|  |  |  |
| --- | --- | --- |
| 南农大  **数据库课程设计报告**  XH2 | | |
|  | 题 目: | 学生作业管理系统 |
|  | 组 长: | 韩秉伸 学号: 19319129 |
|  | 组 员 | 陈宜韬 学号: 19319119 |
|  |  | 臧柯 学号: 19319130 |
|  |  | 范依鹏 学号: 19319120 |
|  | 学 院: | 人工智能学院 |
|  | 专 业: | 网络工程 |
|  | 班 级: | 网工191 |
|  | 指导教师: | 黄芬 职称: 副教授 |
| 2021年5月1日 | | |

目录

**[1 课题分析](#_Toc10280515)** [3](#_Toc10280515)

[1.1 课题背景 3](#_Toc10280516)

[1.2 研究目的及意义 3](#_Toc10280517)

[1.3 国内外研究现状 4](#_Toc10280518)

[1.4 本文所做的工作 4](#_Toc10280519)

**[2 开发技术、开发工具及技术路线](#_Toc10280520)** [5](#_Toc10280520)

[2.1 开发技术及开发工具 5](#_Toc10280521)

**[2.1.1 开发技术](#_Toc10280522)** [5](#_Toc10280522)

**[2.1.2 开发工具](#_Toc10280523)** [5](#_Toc10280523)

[2.2 技术路线 6](#_Toc10280524)

**[3 功能分析与设计](#_Toc10280525)** [7](#_Toc10280525)

[3.1 需求分析 7](#_Toc10280526)

**[3.1.1 用户调研](#_Toc10280527)** [7](#_Toc10280527)

**[3.1.2 初步设计](#_Toc10280528)** [7](#_Toc10280528)

[3.2 系统设计 7](#_Toc10280529)

**[3.2.1 系统流程设计](#_Toc10280530)** [7](#_Toc10280530)

**[3.2.2 系统功能分析](#_Toc10280531)** [7](#_Toc10280531)

[3.3 数据库概念结构设计 8](#_Toc10280532)

[3.4 数据库逻辑结构设计 8](#_Toc10280533)

**[3.4.1 E-R图向关系模型的转换](#_Toc10280534)** [9](#_Toc10280534)

**[3.4.2 利用数据依赖的理论对关系模式进行分析](#_Toc10280535)** [9](#_Toc10280535)

[3.5 数据库物理结构设计 9](#_Toc10280536)

[3.6 模块划分 9](#_Toc10280537)

**[4 实现](#_Toc10280538)** [1](#_Toc10280538)1

[4.1 前端界面实现 11](#_Toc10280539)

**[4.2 用户界面实现](#_Toc10280540)** [11](#_Toc10280540)

[4.3 后端代码实现 12](#_Toc10280542)

**[4.3.1 C#与SQL Server的连接和交互](#_Toc10280543)** [12](#_Toc10280543)

**[4.3.2 C#后台代码文件以及数据库脚本文件关系图](#_Toc10280544)** [14](#_Toc10280544)

**[4.3.3 SQL Server建立数据库](#_Toc10280546)** [1](#_Toc10280546)5

**[5 系统测试与应用](#_Toc10280548)** [1](#_Toc10280548)6

[5.1 系统测试运行环境 1](#_Toc10280549)6

[5.2 数据库测试与运行 1](#_Toc10280550)7

**[5.2.1 用户登录](#_Toc10280551)** [1](#_Toc10280551)7

**[5.2.2 作业任务发布](#_Toc10280552)** [1](#_Toc10280552)7

**[5.2.3 作业查询](#_Toc10280553)** [1](#_Toc10280553)8

**[5.2.4 作业提交](#_Toc10280554)** [1](#_Toc10280554)8

**[5.2.5 教师批改](#_Toc10280555)** [1](#_Toc10280555)9

**[6 总结与展望](#_Toc10280558)** [1](#_Toc10280558)9

[6.1 总结 1](#_Toc10280559)9

[6.2 不足 1](#_Toc10280560)9

[6.3 展望 1](#_Toc10280561)9

学生作业管理系统

摘要： 本课程设计报告详细地描述了本小组开发出的一套“学生作业管理系统”软件，从课题背景入手，分析了开发的目的和意义；接着阐明了本设计的开发技术路线以及功能分析与设计，包括需求分析、系统设计、概念结构设计等；最后依据开发路线和设计实现了管理系统，并对应用程序和数据库进行了测试；

关键词：大学生；管理系统；数据管理；数据上传；C#；SQL server

引言：随着大学生课程科目的丰富，当代大学生所要做的作业也变得种类繁多了起来。随着网络技术的发展以及物质生活水平的提高，人们习惯于借助各种平台来满足自己的需求，但是我们并没有发现一个平台可以帮助我们大学生对自己各科目的作业进行管理以及帮助老师更方便的管理学生的作业。如果我们能够让同学们和老师通过这样一个平台将作业进行合理管理，帮助同学们查找自己的作业完成进度以及提交作业，帮助老师们更方便地向同学们发布作业内容以及对同学们的作业进行收交，那么相信同学们能更从容地安排自己的作业计划，老师们能更快捷地向同学们完成作业互动，提高效率，便利生活。

**1 课题分析**

1.1 课题背景

在校园生活中，同学们有时会遇到一些作业管理方面的小需求，希望能够更方便的找到自己需要完成的作业，并及时向老师完成作业提交，例如作业种类很多，收交的时间也各不相同，希望快速建立一个作业完成计划表；作业写完了但是忘了交，结果错过截止时间；学委想要统计作业收交情况，将信息汇总给老师，希望能够简化其中的步骤；老师突然想布置新的作业……当前大学生类似这样的需求相当普遍。同学们需要一个平台来满足自己的需求，也可以通过解决他人的需求获得收益。

在同学们想要满足这样的需求，通常借助QQ群、微信雨课堂、超星学习通等平台，这样的方式有着操作界面有时稍显繁琐，功能性不强，查找不方便，信息不够集中的缺点。现在没有一个合适的平台来解决同学们的这种需求，于是我们想到能否做出一个学生作业管理系统供同学们更简单的管理自己的作业，并分门别类地规划自己的学习计划，与此同时，这个平台是双向的，老师和学委也可以利用这个平台来进行作业信息的管理和更新，实现高效率学习和作业管理。在帮助学生的同时，也能为老师和学委提供方便，不仅节省了时间，提高效率，并且让用户确实感受到方便后，平台注册人数就会更多，进而继续促进平台的运作。

1.2 研究目的及意义

本课程设计的目的在于综合运用所学程序设计语言以及数据库原理的知识，开发实现上文所述的学生作业管理系统，进行测试并运行，以确保平台的可使用性。在开发的过程中复习所学的知识，并在实践过程中积累项目开发经验。

同时将本平台在同学中进行推广和使用，切实解决同学们的需求，方便同学们的学习，提高学习效率。

## 1.3 国内外研究现状

在国内校园中，有一些网络平台可以实现本平台预期开发的一部分功能，例如微信公众号“雨课堂”，可以实现老师向学生发布作业的功能，再例如手机应用程序“超星学习通”，也可以实现类似的功能并且已经有部分同学开始使用。还有一些应用程序的部分功能与本平台预期开发的功能有一定的重合，例如手机微信小程序“校园e键通”，其中“课表查询”可实现“课表查询”的功能。

在国外校园，从网络调查中我们得知，由于国外计算机起步较早，一些对学生的信息管理系统应运而生，随着时间的推移，这些平台的功能也愈加完善，除了最初“信息管理”的功能以外，也扩展了许多“成绩查询”等服务学生的功能。

总而言之，尽管国内外已有不少平台实现了本平台预取开发部分功能或者与部分预期开发功能重合，但是并没有一个完善的，功能丰富的，能覆盖大部分学生需求的平台，因此，本平台的开发有着一定的创新意义以及较大的现实意义。

1.4 本文所做的工作

本文数据库系统的相关理论用于实践，在需求分析以及系统开发的过程中，具体所做的工作有：

1. 在该系统的可行性分析和规划阶段，对于管理系统的基本功能做了简单的分析了解，和实现平台的技术路线。
2. 在该系统的需求分析和总体设计阶段，给出了系统实现的功能模块和基本架构。
3. 在详细设计和具体实施阶段完成了对于互助平台的功能模块设计、功能设计、数据库设计，并完成实现了具体的功能。
4. 在原有的基础上对互助平台进行错误的修改与功能的优化，提高用户使用的便捷性。

**2 开发技术、开发工具及技术路线**

2.1 开发技术及开发工具

2.1.1 开发技术

本平台是基于C#开发的，后台数据库采用Microsoft SQL server。

（1）C#

C#（C sharp）是[Microsoft公司](https://baike.baidu.com/item/%E5%BE%AE%E8%BD%AF%E5%85%AC%E5%8F%B8/732128" \t "_blank)发布的一种面向对象的、运行于[.NET Framework](https://baike.baidu.com/item/.NET%20Framework" \t "_blank)和.NET Core（完全开源，跨平台）之上的高级程序设计语言。它是Microsoft公司.NET windows网络框架的主角。其语法风格源自C/C++家族：继承了C的过程化编程语言和C++面向对象思想的强大，融合了Visual Basic的高效，具有易上手、功能强大的特性，一经推出便深受世界各地程序员的好评和喜爱。

C#是一种由事件驱动、可视化的编程语言，结合了[Visual B](https://baike.baidu.com/item/VB" \t "_blank)asic简单的可视化操作，以其强大的操作能力、优雅的语法风格、创新的语言特性和便捷的面向[组件](https://baike.baidu.com/item/%E7%BB%84%E4%BB%B6" \t "_blank)编程的支持成为[.NET](https://baike.baidu.com/item/.NET" \t "_blank)开发的首选语言。它使得程序员可以快速地编写基于Microsoft .NET平台的应用程序，Microsoft .NET提供了一系列的[工具](https://baike.baidu.com/item/%E5%B7%A5%E5%85%B7" \t "_blank)和服务来最大程度地开发利用计算与通讯领域。

C#使得C++程序员可以高效的开发程序，且因可调用由 C/C++ 编写的本机原生函数，而绝不损失C/C++原有的强大的功能。因为这种继承关系，C#与C/C++具有极大的相似性，熟悉类似语言的开发者可以很快的转向C#。

（2）SQL server

SQL是英文Structured Query Language的缩写，意思为[结构化查询语言](https://baike.baidu.com/item/%E7%BB%93%E6%9E%84%E5%8C%96%E6%9F%A5%E8%AF%A2%E8%AF%AD%E8%A8%80/10450182" \t "_blank)。SQL语言的主要功能就是同各种数据库建立联系，进行沟通。按照ANSI（[美国国家标准协会](https://baike.baidu.com/item/%E7%BE%8E%E5%9B%BD%E5%9B%BD%E5%AE%B6%E6%A0%87%E5%87%86%E5%8D%8F%E4%BC%9A" \t "_blank)）的规定，SQL被作为[关系型数据库管理系统](https://baike.baidu.com/item/%E5%85%B3%E7%B3%BB%E5%9E%8B%E6%95%B0%E6%8D%AE%E5%BA%93%E7%AE%A1%E7%90%86%E7%B3%BB%E7%BB%9F/696511" \t "_blank)的标准语言。SQL Server是由Microsoft开发和推广的[关系数据库管理系统](https://baike.baidu.com/item/%E5%85%B3%E7%B3%BB%E6%95%B0%E6%8D%AE%E5%BA%93%E7%AE%A1%E7%90%86%E7%B3%BB%E7%BB%9F/11032386" \t "_blank)（RDBMS）。

SQL Server是由Microsoft开发和推广的关系型数据库，它最初是由Microsoft、Sybase和Ashton-Tate三家公司共同开发的，并于1988年推出了第一个OS/2版本。目前最新版本是2017年份推出的SQL Server 2017，也就是本课程设计使用的版本。

所谓数据库，就是存储数据的地方，需要长期存储在计算机中。数据库中的数据按照一定的数据模型组织、存储和描述，具有较小的冗余度、较高的数据独立性和易扩展性、可以为不同的用户所共享。

在诸多类型的数据库中，关系数据库由于其建立在严格的数学概念基础之上，概念单一，结构简单、清晰、用户易懂易用，存取路径等物理操作对用户透明等特点，成为当今的最主流数据库技术。

关系数据库是由数据表和表之间的关联组成的。这其中的数据表通常是一个由列和行组成的二维表。每一行乘坐一个元组，每一列称为属性，属性是数据库表中存储对象的共有属性。

对数据库的操作可通过数据库管理系统（Database Management System，简称DBMS）实现，它是数据库的核心组成部分，用户对数据库的一切操作，比如定义、查询、更新、插入、添加触发器等控制都是通过DBMS来进行操作的。Microsoft公司也为SQL Server开发了一套专属的DBMS——SQL Server Management Studio（SSMS），本课程设计使用的版本是17.9.1。

2.1.2 开发工具

本系统在Windows 10操作系统上开发，采用Microsoft Visual Studio 2017集成开发环境进行开发，数据库管理系统采用Microsoft SQL Server Management Studio 17.9.1。

（1）Microsoft Visual Studio

Microsoft Visual Studio，简称VS。VS是美国[微软公司](https://baike.baidu.com/item/%E5%BE%AE%E8%BD%AF%E5%85%AC%E5%8F%B8" \t "_blank)的开发工具包系列产品。VS是一个基本完整的开发工具集，它包括了整个[软件生命周期](https://baike.baidu.com/item/%E8%BD%AF%E4%BB%B6%E7%94%9F%E5%91%BD%E5%91%A8%E6%9C%9F" \t "_blank)中所需要的大部分工具，如UML工具、代码管控工具、[集成开发环境](https://baike.baidu.com/item/%E9%9B%86%E6%88%90%E5%BC%80%E5%8F%91%E7%8E%AF%E5%A2%83" \t "_blank)（IDE）等等。所写的[目标代码](https://baike.baidu.com/item/%E7%9B%AE%E6%A0%87%E4%BB%A3%E7%A0%81/9407934" \t "_blank)适用于微软支持的所有平台，包括[Microsoft Windows](https://baike.baidu.com/item/Microsoft%20Windows" \t "_blank)、[.NET Framework](https://baike.baidu.com/item/.NET%20Framework" \t "_blank)等。[Visual Studio](https://baike.baidu.com/item/Visual%20Studio" \t "_blank)是目前最流行的[Windows](https://baike.baidu.com/item/Windows" \t "_blank)平台应用程序的[集成开发环境](https://baike.baidu.com/item/%E9%9B%86%E6%88%90%E5%BC%80%E5%8F%91%E7%8E%AF%E5%A2%83" \t "_blank)。最新版本为 Visual Studio 2019版本，基于.NET Framework 4.7。本课程设计使用2017版本。

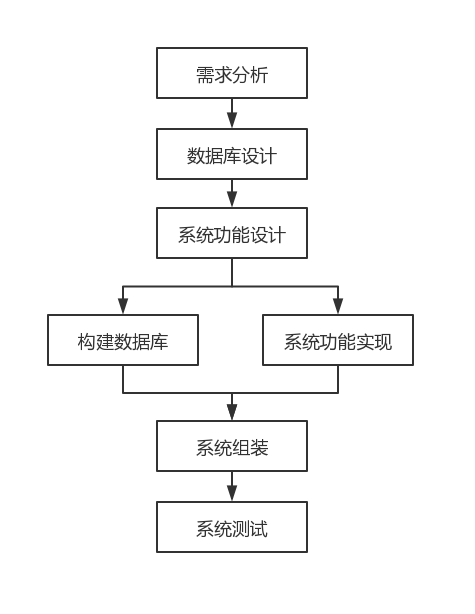
（2）Microsoft SQL Server Management Studio

对数据库的操作可通过数据库管理系统（Database Management System，简称DBMS）实现，它是数据库的核心组成部分，用户对数据库的一切操作，比如定义、查询、更新、插入、添加触发器等控制都是通过DBMS来进行操作的。Microsoft公司也为SQL Server开发了一套专属的DBMS——SQL Server Management Studio（简称SSMS），本课程设计使用的版本是17.9.1。

DBMS可以对[数据库](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E5%BA%93/103728" \t "_blank)进行统一的管理和[控制](https://baike.baidu.com/item/%E6%8E%A7%E5%88%B6/948689" \t "_blank)，以保证[数据库](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E5%BA%93/103728" \t "_blank)的安全性和完整性。用户通过[DBMS](https://baike.baidu.com/item/DBMS" \t "_blank)访问[数据库](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E5%BA%93/103728" \t "_blank)中的数据，[数据库管理员](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E5%BA%93%E7%AE%A1%E7%90%86%E5%91%98/1216449" \t "_blank)也通过DBMS进行数据库的维护工作。它可使多个[应用程序](https://baike.baidu.com/item/%E5%BA%94%E7%94%A8%E7%A8%8B%E5%BA%8F/5985445" \t "_blank)和用户用不同的方法在同时或不同时刻去建立，修改和询问[数据库](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E5%BA%93/103728" \t "_blank)。大部分[DBMS](https://baike.baidu.com/item/DBMS" \t "_blank)提供[数据定义语言](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E5%AE%9A%E4%B9%89%E8%AF%AD%E8%A8%80/104009" \t "_blank)[DDL](https://baike.baidu.com/item/DDL/21997" \t "_blank)（Data Definition Language）和[数据操作语言](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E6%93%8D%E4%BD%9C%E8%AF%AD%E8%A8%80/10826383" \t "_blank)[DML](https://baike.baidu.com/item/DML" \t "_blank)（Data Manipulation Language），供用户定义[数据库](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E5%BA%93/103728" \t "_blank)的模式结构与权限约束，实现对数据的追加、删除等操作。

数据库管理系统是数据库系统的核心，是管理数据库的软件。数据库管理系统就是实现把用户意义下抽象的逻辑数据处理，转换成为计算机中具体的物理数据处理的软件。有了数据库管理系统，用户就可以在抽象意义下处理数据，而不必顾及这些数据在计算机中的布局和物理位置。

2.2 技术路线

本系统在着重分析当前校园二手交易平台、互助平台缺点和不足的基础之上，详细设计了针对校园多功能互助平台的功能，完成诸如代取快递、代拿外卖、转让课本、出售二手等。利用软件工程原理和结构化程序设计思想，结合关系型数据库平台、C#面向对象的程序设计、.NET平台等技术，设计实现了整个多功能互助平台系统。

本平台的开发技术路线如图2-1所示：

图 2-1 开发路线图

**3 功能分析与设计**

3.1 需求分析

**3.1.1 用户调研**

本系统中的用户角色是大学生和平台管理人员，为了能够方便用户操作，让本系统能服务更多的用户，在开发前需要针对实际情况进行详细分析，对各个模块间的跳转进行规划，让系统的功能和用户操作尽可能的符合实际的使用情况。

**3.1.2 初步设计**

根据系统的功能需求分析，系统的主要思路设计如下，这也是我们后续工作设计界面的依据：

（1）用户登录：输入账号密码进行登录

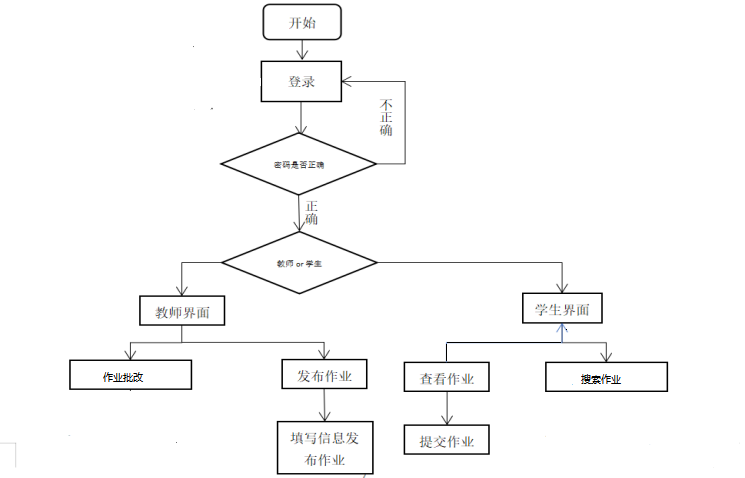
（2）学生用户浏览信息：用户登录后，若为学生用户，则跳转至用户信息界面，可在此界面查看作业并对其进行操作

（3）教师用户浏览信息：用户登录后，若为教师用户，则跳转至教师操作界面，教师可发布作业，也可以搜索作业，批改作业。

3.2 系统设计

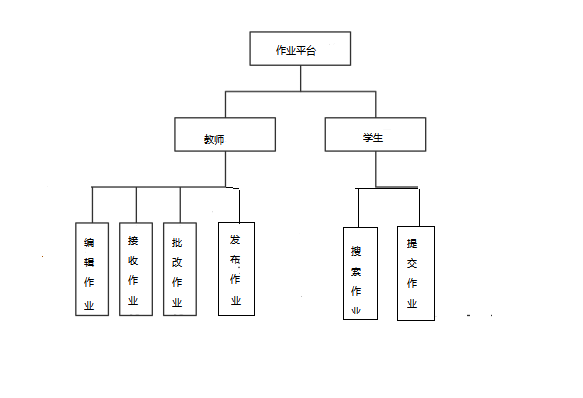
**3.2.1 系统流程设计**

根据对需求分析阶段所得到的信息进行分析，将系统流程细化到如图所示：



**3.2.2 系统功能分析**

根据上述系统流程分析，系统功能概括如下



3.3 数据库概念结构设计

在需求分析阶段所得到的应用需求应该首先抽象为信息世界的结构，然后才能更好、更准确地用某一数据库管理系统实现这些需求。概念模型是各种数据模型的共同基础。描述概念模型的有利工具E-R模型。

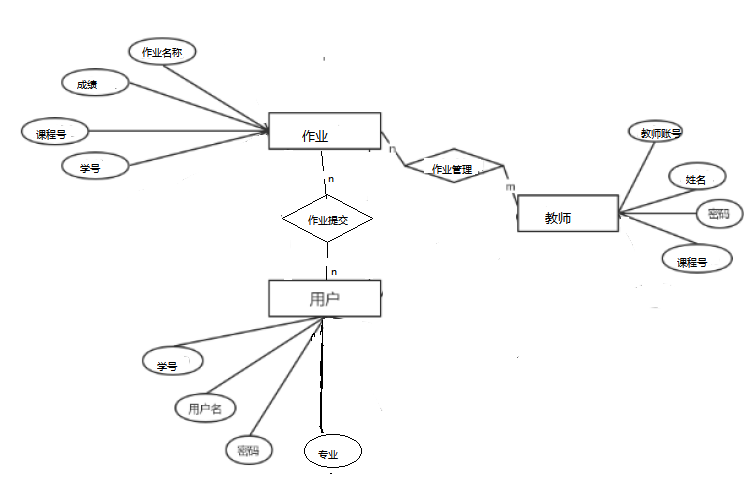
根据需求分析，我们建立了如下的E-R模型，如图3-3所示：

图3-3 E-R模型图

3.4 数据库逻辑结构设计

数据库逻辑结构的设计就是把概念结构设计阶段设计好的基本E-R图转换为与选用DBMS所支持的数据模型相符合的逻辑结构。本课程设计采用的是关系型数据库，因此需要完成E-R图向关系模型的转换。

**3.4.1 E-R图向关系模型的转换**

根据图3-3所示的E-R模型图转换为关系模型，关系的码用下划线标出：

（1）学生信息表（学号、姓名、密码、专业）；

（2）作业总表（学号、课程号、作业名称、作业评分）；

（3）教师信息表（老师账号、密码、姓名、课程号）

**3.4.2 利用数据依赖的理论对关系模式进行分析**

对于上述7种关系模式，我们可以发现每一个非主属性完全依赖于任意一个候选码，并且每一个非主属性不传递依赖于任意一个候选码，所以以上7种关系模式都属于3NF。在数据库的设计中，关系模式满足3NF已经可以满足一定的需求。

3.5 数据库物理结构设计

数据库在物理设备上的存储结构与存取方法称为数据库的物理结构，它依赖于选定的数据库管理系统。为一个给定的逻辑数据模型选取一个最适合应用要求的物理结构的过程，就是数据库的物理设计。

本平台采用的数据库平台是SQL Server，数据库以文件形式存储在计算机磁盘中，其中主要的文件有主数据文件，次数据文件和事物日志文件：

（1）主数据文件

主数据文件是数据库的起点，其中包含了数据库的初始信息，并记录数据库还拥有哪些文件。每个数据库有且只能有一个主数据文件。主数据文件是数据库必须的文件，主数据文件的扩展名是.mdf；

（2）次数据文件

除了主数据文件以外的所有其他数据文件都是次数据库文件。次数据文件不是数据库必需的文件。次数据文件的扩展名是.ndf；

（3）事务日记文件

每个数据库至少拥有一个自己的日记文件（也可以拥有多个日记文件）。日记文件的大小最少是1MB，默认扩展名是.ldf,用来记录数据库的事务以及每个数据对数据库所做的修改。

在SQL Server中，页是数据存储的基本单位，为数据库中的数据文件分配的磁盘空间可以从逻辑上划分带有连续编号的（编号从0开始）磁盘I/O操作在页级执行，SQL Server读取或写入的是所有的数据页。

3.6 模块划分

按照上述分析将系统最终划分为如下4个模块：

（1）用户注册模块：

用户如果希望使用本平台的功能则必须完成注册，需要设定账号，密码，并且上传一张自己校园卡的照片供管理员查验，信息填写完整后即可提交注册申请并使用本平台；

（2）用户/管理员登录模块：

* 学生填写学号、密码登录平台，进入用户界面；
* 教师填写教师账号、密码来进入教师界面；

（3）教师模块：

教师模块又分成如下3个子模块：

a. 发布作业子模块：

用户首先选择自己的需求类别，进入相应的表单填写相关信息、上传图片（如果有需要）并提交表单；

b. 批改作业子模块：

用户可以根据需求的类别等信息来筛选已经发布的需求，并根据用户指定的方式（例如设定类别，任务状态，发布者ID，是否按发布时间降序等条件）进行筛选排序，并且可以查看需求的详细信息，在任务详细信息界面可以选择接受此需求；

c. 个人信息模块：

* 查看并修改个人信息；
* 退出当前账号，回到登录界面；

（4）学生模块：

* 查看并修改个人信息；
* 退出当前账号，回到登录界面；
* 搜索自己收到的作业;
* 提交作业;

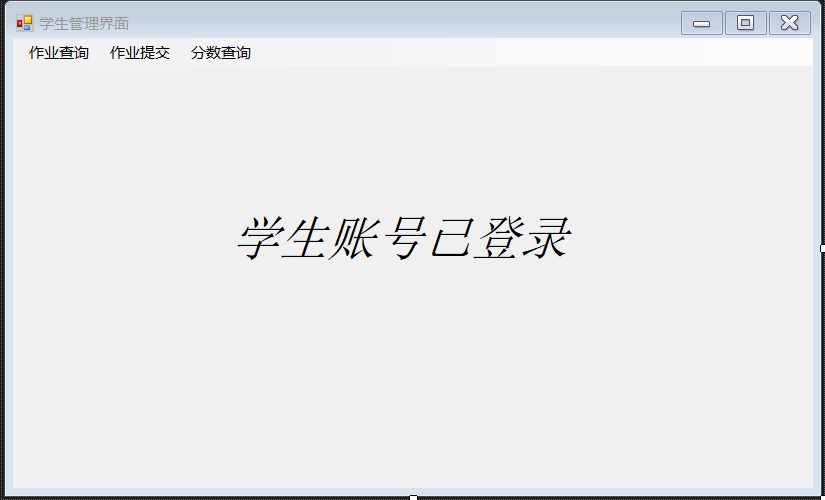
**4 实现**

4.1 前端界面实现

本平台前端界面开发主要借助Visual Studio 2017集成开发环境中针对C#窗体应用程序的设计器来完成前端界面的设计与开发工作。开发界面如图所示。

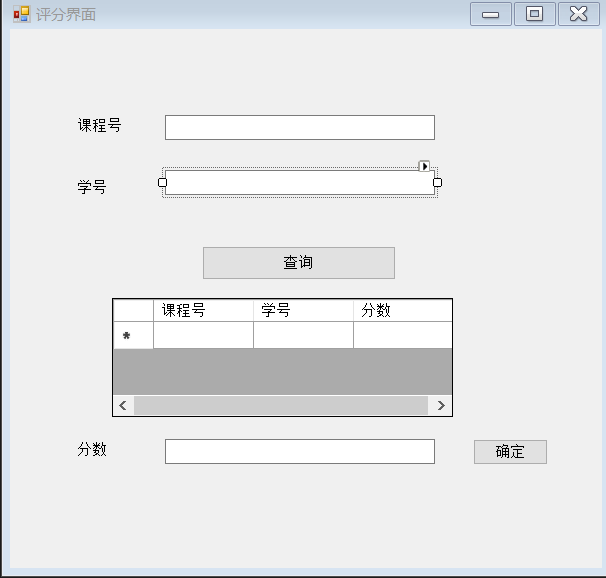
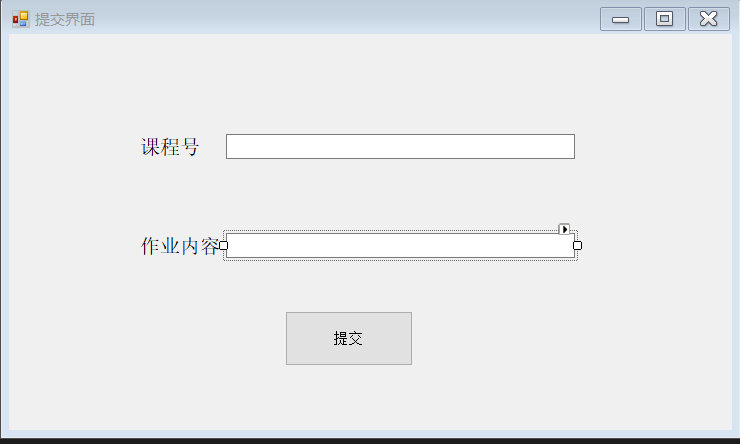
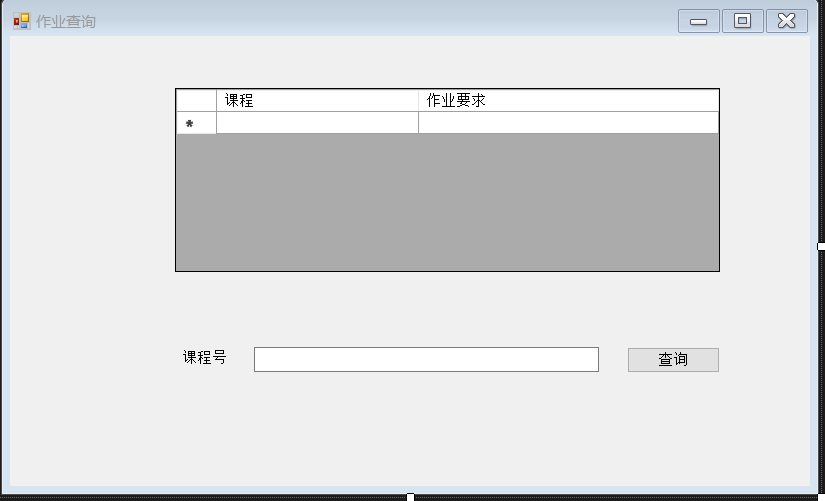
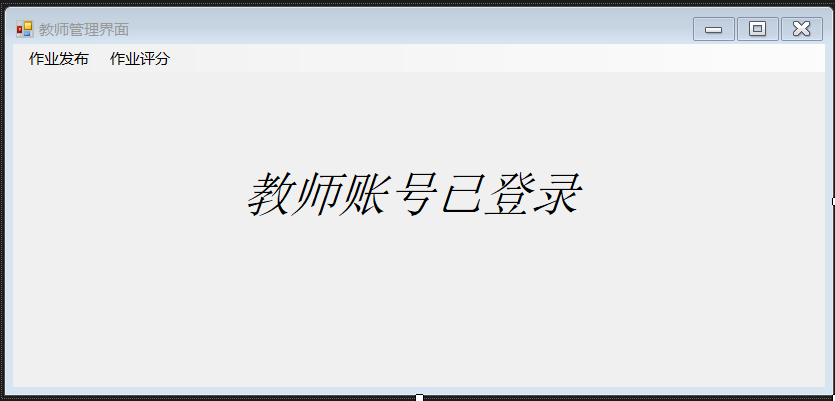
利用这个设计器，可以所见即所得地、便捷地完成窗体的设计，只需要将所需的空间拖拽至窗体，再设置属性即可，设计器将自动地生成后缀名为.Designer.cs的文件，记录窗体以及控件属性信息，在编译时会自动构建窗体，不需要程序员手动编写代码来进行窗体设计，这为我们的开发工作提供了极大的方便。



**4.2 用户界面实现**

用户界面如图所示





4.3后端代码实现

由于本平台后端有一定的代码量，因此仅选取核心的和关键的部分代码进行说明。

**4.3.1 C#与SQL Server的连接和交互**

C#是基于.NET平台的，.NET平台提供了许多命名空间，其中允许同本地或远程数据库进行交互式操作时，需要用到的命名空间是ADO. NET，其大部分由System. Data. dll核心程序集来表示，这可以完成应用程序对数据库的不同操作。

以登录界面代码为例

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.SqlClient;

namespace WindowsFormsApp1

{

public partial class login : Form

{

public login()

{

InitializeComponent();

}

private void button2\_Click(object sender, EventArgs e)

{

if(textBox1.Text!=""&&textBox2.Text!="")

{

denglu();

}

else

{

MessageBox.Show("输入不能为空");

}

}

private void 登录\_Load(object sender, EventArgs e)

{

}

public void denglu()

{

//教师

if(radioButton1.Checked==true)

{

Dao dao = new Dao();

string sql = "select \* from 老师 where 老师账号='" + textBox1.Text + "'and 密码='" + textBox2.Text + "'";

IDataReader dc = dao.read(sql);

if(dc.Read())

{

MessageBox.Show("登陆成功");

teacher t = new teacher();

this.Hide();

t.ShowDialog();

this.Show();

}

else

{

MessageBox.Show("登陆失败");

}

dao.DaoClose();

}

//学生

if (radioButton2.Checked == true)

{

Dao dao = new Dao();

string sql = "select \* from 学生 where 学号='" + textBox1.Text + "'and 密码='" + textBox2.Text + "'";

IDataReader dc = dao.read(sql);

if (dc.Read())

{

MessageBox.Show("登陆成功");

student s = new student();

this.Hide();

s.ShowDialog();

this.Show();

}

else

{

MessageBox.Show("登陆失败");

}

dao.DaoClose();

}

}

private void button3\_Click(object sender, EventArgs e)

{

textBox1.Text = "" ;textBox2.Text = "";

}

}

}

**4.3.2 C#后台代码文件以及数据库脚本文件关系图**

using System;  
using System.Collections.Generic;  
using System.ComponentModel;  
using System.Data;  
using System.Drawing;  
using System.Linq;  
using System.Text;  
using System.Threading.Tasks;  
using System.Windows.Forms;  
  
namespace WindowsFormsApp1  
{  
    public partial class student1 : Form  
    {  
        public student1()  
        {  
            InitializeComponent();  
        }  
  
        private void textBox1\_TextChanged(object sender, EventArgs e)  
        {  
           
        }  
        public void classID()  
        {  
            dataGridView1.Rows.Clear();  
            Dao dao = new Dao();  
            string sql = $"select \* from 作业 where 作业要求='{textBox1.Text}'";  
            IDataReader dc = dao.read(sql);  
            while (dc.Read())  
            {  
                dataGridView1.Rows.Add(dc[0].ToString(),dc[1].ToString());  
  
            }  
            dc.Close();  
            dao.DaoClose();  
        }  
        private void button1\_Click(object sender, EventArgs e)  
        {  
            classID();  
        }  
    }  
}

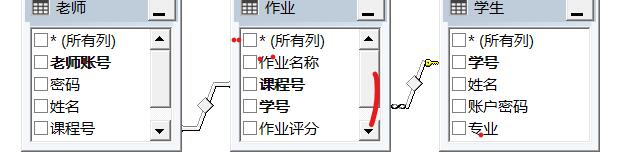
C#的源文件的后缀名为.cs，由于源文件数量较多，所以将源文件根据功能进行了分门别类地整理。另外由于本平台是Windows窗体应用程序，所以窗体与源代码有对应关系。图4-4显示了各个源文件之间的关联以及调用逻辑关系。

用于在SQL Server中建立基本表以及触发器的SQL脚本文件共2份，也在图4-4上进行了标注。

**4.3.3 SQL Server建立数据库**

CREATE TABLE 学生(  
学号 CHAR(20),  
姓名 CHAR(10),  
    账户密码 CHAR(20),  
专业 CHAR(10),  
PRIMARY KEY (学号)  
)  
GO  
CREATE TABLE 老师(  
老师账号 CHAR(20),  
密码 CHAR(20),  
姓名 CHAR(10),  
课程号 CHAR(10),  
课程名称 CHAR(10),  
PRIMARY KEY (老师账号)  
  
)  
GO  
CREATE TABLE 作业(  
作业名称 CHAR(20),  
课程号 CHAR(10),  
学号 CHAR(20),  
作业评分 CHAR(10),  
作业内容 VARCHAR(300),  
PRIMARY KEY (学号,课程号),  
FOREIGN KEY (学号) REFERENCES 学生(学号)  
  
)  
GO

数据库关系图如图所示：



**5 系统测试与应用**

5.1 系统测试运行环境

测试系统软件环境：操作系统为Windows 10并且已经安装.Net Framework 4.6.1，已经安装数据库SQL Server 2017和SQL Server Management Studio 17.9.1；

测试系统硬件配置：处理器Intel Core i5-7200U，内存8G，外存256G固态硬盘，显示适配器AMD Radeon R7 M440。

5.2 数据库测试与运行

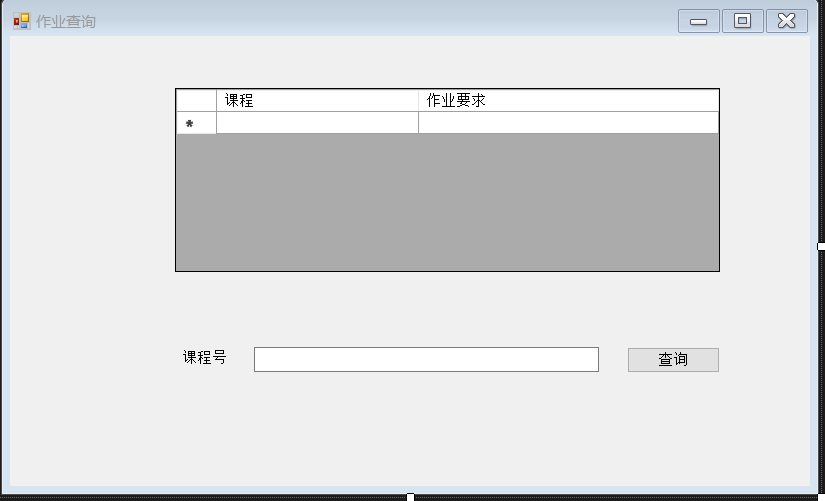
**5.2.1 用户登录**



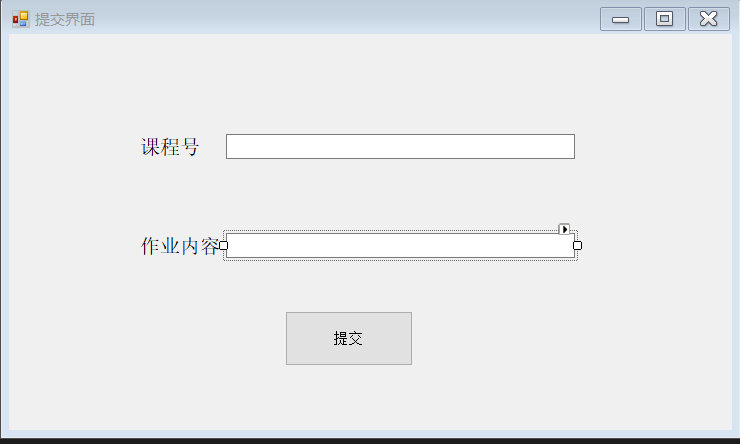
**5.2.2 作业任务发布**



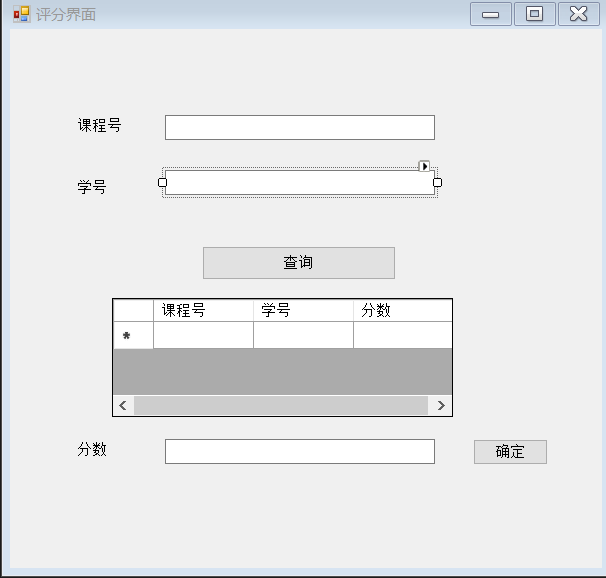
**5.2.3 作业查询**



**5.2.4 作业提交**



**5.2.5 教师评分**



**6 总结与展望**

6.1 总结

本系统在着重考察研究目前大学生生活学习习惯的基础之上实现了如下核心功能。

* 用户注册登录：只有在系统内存在的用户才可进行登录操作，忘记密码的用户可根据注册时的手机号重新修改密码。
* 接收作业功能：用户登录之后可以前往作业界面，查看自己的作业。
* 用户中心：用户在登录后可以在用户中心修改个人信息、修改密码、查看自己接收与发布的作业。

6.2 不足

目前，本系统的大体核心功能都基本实现，但是整个系统的应用上来说还是有很多的不足，包括：

* 用户输入个人信息的操作需要按照指定格式输入，用户如果不按照规定的格式输入时间则会报错，用户使用上则会产生一些不便；
* 对信息的处理结构不够严谨，没有额外增加容错选项；
* 有些功能时显得有些粗糙，可能对用户造成不便；
* 并未真正地实现将图片上传到网络服务器，平台暂时只能在本地运行，未能上线投入真正的使用；
* 整体界面设计有待改进，可以设计得更加美观。

6.3 展望

本系统面向大学学生这一个群体，希望能为广大学生节省时间、提高效率，同时也能在学习时将更少的时间花费在准备工作上，有效延长大家的有效学习时间。

本系统是对于“学生作业管理系统”的一次尝试，仅仅只能做出一个雏形。希望今后能以本次课程设计为基础，在已有的工作成果之上优化平台的功能，美化前端界面，争取能够真正上线运行，相信这样的学生作业管理系统有着美好的前景和极大的潜力。