

# 南京大学教务系统软件体系结构描述文档

南京大学软件学院

2013 年 10 月

韩旭 (121250041)  
郭天昊 (121250039)  
陈俊杉 (121250011)  
施周利 (121250126)

# 目录

- 1 引言 ..... 2
  - 1.1 编制目的 ..... 2
  - 1.2 词汇表 ..... 2
  - 1.3 参考资料 ..... 2
- 2 产品概述 ..... 2
- 3 逻辑视角 ..... 2
- 4 组合视角 ..... 3
  - 4.1 开发包图 ..... 3
  - 4.2 运行时进程 ..... 6
  - 4.3 物理部署 ..... 7
- 5 接口视角 ..... 7
  - 5.1 模块的职责 ..... 7
  - 5.2 用户界面层的分解 ..... 10
    - 5.2.1 用户界面层模块的职责 ..... 12
    - 5.2.2 用户界面层模块的接口规范 ..... 12
    - 5.2.3 用户界面模块设计原理 ..... 12
  - 5.3 业务逻辑层的分解 ..... 12
    - 5.3.1 业务逻辑层模块的职责 ..... 12
    - 5.3.2 业务逻辑层模块接口规范 ..... 12
  - 5.4 数据层的分解 ..... 18
    - 5.4.1 数据层模块的职责 ..... 18
    - 5.4.2 数据层模块的接口规范 ..... 19
- 6 信息视角 ..... 20
  - 6.1 数据持久化对象 ..... 20
  - 6.2 数据库表 ..... 20

## 1. 引言

### 1.1 编制目的

本报告详细完成对教务系统的概要设计，达到指导详细设计和开发的目的，同时实现和测试人员及用户的沟通。

本报告面向开发人员，测试人员及最终用户而编写，是了解系统的导航。

### 1.2 词汇表

### 1.3 参考资料

## 2. 产品概述

参考教务系统用例文档和教务系统需求规格说明文档中对产品的概括描述。教务系统体系结构采用分层风格。

## 3. 逻辑视角

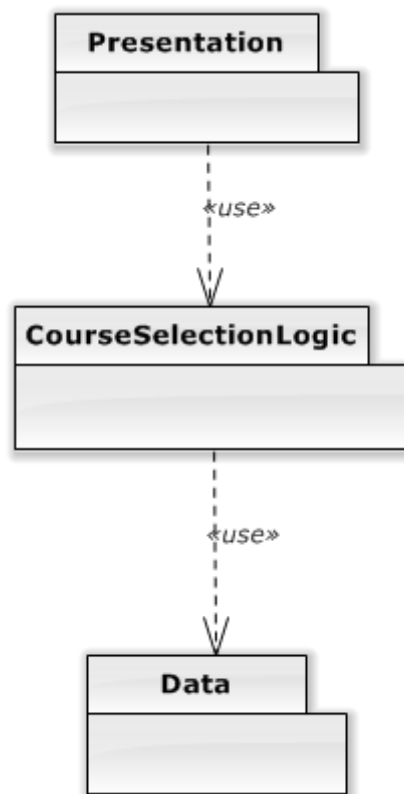


图 1 参考体系结构风格的包图表达逻辑视角

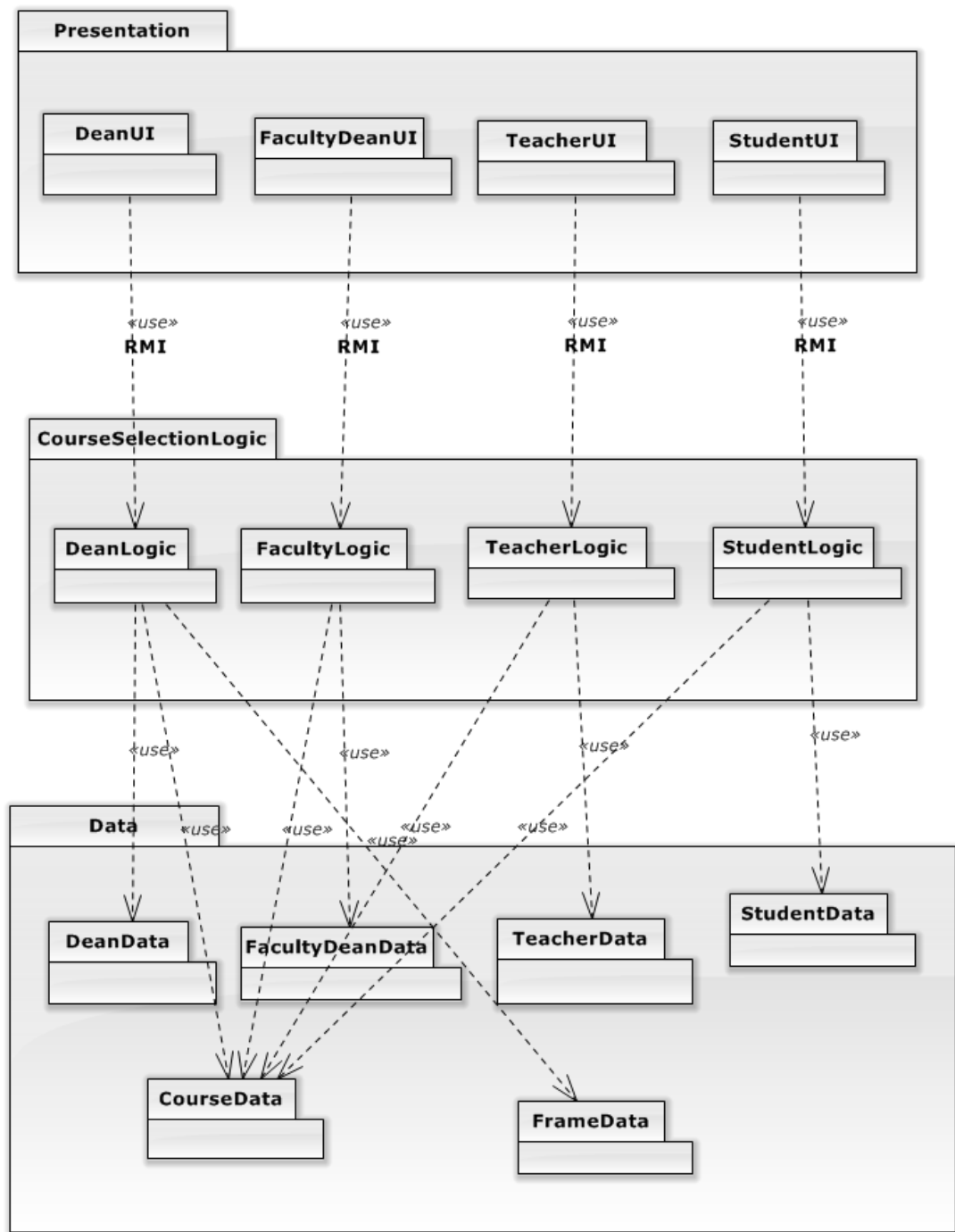


图 2 软件体系结构逻辑设计方案

#### 4. 组合视角

##### 4.1 开发包图

表 1 教务系统的最终开发包设计

开发（物理）包	依赖的其他开发包
com.client.UI.mainUI	loginUI,deanUI,facultyDeanUI,teacherUI,studentUI,vo
com.client.UI.loginUI	vo

com.client.UI.deanUI	logicService,vo
com.client.UI.facultyDeanUI	logicService,vo
com.client.UI.teacherUI	logicService,vo
com.client.UI.studentUI	logicService,vo
com.server.logic.logicService	Java RMI,po
com.server.logic.deanImpl	com.server.logic.deanImpl.method,po
com.server.logic.facultyDeanImpl	com.server.logic.facultyDeanImpl.method,po
com.server.logic.teacherImpl	com.server.logic.teacherImpl.method,po
com.server.logic.StudentImpl	com.server.logic.StudentImpl.method,po
com.server.logic.deanImpl.method	dataService,po
com.server.logic.facultyDeanImpl.method	dataService,po
com.server.logic.teacherImpl.method	dataService,po
com.server.logic.StudentImpl.method	dataService,po
com.server.data.dataService	po
com.server.data.dataImpl	com.server.data.dataImpl.method,po
com.server.data.dataImpl.method	po,JDBC
com.server.data.dataImpl.method.dean	po,JDBC
com.server.data.dataImpl.method.facultyDean	po,JDBC
com.server.data.dataImpl.method.teacher	po,JDBC
com.server.data.dataImpl.method.student	po,JDBC
com.data.po	
com.data.vo	

教务系统的客户端开发包图如图 3 所示，服务器端的开发包图如图 4 所示

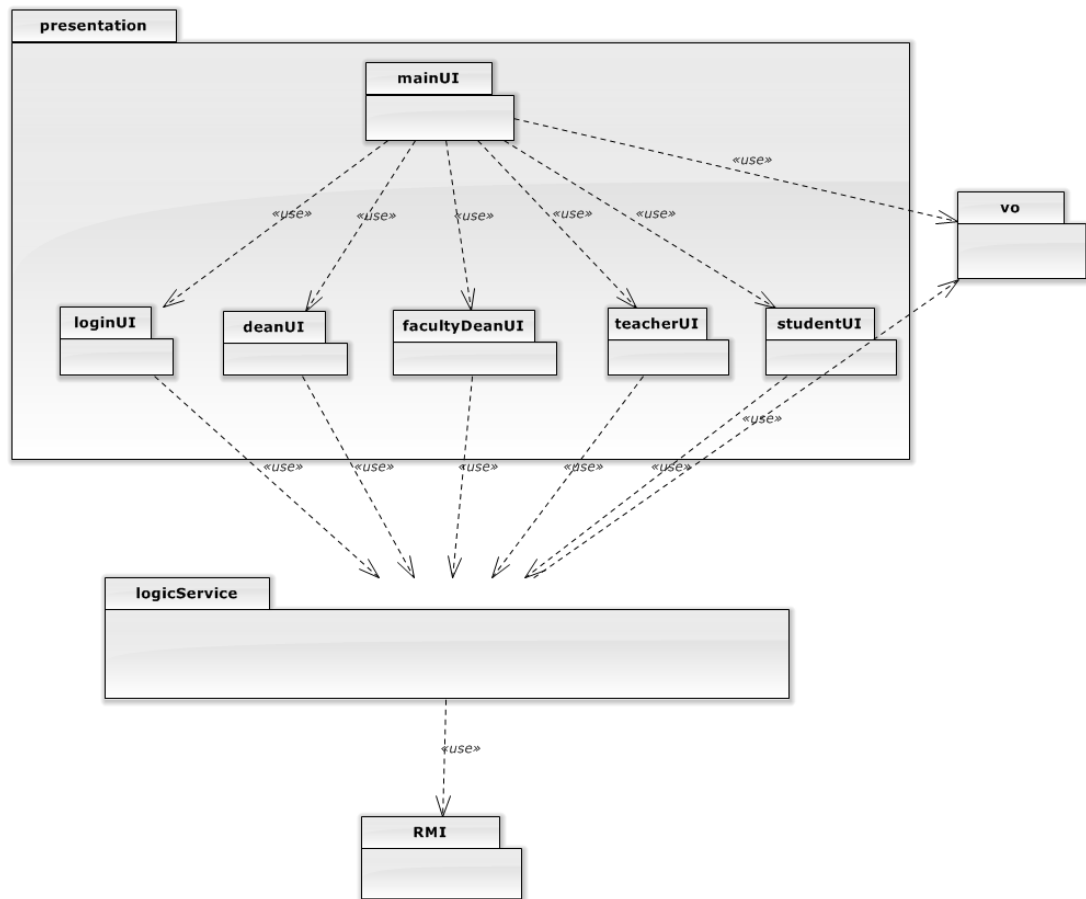


图 3 教务系统客户端开发包图

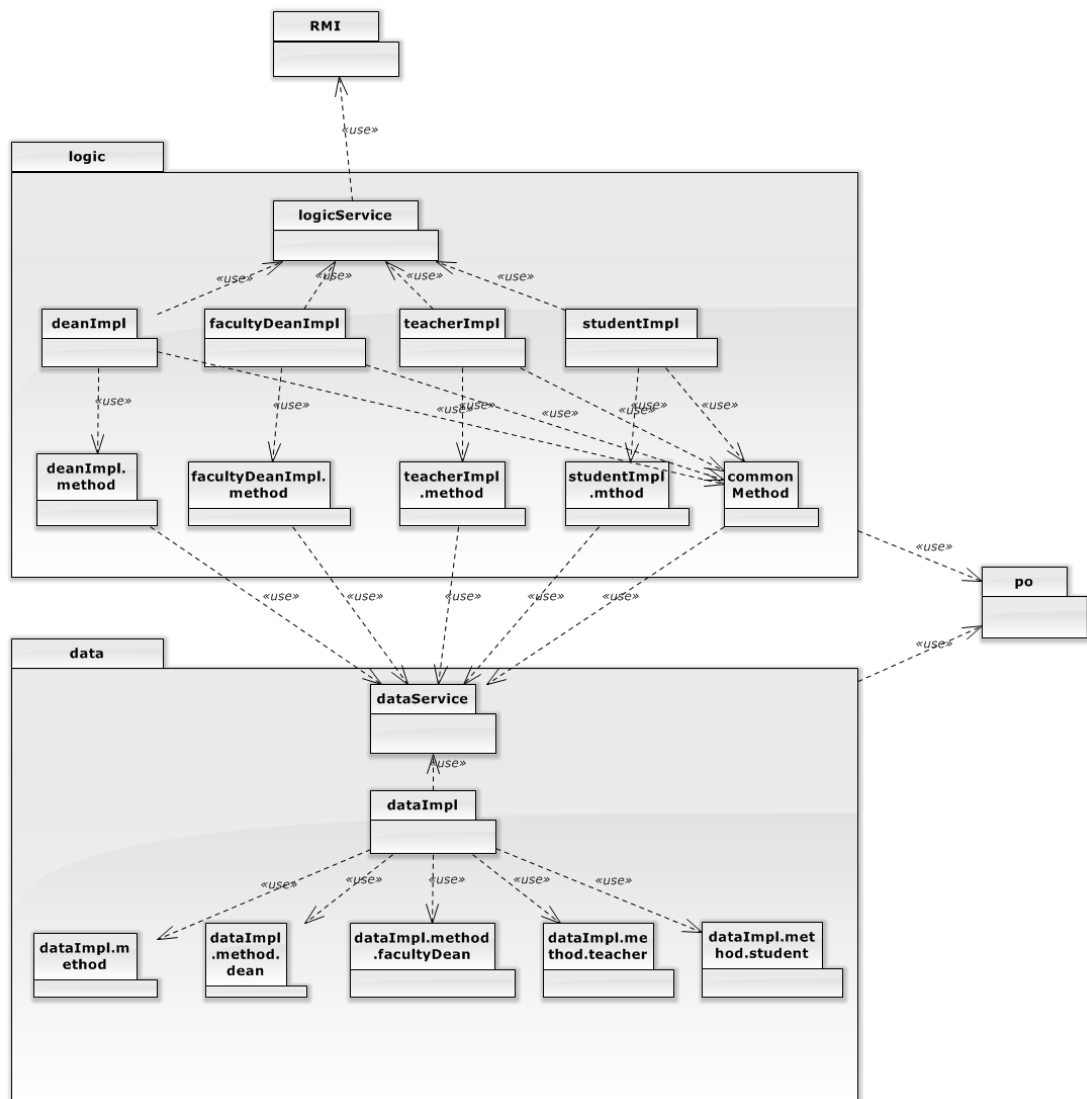


图 4 教务系统服务器端开发包图

## 4.2 运行时进程

在教务系统中，会有多个客户端进程和一个服务器端进程，其进程图如图 5 所示。结合部署图，客户端进程是在客户端机器上运行，服务器端进程是在服务器机器上运行。

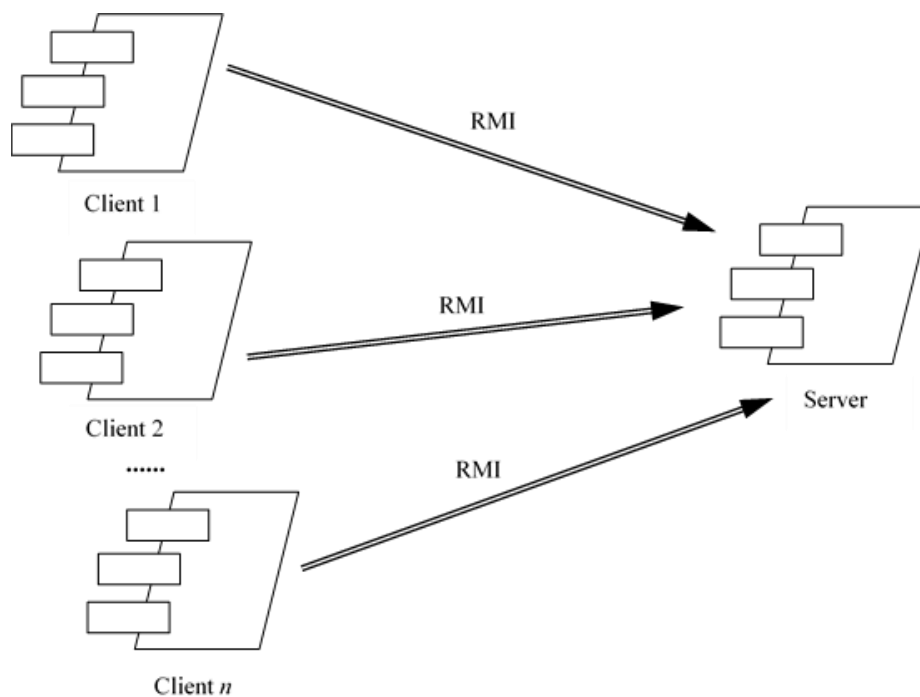


图 5 进程图

### 4.3 物理部署

教务系统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 Java RMI 构件是属于 JDK 6.0 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 6 所示。

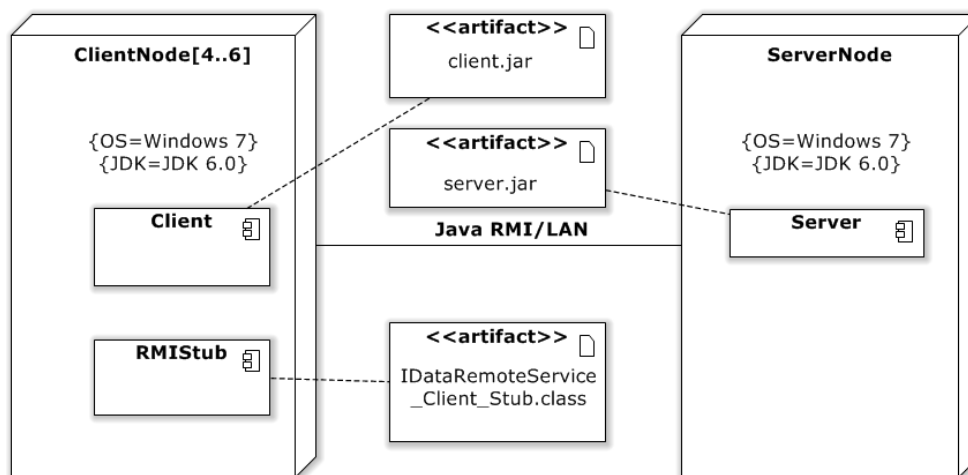


图 6 部署图

## 5. 接口视角

### 5.1 模块的职责

客户端模块和服务器模块试图分别如图 7 和图 8 所示。客户端和服务器端各层的职责分别如表 2 和表 3 所示。



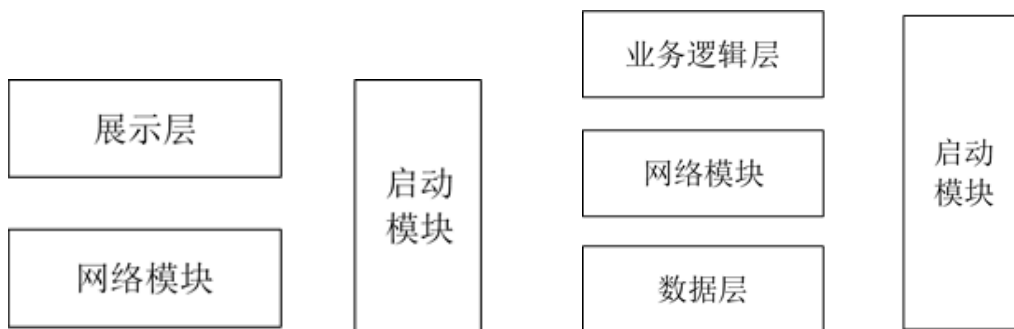


图 7 客户端模块视图

图 8 服务器端模块视图

表 2 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	基于窗口的教务系统客户端用户界面，并对用户的输入进行响应
客户端网络模块	利用 Java RMI 机制查找 RMI 服务

表 3 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
业务逻辑层	对 RMI 传递过来的数据进行业务处理逻辑
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

表 4 层之间调用的接口

接口	服务调用方	服务提供方
logicService	客户端展示层	服务器端逻辑层
dataService	服务器端逻辑层	服务器端数据层

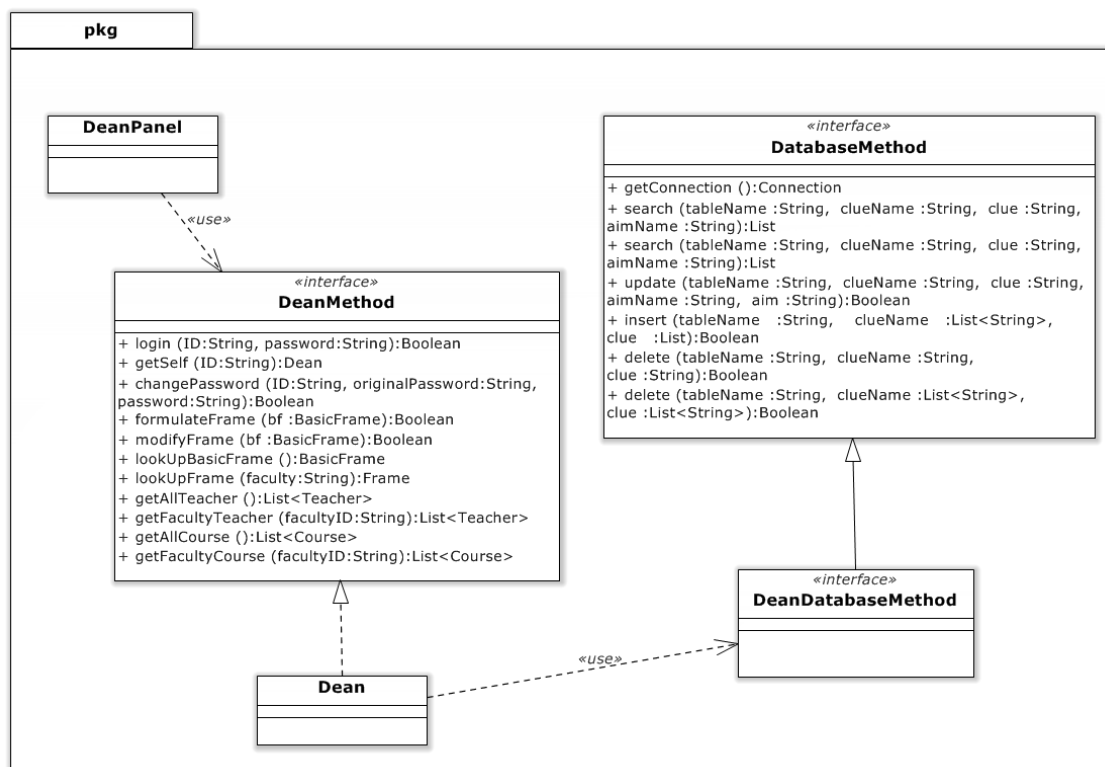


图 9 教员

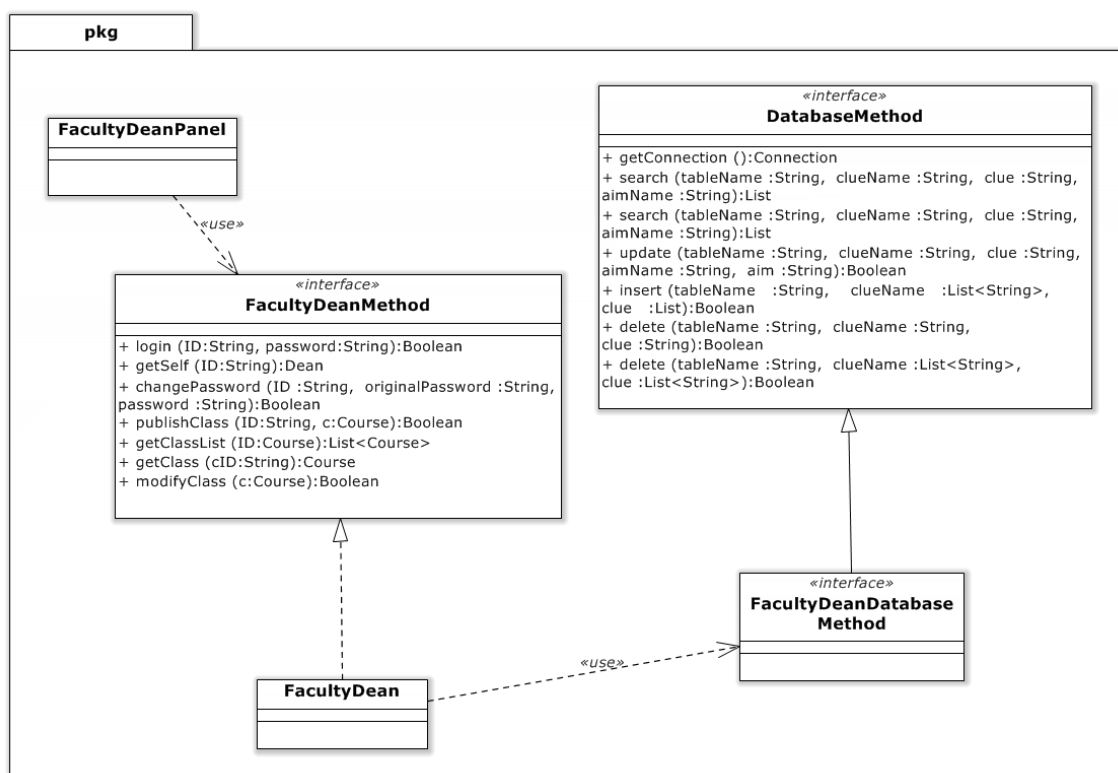


图 10 院系教员

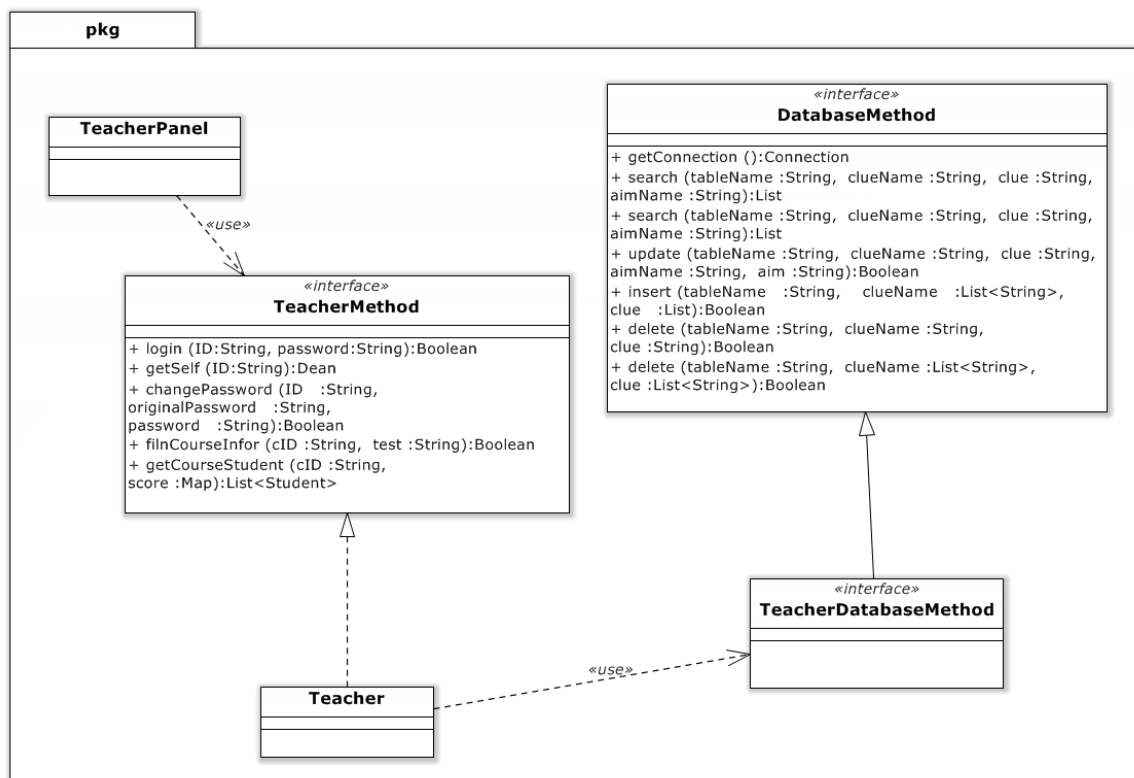


图 11 老师

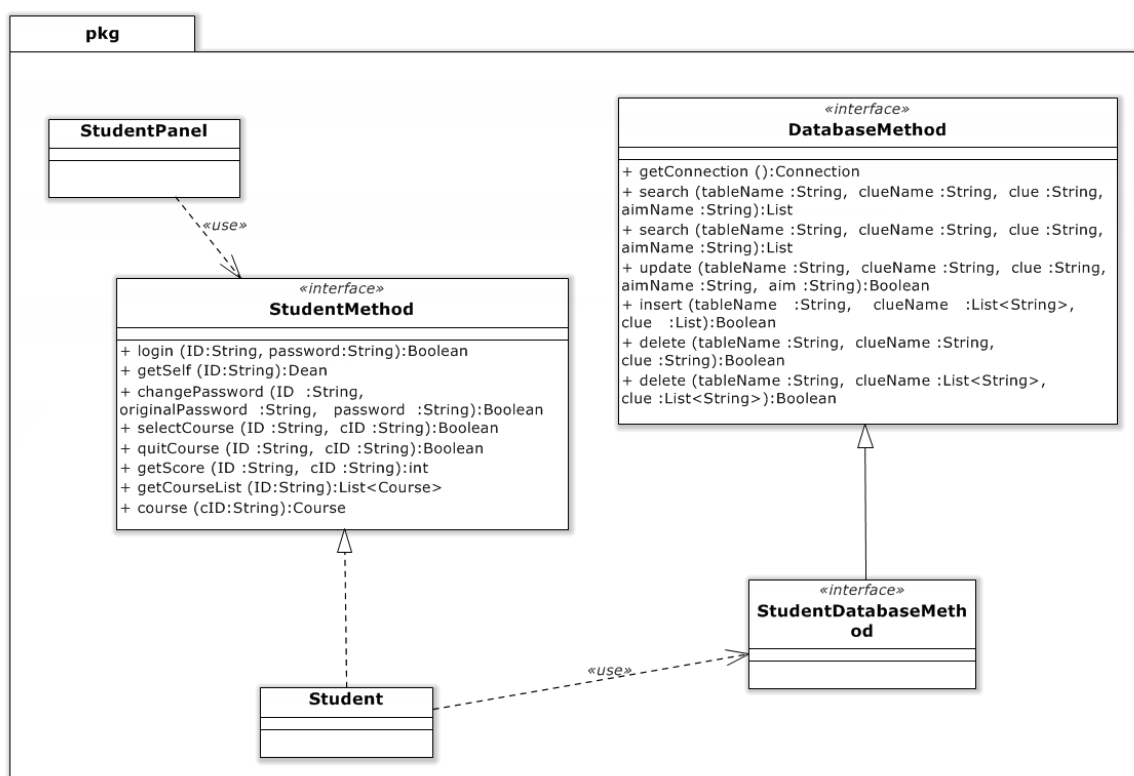


图 12 学生

## 5.2 用户界面层的分解

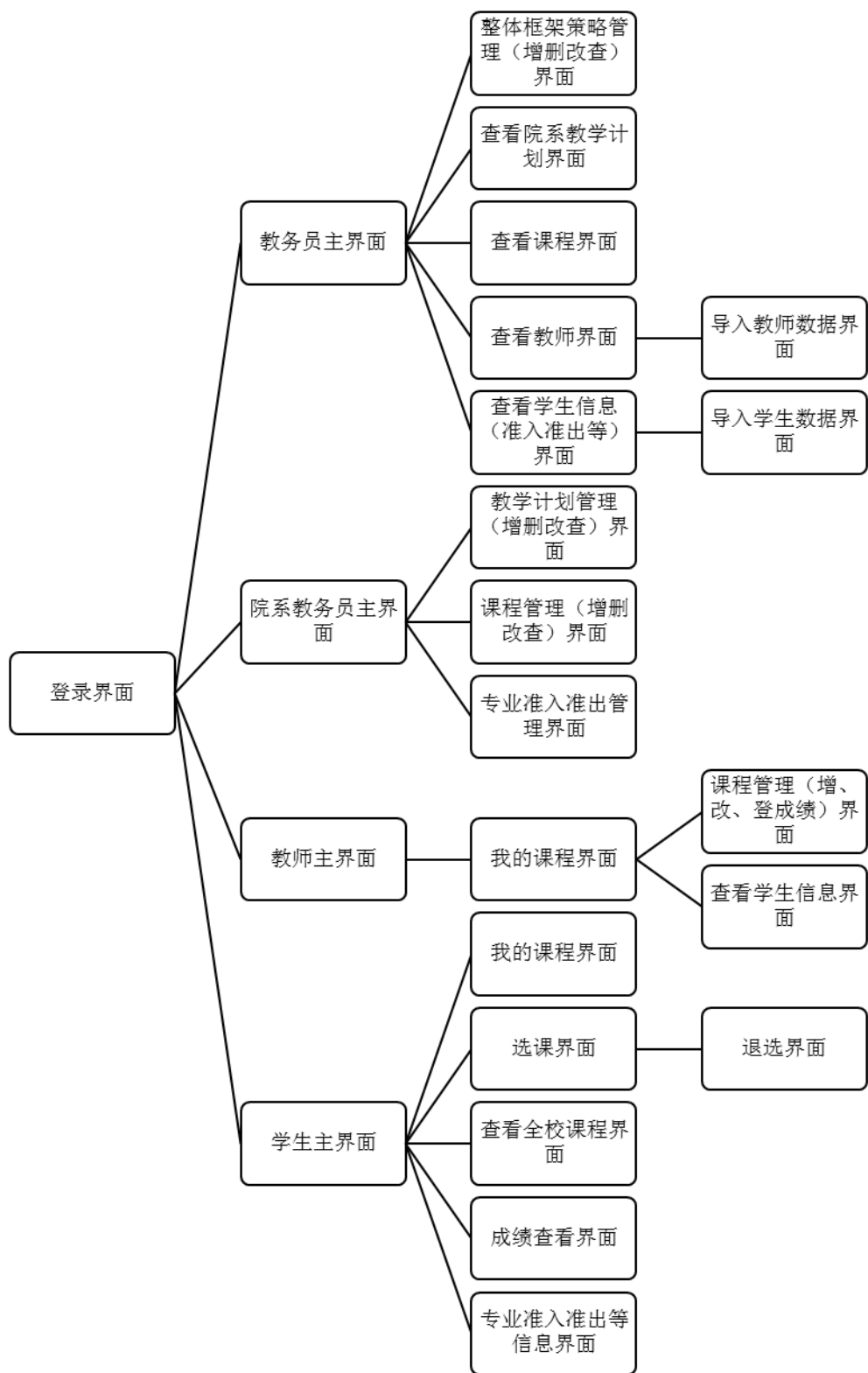


图 13 用户界面跳转

用户界面类如图 14 所示。

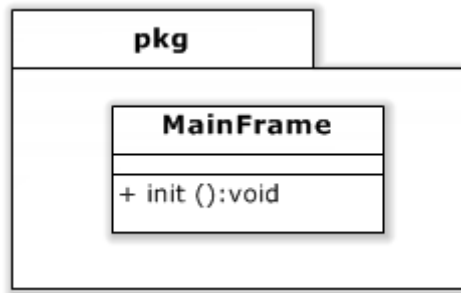


图 14 用户界面类

### 5.2.1 用户界面层模块的职责

表 5

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

表 6

MainFrame	语法	init(args:String[])
	前置条件	无
	后置条件	显示 Frame 和 LoginPanel

表 7

服务名	服务
logicService.DeanMethod	教务员页面的业务逻辑接口
logicService.FacultyDeanMethod	院系教务员页面的业务逻辑接口
logicService.TeacherMethod	教师页面的业务逻辑接口
logicService.StudentMethod	学生页面的业务逻辑接口

### 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。

### 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 8 所示。

表 8 业务逻辑层模块的职责

模块	职责
logicService.DeanMethod	负责教务员任务界面所需的服务
logicService.FacultyDeanMethod	负责院系教务员任务界面所需的服务
logicService.TeacherMethod	负责老师任务界面所需的服务
logicService.StudentMethod	负责学生任务界面所需的服务

### 5.3.2 业务逻辑层模块接口规范

表 9 DeanMethod 模块接口规范

提供的服务（供接口）		
Dean.login	语法	public interface DeanMethod extends Remote
	前置条件	用户输入有效的账号和密码。
	后置条件	根据账号查找是否存在相应的用户，根据账号验证

		密码是否正确，如果通过验证，则登录成功，返回 True 否则，返回 False
Dean.getSelf	语法	public Dean getSelf(String ID) throws RemoteException;
	前置条件	学校教务员已经成功登录
	后置条件	返回该院系教务员信息
Dean.changePassword	语法	public boolean changePassword(String ID, String originalPassword,String password)throws RemoteException;
	前置条件	学校教务员已经登录。且用户输入有效的账号，原密码和现密码。
	后置条件	院系教务员修改密码。根据账号查找是否存在相应的用户，根据账号验证旧密码是否正确，如果通过验证，则将密码修改为现密码，且返回 True，否则，返回 False
Dean.formulateFrame	语法	public boolean formulateFrame(BasicFrame bf)throws RemoteException;
	前置条件	学校教务员已经登录，且用户根据内容进行了有效的输入
	后置条件	建立整体框架，如果创建成功，则返回 True，否则返回 False
Dean.modifyFrame	语法	public boolean modifyFrame(BasicFrame bf)throws RemoteException;
	前置条件	学校教务员已经登录，且用户根据内容进行了有效的输入
	后置条件	学校教务员修改整体框架，如果修改成功，则返回 True，否则返回 False
Dean. lookUpBasicFrame	语法	public BasicFrame lookUpBasicFrame()throws RemoteException;
	前置条件	学校教务员已经登录
	后置条件	学校教务员请求查看整体框架，如果整体框架已创建，则返回该框架。反之，则系统提示错误
Dean.lookUpFrame	语法	public Frame lookUpFrame(String facultyID)throws RemoteException;
	前置条件	学校教务员已经登录，且输入了有效的院系 ID
	后置条件	教务员查看院系教学计划，根据 ID 找到对应的院系，并返回该院系计划，否则系统提示 ID 输入有误
Dean.getAllTeacher	语法	public List<Teacher> getAllTeacher()throws RemoteException;
	前置条件	学校教务员已经成功登录
	后置条件	学校教务员查看全校老师列表，返回一个装有信息的列表
Dean.getFacultyTeacher	语法	public List<Teacher> getFacultyTeacher(String

		facultyID)throws RemoteException;
	前置条件	学校教务员已经登录，且输入了有效的院系 ID
	后置条件	学校教务员查看院系所有老师，根据 ID 找到对应的院系，并返回该院系所有老师列表，否则系统提示 ID 输入有误
Dean.getAllCourse	语法	public List<Course> getAllCourse()throws RemoteException;
	前置条件	学校教务员已经成功登录
	后置条件	学校教务员查看全校课程列表，返回一个装有信息的列表
Dean.getFacultyCourse	语法	public List<Course> getFacultyCourse(String facultyID)throws RemoteException;
	前置条件	学校教务员已经登录，且输入了有效的院系 ID
	后置条件	学校教务员查看院系所有老师，根据 ID 找到对应的院系，并返回该院系所有老师列表，否则系统提示 ID 输入有误
Dean.publishCourse	语法	public boolean publishCourse(Course c)throws RemoteException;
	前置条件	学校教务员已登录，正确输入课程信息
	后置条件	发布课程成功，系统返回 true，否则提示发布失败，返回 false
Dean.modifyCourse	语法	public boolean modifyCourse(Course c)throws RemoteException;
	前置条件	学校教务员已登录，正确查找已发布课程
	后置条件	修改成功，系统返回 true，否则返回 false

需要的服务（需接口）

服务名	服务
DatabaseMethod.getConnection	得到数据库的服务引用
DatabaseMethod.search(String tableName,List clueName,List clue,List aimName)	在数据库中的数据进行搜索
DatabaseMethod.update(String tableName,String clueName,String clue,String aimName,String aim)	更新数据库中信息
DatabaseMethod.insert(String tableName,List<String> clueName,List clue)	向数据库中插入条目
DatabaseMethod.delete(String tableName,List<String> clueName,List<String> clue)	删除数据库中的条目

表 10 FacultyDeanMethod 模块的接口规范

提供的服务（供接口）

FacultyDean.login	语法	public boolean login(String ID,String password) throws RemoteException;
	前置条件	ID,password 符合输入规则
	后置条件	系统判断，当 ID 和 password 匹配 return true, 反之 return false;

FacultyDean.getSelf	语法	public FacultyDean getSelf(String ID) throws RemoteException;
	前置条件	一位院系教务员已经登录
	后置条件	返回该院系教务员信息
FacultyDean.changePassword	语法	public boolean changePassword(String ID,String originalPassword,String password) throws RemoteException;
	前置条件	一位院系教务员已经登录
	后置条件	院系教务员修改密码。传入的原密码若正确 return true, 若原密码错误 return false。
FacultyDean.publishClass	语法	public boolean publishClass(String ID,Course c) throws RemoteException;
	前置条件	一位院系教务员已经登录
	后置条件	发布课程, 若课程已存在, 则发布失败 return false, 反之 return true。
FacultyDean.getClassList	语法	public List<Course> getClassList(String ID) throws RemoteException;
	前置条件	一位院系教务员已经登录
	后置条件	院系教务员查看本院系课程列表 return
FacultyDean.getClass	语法	public Course getClass(String cID) throws RemoteException;
	前置条件	一位院系教务员已经登录
	后置条件	查看任何课程信息
FacultyDean.modifyClass	语法	Public Boolean modifyClass(Course c) throws RemoteException;
	前置条件	一位院系教务员已经登录
	后置条件	修改课程信息

#### 需要的服务（需接口）

服务名	服务
DatabaseMethod.getConnection	得到数据库的服务引用
DatabaseMethod.search(String tableName,List clueName,List clue,List aimName)	在数据库中的数据进行搜索
DatabaseMethod.update(String tableName,String clueName,String clue,String aimName,String aim)	更新数据库中信息
DatabaseMethod.insert(String tableName,List<String> clueName,List clue)	向数据库中插入条目
DatabaseMethod.delete(String tableName,List<String> clueName,List<String> clue)	删除数据库中的条目

表 11 TeacherMethod 模块的接口规范

#### 提供的服务（供接口）

TeacherMethod.login	语法	public boolean login(String ID,String password) throws RemoteException;
	前置条件	无



	后置条件	系统判断, 当 ID 和 password 匹配 return true, 反之 return false;
TeacherMethod.getSelf	语法	public Teacher getSelf(String ID) throws RemoteException;
	前置条件	一位教师已经登录
	后置条件	返回该院系教务员信息
TeacherMethod.changePassword	语法	public boolean changePassword(String ID,String originalPassword,String password) throws RemoteException;
	前置条件	一位教师已经登录
	后置条件	教师修改密码。传入的原密码若正确 return true, 若原密码错误 return false。
TeacherMethod.filnCourseInfor	语法	public boolean filnCourseInfor(String cID,String text) throws RemoteException;
	前置条件	一位老师已经登录
	后置条件	老师填写课程信息, 若成功 return true, 反之 return false
TeacherMethod.getCourseStudent	语法	public List<Student> getCourseStudent(String cID) throws RemoteException;
	前置条件	一位老师已经登录
	后置条件	老师查看选课学生, 返回选择该课程的学生列表。
TeacherMethod.recordScore	语法	public boolean recordScore(String cID,Map score) throws RemoteException;
	前置条件	一位老师已经登录
	后置条件	老师录入成绩

需要的服务（需接口）

服务名	服务
DatabaseMethod.getConnection	得到数据库的服务引用
DatabaseMethod.search(String tableName,List clueName,List clue,List aimName)	在数据库中的数据进行搜索
DatabaseMethod.update(String tableName,String clueName,String clue,String aimName,String aim)	更新数据库中信息
DatabaseMethod.insert(String tableName,List<String> clueName,List clue)	向数据库中插入条目
DatabaseMethod.delete(String tableName,List<String> clueName,List<String> clue)	删除数据库中的条目
DatabaseMethod.getConnection	得到数据库的服务引用

表 12 StudentMethod 模块接口

提供的服务（供接口）

StudentMethod.login	语法	public boolean login(String ID,String password) throws RemoteException;
	前置条件	用户输入有效的账号和密码。
	后置条件	根据账号查找是否存在相应的用户，根据账号验证密码是否正确，如果通过验证，则登录成功，返回 True 否则，返回 False
StudentMethod.getSelf	语法	public Student getSelf(String ID) throws RemoteException;
	前置条件	一位学生已经登录
	后置条件	返回该学生信息
StudentMethod.changePassword	语法	public boolean changePassword(String ID,String originalPassword,String password) throws RemoteException;
	前置条件	一位学生已经登录，且用户输入有效的账号，原密码和现密码。
	后置条件	学生修改密码。根据账号查找是否存在相应的用户，根据账号验证旧密码是否正确，如果通过验证，则将密码修改为现密码，且返回 True，否则，返回 False
StudentMethod.selectCourse	语法	public boolean selectCourse(String ID,String cID) throws RemoteException;
	前置条件	一位学生已经登录，且用户输入有效地学生 ID 和课程 ID
	后置条件	学生选择课程，根据学生 ID 查找是否存在相应的用户，根据课程 ID 查找是否存在相应的课程，如果通过验证，则学生选课成功，且返回 True，否则，返回 False
StudentMethod.quitCourse	语法	public boolean quitCourse(String ID,String cID) throws RemoteException;
	前置条件	一位学生已经登录，且用户输入有效地学生 ID 和课程 ID
	后置条件	学生退选课程，根据学生 ID 查找是否存在相应的用户，根据课程 ID 查找是否存在相应的课程，如果该学生已选择该课程，则退课成功，返回 True，如果该学生未选择该课程，则系统提示该学生未选择该课程，返回 False，其余情况均返回 False
StudentMethod.getScore	语法	public int getScore(String ID,String cID) throws RemoteException;
	前置条件	一位学生已经登录，且用户输入有效地学生 ID 和课程 ID
	后置条件	学生查看课程成绩，根据学生 ID 查找是否存

		在相应的用户，根据课程 ID 查找是否存在相应的课程，如果该学生已选择该课程，则显示学生成绩，返回 True，如果该该课程未登记成绩，则系统提示未登记成绩，返回 False，如果该学生未选择该课程，则系统提示该学生未选择该课程，返回 False，其余情况均返回 False
StudentMethod.getCourseList	语法	public List<Course> getCourseList(String ID) throws RemoteException;
	前置条件	一位学生已经登录，且用户输入有效地学生 ID
	后置条件	学生查看自己的选课列表，根据账号查找是否存在相应的用户，通过验证，则返回选课列表
StudentMethod.getCourseList	语法	public Course course(String cID) throws RemoteException;
	前置条件	一位学生已经登录，且用户输入有效地课程 ID
	后置条件	学生查看自己的选课列表，根据课程 ID 查找是否存在相应的课程，通过验证，则返回课程信息

#### 5.4 数据层的分解

数据层主要给业务逻辑层提供数据访问服务，包括对持久画数据的增、删、改查。

Dean 业务逻辑需要的服务由 DeanDatabaseMethod 接口提供，FacultyDean 业务逻辑需要的服务由 FacultyDeanDatabaseMethod 接口提供，Teacher 业务逻辑需要的服务由 TeacherDatabaseMethod 提供，Student 业务逻辑需要的服务由 StudentDatabaseMethod 提供。数据层模块的描述具体如图 15 所示。

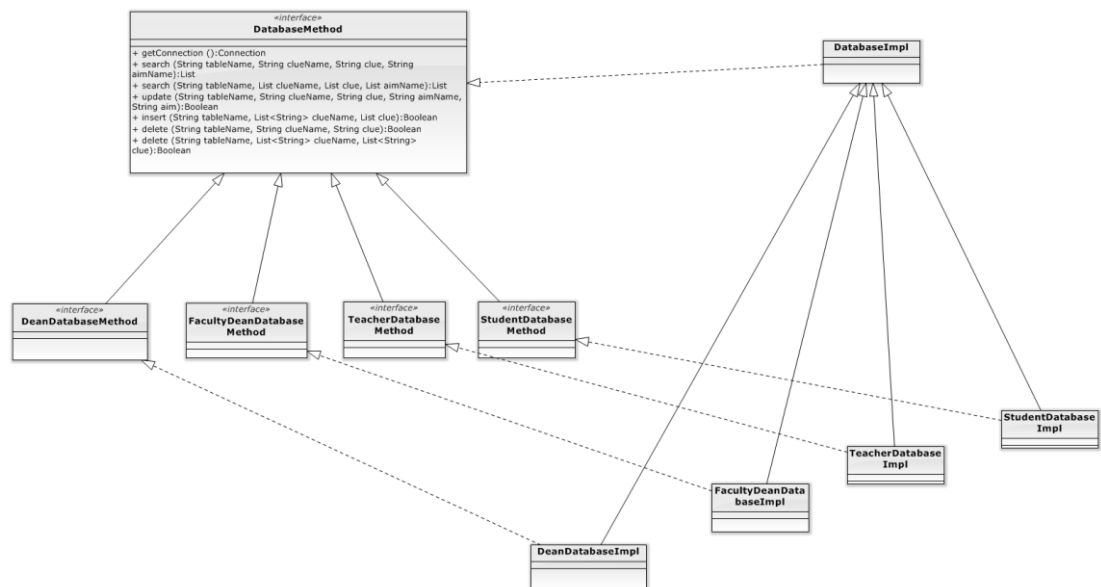


图 15

##### 5.4.1 数据层模块的职责

数据层模块的职责如表 13 所示

表 13 数据层模块的职责

模块	职责
DatabaseMethod	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
DeanDatabaseMethod	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
FacultyDeanDatabaseMethod	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
TeacherDatabaseMethod	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
StudentDatabaseMethod	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
DatabaseImpl	基于 MySQL 数据库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
DeanDatabaseImpl	基于 MySQL 数据库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
FacultyDeanDatabaseImpl	基于 MySQL 数据库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
TeacherDatabaseImpl	基于 MySQL 数据库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
StudentDatabaseImpl	基于 MySQL 数据库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

数据层模块的接口规范如表 14 所示。

表 14 数据层模块的接口规范

提供的服务（供接口）		
DatabaseMethod.getConnection	语法	public Connection getConnection()
	前置条件	无
	后置条件	初始化持久数据库
DatabaseMethod.search	语法	public List search(String tableName,String clueName,String clue,String aimName)
	前置条件	无
	后置条件	按搜索条件返回相应的结果
DatabaseMethod.search	语法	public List search(String tableName,List clueName,List clue,List aimName)
	前置条件	无
	后置条件	按搜索条件返回相应的结果
DatabaseMethod.update	语法	public boolean update(String tableName,String clueName,String clue,String aimName,String aim)
	前置条件	数据库存在 aimName 属性上与 aim 相同的数据

	后置条件	更新在 aimName 属性上与 aim 相同的数据
DatabaseMethod.insert	语法	public boolean insert(String tableName,List<String> clueName,List clue)
	前置条件	数据库里不存在相同的数据
	后置条件	在数据库中插入一个纪录
DatabaseMethod.delete	语法	public boolean delete(String tableName,String clueName,String clue)
	前置条件	无
	后置条件	数据库删除与条件匹配的纪录
DatabaseMethod.delete	语法	public boolean delete(String tableName,List<String> clueName,List<String> clue)
	前置条件	无
	后置条件	数据库删除与条件匹配的纪录

## 6. 信息视角

### 6.1 数据持久化对象

Dean 类包含教务员的 ID、姓名。

FacultyDean 类包含院系教务员的 ID、姓名、院系。

Teacher 类包含老师的 ID、姓名、院系。

Student 类包含学生的 ID、姓名、院系、入学年份。

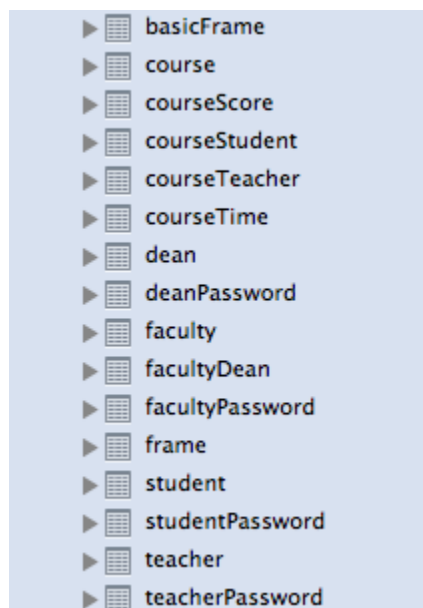
BasicFrame 类包含 A、B、C、D、E、F、L 的学分范围和时间范围。

Frame 类包含 A、B、C、D、E、F、L 的学分。

持久化用户对象的定义参见 com.data.po 包。

### 6.2 数据库表

表 15 数据库表



▶ basicFrame
▶ course
▶ courseScore
▶ courseStudent
▶ courseTeacher
▶ courseTime
▶ dean
▶ deanPassword
▶ faculty
▶ facultyDean
▶ facultyPassword
▶ frame
▶ student
▶ studentPassword
▶ teacher
▶ teacherPassword