> jobserveAllCour(); | |
| 前置条件 |  | |
| 后置条件 | 无 | |
| Course.cresteTerm | 语法 | public void createTerm(String newTerm); | |
| 前置条件 |  | |
| 后置条件 | 无 | |
| Course.jDelete | 语法 | public ResultMessage jdelete(CourseVO v); | |
| 前置条件 |  | |
| 后置条件 | 无 | |
| Course.jcreateCourse | 语法 | public ResultMessage jcreateCourse(CourseVO course); | |
| 前置条件 |  | |
| 后置条件 | 无 | |
| Course.jCreteCourse | 语法 | public ResultMessage jcreateCourse(CourseVO course); | |
| 前置条件 |  | |
| 后置条件 | 无 | |
| Course.jUpdate | 语法 | public ResultMessage jupdateCourse(CourseVO v); | |
| 前置条件 |  | |
| 后置条件 | 无 | |
| Course.eateTerm | 语法 | public void jcreateTerm(String newTerm); | |
| 前置条件 |  | |
| 后置条件 | 无 | |
| Course.getAllSelected | 语法 | public ArrayList<CourseVO> getAllSelectedCour(); | |
| 前置条件 |  | |
| 后置条件 | 无 | |
| 需要的服务（需接口） | | | |
| 服务名 | | | 服务 |
| public ResultMessage insert(CoursePO cpo) throws RemoteException; | | | 添加PO |
| public ResultMessage update(CoursePO cpo) throws RemoteException; | | | 更新课程信息 |
| public ResultMessage delete(String courseID) throws RemoteException; | | | 删除课程 |
| public CoursePO find(String courseID) throws RemoteException; | | | 根据课程号查找课程 |
| public ArrayList<CoursePO> finds(String field, String value) throws RemoteException; | | | 根据关键信息查找课程列表 |
| public void readData() throws RemoteException; | | | 读取数据 |
| public void writeData() throws RemoteException; | | | 写数据 |
| public ArrayList<CoursePO> getTeaAllCour(String teaID) throws RemoteException; | | | 得到老师教授的所有课程 |
| public ArrayList<CoursePO> getInsAllCour(String depart) throws RemoteException; | | | 得到学生修读的所有课程 |
| public ArrayList<CoursePO> getInsLeanCour(String ins) throws RemoteException; | | | 得到院系修读的课程列表 |
| public String[] getAllTerms() throws RemoteException; | | | 得到所有的学期列表 |
| public CoursePO addCourses(ArrayList<CoursePO> pos) throws RemoteException; | | | 添加课程 |
| public String[] jgetAllTerms() throws RemoteException; | | | 教务处老师得到学期 |
| public ArrayList<CoursePO> jgetTermCourses() throws RemoteException; | | | 教务处老师得到学期课程列表 |
| public ResultMessage jAddCourse(CoursePO p) throws RemoteException; | | | 教务处老师添加课程 |
| public ResultMessage jupdateCourse(CoursePO p) throws RemoteException; | | | 教务处老师更新课程 |
| public CoursePO jAddCourses(ArrayList<CoursePO> pos) throws RemoteException; | | | 教务处老师添加课程列表 |
| public void jcreateTerm(String newTerm) throws RemoteException; | | | 教务处老师创建学期 |
| public ResultMessage jdelete(CoursePO p) throws RemoteException; | | | 教务处老师删除课程 |
| public ArrayList<CoursePO> getInsCreateCour(String depart) throws RemoteException; | | | 教务处老师得到院系修读列表 |
| public ArrayList<CoursePO> getAllSelectedCour() throws RemoteException; | | | 得到所有选课学生列表 |

表5.3.2-5 gradebl模块的接口规范

|  |  |  |  |
| --- | --- | --- | --- |
| 提供的服务（供接口） | | | |
| Grade.getCourseRange | 语法 | public String getCourseRange(Stirng ID) | |
| 前置条件 | 用户被正确识别 | |
| 后置条件 | 显示该用户的可以选择的范围 | |
| Grade.getCourseScore | 语法 | public ArrayList<GradeVo> getCourseScore(String ID,String range) | |
| 前置条件 | 学生已修完响应的课程 | |
| 后置条件 | 显示该用户选择范围内的课程成绩 | |
| Grade.getGPA | 语法 | public double getGPA(String ID,String range) | |
| 前置条件 | Range内的课程成绩有效 | |
| 后置条件 | 显示该用户range内的GPA | |
| Grade.getGradeSum | 语法 | public int getGradeSum(String ID,String range) | |
| 前置条件 | 用户选择相应的range | |
| 后置条件 | 显示该用户选择range内的总学分 | |
| Grade.record | 语法 | Public RM record(ArrayList<Integer>) | |
| 前置条件 | 启动登记成绩 | |
| 后置条件 | 返回登记成绩结果 | |
| Grade.observe | 语法 | Public RM observe(String courseNum) | |
| 前置条件 | 课程成绩已登记 | |
| 后置条件 | 返回课程成绩单和统计信息 | |
| Grade.hasRecorded | 语法 | Pubilc RM hasRecorded(String courseNum) | |
| 前置条件 | 启动检验，选择课程 | |
| 后置条件 | 返回课程是否已登记成绩 | |
| 需要的服务（需接口） | | | |
| 服务名 | | | 服务 |
| public void updateGrade(ArrayList<GradePO> gradelist) throws RemoteException; | | | 得到成绩列表 |
| public Grade\_StudentPO findStudent\_Grade(String ID) throws RemoteException; | | | 得到学生成绩 |
| public Grade\_TeacherPO findTeacher\_Grade(String ID) throws RemoteException; | | | 得到老师成绩 |
| public ArrayList<GradePO> findCourseInfo(String courseNum, String term) throws RemoteException; | | | 根据课程得到成绩列表 |
| public ArrayList<GradePO> findStudentInfo(String stuID) throws RemoteException; | | | 根据学生得到成绩列表 |

表5.3.2-6 planbl模块的接口规范

|  |  |  |  |
| --- | --- | --- | --- |
| 提供的服务（供接口） | | | |
| Plan.add | 语法 | | public void add(PlanVO pvo); |
| 前置条件 | | 启动查看（发布）教学计划 |
| 后置条件 | | （若发布，初始化教学计划）返回教学计划（初始）信息 |
| Plan.modify | 语法 | | public void modify(PlanVO pvo); |
| 前置条件 | | 启动发布教学计划 |
| 后置条件 | | 保存教学计划，返回教学计划 |
| Plan.release | 语法 | | public void delete(String courseID,String institute); |
| 前置条件 | | 教学计划已完整输入，达到发布要求 |
| 后置条件 | | 教学计划设置为公开状态 |
| Plan.observe | 语法 | | public ArrayList<PlanVO> observePlan(String field,String value) |
| 前置条件 | | 教学计划已发布（非院系教务老师）  教学计划已输入（院系教务老师） |
| 后置条件 | | 无 |
|  | 语法 | | public int observeCredit(String field,String value); |
| Plan.ObserveCredit | 前置条件 | | 已发布教学计划 |
| 后置条件 | | 无 |
| 需要的服务（需接口） | | | |
| 服务名 | | 服务 | |
| PlanDataService.find(String institute) | | 根据ID进行查找单一持久化对象 | |
| PlanDataService.insert(PlanPO ppo) | | 插入单一持久化对象 | |
| PlanDataService.update(PlanPO ppo) | | 更新单一持久化对象 | |

表5.3.2-7 selectionbl模块的接口规范

|  |  |  |  |
| --- | --- | --- | --- |
| 提供的服务（供接口） | | | |
| Selection.addCourse | 语法 | public boolean addCourse(SelectionVO selectionvo); | |
| 前置条件 | 启动添加选课记录 | |
| 后置条件 | 保存选课记录，返回选课记录 | |
| Selection.addDirectCourse | 语法 | public boolean addDirectCourse(SelectionVO selectionvo); | |
| 前置条件 | 启动直选 | |
| 后置条件 | 无 | |
| Selection.deleteCourse | 语法 | public boolean deleteCourse(SelectionVO selectionvo); | |
| 前置条件 | 启动删除 | |
| 后置条件 | 无 | |
| Selection.quit | 语法 | public boolean quitCourse(SelectionVO selectionvo); | |
| 前置条件 | 登录学生满足选课要求  课程可以推选 | |
| 后置条件 | 是否成功 | |
| Selection.getList | 语法 | public ArrayList<CourseVO> getCourseList(); | |
| 前置条件 | 无 | |
| 后置条件 | 返回课程列表 | |
| Selection.getSeleList | 语法 | public ArrayList<CourseVO> getSelCourseList(); | |
| 前置条件 | 无 | |
| 后置条件 | 返回选课列表 | |
| Selection.getMyList | 语法 | publicArrayList<CourseVO>getMyCourseList(String stuID); | |
| 前置条件 | 无 | |
| 后置条件 | 返回个人选课列表 | |
| Selection.getRelist | 语法 | public ArrayList<CourseVO> getReSelCourseList(); | |
| 前置条件 | 已经补选成功 | |
| 后置条件 | 返回补选课程列表 | |
| Selection.verify | 语法 | public boolean verify(); | |
| 前置条件 | 启动检验选课，选择课程 | |
| 后置条件 | 返回检验结果 | |
| Selection.getTempList | 语法 | publicArrayList<CourseVO>getTempCourseList(String stuID); | |
| 前置条件 | 到达排课时间点 | |
| 后置条件 | 返回排课结果 | |
| Selection.addCompusoryCourse | 语法 | public boolean addCompulsoryCourse(CourseVO course, String institute, String stuGrade, String term); | |
| 前置条件 | 添加必修课程 | |
| 后置条件 | 无 | |
| 需要的服务（需接口） | | | |
| 服务名 | | | 服务 |
| public void insertTempSelection(ArrayList<SelectionPO> selectionlist) throws RemoteException; | | | 插入未确认选课记录 |
| public void insertSelection(ArrayList<SelectionPO> selectionlist) throws RemoteException; | | | 插入选课记录 |
| public void deleteTempSelection(ArrayList<SelectionPO> selectionlist) throws RemoteException; | | | 删除未确认选课记录 |
| public void deleteSelection(ArrayList<SelectionPO> selectionlist) throws RemoteException; | | | 删除选课记录 |
| public Selection\_StudentPO findStudent\_Selection(String ID) throws RemoteException; | | | 得到学生选课记录 |
| public ArrayList<TempSelectionPO> getTempSelection() throws RemoteException; | | | 得到临时选课记录列表 |
| public void saveTempSelection(ArrayList<TempSelectionPO> tempSelectionList) throws RemoteException; | | | 存储排课结果 |
| public boolean directInsertSelection(SelectionPO selectionpo) throws RemoteException; | | | 直接插入选课记录 |
| public boolean stuToUpperBound(String course, String term) throws RemoteException; | | | 达到上限 |

## 5.4数据层的分解

数据层主要给业务逻辑层提供数据防伪服务，包括对于持久化数据的增、删、改、查。User业务逻辑需要的服务由UserDataService接口提供。由于持久化数据的保存可能存在多种形式：Txt文件、序列化文件、数据库等，所示抽象了数据服务。数据层模块的具体描述如图5.4所示。

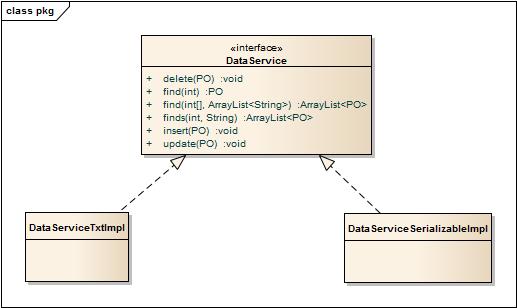


图5.4 数据层模块的描述

### 5.4.1数据层模块的职责

数据层模块的职责如表5.4.1所示。

表5.4.1 数据层模块的职责

|  |  |
| --- | --- |
| 模块 | 职责 |
| DataService | 持久化数据库的接口，提供集体载入，集体保存、增、删、改、查服务。 |
| DataServiceTextImpl | 基于TXT文件的持久化数据库的接口，提供集体载入、集体保存、增、删、改、查服务 |
| DataServiceSerializableImpl | 基于序列化文件的持久化数据库的接口，提供集体载入、集体保存、增、删、改、查服务 |

### 5.4.2数据层模块的接口规范

表5.4.2-1 数据层user模块的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| UserInfoDataService.find | 语法 | public ArrayList<UserInfoPO> find(String type); |
| 前置条件 | 无 |
| 后置条件 | 按ID进行查找返回相应的UserInfoPO结果 |
| UserInfoDataService.insert | 语法 | public ResultMessage insert(ArrayList<UserInfoPO> pos); |
| 前置条件 | 同样ID的po在Mapper中不存在 |
| 后置条件 | 在数据文件中增加一个po记录 |
| UserInfoDataService.delete | 语法 | public ResultMessage delete(UserInfoPO po); |
| 前置条件 | 不需要数据文件中的某个po |
| 后置条件 | 删除一个po |
| UserInfoDataService.update | 语法 | public ResultMessage update(String ID, UserInfoPO po); |
| 前置条件 | 在数据文件中存在同样ID的po |
| 后置条件 | 更新一个po |
| UserInfoDataService.find | 语法 | public ResultMessage find(UserLoginPO po); |
| 前置条件 | 无 |
| 后置条件 | 按ID进行查找返回相应的UserLoginPO结果 |

表5.4.2-2 数据层frame模块的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| FrameDataService.find | 语法 | public FramePO find() throws RemoteException; |
| 前置条件 | 无 |
| 后置条件 | 按ID进行查找返回相应的FramePO结果 |
| FrameDataService.insert | 语法 | public void insert(FramePO fpo) throws RemoteException; |
| 前置条件 | 同样ID的po在Mapper中不存在 |
| 后置条件 | 在数据文件中增加一个po记录 |
| FrameDataService.update | 语法 | public void update(FramePO fpo) throws RemoteException; |
| 前置条件 | 在数据文件中存在同样ID的po |
| 后置条件 | 更新一个po |

表5.4.2-3 数据层static模块的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| StaticDataService.find | 语法 | public TeacherStaticPO find(TeacherStaticPO tpo)throws RemoteException; |
| 前置条件 | 无 |
| 后置条件 | 按ID进行查找返回相应的TeacherPO结果 |
| StaticDataService.find | 语法 | public StudentStaticPO find(StudentStaticPO spo)throws RemoteException; |
| 前置条件 | 无 |
| 后置条件 | 按ID进行查找返回相应的StudentPO结果 |
| StaticDataService.find | 语法 | public CourseStaticPO find(CourseStaticPO cpo)throws RemoteException; |
| 前置条件 | 无 |
| 后置条件 | 按ID进行查找返回相应的CoursePO结果 |

表5.4.2-4 数据层course模块的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| CourseDataService.find | 语法 | Public CoursePO find( ) throws RemoteException; |
| 前置条件 | 无 |
| 后置条件 | 按ID进行查找返回相应的FramePO结果 |
| CourseDataService.insert | 语法 | Public void insert(CoursePO cpo) throws RemoteException; |
| 前置条件 | 同样ID的po在Mapper中不存在 |
| 后置条件 | 在数据文件中增加一个po记录 |
| CourseDataService.delete | 语法 | Public void delete（CoursePO cpo）throws RemoteException; |
| 前置条件 | 不需要数据文件中的某个po |
| 后置条件 | 删除一个po |
| CourseDataService.update | 语法 | Public void update(CoursePO cpo) throws RemoteException; |
| 前置条件 | 在数据文件中存在同样ID的po |
| 后置条件 | 更新一个po |

表5.4.2-5 数据层grade模块的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| GradeDataService.find | 语法 | Public GradePO find(String ID,String Num) throws RemoteException; |
| 前置条件 | 无 |
| 后置条件 | 按ID进行查找返回相应的GradePO结果 |
| GradeDataService.insert | 语法 | Public void insert(GradePO gpo) throws RemoteException; |
| 前置条件 | 同样ID的po在Mapper中不存在 |
| 后置条件 | 在数据文件中增加一个po记录 |

表5.4.2-6 数据层plan模块的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| PlanDataService.find | 语法 | Public PlanPO find(String institude) throws RemoteException; |
| 前置条件 | 无 |
| 后置条件 | 按ID进行查找返回相应的PlanPO结果 |
| PlanDataService.insert | 语法 | Public void insert(PlanPO ppo) throws RemoteException; |
| 前置条件 | 同样ID的po在Mapper中不存在 |
| 后置条件 | 在数据文件中增加一个po记录 |
| FrameDataService.update | 语法 | Public void update(PlanPO ppo) throws RemoteException; |
| 前置条件 | 在数据文件中存在同样ID的po |
| 后置条件 | 更新一个po |

表5.4.2-7 数据层selection模块的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| SelectionDataService.find | 语法 | Public SelectionPO find(String ID,String Num) throws RemoteException; |
| 前置条件 | 无 |
| 后置条件 | 按ID进行查找返回相应的SelectionPO结果 |
| SelectionDataService.insert | 语法 | Public void insert(SelectionPO po) throws RemoteException; |
| 前置条件 | 同样ID的po在Mapper中不存在 |
| 后置条件 | 在数据文件中增加一个po记录 |
| SelectionDataService.delete | 语法 | Public void delete（String ID,String Num）throws RemoteException; |
| 前置条件 | 不需要数据文件中的某个po |
| 后置条件 | 删除一个po |

## 5.5模块的关键类图

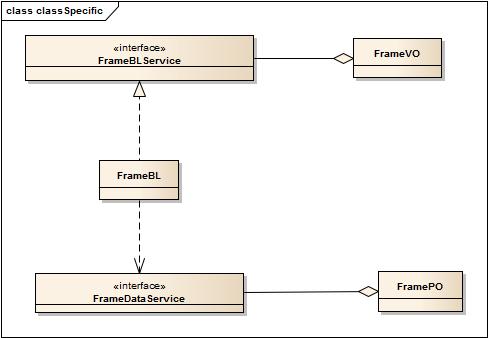


图5.5-1 framebl模块的关键类图

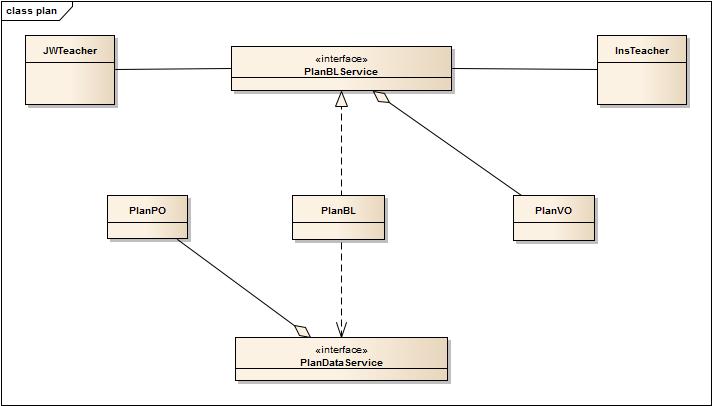


图5.5-2 planbl模块的关键类图

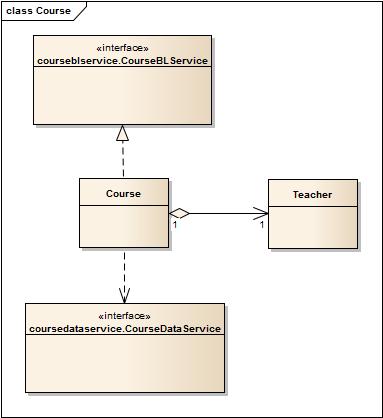


图5.5-3 coursebl模块的关键类图

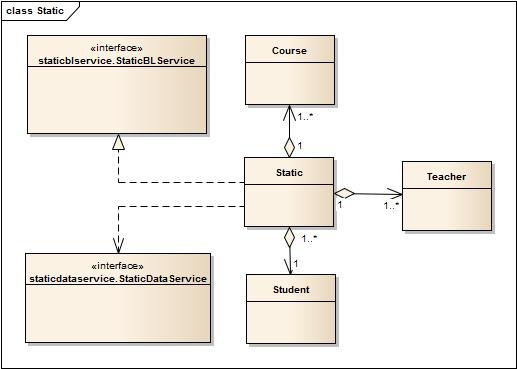


图5.5-4 staticbl模块的关键类图

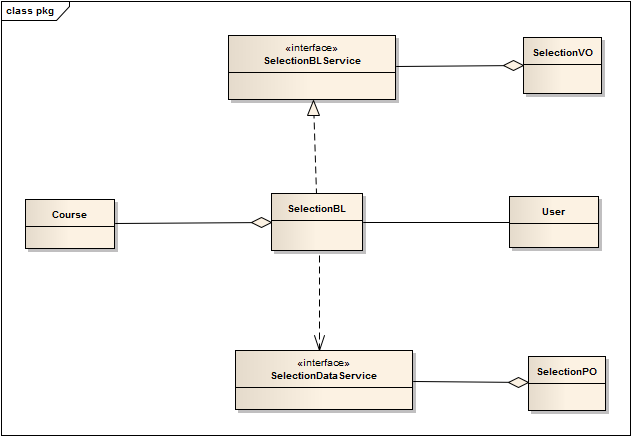


图5.5-5 selectionbl模块的关键类图

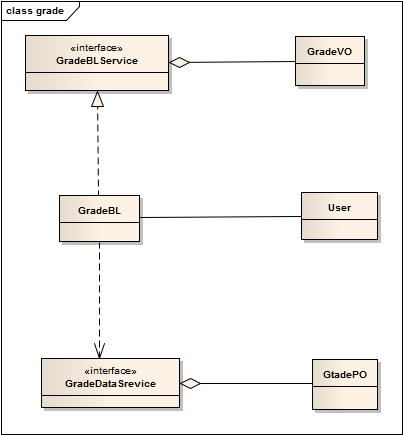


图5.5-6 gradebl模块的关键类图

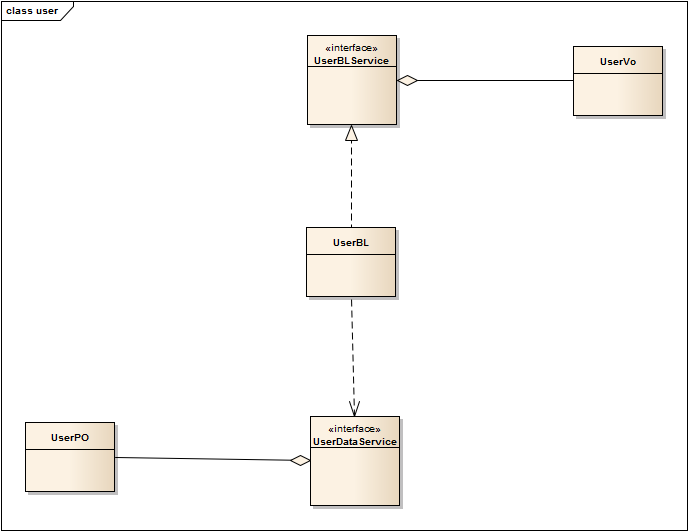


图5.5-7 userbl模块的关键类图

**6．信****息视角**

## 6.1数据持久化对象

系统的PO类就是对应的相关的实体类，在此只做简单的介绍，如表6.1所示：

表6.1 PO类简单介绍

|  |  |
| --- | --- |
| 类名 | 包含的属性 |
| UserPO | 用户名、用户类型 |
| FramePO | 框架的总体描述，总学分，模块属性 |
| BlockPO | 课程模块的描述，学分上限，学分下限，名字属性 |
| StaticPO | 查看的对象主体，对象范围属性 |
| PlanPO | 院系，课程列表属性 |
| CoursePO | 课程名，课程号，授课老师，课程属性，周学时，学分，参考书目，助教，教学大纲，授课时间属性 |
| GradePO | 学生，课程，成绩属性 |
| SelectionPO | 学生，课程，时间属性 |

持久化用户对象如UserPOrcupine的定义如图6.1。

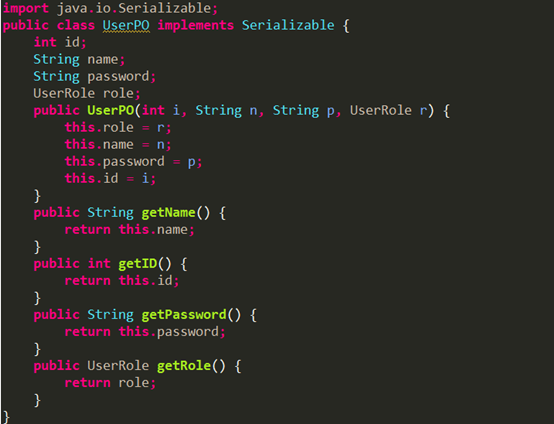


图6.1 持久化用户对象UserPO的定义

## 6.2 Txt持久化格式

表6.2-1 Frame.txt数据格式示意表

|  |  |  |
| --- | --- | --- |
| Frame.txt | 格式 | 框架描述：总学分  、\n模块名：模块描述：模块学分下界：模块学分上界 |
| 举例 | [N][P]  NJWU为了建设世界一流大学，实行了三三制教学改革。：180  通识通修：大类培养。:30:40 |

表6.2-2 Plan.txt数据格式示意表

|  |  |  |
| --- | --- | --- |
| Plan.txt | 格式 | 院系  所属模块：课程名：课程号：学分：周学时 |
| 举例 | 软件学院  平台指选：计算机组成原理：001002:3:3  通识通修：认识计算机：003002:2:2 |

表6.2-3 Course.txt数据格式示意表

|  |  |  |
| --- | --- | --- |
| Course.txt | 格式 | 每行对应课程号，课程名，课程属性，课程学分，周学时,开课院系,修读院系,授课老师(工号),授课时间,助教,课程大纲,教材,参考资料.中间以“：”分开 |
| 举例 | 00101101:计算与软件工程2:核心:3:4:软件学院:软件学院:061250001:周一,1-2;周四,5-6:吴小宝:课程大纲:《软件开发的技术基础》:《软件开发实践者的研究方法》 |

表6.2-4 Selection.txt数据格式示意表

|  |  |  |
| --- | --- | --- |
| Selection.txt | 格式 | 每行对应课程号，选课学生，学生学号，选课时间，成绩  中间以：分开 |
| 举例 | 0040031 张三 121250333 2013-9-1 94 |

表6.2-5 User.txt数据格式示意表

|  |  |  |
| --- | --- | --- |
| User.txt | 格式 | 每行对应用户ID，用户密码，用户名，用户类型，学院，性别，身份证号.中间以“：”分开 |
| 举例 | 121250123：123456：张三：学生：软件学院：男：342343452362345 |