**NJWU选课系统**

**(Course Selection System)**

**详细设计描述文档**



**南京大学软件学院Aurora小组**

**2013-10-21**

**目录** 目录有链接、还有导航的哟~

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

[2.中层设计](#中层设计)···················································································································································

[2.1 frame模块静态结构和动态行为](#frame)····························································································

[2.2 statics模块静态结构和动态行为](#statics)····························································································

[2.3 plan模块静态结构和动态行为](#plan)································································································

[2.4 course模块静态结构和动态行为](#course)····························································································

[2.5 grade模块静态结构和动态行为](#grade)······························································································

[2.6 selection模块静态结构和动态行为](#selection)·······················································································

[2.7 user模块静态结构和动态行为](#user)································································································

[2.8 用户界面层的行为](#用户界面)······················································································································

[3.底层设计模型](#底层设计)············································································································································

[4.设计模型之间的映射](#映射)·······························································································································

[5.详细设计的原理](#详细设计原理)·······································································································································

**更新历史**

|  |  |  |  |
| --- | --- | --- | --- |
| 修改人员 | 日期 | 变更原因 | 版本号 |
| 陈姿丽 | 2013-10-21 | 最初草稿（创建文档模板） | V1.0 |
| 刘璟 | 2013-10-28 | 添加frame模块和statics模块内容 | V1.0.1 |
| 佘昀峰 | 2013-10-29 | 添加grade模块和selection模块内容 | V1.0.2 |
| 任天 | 2013-10-29 | 添加user模块内容 | V1.0.3 |
| 陈姿丽 | 2013-10-29 | 添加plan模块和course模块内容 | V1.0.4 |
| 刘璟 | 2013-10-30 | 添加processmng模块内容 |  |

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 1016-1998 《IEEE Recommended Practice for Software Design Descriptions》

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

1. **中层设计**

**2.1** **frame模块静态结构和动态行为**

**2.1.1 frame模块概述**

frame模块承担与教学框架相关的一系列创建，修改，发布和查看功能。

**2.1.2 frame模块局部模块的职责**

**2.1.2.1结构视角**

（1）模块概述

framebl模块承担的需求见需求规格说明书文档功能需求及相关非功能需求。

framebl模块的职责及接口参见软件体系结构描述文档表。

（2）整体结构

根据体系结构的设计，采用分层风格，将系统分为展示层，业务逻辑层，数据层。每一层之间为了灵活性，添加了接口，以实现针对接口编程，隔离数据传输的职责，降低层与层之间耦合，添加了frameblservice,framedataservice两个接口。为了隔离业务逻辑职责和逻辑控制职责，我们添加了frameController，这样frameController将会将发布整体框架策略的业务逻辑职责和逻辑控制委托给framebl对象。framePO是做为框架信息的持久化对象被添加到设计模型中的，而BlockList封装了Block数据集合的数据结构和组织形式。

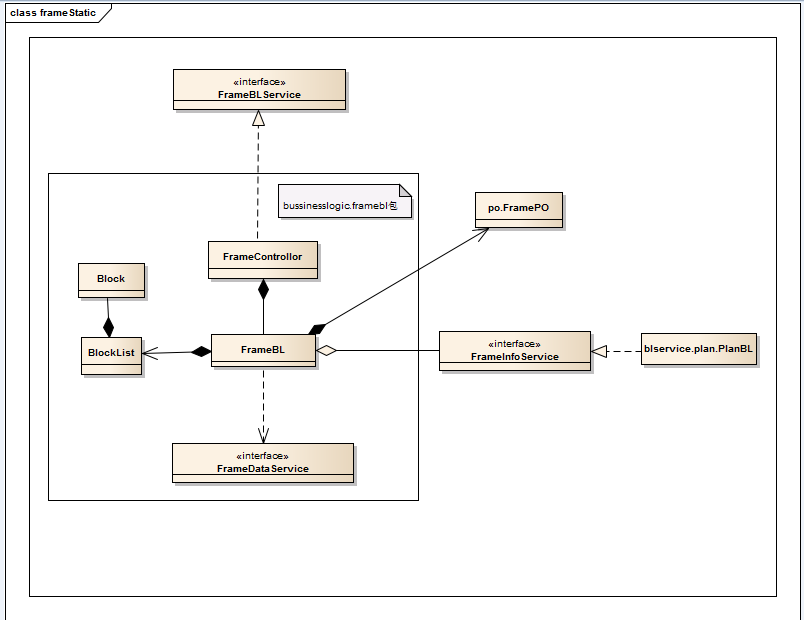


图2.1.2.2 framebl模块各个类的设计

表2.1.2.2 framebl模块各个类的职责

|  |  |
| --- | --- |
| 模块 | 职责 |
| FrameController | 负责实现框架管理界面所需要的服务 |
| Frame | 系统教学框架的领域模型对象，拥有教学框架的总学分(int)，描述(String)，模块列表(BlockList) |
| Block | 系统教学模块的领域模型对象，拥有教学模块的名称(String)，描述(String)，学分下限(int)，学分上限(int)。 |
| BlockList | 系统教学模块集合的领域模型对象，拥有一系列教学模块，将其组织起来，提供一个整体的接口进行访问。 |

**2.1.2.2依赖视角**

**2.1.3 frame模块内部类接口规范**

表2.1.3-1 FrameController的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| createFrame | 语法 | public FrameVO createFrame(int total,String description) |
| 前置条件 | 启动一个框架管理回合 |
| 后置条件 | 调用Frame领域对象的createFrame方法 |
| createBlock | 语法 | public BlockVO createBlock(String name, String description, int lower, int upper) |
| 前置条件 | 启动一个框架管理回合 |
| 后置条件 | 调用Frame领域对象的createBlock方法 |
| look | 语法 | public FrameVO look() |
| 前置条件 | 启动一个框架管理回合 |
| 后置条件 | 调用Frame领域对象的look方法 |
| release | 语法 | public boolean release() |
| 前置条件 | 启动一个框架管理回合 |
| 后置条件 | 调用Frame领域对象的release方法 |
| modifyBase | 语法 | public FrameVO modifyBase(int total, String description) |
| 前置条件 | 启动一个框架管理回合 |
| 后置条件 | 调用Frame领域对象的modifyBase方法 |
| modifyBlock | 语法 | public BlockVO modifyBlock(int numOfBlock, String name, String description, int lower, int upper) |
| 前置条件 | 启动一个框架管理回合 |
| 后置条件 | 调用Frame领域对象的modifyBlock方法 |
| 需要的服务（需接口） | | |
| 服务名 | 服务 | |
| createFrame | 创建框架的具体实现 | |
| createBlock | 创建模块的具体实现 | |
| look | 查看教学框架的具体实现 | |
| release | 发布框架的具体实现 | |
| modifyBase | 修改框架基础信息的具体实现 | |
| modifyBlock | 修改模块信息的具体实现 | |

表2.1.3-2 Frame的接口规范

|  |  |  |  |
| --- | --- | --- | --- |
| 提供的服务（供接口） | | | |
| createFrame | 语法 | public FrameVO createFrame(int total,String description) | |
| 前置条件 | 已创建一个Frame对象，并且输入符合输入规则 | |
| 后置条件 | 持久化所涉及的新创建的领域对象的数据 | |
| createBlock | 语法 | public BlockVO createBlock(String name, String description, int lower, int upper) | |
| 前置条件 | 已创建一个Frame对象，并且输入符合输入规则 | |
| 后置条件 | 持久化所涉及的新创建的领域对象的数据 | |
| look | 语法 | public FrameVO look() | |
| 前置条件 | 已创建一个Frame对象，并且输入符合输入规则 | |
| 后置条件 | 无 | |
| release | 语法 | public boolean release() | |
| 前置条件 | 已创建一个Frame对象，并且输入符合输入规则 | |
| 后置条件 | 设置框架的领域对象的相应属性，进行数据持久化 | |
| modifyBase | 语法 | public FrameVO modifyBase(int total, String description) | |
| 前置条件 | 已创建一个Frame对象，并且输入符合输入规则 | |
| 后置条件 | 更新对应领域对象的持久化数据 | |
| modifyBlock | 语法 | public BlockVO modifyBlock(int numOfBlock, String name, String description, int lower, int upper) | |
| 前置条件 | 已创建一个Frame对象 | |
| 后置条件 | 更新对应领域对象的持久化数据 | |
| 需要的服务（需接口） | | | |
| 服务名 | | | 服务 |
| FrameDataService.insert(FramePO fpo ) | | | 插入新的Frame信息至持久化对象 |
| FrameDataService.update(FramePO fpo) | | | 更新Frame的信息至持久化对象 |
| FrameDataService.find() | | | 查找返回Frame的信息至持久化对象 |
| DataFactory.getFrameData() | | | 得到Frame数据存储系统的服务的引用 |
| BlockList.addBlock(String name, String descriptioin, int lower, int upper) | | | 在模块列表对象中插入新模块 |
| BlockList.modifyBlock(int num, String name, String descriptioin, int lower, int upper) | | | 更新模块列表对象中模块num的模块信息 |
| FramePO.setPublic(bool isPublic) | | | 更新框架持久化数据的状态信息 |

表2.1.3-3 Block的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| modifyBlock | 语法 | public boolean modifyBlock(String name, String description, int lower, int upper) |
| 前置条件 | Block对象已经被创建 |
| 后置条件 | 无 |
| 需要的服务（需接口） | | |
| 服务名 | 服务 | |
| 无 | 无 | |

表2.1.3-4 BlockList的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| addBlock | 语法 | public boolean addBlock(String name, String description, int lower, int upper); |
| 前置条件 | Frame中已创建一个Block对象 |
| 后置条件 | 更新对应领域对象的持久化数据 |
| modifyBlock | 语法 | public boolean modifyBlock(int num, String name, String description, int lower, int upper); |
| 前置条件 | 在Frame中已创建一个Block对象 |
| 后置条件 | 更新对应领域对象的持久化数据 |
| deleteBlock | 语法 | public boolean deleteBlock(int num); |
| 前置条件 | 在Frame中已创建一个Block对象 |
| 后置条件 | 更新对应领域对象的持久化数据 |
| 需要的服务（需接口） | | |
| 服务名 | | 服务 |
| BlockPO.modifyPO | | 更新BlockPO持久化数据的内容 |
| Block.modifyPO | | 更新Block的内容 |

**2.1.4动态模型**

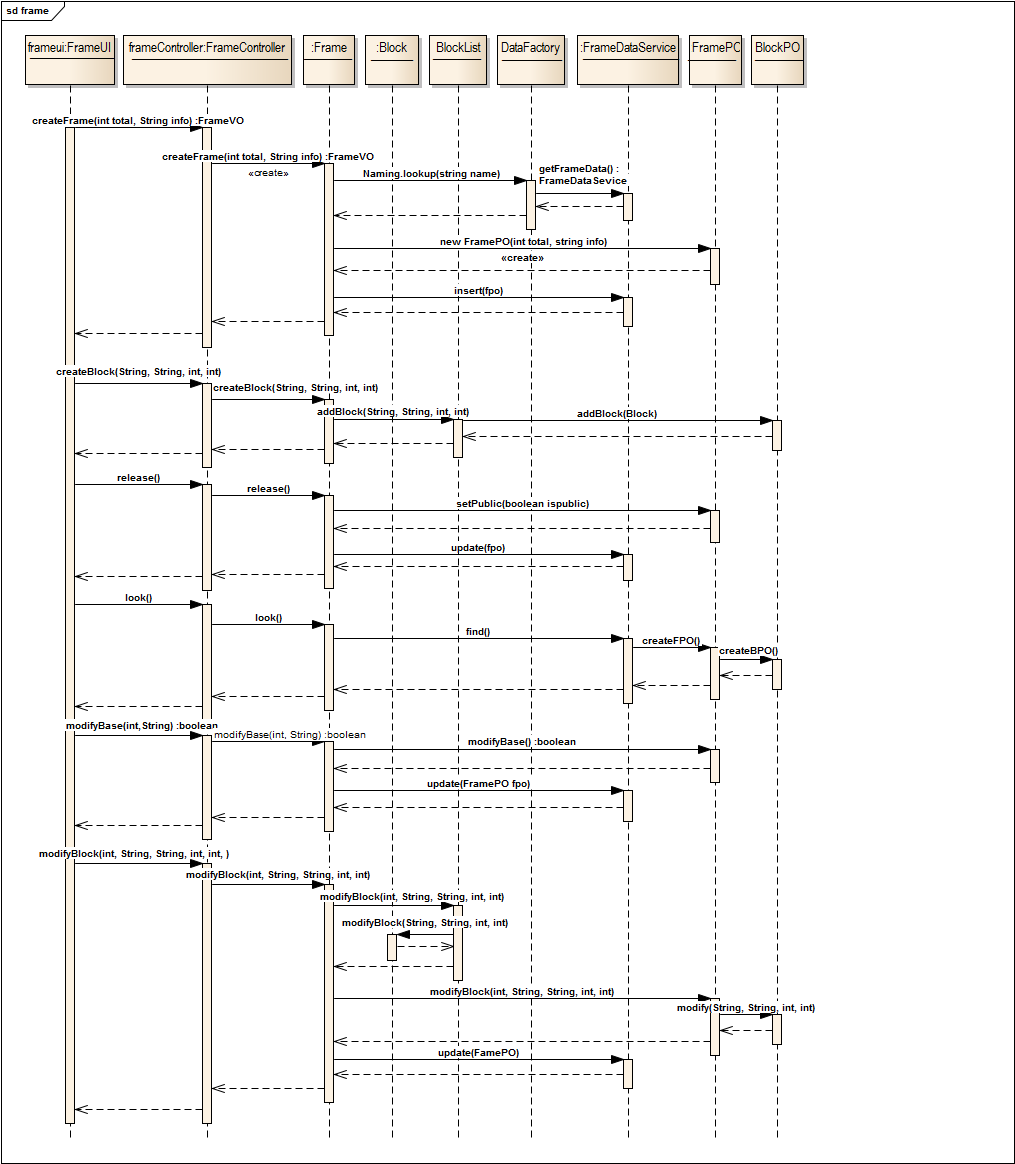


图2.1.4-1 教务处老师与整体框架的顺序图

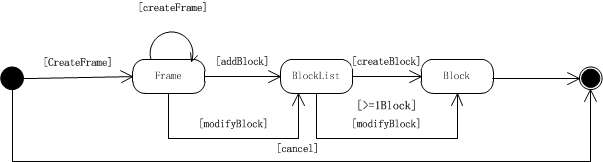


图2.1.4-2 frame对象的状态图

**2.1.5实现注解**

**2.1.6设计原理**

**2.2 st****atics模块静态结构和动态行为**

**2.2.1 statics模块概述**

statics模块承担与统计信息相关的一系列查看功能。

**2.2.2 statics模块局部模块的职责**

**2.2.2.1结构视角——staticsbl模块的分解**

（1）模块概述

staticsbl模块承担的需求见需求规格说明书文档功能需求及相关非功能需求。

staticsbl模块的职责及接口参见软件体系结构描述文档表。

（2）整体结构

根据体系结构的设计，采用分层风格，将系统分为展示层，业务逻辑层，数据层。每一层之间为了灵活性，添加了接口，以实现针对接口编程，隔离数据传输的职责，降低层与层之间耦合，添加了CourseStatiscBLService和CourseDataService两个接口。为了隔离业务逻辑职责和逻辑控制职责，我们添加了CourseStaticsController，这样CourseStsaticsController将会将销售的业务逻辑职责和逻辑控制委托给CourseStatics对象。CourseStaticsPO是做为框架信息的持久化对象被添加到设计模型中的，CourseStsaticsItem封装了具体的统计信息内容，只提供结果做为对外的接口，CourseSaticsItemCalculator封装了算法族，将统计信息的计算任务隔离。（注:以上对CourseStaticsBL模块的描述同样适用于TeacherStaticsBL和StudentStatics）

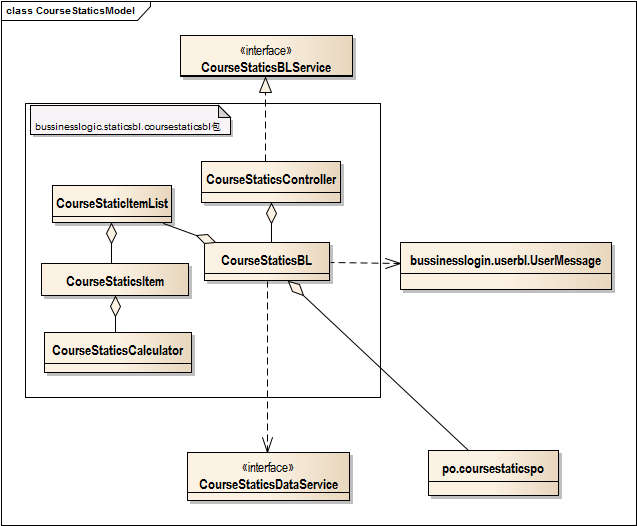


图2.2.2.1-1 courseStaticsbl模块各个类的设计

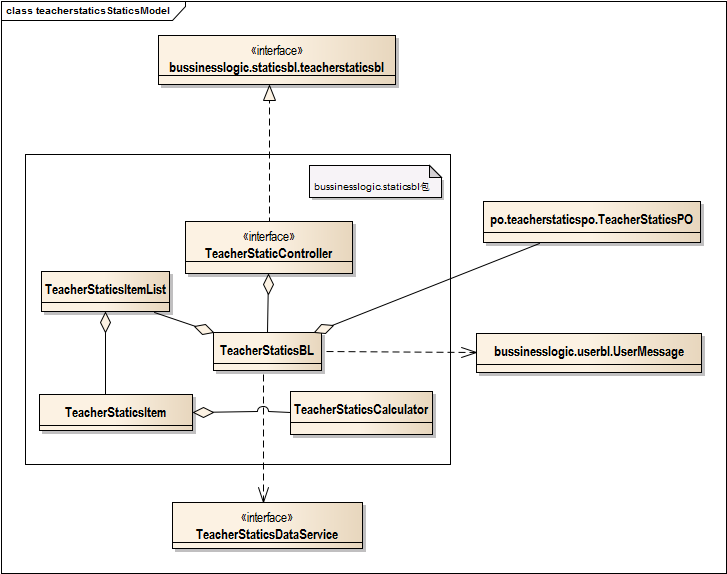


图2.2.2.1-2 teacherStaticsbl模块各个类的设计

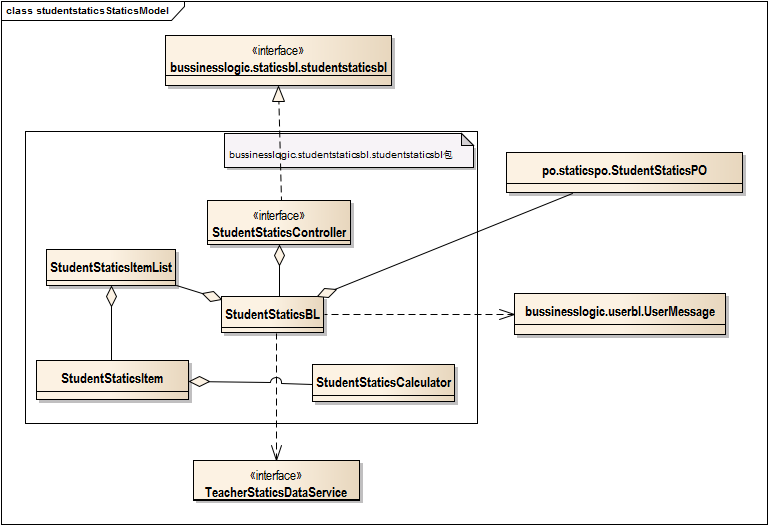


图2.2.2.1-3 studentStaticsbl模块各个类的设计

**2.2.2.2依赖视角**

**2.2.3 statics模块内部类接口规范**

表2.2.3-1 StaticsController的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| show | 语法 | public StaticVO show() |
| 前置条件 | 启动一个查看统计信息回合 |
| 后置条件 | 调用StaticsBL的show方法 |
| show(String ins) | 语法 | public StaticsVO show(String ins) |
| 前置条件 | 启动一个查看统计信息回合 |
| 后置条件 | 调用StaticsBL的show(String ins)方法 |
| 需要的服务（需接口） | | |
| 服务名 | 服务 | |
| StaticsBL.show() | 显示全校的某项统计信息 | |
| StaticsBL.show(String ins) | 显示某院系的某项具体统计信息 | |

表2.2.3-2 StaticsBL类的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| show() | 语法 | public StaticsVO sho() |
| 前置条件 | 已创建一个Statics对象 |
| 后置条件 | 无 |
| show(String ins) | 语法 | public StaticsVO show(String ins) |
| 前置条件 | 已创建一个Satics对象 |
| 后置条件 | 无 |
| 需要的服务（需接口） | | |
| 服务名 | 服务 | |
| ItemList.getStaticsItems | 得到所有的统计项目的条目 | |

表2.2.3-3 StaticsItemList的接口规范

|  |  |  |  |
| --- | --- | --- | --- |
| 提供的服务（供接口） | | | |
| getStaticsItems | 语法 | public StaticsItemList getStaticsItemss() | |
| 前置条件 | ItemList对象已创建 | |
| 后置条件 | 无 | |
| 需要的服务（需接口） | | | |
| 服务名 | | | 服务 |
| StaticsItem.getStaticsItem | | | 得到某个统计项目的数据 |

表2.2.3-4 Statcsitem的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| getStaticsItem | 语法 | public StaticsItem getStaticsItem() |
| 前置条件 | StaticsItemList对象已经被创建 |
| 后置条件 | 无 |
| 需要的服务（需接口） | | |
| 服务名 | | 服务 |
| StaticsItemCalculator.calculateRate | | 计算出对应条目的比率 |
| StaticsDataService.count(int field,String key) | | 返回对应条目在数据存储系统出现的次数，用以做为下步计算二道分子或分母。 |

表2.2.3-5 StaticsCalculator的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| calculateRate | 语法 | public double calculateRate() |
| 前置条件 | Calculator已经被创建 |
| 后置条件 | 更新持久化数据信息 |
| 需要的服务（需接口） | | |
| 服务名 | 服务 | |
| 无 | 无 | |

**2.2.4动态模型**

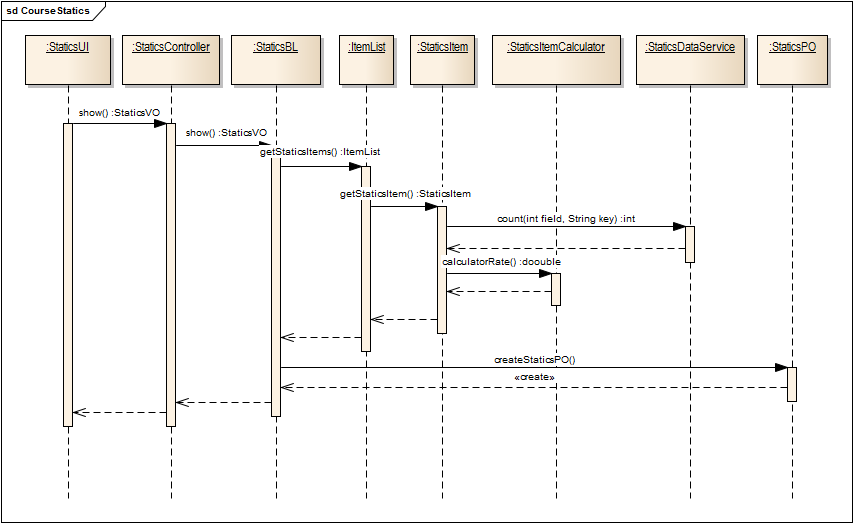


图2.2.4-1 查看统计信息的顺序图

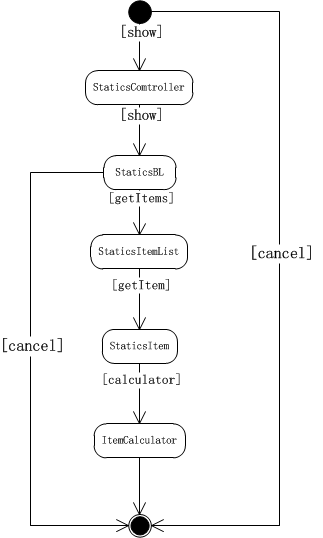


图2.2.4-2 Statics对象的状态图

**2.2.5实现注解**

**2.2.6设计原理**

**2.3 plan****模块静态结构和动态行为**

**2.3.1 plan模块概述**

plan模块承担与教学计划相关的一系列创建，修改，发布和查看功能。

**2.3.2 plan模块局部模块的职责**

**2.3.2.1结构视角**

（1）模块概述

planbl模块承担的需求见需求规格说明书文档功能需求及相关非功能需求。

planbl模块的职责及接口参见软件体系结构描述文档表。

（2）整体结构

根据体系结构的设计，采用分层风格，将系统分为展示层，业务逻辑层，数据层。每一层之间为了灵活性，添加了接口，以实现针对接口编程，隔离数据传输的职责，降低层与层之间耦合，添加了PlanBLService,PlanDataService两个接口。为了隔离业务逻辑职责和逻辑控制职责，我们添加了PlanController，这样PlanController将会将教学计划的业务逻辑职责和逻辑控制委托给Plan对象。PlanPO是做为教学计划信息的持久化对象被添加到设计模型中的。PlanLineItem保有某模块的课程数量的数据，及相应的计算统计数据的职责，而PlanList封装了PlanLineItem数据集合的数据结构和组织形式。

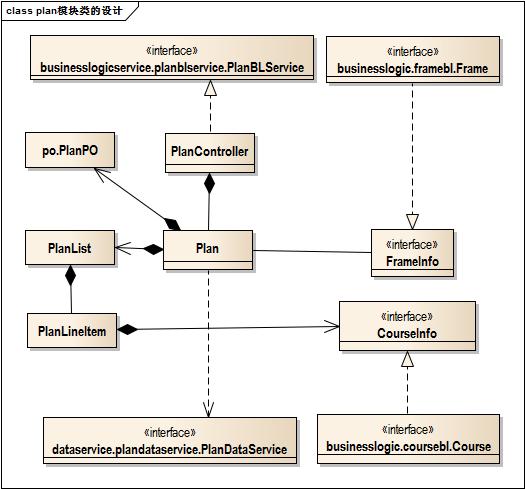


图2.3.2.1 plan模块各个类的设计

表2.3.2.1 plan模块各个类的职责

|  |  |
| --- | --- |
| 类 | 职责 |
| PlanController | 负责实现教学计划管理界面所需要的服务 |
| Plan | 教学计划的领域模型对象，拥有PlanList |
| PlanList | 封装了PlanLineItem数据集合的数据结构和组织形式。 |
| PlanLineItem | 保有某模块的课程数量的数据，及相应的计算统计数据的职责 |

**2.3.2.2依赖视角**

**2.3.3 plan模块内部接口规范**

表2.3.3-1 PlanController的接口规范

|  |  |  |  |
| --- | --- | --- | --- |
| 提供的服务（供接口） | | | |
| addCourse | 语法 | | public PlanVO addCourse(CourseVO cvo) |
| 前置条件 | | 启动一个录入教学计划回合 |
| 后置条件 | | 调用Plan领域对象的addCourse方法 |
| modifyCourse | 语法 | | public PlanVO modifyCourse(CourseVO src, CourseVO target) |
| 前置条件 | | 启动一个修改教学计划回合 |
| 后置条件 | | 调用Plan领域对象的modifyCourse方法 |
| observe | 语法 | | public PlanVO observe() |
| 前置条件 | | 启动一个查看教学计划回合，教学计划相关统计数据已计算 |
| 后置条件 | | 调用Plan领域对象的observe方法 |
| release | 语法 | | public PlanVO release() |
| 前置条件 | | 启动一个发布回合，教学计划已录入，并且录入符合规范 |
| 后置条件 | | 调用Plan领域对象的release方法 |
| deleteCourse | 语法 | | public PlanVO deleteCourse(CourseVO cvo) |
| 前置条件 | | 启动一个删除教学计划回合，并且待删除的计划存在 |
| 后置条件 | | 调用Plan领域对象的deleteCourse方法 |
| selectIns | 语法 | | public PlanVO selectIns(String institute) |
| 前置条件 | | 启动一个选择院系回合，选择的院系存在 |
| 后置条件 | | 调用Plan领域对象的selectIns方法 |
| 需要的服务（需接口） | | | |
| 服务名 | | 服务 | |
| addCourse | | 录入教学计划的具体实现 | |
| modifyCourse | | 修改教学计划的具体实现 | |
| observe | | 查看教学计划的具体实现 | |
| release | | 发布教学计划的具体实现 | |
| deleteCourse | | 删除课程的具体实现 | |
| selectIns | | 选择院系的具体实现 | |

表2.3.3-2 Plan的接口规范

|  |  |  |  |
| --- | --- | --- | --- |
| 提供的服务（供接口） | | | |
| addCourse | 语法 | public PlanVO addCourse(CourseVO cvo) | |
| 前置条件 | 已创建一个Plan对象 | |
| 后置条件 | 添加课程，更新数据库 | |
| modifyCourse | 语法 | public PlanVO modifyCourse(CourseVO src, CourseVO target) | |
| 前置条件 | 已创建一个Plan对象，并且待修改的课程存在 | |
| 后置条件 | 修改并保存课程信息，更新数据库 | |
| observe | 语法 | public PlanVO observe() | |
| 前置条件 | 已创建一个Plan对象，教学计划相关统计数据已计算 | |
| 后置条件 | 返回教学计划统计数据及信息 | |
| release | 语法 | public PlanVO release() | |
| 前置条件 | 已创建一个Plan对象，且输入符合规范 | |
| 后置条件 | 教学计划对其他人公开 | |
| deleteCourse | 语法 | public PlanVO deleteCourse(CourseVO cvo) | |
| 前置条件 | 已创建一个Plan对象，并且待删除的计划存在 | |
| 后置条件 | 删除该教学计划，更新 | |
| selectIns | 语法 | public PlanVO selectIns(String institute) | |
| 前置条件 | 已创建一个Plan对象，且选择的院系存在 | |
| 后置条件 | 保存所选择的院系名称 | |
| 需要提供的服务（需接口） | | | |
| 服务名 | | | 服务 |
| PlanDataService. insert | | | 插入单一持久化对象 |
| PlanDataService. update | | | 更新单一持久化对象 |
| PlanDataService. find(String institute) | | | 根据院系进行查找单一持久化对象 |
| PlanDataService. finds(String field,String value) | | | 根据字段名和值进行查找多个持久化对象 |

表2.3.3-3 PlanList的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| addCourse | 语法 | public PlanVO addCourse(CourseVO cvo) |
| 前置条件 | Plan领域对象调用PlanList的addCourse方法 |
| 后置条件 | 根据持有的信息创建PlanLineItem |
| modifyCourse | 语法 | public PlanVO modifyCourse(CourseVO src, CourseVO target) |
| 前置条件 | Plan领域对象调用PlanList的modifyCourse方法 |
| 后置条件 | 根据持有的信息创建PlanLineItem |
| observe | 语法 | public PlanVO observe() |
| 前置条件 | Plan领域对象调用PlanList的observe方法 |
| 后置条件 | 根据持有的信息创建PlanLineItem |
| release | 语法 | public PlanVO release() |
| 前置条件 | Plan领域对象调用PlanList的release方法 |
| 后置条件 | 根据持有的信息创建PlanLineItem |
| deleteCourse | 语法 | public PlanVO deleteCourse(CourseVO cvo) |
| 前置条件 | Plan领域对象调用PlanList的deleteCourse方法 |
| 后置条件 | 根据持有的信息创建PlanLineItem |
| selectIns | 语法 | public PlanVO selectIns(String institute) |
| 前置条件 | Plan领域对象调用PlanList的selectIns方法 |
| 后置条件 | 根据持有的信息创建PlanLineItem |
| 需要的服务（需接口） | | |
| 服务名 | | 服务 |
| 无 | | 无 |

表2.3.3-4 PlanLineItem的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| addCourse | 语法 | public PlanVO addCourse(CourseVO cvo) |
| 前置条件 | PlanList发送addCourse信息 |
| 后置条件 | 向Course数据文件中添加课程 |
| modifyCourse | 语法 | public PlanVO modifyCourse(CourseVO src, CourseVO target) |
| 前置条件 | PlanList发送modifyCourse信息 |
| 后置条件 | 更新Course文件中的课程信息 |
| observe | 语法 | public PlanVO observe() |
| 前置条件 | PlanList发送obeserve信息 |
| 后置条件 | 检索Course文件中的课程 |
| release | 语法 | public PlanVO release() |
| 前置条件 | PlanList发送release信息 |
| 后置条件 | 更新数据文件中教学计划的公布状态 |
| deleteCourse | 语法 | public PlanVO deleteCourse(CourseVO cvo) |
| 前置条件 | PlanList发送deleteCourse信息 |
| 后置条件 | 删除数据文件中的课程，更新Plan数据文件 |
| selectIns | 语法 | public PlanVO selectIns(String institute) |
| 前置条件 | PlanList发送selectIns信息 |
| 后置条件 | 无 |
| 需要的服务（需接口） | | |
| 服务名 | | 服务 |
| 无 | | 无 |

**2.3.4动态模型**

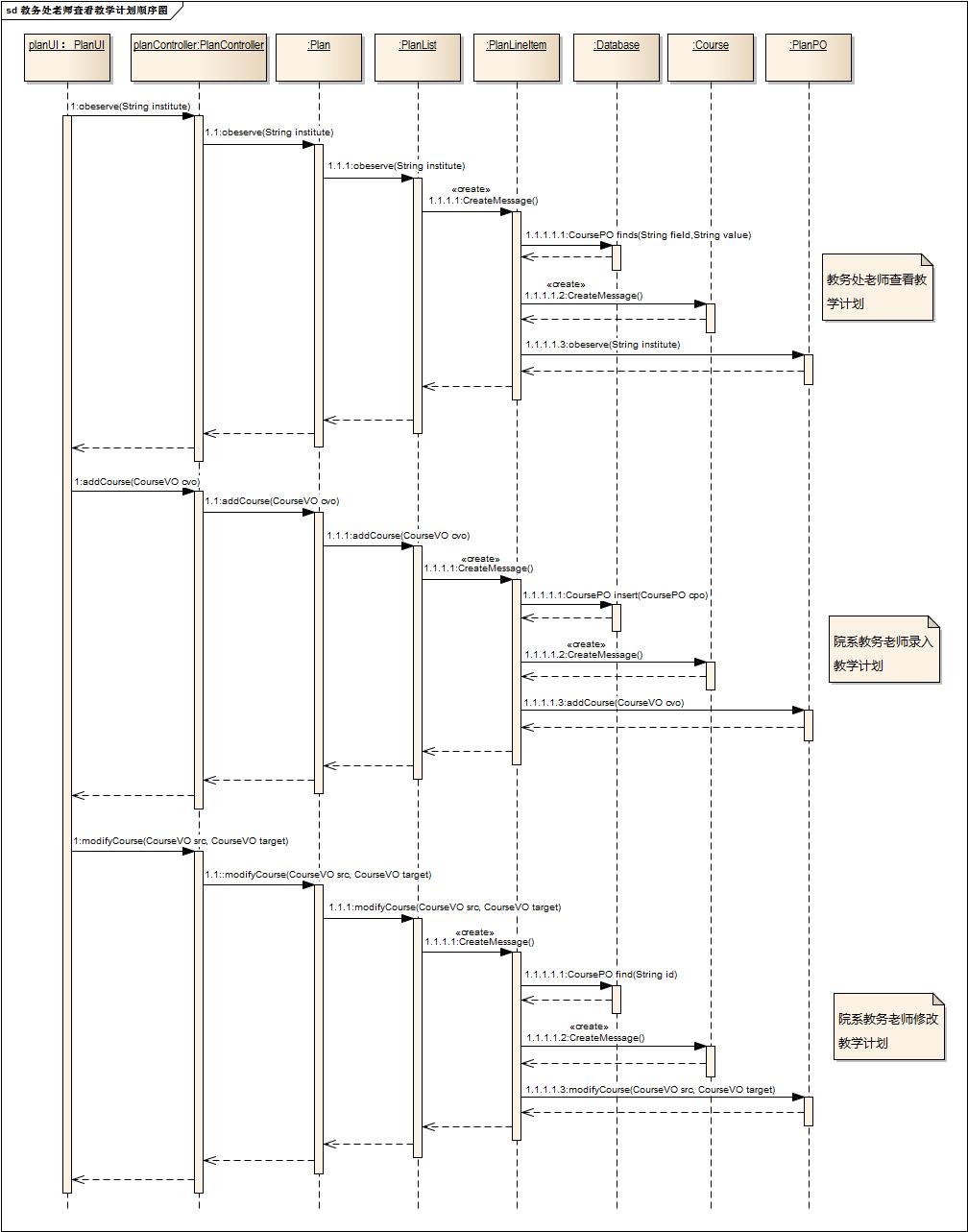


图2.3.4-1 教学计划的顺序图

**2.3.5实现注解**

**2.3.6设计原理**

**2.4 c****ourse模块静态结构和动态行为**

**2.4.1 course模块概述**

course模块承担与课程相关的一系列创建，修改，发布和查看功能。

**2.4.2 course模块局部模块的职责**

**2.4.2.1结构视角——coursebl模块的分解**

（1）模块概述

course模块承担的需求见需求规格说明书文档功能需求及相关非功能需求。

course模块的职责及接口参见软件体系结构描述文档表。

（2）整体结构

根据体系结构的设计，采用分层风格，将系统分为展示层，业务逻辑层，数据层。每一层之间为了灵活性，添加了接口，以实现针对接口编程，隔离数据传输的职责，降低层与层之间耦合，添加了CourseBLService和CourseDataService两个接口。为了隔离业务逻辑职责和逻辑控制职责，我们添加了CourseController，这样CourseController将会将课程的业务逻辑职责和逻辑控制委托给Course对象。CoursePO是作为课程信息的持久化对象被添加到设计模型中的。

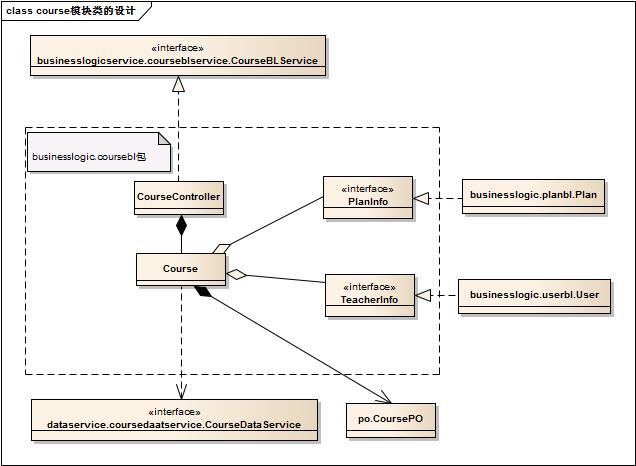


图2.4.2.1-1 course模块类的设计

表2.4.2.1 course模块各个类的职责

|  |  |
| --- | --- |
| 类 | 职责 |
| CourseController | 负责实现课程管理界面所需要的服务 |
| Course | 课程的领域模型对象 |

**2.4.2.2依赖视角**

**2.4.3 course模块内部类接口规范**

表2.4.3-1 CourseController的接口规范

|  |  |  |  |
| --- | --- | --- | --- |
| 提供的服务（供接口） | | | |
| createCourse | 语法 | | public CourseVO createCourse(String courseName, String courseID,String courseProperity,ArrayList<String> teachers,int courseTimePerWeek,int credit,String department,String learnIns) |
| 前置条件 | | 启动一个创建课程回合 |
| 后置条件 | | 调用Course领域对象中的createCourse方法 |
| release | 语法 | | public void release() |
| 前置条件 | | 启动一个发布课程回合 |
| 后置条件 | | 调用Course领域对象中的release方法 |
| delete | 语法 | | public void delete(String courseID) |
| 前置条件 | | 启动一个删除课程回合 |
| 后置条件 | | 调用Course领域对象中的delete方法 |
| modify | 语法 | | public CourseVO modify(String oldCourseID,String newCourseID,String newName,String newProperity,ArrayList<String> teachers,int newTimePerWeek,int newCredit,ArrayList<String> references,ArrayList<String> assistants,String newSummary,String newTime,String newDepartment) |
| 前置条件 | | 启动一个修改课程回合 |
| 后置条件 | | 调用Course领域对象中的modify方法 |
| observe | 语法 | | public CourseVO observe() |
| 前置条件 | | 启动一个查看课程回合 |
| 后置条件 | | 调用Course领域对象中的observe方法 |
| 需要的服务（需接口） | | | |
| 服务名 | | 服务 | |
| createCourse | | 创建课程的具体实现 | |
| release | | 发布课程的具体实现 | |
| delete | | 删除课程的具体实现 | |
| modify | | 修改课程信息的具体实现 | |
| observe | | 查看课程信息的具体实现 | |

表2.4.3-2 Course的接口规范

|  |  |  |  |
| --- | --- | --- | --- |
| 提供的服务（供接口） | | | |
| createCourse | 语法 | public CourseVO createCourse(String courseName, String courseID,String courseProperity,ArrayList<String> teachers,int courseTimePerWeek,int credit,String department,String learnIns) | |
| 前置条件 | 已创建一个Course领域对象，课程信息输入符合规范 | |
| 后置条件 | 添加课程，并更新数据文件 | |
| release | 语法 | public void release() | |
| 前置条件 | 已创建一个Course领域对象，待发布课程合法 | |
| 后置条件 | 更改数据文件中课程的公布状态 | |
| delete | 语法 | public void delete(String courseID) | |
| 前置条件 | 已创建一个Course领域对象，待删除课程存在 | |
| 后置条件 | 删除数据文件中的课程 | |
| modify | 语法 | public CourseVO modify(String oldCourseID,String newCourseID,String newName,String newProperity,ArrayList<String> teachers,int newTimePerWeek,int newCredit,ArrayList<String> references,ArrayList<String> assistants,String newSummary,String newTime,String newDepartment) | |
| 前置条件 | 已创建一个Course领域对象，输入符合规范 | |
| 后置条件 | 修改并保存课程信息，更新数据文件 | |
| observe | 语法 | public CourseVO observe() | |
| 前置条件 | 已创建一个Course领域对象 | |
| 后置条件 | 返回课程信息 | |
| 需要的服务（需接口） | | | |
| 服务名 | | | 服务 |
| CourseDataService.insert | | | 插入单一持久化对象 |
| CourseDataService.update | | | 更新单一持久化对象 |
| CourseDataService.delete | | | 删除单一持久化对象 |
| CourseDataService.find（String id） | | | 根据ID进行查找单一持久化对象 |
| CourseDataService.finds（String field，String value） | | | 根据字段名和值进行查找多个持久化对象 |

**2.4.4动态模型**

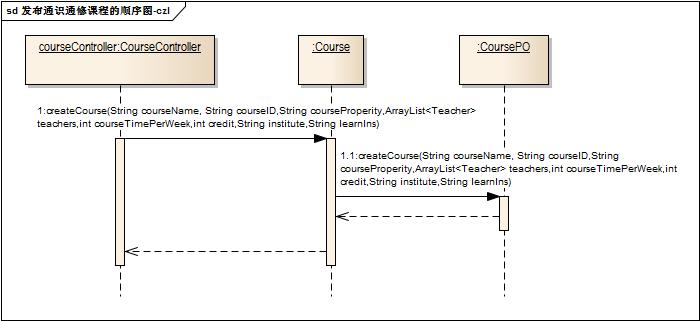


图2.4.4-1 教务处老师发布通识通修课的顺序图

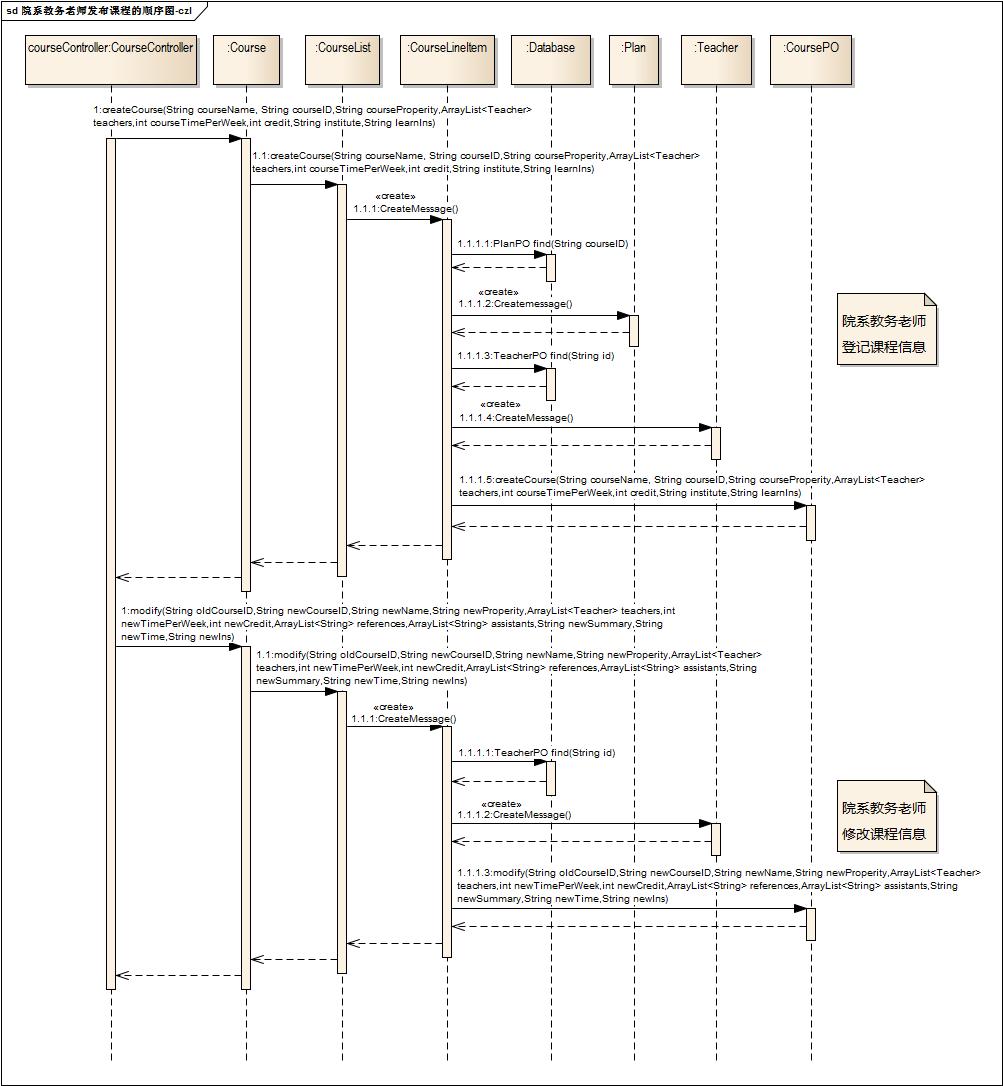


图2.4.4-2 院系教务老师与课程的顺序图

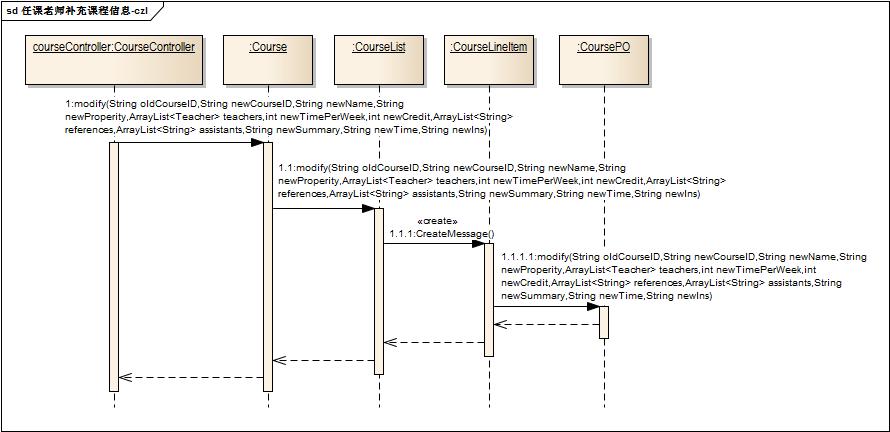


图2.4.4-3 任课老师补充课程信息的顺序图

**2.4.5实现注解**

**2.4.6设计原理**

**2.5 grade模块静态结构和动态行为**

**2.5.1 grade模块概述**

grade模块承担与成绩相关的一系列录入，修改，查看功能。

**2.5.2 grade模块局部模块的职责**

**2.5.2.1结构视角——grade模块的分解**

（1）模块概述

Gradebl模块承担的需求见需求规格说明书文档功能需求及相关非功能需求。

Gradebl模块的职责及接口参见软件体系结构描述文档表？

（2）整体结构

根据体系结构的设计，采用分层风格，将系统分为展示层，业务逻辑层，数据层。每一层之间为了灵活性，添加了接口，以实现针对接口编程，隔离数据传输的职责，降低层与层之间耦合，添加了gradeblservice,gradedataservice两个接口。为了隔离业务逻辑职责和逻辑控制职责，我们添加了gradeController，这样gradeController将会将成绩管理的业务逻辑职责和逻辑控制委托给gradebl对象。GradePO是做为框架信息的持久化对象被添加到设计模型中。

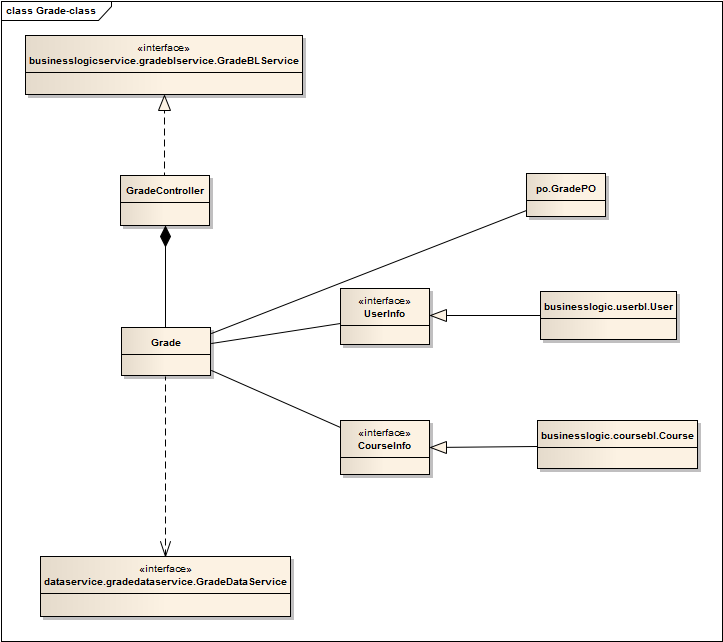


图2.5.2.1 grade模块类的设计

表2.5.2.1 grade模块各个类的职责

|  |  |
| --- | --- |
| 类 | 职责 |
| GradeController | 负责实现成绩管理界面所需要的服务 |
| Grade | 成绩管理的领域模型对象，拥有老师，学生，课程，成绩等信息 |

**2.5.2.2依赖视角**

**2.5.3 grade模块内部接口规范**

表2.5.3-1 GradeController的接口规范

|  |  |  |  |
| --- | --- | --- | --- |
| 提供的服务（供接口） | | | |
| getCoruseRange | 语法 | public String getCourseRange(Stirng ID) | |
| 前置条件 | 已创建一个Grade领域对象 | |
| 后置条件 | 调用Grade领域对象的getCoruseRange方法 | |
| getCorusesScore | 语法 | public ArrayList<GradeVo> getCourseScore(String ID,String range) | |
| 前置条件 | 已创建一个Grade领域对象 | |
| 后置条件 | 调用Grade领域对象的getCorusesScore方法 | |
| getGPA | 语法 | public double getGPA(String ID,String range) | |
| 前置条件 | 已创建一个Grade领域对象 | |
| 后置条件 | 调用Grade领域对象的getGPA方法 | |
| getGradeSum | 语法 | public int getGradeSum(String ID,String range) | |
| 前置条件 | 已创建一个Grade领域对象 | |
| 后置条件 | 调用Grade领域对象的getGradeSum方法 | |
| recordScore | 语法 | public boolean recordScore(String course, ArrayList<GradeVo> scores) | |
| 前置条件 | 已创建一个Grade领域对象 | |
| 后置条件 | 调用Grade领域对象的recordScore方法 | |
| getScoreList | 语法 | public ArrayList<GardeVo> getScoreList(String course) | |
| 前置条件 | public String getCourseRange(Stirng ID) | |
| 后置条件 | 已创建一个Grade领域对象 | |
| 需要的服务（需接口） | | | |
| 服务名 | | | 服务 |
| getCourseRange(Stirng ID) | | | 查看学生已登记成绩的学期 |
| getCourseScore(String ID,String range) | | | 查看学期成绩单 |
| getGPA(String ID,String range) | | | 查看GPA |
| getGradeSum(String ID,String range) | | | 发布学期总学分 |
| recordScore(String course, ArrayList<GradeVo> scores) | | | 录入成绩 |
| getScoreList(String course) | | | 查看课程修读学生成绩单 |

表2.5.3-2 Grade的接口规范

|  |  |  |  |
| --- | --- | --- | --- |
| 提供的服务（供接口） | | | |
| getCoruseRange | 语法 | public String getCourseRange(Stirng ID) | |
| 前置条件 | 启动查看成绩任务 | |
| 后置条件 | 返回可查看成绩的学期 | |
| getCorusesScore | 语法 | public ArrayList<GradeVo> getCourseScore(String ID,String range) | |
| 前置条件 | 选择一个可查看成绩的学期 | |
| 后置条件 | 显示该学期的成绩单 | |
| getGPA | 语法 | public double getGPA(String ID,String range) | |
| 前置条件 | 选择一个可查看成绩的学期 | |
| 后置条件 | 显示该学期的成绩单 | |
| getGradeSum | 语法 | public int getGradeSum(String ID,String range) | |
| 前置条件 | 选择一个可查看成绩的学期 | |
| 后置条件 | 显示该学期的总学分 | |
| recordScore | 语法 | public boolean recordScore(String course, ArrayList<GradeVo> scores) | |
| 前置条件 | 课程为该老师授课且未录入成绩，录入成绩信息完整 | |
| 后置条件 | 返回录入成绩结果 | |
| getScoreList | 语法 | public ArrayList<GardeVo> getScoreList(String course) | |
| 前置条件 | 选择一门老师教授的课程 | |
| 后置条件 | 显示该课程的学生成绩单 | |
| 需要的服务（需接口） | | | |
| 服务名 | | | 服务 |
| GradeDataService.insert(GradePO gpo ) | | | 插入新的Grade信息至持久化对象 |
| GradeDataService.update(GradePO fpo) | | | 更新Frame的信息至持久化对象 |
| GradeDataService.find() | | | 查找返回Frame的信息至持久化对象 |
| DataFactory.getGradeData() | | | 得到Frame数据存储系统的服务的引用 |

**2.5.4动态模型**

**2.5.5实现注解**

**2.5.6设计原理**

**2.6** **selection模块静态结构和动态行为**

**2.6.1 selection模块概述**

selection模块承担与选课相关的一系列排课，记录，查看功能。

**2.6.2 selection模块局部模块的职责**

**2.6.2.1结构视角——selection模块的分解**

（1）模块概述

Selectionbl模块承担的需求见需求规格说明书文档功能需求及相关非功能需求。

Selectionbl模块的职责及接口参见软件体系结构描述文档表？

（2）整体结构

根据体系结构的设计，采用分层风格，将系统分为展示层，业务逻辑层，数据层。每一层之间为了灵活性，添加了接口，以实现针对接口编程，隔离数据传输的职责，降低层与层之间耦合，添加了selectionblservice,selectiondataservice两个接口。为了隔离业务逻辑职责和逻辑控制职责，我们添加了selectionController，这样selectionController将会将成绩管理的业务逻辑职责和逻辑控制委托给selectionbl对象。selectionPO是做为框架信息的持久化对象被添加到设计模型中的，而CourseList封装了Course数据集合的数据结构和组织形式。

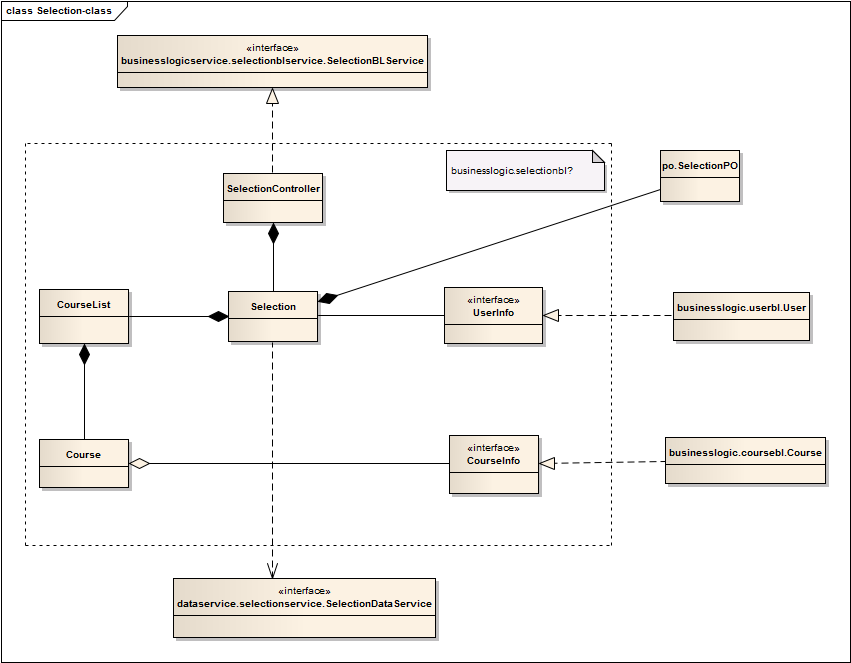


图2.6.2.2 selectionbl模块类的设计

表2.6.2.2 selectionbl模块各个类的职责

|  |  |
| --- | --- |
| 类 | 职责 |
| SelectionController | 负责实现对应于选课界面所需要的服务 |
| Selection | 选课的领域模型对象，拥有选课所持有的学生，课程等信息，可以帮助完成选课界面所需要的服务 |
| Course | 课程模块的领域模型对象，拥有课程号，课程名称，学分，授课时间，上课时间，要求等信息 |
| CourseList | 课程模块集合的领域模型对象，拥有一系列课程模块，将其组织起来，提供一个整体的接口进行访问。 |

**2.6.2.2依赖视角**

**2.6.3 selection模块内部接口规范**

表2.6.3-1 SelectionController的接口规范

|  |  |  |  |
| --- | --- | --- | --- |
| 提供的服务（供接口） | | | |
| addCourse | 语法 | public boolean addCourse(String stuID, String courseNum) | |
| 前置条件 | 已创建一个Selection领域对象 | |
| 后置条件 | 调用Selection领域对象的addCourse方法 | |
| deleteCourse | 语法 | public boolean deleteCourse(String stuID, String courseNum) | |
| 前置条件 | 已创建一个Selection领域对象 | |
| 后置条件 | 调用Selection领域对象的deleteCourse方法 | |
| quitCourse | 语法 | public boolean quitCourse() | |
| 前置条件 | 已创建一个Selection领域对象 | |
| 后置条件 | 调用Selection领域对象的-quitCourse方法 | |
| verify | 语法 | public boolean verify() | |
| 前置条件 | 已创建一个Selection领域对象 | |
| 后置条件 | 调用Selection领域对象的-verify方法 | |
| 需要的服务（需接口） | | | |
| 服务名 | | | 服务 |
| Selection.addCourse(String id, String num) | | | 添加一门课程 |
| Selection.deleteCourse(String id, String num) | | | 删除一门课程 |
| Selection.quitCourse(String id, String num) | | | 退选一门课程 |
| Selection.erify() | | | 确认选课 |

表2.6.3-2 Selection的接口规范

|  |  |  |  |
| --- | --- | --- | --- |
| 提供的服务（供接口） | | | |
| addCourse | 语法 | public boolean addCourse(String stuID, String courseNum) | |
| 前置条件 | 启动一个选课任务 | |
| 后置条件 | 返回选课结果 | |
| deleteCourse | 语法 | public boolean deleteCourse(String stuID, String courseNum) | |
| 前置条件 | 已添加选课且系统未开始受理选课任务 | |
| 后置条件 | 返回删除选课结果 | |
| quitCourse | 语法 | public boolean quitCourse() | |
| 前置条件 | 启动退课任务 | |
| 后置条件 | 返回退课结果 | |
| verify | 语法 | public boolean verify() | |
| 前置条件 | 已添加选课且系统未开始受理选课任务 | |
| 后置条件 | 返回选课确认结果 | |
| 需要的服务（需接口） | | | |
| 服务名 | | | 服务 |
| SelectionDataService.insert(SelectionPO spo ) | | | 插入单一持久化对象 |
| SelectionDataService.delete(SelectionPO spo ) | | | 删除单一持久化对象 |
| SelectionFactory.getFrameData() | | | 得到Selection数据存储系统的服务的引用 |
| CourseList.addCourse(String courseNum) | | | 在课程列表对象中插入新课程 |

表2.6.3-3 CourseList的接口规范

|  |  |  |  |
| --- | --- | --- | --- |
| 提供的服务（供接口） | | | |
| addCourse | 语法 | public boolean addCourse(String stuID, String courseNum) | |
| 前置条件 | Selection中学生满足该门课程选课要求 | |
| 后置条件 | 更新学生已选课程列表 | |
| deleteCourse | 语法 | public boolean deleteCourse(String stuID, String courseNum) | |
| 前置条件 | 已选课程中包含该课程 | |
| 后置条件 | 更新学生已选课程列表 | |
| verify | 语法 | public boolean verify() | |
| 前置条件 | 已对选课列表进行更新 | |
| 后置条件 | 更新持久化数据 | |
| 需要的服务（需接口） | | | |
| 服务名 | | | 服务 |
| StudentPO.addCourse | | | 更新StudentPO持久化数据的内容 |
| CoursePO.addStudent | | | 更新CoursePO持久化数据的内容 |

**2.6.4动态模型**

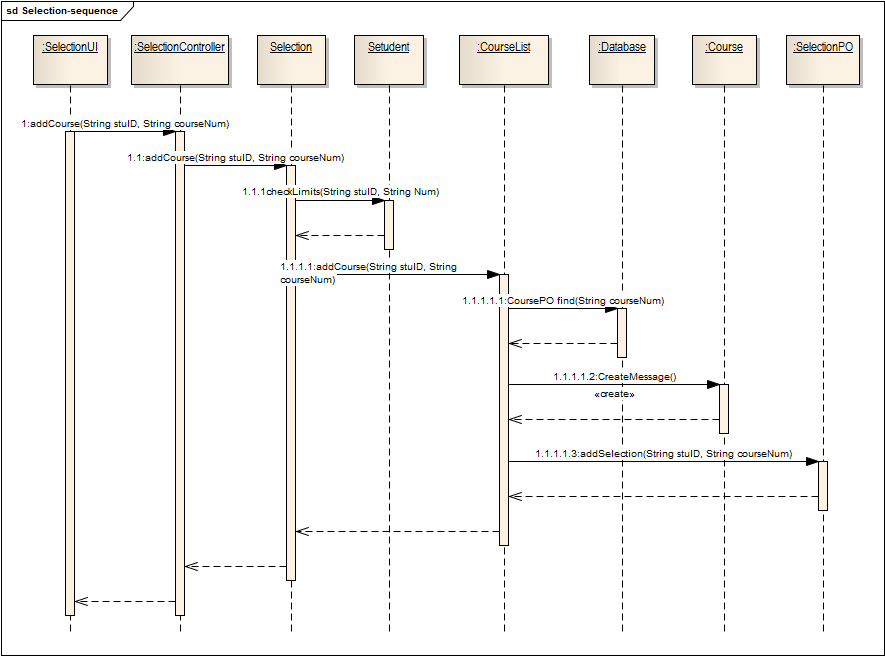


图2.6.4-1选课的顺序图

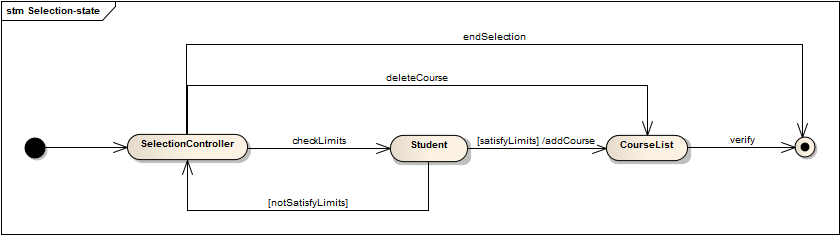


图2.6.4-2选课的状态图

**2.6.5实现注解**

**2.6.6设计原理**

**2.7 us****er模块静态结构和动态行为**

**2.7.1 user模块概述**

user模块承担与用户相关的一系列创建，修改，登录和查看功能。

**2.7.2 user模块局部模块的职责**

**2.7.2.1结构视角——user模块的分解**

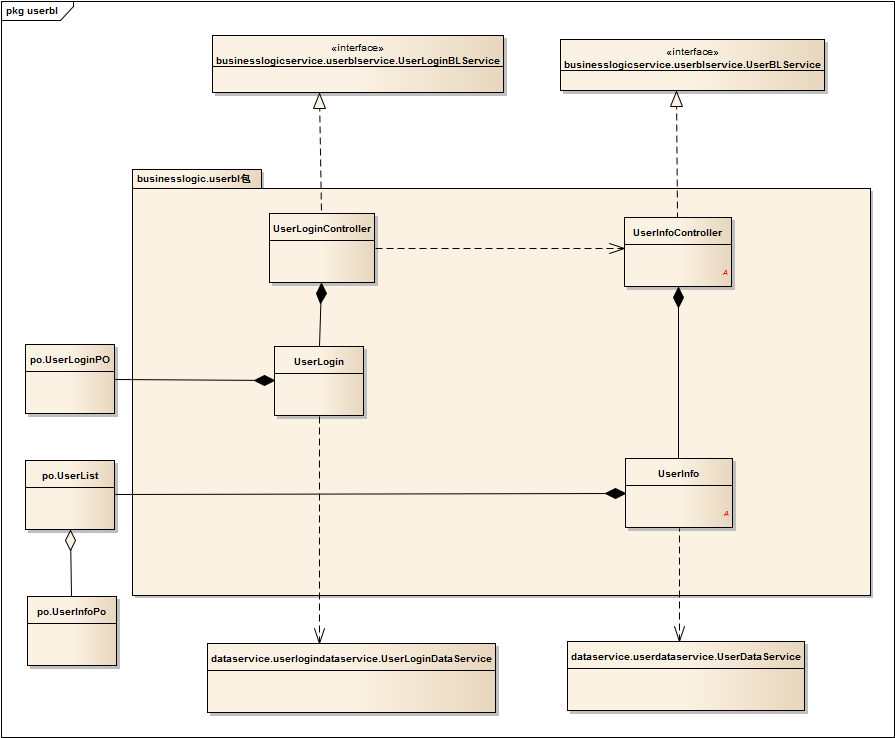


图2.7.2.1 user模块类的设计

表2.7.2.1 user模块各个类的职责

**2.7.2.2依赖视角**

**2.7.3 user模块内部接口规范**

**2.7.4动态模型**

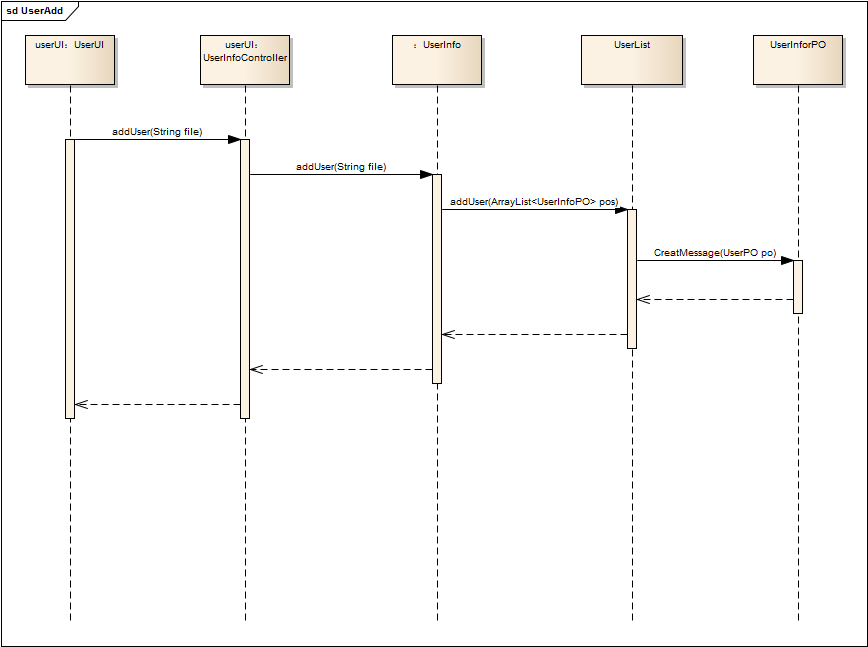


图2.7.4-1添加用户的顺序图

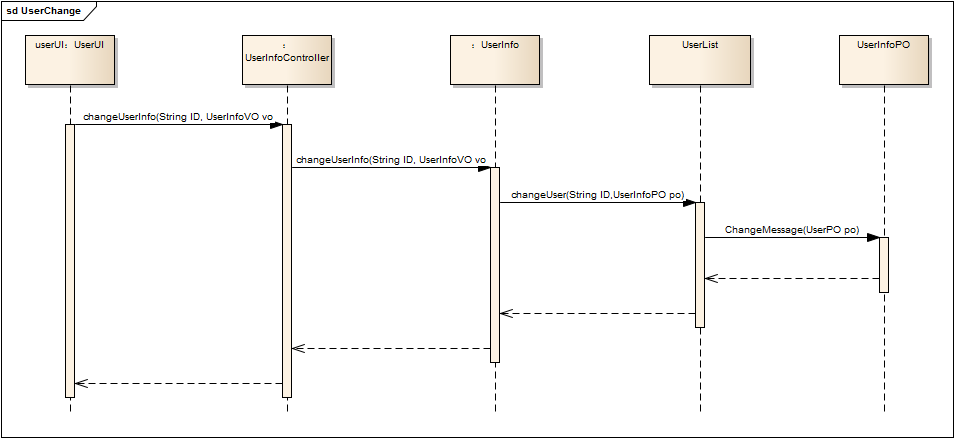


图2.7.4-2更改用户的顺序图

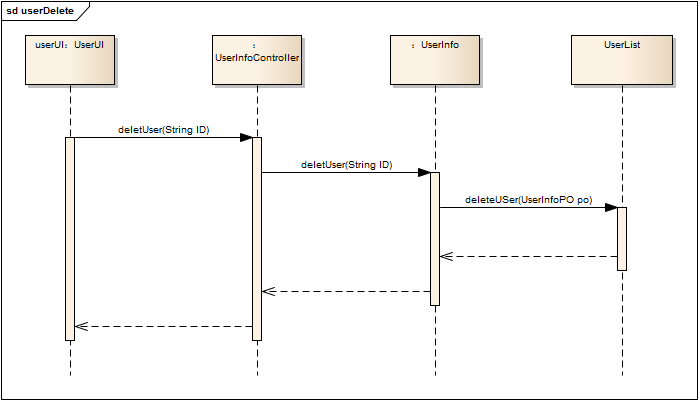


图2.7.4-3删除用户的顺序图

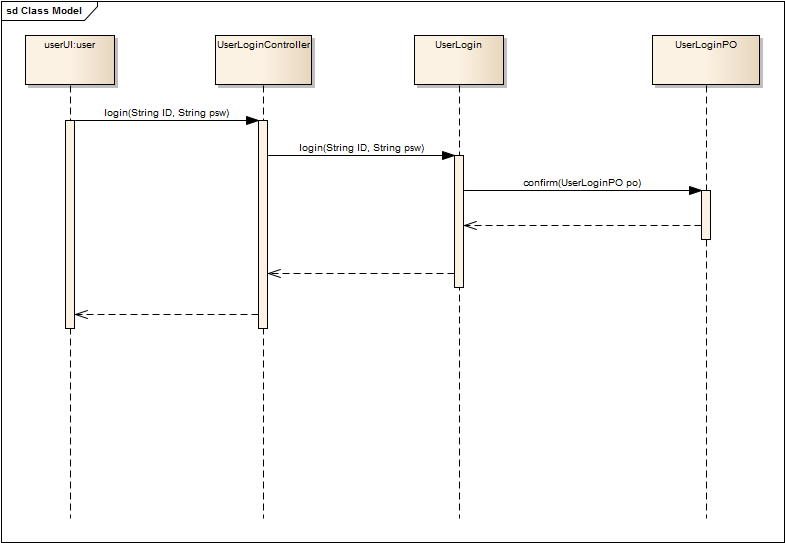


图2.7.4-4用户登录的顺序图

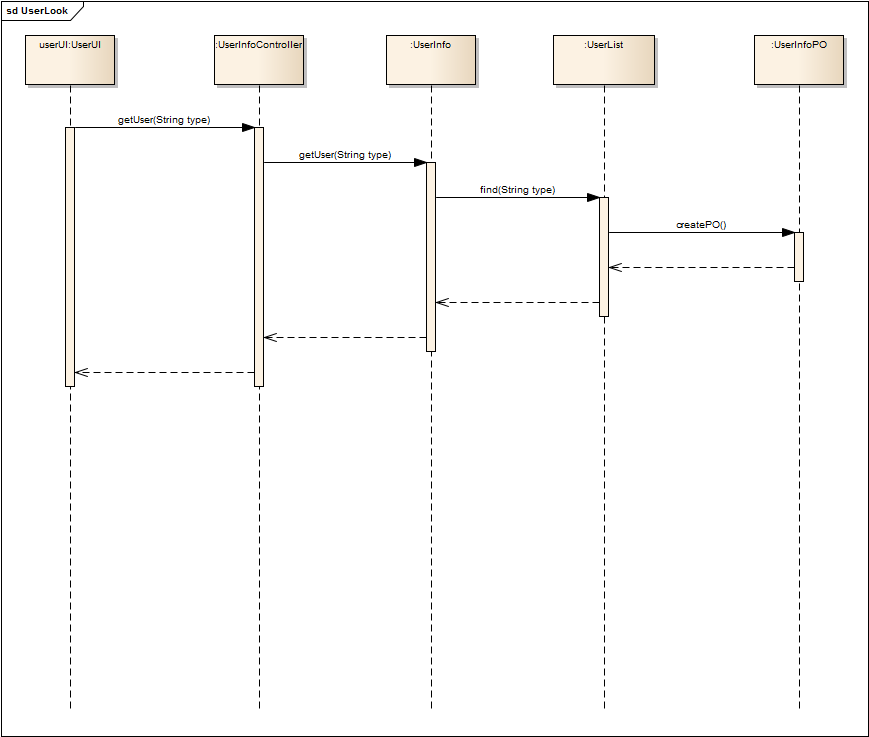


图2.7.4-5查看用户的顺序图

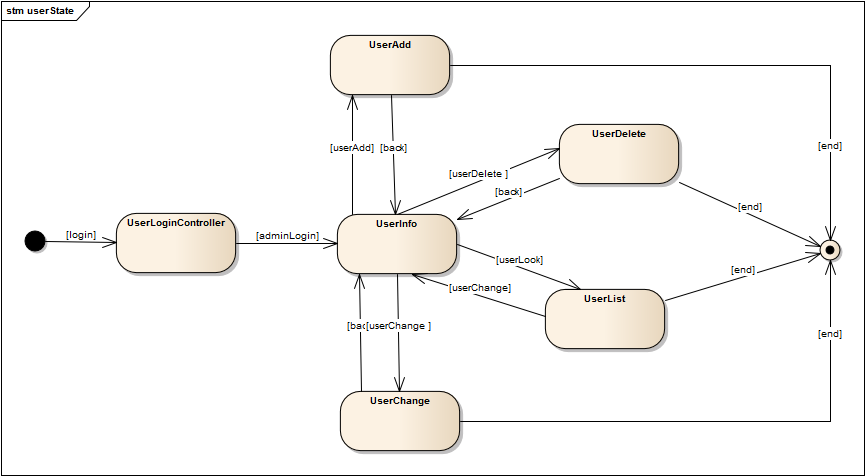


图2.7.4-6 User对象的状态图

**2.7.5实现注解**

**2.7.6设计原理**

**2.8****用户界面层的行为**

1. **底****层设计模型**
2. **设计模型之间的映射**

**5.****详细设计的原理**