**设计说明书**

1. **引言**
   1. **编写目的**

由前面的需求分析，得出了系统的基本需求，要实现整个系统，需要对用户的需求进行设计，概要设计主要是利用比较抽象的语言对整个需求进行概括，确定对系统的物理配置，确定整个系统的处理流程和系统的数据结构，接口设计，实现对系统的初步设计。预期读者为客户代表及程序设计人员。

* 1. **背景**

随着信息化时代的到来，我校正准备筹建课程作业管理平台。为了能够更好的方便老师，学生日常提交，批阅作业的功能。

* 1. **定义**

SQL Server：数据库管理软件

DBMS：数据库管理系统

Windows XP及以上版本：运行环境

Java：软件开发语言

* 1. **参考资料**

《软件工程》 舒坚、陈斌全 编 高等教育出版社

1. **总体设计**
   1. **需求规定**

课程作业管理平台使用者为学生与教师、管理人员三类。学生通过它来提交、查询自己的作业，教师则通过它来达到统计作业详情、发布、批改、取消、通知等目的。系统主要包括作业发布。作业提交、作业处理、作业查询、作业统计、用户管理。

* 1. **运行环境**

数据库管理系统软件：MySQL

运行平台：Windows XP及以上版本

分辨率：最佳效果1027\*768像素

* 1. **基本设计概念和处理流程**

**顶层数据流图：**



* 1. **体系结构**



针对课程作业管理系统，可以分为两大模块：用户管理子系统、课程管理子系统、作业管理子系统。

**在用户管理子系统中：**

1. 管理员由校方注册，老师、学生由管理员注册。
2. 管理员由校方注销，老师、学生由管理员注销。
3. 用户类型设置模块：即设置该用户是管理员还是老师还是学生。

**在课程管理子系统中：**

1. 添加课程模块：

该模块主要是处理管理员添加课程，指定老师、学生、班级与课程之间的关系。

1. 删除课程模块：

该模块用于删除一个错误的课程。

1. 修改课程模块：

该模块用于对一门已添加的课程进行修改：修改任课老师、学生、班级等等。

1. 查询课程模块：

该模块主要是处理用户对一门课程的详细信息进行查询时信息。

**在作业管理子系统中：**

1. 发布作业模块：

该模块主要是处理老师发布指定时间、内容，规定作业人群的作业的问题。

1. 批改作业模块：

该模块主要是处理老师对一份作业进行批改、评分。

1. 统计作业模块：

该模块主要是对提交作业数、作业分数、作业排名信息的统计。

1. 查询作业模块：

该模块主要是对作业内容、作业成绩的查询。

* 1. **人工处理过程**

无。

* 1. **尚未问决的问题**

无。

1. **接口设计**
   1. **用户接口**

采用窗口化，菜单式进行设计，在操作时响应热键。

* 1. **外部接口**

**硬件接口**

无。

**软件接口**

通过JDBC对MYSQL数据库的连接

* 1. **内部接口**

通过面向对象语言设计类，在public类中实现调用，类间实现严格封装。

1. **运行设计**
   1. **运行模块组合**

本程序主要是以一个窗口为模块，一般一个窗口完成一个特定的功能，主窗口通过打开另一个子窗口来实现各个模块之间不同功能的连接和组合。各模块之间相对独立，程序的可移植性好。各模块之间主要以传递数据项的引用来实现模块之间的合作和数据共享。

* 1. **运行控制**

只要符合操作说明书，用户可自由控制，不额外限定用户输入，异常由程序内部进行处理，给出相应的提示信息。

* 1. **运行时间**

由用户决定，但每次操作响应时间上限应控制在1S以内。

* 1. **系统数据结构设计**
     1. **逻辑结构设计要点**
        1. **管理员文件表**

管理员（管理员编号，姓名，性别，联系方式，注册时间）

* + - 1. **老师文件表**

老师（老师编号，姓名，性别，联系方式，注册时间）

* + - 1. **学生文件表**

学生（学生编号，姓名，性别，联系方式，注册时间）

* + - 1. **班级文件表**

班级（班级编号，学生编号）

* + - 1. **课程文件表**

课程（课程编号，课程名，班级编号，老师编号）

* + - 1. **作业文件表**

作业（作业编号，作业主题，作业内容，学生编号，成绩）

* 1. **数据结构与程序的关系**

数据结构为关系型数据库，所以，在程序中可以通过标准的SQL语句与数据结构进行交互，交互过程中采用通用的数据访问接口。为了保持良好的程序架构，对数据库访问采用DAO设计模式实现，提高维护性及扩展性。

1. **类设计**
   1. **领域模型**



* 1. **UML类图**
     1. **用户类：**



* + 1. **管理员类：**

继承自用户类

* + 1. **老师类：**

继承自用户类

* + 1. **学生类：**

继承自用户类

* + 1. **班级类：**



* + 1. **课程类：**



* + 1. **作业类：**



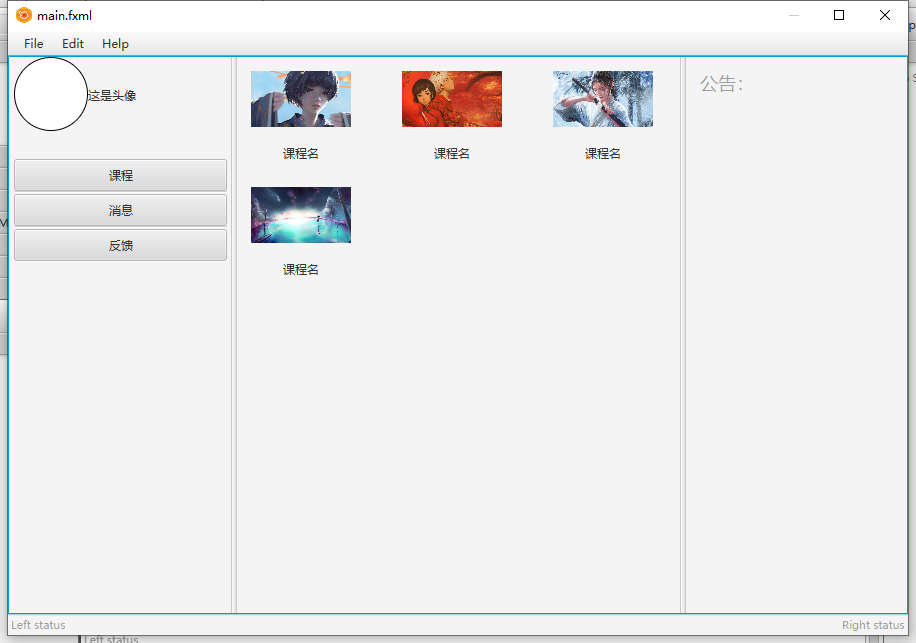
* + 1. **类之间关系：**



1. **界面设计**
   1. **登陆界面：**



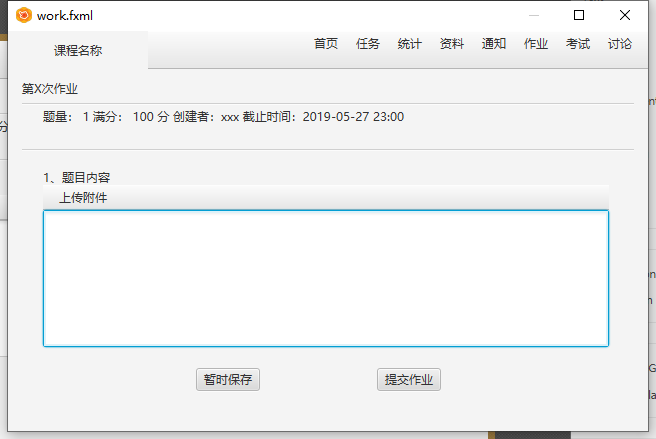
* 1. **主界面：**



* 1. **课程界面：**



* 1. **作业界面：**



1. **系统出错处理设计**
   1. **出错信息**

本程序多处采用了异常处理的机制，当遇到异常时不但能及时的处理，保证程序的安全性和稳定性，而且各种出错信息能通过弹出对话框的形式，及时的告诉用户出错的原因及解决的方法，使用户以后能够减少错误的发生。程序的大部分地方还采取了出错保护，如输入内容的长度和类型等减少了用户出错的可能。

* 1. **补救措施**

我们对于本程序的几种可能的错误进行了分析，分别进行了不同的处理。

主要的错误可能有：

* 1. 数据库连接错误：

这类错误主要时数据库设置不正确，或SQL Server异常引起的，我们只要取消本次操作，提醒用户检查数据库问题即可。

* 1. 输入错误：

这主要是用户输入不规范造成的，我们在尽量减少用户出错的条件的情况下，主要也是通过对话框，提醒用户，然后再次操作。

* 1. 其他操作错误：

对于用户的不正当操作，有可能使程序发生错误。我们主要是中止操作，并提醒用户中止的原因和操作的规范。

* 1. 其他不可预知的错误：

程序也会有一些我们无法预知或没考虑完全的错误，我们对此不可能作出完全的异常处理，这时我们主要要保证数据的安全，所以要经常的进行数据库备份，并能及时的和我们联系，以逐步的完善我们的程序。

* 1. **系统维护设计**

软件的维护主要包括，数据库的维护和软件功能的维护。

对于数据库的维护，本软件已经提供了数据库的备份和恢复的功能，可以方便的实现数据库的维护管理。

对于软件功能方面的维护，由于我们采用的是模块化设计方法，每个模块（窗口）之间相互独立性较高，这样对软件的维护带来了很大的方便，对于单独功能的修改只需修改一个窗口就行了。而对于功能的添加，只要再添加菜单项的内容即可。