

目 录

摘要:	3
关键词:	3
Abstract:.....	3
Key words:	3
1 课题分析.....	4
1.1 课题背景.....	4
1.1.1 国内研究概况.....	4
1.1.2 国外研究概况.....	4
1.2 研究目的及意义.....	4
1.3 本文所做的工作.....	4
2 技术路线及开发工具.....	5
2.1 技术路线.....	5
2.1.1 技术思路.....	5
2.1.2 技术路线图.....	5
2.2 开发技术及工具.....	6
2.2.1 开发技术.....	6
2.2.2 开发工具.....	6
3 功能分析与设计.....	7
3.1 需求分析.....	7
3.1.1 用户调研.....	7
3.1.2 初步设计.....	7
3.2 系统设计.....	8
3.2.1 系统流程设计.....	8
3.2.2 系统功能分析.....	9
3.3 概念结构设计.....	10
3.3.1 数据库设计.....	10
3.3.2 数据库 E-R 图.....	12
3.4 系统核心功能设计.....	17
3.4.1 核心功能顺序图.....	17
3.5 逻辑结构设计及物理设计.....	17
3.5.1 数据库关系表.....	17
3.5.2 数据库视图.....	26
3.5.1 数据库触发器.....	28
3.5.1 数据库整体关系图.....	29
3.6 模块划分.....	30
4 开发与实现.....	30
4.1 系统核心程序编写.....	30
4.1.1 C#访问 SQL Server 数据库.....	30
4.1.2 验证用户登录的流程图.....	30

4.1.3 实现用户拜师功能的流程图.....	31
4.2 系统实施.....	32
4.2.1 管理员登录模块.....	32
4.2.2 管理员管理模块.....	32
4.2.3 用户注册模块.....	33
4.2.4 用户登录模块.....	33
4.2.5 用户信息模块.....	34
4.2.6 用户拜师学习模块.....	34
5 系统测试与应用.....	35
5.1 系统测试运行环境.....	35
5.1.1 服务器端运行环境.....	35
5.1.2 客户端运行环境.....	35
5.2 数据库测试与运行.....	35
5.2.1 管理员登录模块.....	35
5.2.2 管理员管理模块.....	35
5.2.3 用户注册模块.....	41
5.2.4 用户登录模块.....	42
5.2.5 用户信息模块.....	42
5.2.6 用户拜师学习模块.....	46
6 总结与展望.....	50
6.1 总结.....	50
6.2 展望.....	50
参考文献:	50

南农师徒信息管理系统

摘要：临近开学季，大一新生即将面对从未经历过的大学生活，对于大部分新生来说，无论是学习还是生活上，甚至是兴趣爱好，艺术娱乐等方面，他们都难以快速适应。而对于已经适应大学生活的学长学姐或者适应能力较高的同学们，在他们擅长的方面去帮助或者指导需要帮助的同学，既可以锻炼自己，也可以拓宽自己的社交范围，也帮助了有困难的同学。由此可以看出，在我校范围内建立有效的师徒系统，给有困难的同学以寻求解决问题的途径，给能力强的同学以进一步锻炼自己的方法，是必要且有意义的。本人在了解需求后，基于 SqlServer 与 C#，开发出一款面向广大南农学子的师徒信息管理系统。

关键词：南农 师徒系统 学习 拜师 大学生活

NJAU master and apprentice information management system

Abstract: Near the beginning of the school season, freshmen are about to face college life that they have never experienced before. For most freshmen, it is difficult for them to adapt to their study, life, even hobbies, art and entertainment. For the elder and elder sisters who have adapted to college life or the students with higher adaptability, they can help or guide the students who need help in the aspects they are good at, which can not only exercise themselves, but also broaden their social scope, and also help the students who have difficulties. It can be seen from this that it is necessary and meaningful to establish an effective apprenticeship system within the scope of our school, to seek solutions for students with difficulties, and to further exercise their own methods for students with strong abilities. Based on SqlServer and C#, I developed an apprentice information management system for the students of NJAU.

Key words: NJAU master and apprentice system Learning take sb. as one's teacher College life

古有孔子云：“三人行，必有我师焉。”随着互联网的普及，有需求的人可以通过各种渠道寻找到自己的老师，你可以选择在大学课堂中与传道授业的老师畅游学术海洋；你也可以选择在网络直播课上跟随网络老师拓展学习；你甚至还可以向你的朋友，学长学姐学得生活经验与知识技能。虽然在知识流量越来越大的今天，“师徒系统”显得越发重要且必要，但是以往的师徒系统常常是某一软件的附属产品，市面上很少能够见到以“师徒管理”为主要功能的，专门开发的系统，所以为了帮助本校同学们能够更方便得获取知识，各个年级，各个专业之间的同学们能够有更广泛的交流，我开发了这一款可以根据用户需求寻找筛选合适老师，允许用户发布问题寻求帮助，通过师生互评营造良好学习氛围，通过排行榜等机制来鼓励师生交互学习的南农师徒信息管理系统。

1 课题分析

1.1 课题背景

随着互联网技术的发展,网络的普及以及网速的提高,人们学习的途径越发的广泛,当今的“师徒”概念早已不是古代“传道授业”般那样单一,而是变得更加多元化。人们可以从自己喜欢的视频平台上观看视频进行学习,人们也可以从有效的学习平台上观看录播进行学习,无论是哪种途径,都是知识从“师傅”向“徒弟”完成了传递。

同时,在许多游戏中也会设立“师徒系统”,徒弟会在游戏内虚拟师傅的带领下完成某一项自己单独不可能完成的任务而获得奖励,师傅也会在徒弟等级提升后获得相应的奖励,从而形成良性循环,促进整个游戏的生态和可持续性,可玩性的提高。

但是反观当下的大学生活中,许多“师徒系统”往往都依附于某个学习平台,使得其使用效果并不是很理想,用户难以第一时间,快速而方便地获得解决方案,同时也缺少了某种机制来促进师生的交互。所以我决定基于 `SqlServer` 与 `C#` 开发这样一款“师徒信息管理系统”,方便师生交互。同时,系统主要面对两个群体——用户和管理员。管理员有权限查看用户的关键信息,冻结或解冻某个用户;查看系统的基本信息;查看或清空登录日志;添加新的管理员。用户则能查看或修改个人信息;查看排行榜;查看历史记录;发布问题;筛选合适的老师;发出拜师申请;同意拜师申请并建立师生关系开始学习;师生互评并结束学习。

本着“人人皆是老师”的原则,本系统并没有刻意去区分老师与学生,目的是为了形成良好的学习氛围。

1.1.1 国内研究概况

中国目前的学习平台比比皆是,其附带的师徒系统也是极其丰富。比如师徒在线,师徒在线主面向毕业学生、职业人和基础教育学生,通过功能强大 `LMS` 学习管理系统和优质课程视频,整合专业培训机构和优秀师傅资源,致力于成为国内至佳的中小学校教学辅助工具、企业网络教育云运营商和全方位的终生免费在线学习求职平台。

1.1.2 国外研究概况

目前国外的主流学习网站中, `Udemy` 与 `Coursera` 功能较为齐全。`Udemy` 是全球最著名的在线教育网站,网站提供令人难以置信的流行和宝贵的课程资源。用户在 `Udemy` 里可以搜索到超过 40000 个不同主题的课程,以及还可以下载 `Udemy` 应用程序,使用手机快速的学习课程。`Udemy` 课程不是免费提供的,课程最低为 12 美元,用户也可以创建和发布自己的课程。

而 `Coursera` 是一个提供超过全球 140 个国家顶级大学和组织课程的教育网站。`Coursera` 与美国宾夕法尼亚大学,斯坦福大学,密歇根大学和其他机构合作来提供世界上最好的教育。用户可以在网站里找到计算机科学,商业,社会科学等领域的有酬和无酬课程。`Coursera` 也提供移动应用程序,所以用户可以按自己的节奏来学习。

1.2 研究目的及意义

随着信息时代与数据时代的推进,当下大学生的拓展学习需求越发丰富且在不断扩大,高校也急需一个平台来促进同学们之间的交流,养成良好的学术氛围,所以我在本数据库课程设计中,基于所学,利用 `SqlServer` 作为后端,自学 `C#` 作为前端,开发出这样一款师徒信息管理系统,尽可能地去适应同学们的需求,促进校园良好学风氛围的形成。

1.3 本文所做的工作

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

(1) 在该系统的可行性分析和规划阶段,对于师徒信息管理系统的现状做了简单的介

绍和实现系统的技术路线。

- (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师徒信息管理系统的功能模块设计、功能设计、数据库设计，并完成实现了具体的功能。

2 技术路线及开发工具

2.1 技术路线

2.1.1 技术思路

本系统在了解分析目前国内外主流的师徒系统后，借鉴了游戏中师徒系统的部分功能，引入潜力值概念，通过师生互评与排行榜来增强学习动力，使用 SqlServer 完成了数据库的设计，使用 C#完成了 Windows 窗体程序的设计，并在 Visual Studio 2015 的集成开发环境中完成系统的代码编写，调试，测试以及最后的软件发布。

2.1.2 技术路线图

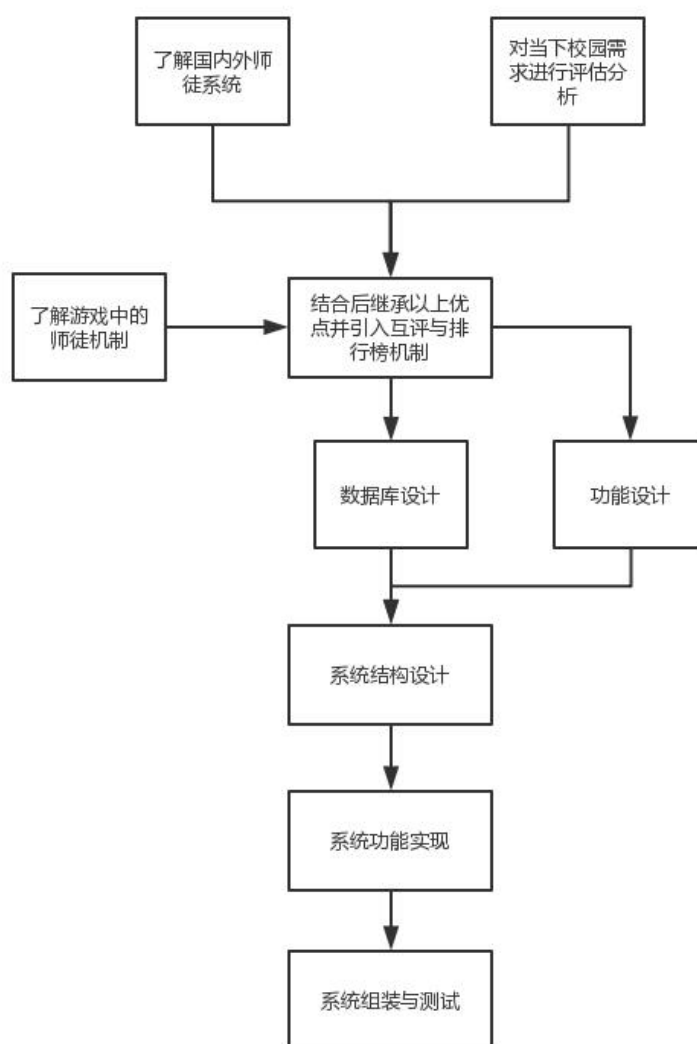


图 2-1 技术路线图

2.2 开发技术及工具

2.2.1 开发技术

本系统前端开发采用 WinForm 技术来设计 Windows 窗体应用程序，采用 C#.NET 技术作为前端与数据库的连接，后端采用 Sql Server 来设计数据库。

.NET 是微软当代的操作平台，它允许人们在其上构建各种应用方式，使人们尽可能通过简单的方式，多样化地、最大限度地从网站获取信息，解决网站之间的协同工作，并打破计算机、设备、网站、各大机构和工业界间的障碍——即所谓的“数字孤岛”，从而实现因特网的全部潜能，搭建起第三代互联网平台。后缀为 net 是网络服务公司，为个人或商业提供服务。而 .NET Framework 是用于 Windows 的新托管代码编程模型。它强大功能与新技术结合起来，用于构建具有视觉上引人注目的用户体验的应用程序，实现跨技术边界的无缝通信，并且能支持各种业务流程。

WinForm 是 .Net 开发平台中对 Windows Form 的一种称谓。Windows 窗体的一些重要特点如下：功能强大，Windows 窗体可用于设计窗体和可视控件，以创建丰富的基于 Windows 的应用程序；操作方便，新的数据提供程序管理：数据提供程序管理提供易于连接 OLEDB 和 ODBC 数据源的数据控件，包括 Microsoft SQL Server、Microsoft Access、Jet、DB2 以及 Oracle 等；使用安全，Windows 窗体充分利用公共语言运行库的安全特性。这就意味着，一切都可以通过 Windows 窗体来实现，包括在浏览器中运行的不可信控件和用户硬盘上安装的完全可信的应用程序。

同时，Windows 窗体提供了一套丰富的控件，并且开发人员可以定义自己有特色的新的控件。

WinForm 控件是指以输入或操作数据的对象。比如 ComponentOne 是 .net 平台下对数据和方法的封装。有自己的属性和方法。属性是控件数据的简单访问者。方法则是控件的一些简单而可见的功能。包含在 .NET Framework 中的 Windows 窗体类旨在用于 GUI 开发。您可以轻松创建具有适应多变的商业需求所需的灵活性的命令窗口、按钮、菜单、工具栏和其他屏幕元素。在数据管理方面，它有方便的数据显示和操作：应用程序开发中最常见的情形之一是在窗体上显示数据。Windows 窗体对数据库处理提供全面支持。可以访问数据库中的数据，并在窗体上显示和操作数据。它也向用户提供创建窗体、数据处理、打包和部署等的分布指导。

数据库管理系统是一种操纵和管理数据库的大型软件，用于建立、使用和维护数据库，简称 DBMS。它对数据库进行统一的管理和控制，以保证数据库的安全性和完整性。用户通过 DBMS 访问数据库中的数据，数据库管理员也通过 DBMS 进行数据库的维护工作。它可以支持多个应用程序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。大部分 DBMS 提供数据定义语言 DDL 和数据操作语言 DML，供用户定义数据库的模式结构与权限约束，实现对数据的追加、删除等操作。DBMS 可以保证控制数据的冗余，保证数据的一致性，提高数据的共享。

关系数据库，是建立在关系数据库模型基础上的数据库，借助于集合代数等概念和方法来处理数据库中的数据，同时也是一个被组织成一组拥有正式描述性的表格，该形式的表格作用的实质是装载着数据项的特殊收集体，这些表格中的数据能以许多不同的方式被存取或重新召集而不需要重新组织数据库表格。主流的关系数据库有 oracle、db2、sqlserver、sybase、mysql 等。

2.2.2 开发工具

本系统在 Windows 10 操作系统上开发，C#集成开发环境采用 Microsoft Visual Studio 2015，数据库服务器采用 SqlServer 2008R2。

C#是微软公司发布的一种由 C 和 C++ 衍生出来的面向对象的编程语言、运行于 .NET Framework 和 .NET Core(完全开源，跨平台)之上的高级程序设计语言。并定于在微软职

业开发者论坛(PDC)上登台亮相。C#是微软公司研究员 Anders Hejlsberg 的最新成果。C#看起来与 Java 有着惊人的相似；它包括了诸如单一继承、接口、与 Java 几乎同样的语法和编译成中间代码再运行的过程。但是 C#与 Java 有着明显的不同，它借鉴了 Delphi 的一个特点，与 COM(组件对象模型)是直接集成的，而且它是微软公司 .NET windows 网络框架的主角。

Microsoft Visual Studio 简称 VS，它是美国微软公司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(IDE)等等。所写的目标代码适用于微软支持的所有平台。

SQL Server 系列软件是 Microsoft 公司推出的关系型数据库管理系统。SQL Server 2008 版本可以将结构化、半结构化和非结构化文档的数据直接存储到数据库中。可以对数据进行查询、搜索、同步、报告和分析之类的操作。数据可以存储在各种设备上，从数据中心最大的服务器一直到桌面计算机和移动设备，它都可以控制数据而不用管数据存储在哪里。

此外，SQL Server 2008 允许使用 Microsoft .NET 和 Visual Studio 开发的自定义应用程序中使用数据，在面向服务的架构(SOA)和通过 Microsoft BizTalk Server 进行的业务流程中使用数据。信息工作人员可以通过日常使用的工具直接访问数据。该平台具有以下特点：可信任，高效的，智能的。在实际应用中取得了不错的效果。

3 功能分析与设计

3.1 需求分析

3.1.1 用户调研

本系统中的主要面向群体是管理员和用户，为了能够更加清晰的体现本系统在师徒管理中的运用，并且使得本系统可以被用户和管理员最大限度地使用，在实际开发以前要针对用户和管理员做一定的相关资料收集，一定的考察，让系统的功能、环境等都可能符合实际使用情况。

3.1.2 初步设计

根据系统的功能需求分析，可以将系统大体分为以下功能：

(1) 用户注册登录找回密码：如果用户要使用系统功能，就必须注册一个唯一账号来登录系统，如果用户忘记密码，系统也会协助用户找回密码。

(2) 用户查看或修改个人信息：用户可以查看自己的个人信息，并可以修改它们，也可以在系统内修改自己的密码。

(3) 用户查看历史记录：用户可以在系统内查看自己的各种历史记录。

(4) 用户查看排行榜：用户可以查看其他评分优秀的用户和高频问题。

(5) 用户发出拜师申请：用户可以向符合要求的其他用户发出拜师申请。

(6) 用户开始学习：申请通过后，即建立师徒关系，开始学习。

(7) 用户结束学习：只有师生互评之后才会结束学习。

(8) 用户发布问题：用户可以向系统发布问题，以寻求帮助。

(9) 管理员查看系统基本信息：管理员可以查看数据库中的基本信息。

(10) 管理员查看登录日志：管理员可以查看所有的登录记录，包括允许与拒绝的登录，还可以清空所有登录日志。

(11) 管理员添加新的管理员：管理员有权限添加新的管理员。

(12) 管理员管理用户：管理员可以查看或搜索用户的密码，电话号，并且可以冻结或者解冻某个用户。

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，将系统流程细化如图 3-1：

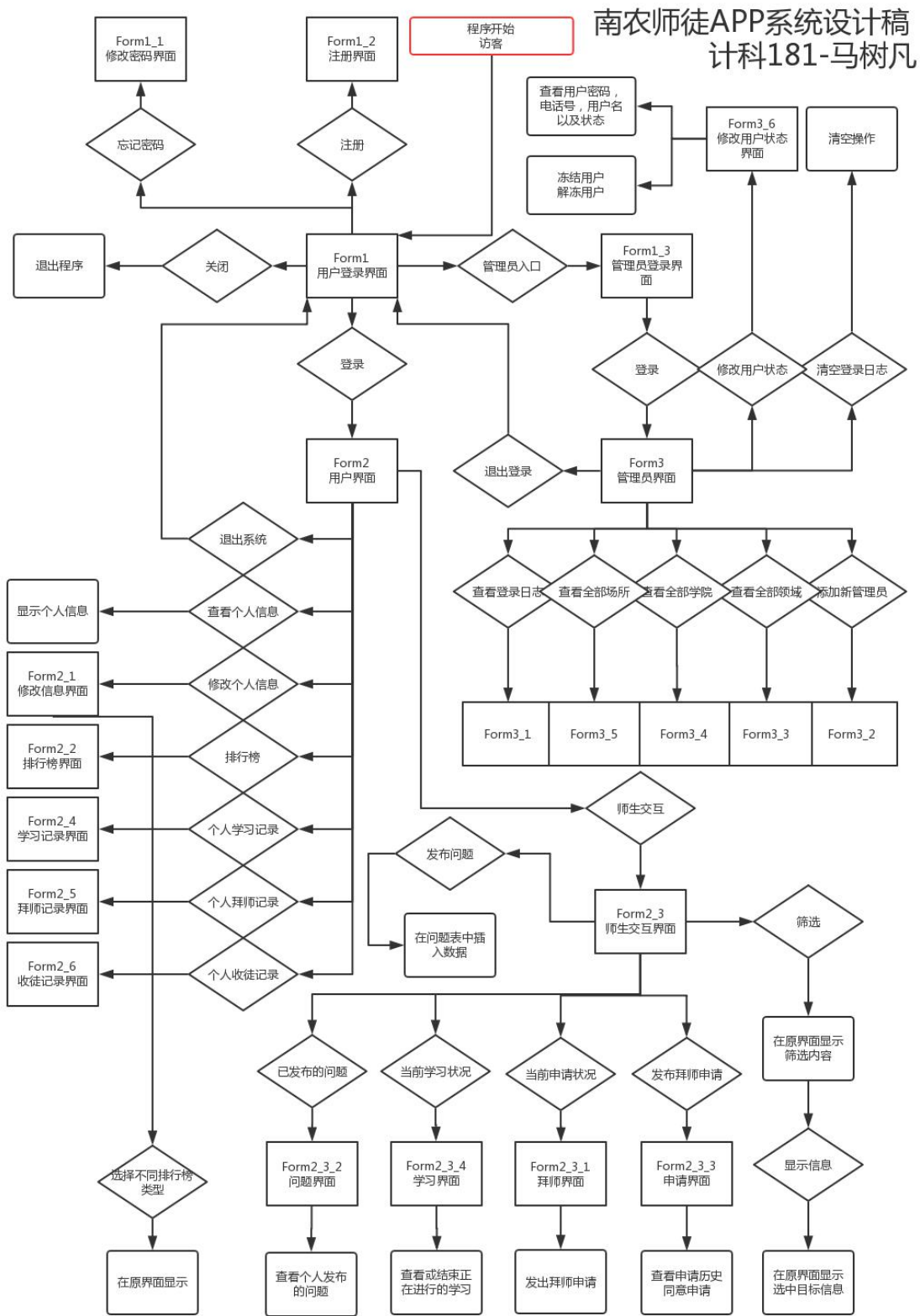


图 3-1 系统流程图

3.2.2 系统功能分析

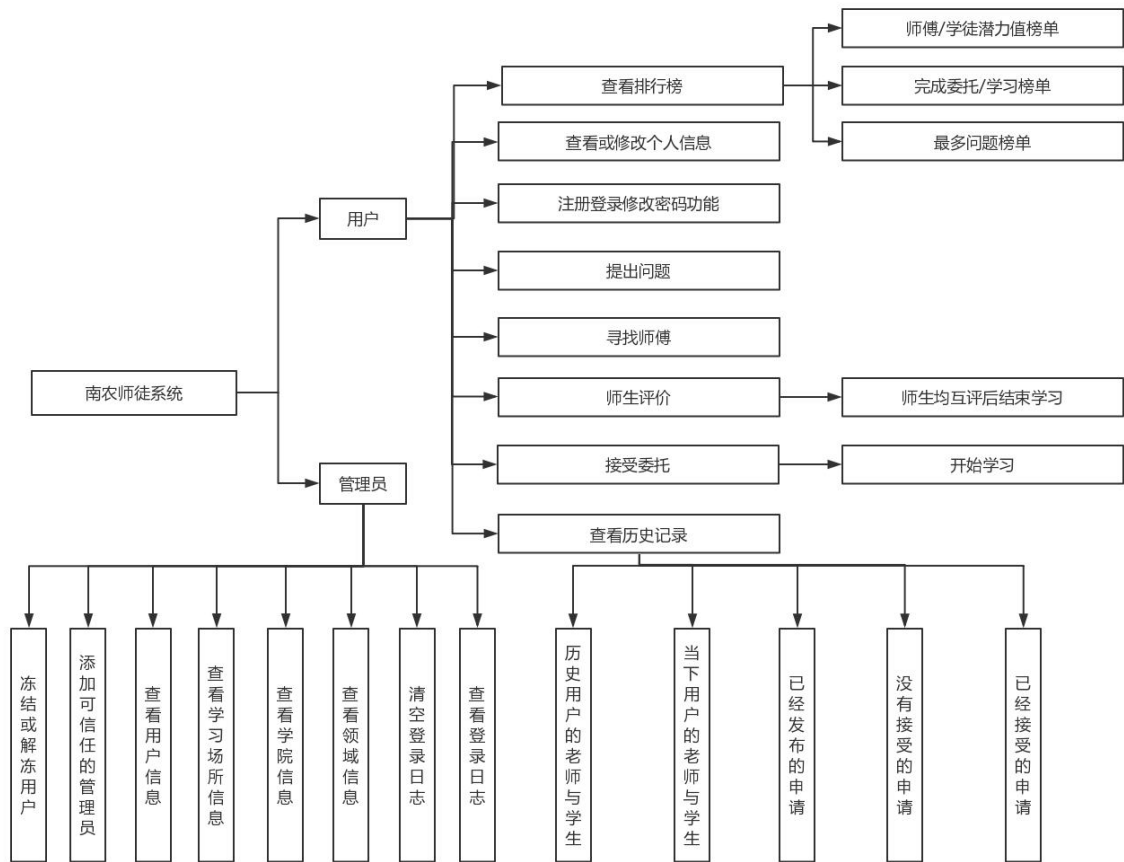


图 3-2 系统功能图

本系统主要功能分为两大部分，用户部分和管理员部分，每一部分又分出如下功能。
管理员部分：

- (1) 查看系统基本信息，包括领域及其分类信息，学院信息，学习场所信息。
- (2) 管理用户，包括搜索或查看用户的密码，电话号，账户状态，冻结或解冻用户。
- (3) 查看或清空登录日志。

用户部分：

- (1) 注册，登录，忘记密码可以找回密码的功能。
- (2) 查看或修改个人基本信息。
- (3) 查看排行榜，包括“师傅潜力值”，“学徒潜力值”，“完成委托数”，“完成学习数”与“高频问题”五种排行。
- (4) 查看历史记录，包括已经完成的学习记录，用户的拜师记录，用户的收徒记录，正在进行的学习（包括老师与学生），用户收到的申请（包括已经批准的和未批准的），用户曾经发出的申请（显示批准状态）。
- (5) 用户发布问题。
- (6) 用户发出拜师申请。
- (7) 用户接受拜师申请，并开始学习。
- (8) 用户结束学习，并进行师生互评。
- (9) 用户查看问题记录，包括自己提出的问题（解决和未解决），以及系统中存储的所有用户提出的问题，这里会给出提出问题用户的联系方式，方便有能力的用户

去联系。

基于以上功能，系统运行时对用户和管理员提出以下约束：

管理员约束：

- (1) 管理员账户名和密码均不可更改
- (2) 管理员账户名不可重复，且不能包含空格
- (3) 登录日志一旦清除即全部清除，不可恢复

用户约束：

- (1) 用户名不可重复，一旦确定便不可更改
- (2) 同一领域内，每一个用户只能同时存在一条未解决的问题，但可以同时存在多条已经解决的问题。
- (3) 用户只能在老师擅长的领域内发出拜师申请，不能跨领域拜师。
- (4) 同一领域内，对于同一个老师，每一个用户只能同时存在一条未通过的拜师申请，但可以同时存在多条通过的拜师申请记录。
- (5) 同一领域内，对于同一个老师，每一个用户只能同时存在一条正在进行的学习，但可以同时存在多条完成的学习记录。
- (6) 当用户发出拜师申请时，系统会首先检索是否存在该用户发布的在该领域的未解决的问题，若存在，则修改属性为“已解决”再新建拜师申请记录，若不存在，则新建一条“已解决”的问题记录再新建拜师申请记录。

3.3 概念结构设计

3.3.1 数据库设计

基本表设计（每张表的主码使用下划线标明，外码使用加粗标明）：

中文格式（方便理解）：

- (1) 管理员表（管理员名，管理员密码）
- (2) 用户表（标识码，用户名，登录密码，真实姓名，性别，年龄，**学院代号**，联系方式，**擅长领域代号**，**常去的学习场所代号**，讲师潜力值，学徒潜力值，完成委托数目，完成学习数目，用户状态）
- (3) 学院表（学院 id，学院名）
- (4) 领域表（领域 id，**类代号**，领域名）
- (5) 领域分类表（类 id，类名）
- (6) 学习场所表（场所 id，场所名）
- (7) 学习记录表（标识符，**老师用户名**，**学生用户名**，**领域 id**，开始时间，结束时间，学生对老师的评价，老师对学生的评价）
- (8) 拜师记录表（标识符，**申请人用户名**，**被申请人用户名**，**领域代号**，是否通过）
- (9) 问题记录表（标识符，**提出问题者**，提问时间，**问题领域**，具体描述，处理状态）
- (10) 登录日志表（标识符，**登录者名**，登陆类型，登录时间，登录状态）

英文格式（系统中）与上文一一对应：

- (1) ADM_Table(ADMName, ADMCode)
- (2) USER_Table(UNo, UName, UCode, UTname, UGender, UAge, **UCid**, UPnum, **UFid**, **UPid**, UTvalue, USvalue, UEntrust, ULearn, UStatus)
- (3) COLLEGE_Table(COid, COName)
- (4) FIELD_Table(Fid, **FCid**, FName)
- (5) FIELD_CLASS_Table(Cid, CName)
- (6) PLACE_Table(Pid, PName)

(7) LEARN_RECORD_Table(LRno, LTName, LSName, LRFid, StartTime, EndTime, STEva, TSEva)

(8) MASTER_RECORD_Table(MRno, MSName, MTName, MFid, Flag)

(9) QUESTION_Table(Qno, QName, QTime, QFid, QDescribe, QSolve)

(10) SIGNLOG_Table(SLno, SLName, SLStatus, SLTime, SLReturn)

各个表中的键与约束条件（与上文一一对应）：

(1) 管理员名为主键；管理员密码初值为‘123456’，不唯一，只能是六位数字；所有属性均不能为空。

(2) 标识码设置为标识量，用户名为主键；登录密码初值为‘123456’，只能是六位数字；年龄只能是 0-150 之间的整数；性别用 bit 类型存储，1 表示男性，0 表示女性；联系方式只能是 11 位数字；学院代号为外键，引用自学院表的学院 id，更新方式为级联，删除方式为置空；擅长领域代号为外键，引用自领域表的领域 id，更新方式为级联，删除方式为置空；常去的学习场所代号为外键，引用自学习场所表的场所 id，更新方式为级联，删除方式为置空；讲师潜力值和学徒潜力值初值为 60；完成委托数目与完成学习数目初值为 0；用户状态为 bit 类型，初值设置为 False，True 表示被冻结，False 表示没有被冻结；除了用户名与联系方式外均不唯一，除了学员代号、擅长领域代号、常去场所代号外其他属性均不能为空。

(3) 学院 id 为主键，是标识量，增量为 1，种子为 1；所有属性均不能为空且均唯一。

(4) 领域 id 为主键，是标识量，增量为 1，种子为 1；类代号为外键，引用自领域分类表的类 id，更新方式为级联，删除方式为置空；除类代号外，其他属性均唯一且不能为空。

(5) 类 id 为主键，是标识量，增量为 1，种子为 1；所有属性均不能为空且均唯一。

(6) 场所 id 为主键，是标识量，增量为 1，种子为 1；所有属性均不能为空且均唯一。

(7) 标识符为主键，是标识量，增量为 1，种子为 1；老师用户名和学生用户名均引用自用户表的用户名，更新删除方式为不进行任何操作；领域代号引用自领域表的领域 id，更新方式为级联，删除方式为置空；除了结束时间，学生对老师的评价，老师对学生的评价，领域代号外均不能为空，除了标识符外均不唯一。

(8) 标识符为主键，是标识量，增量为 1，种子为 1；申请人用户名与被申请人用户名均引用自用户表的用户名，更新删除方式为不进行任何操作；领域代号引用自领域表的领域 id，更新方式为级联，删除方式为置空；是否通过用 bit 类型存储，1 表示通过，0 表示不通过；除了标识符外均不唯一，所有属性均不能为空。

(9) 标识符为主键，是标识量，增量为 1，种子为 1；提出问题者引用自用户表的用户名，更新方式为级联，删除方式为不进行任何操作；问题领域引用自领域表的领域 id，更新方式为级联，删除方式为置空，处理状态有“1”，“0”两种转态；初标识符外，其他属性均不唯一，只有问题领域允许是空值。

(10) 标识符为主键，是标识量，增量为 1，种子为 1；登陆者名引用自用户表的用户名与管理员表的管理员名，更新与删除方式均不作任何操作；登录类型约束在“用户”与“管理员”两种值，登录状态约束在“允许”和“拒绝”两种值；除了标识符外均不唯一，所有属性均不能为空。

视图设计：

共设计了 5 个视图，分别为：FIELD_FIELDCLASS_VIEW, USER_MORE_VIEW, QUESTION_MORE_VIEW, LEARN_MORE_VIEW, MASTER_MORE_VIEW。其主要作用就是将各个常用基本表中的领域代号与 FIELD_Table 中的 Fid 相连，显示出 FName

属性值，以方便前端使用，其具体代码与形式会在下文呈现。

触发器设计：

设计了两个触发器，为 LEARN_EndTime_TRIGGER 与 LEARN_Eva_TRIGGER，前者其作用是在 LEARN_RECORD_Table 中进行更新 STEva 或者 TSEva 属性值操作的时候，检查插入后的师生互评值是否均不为空，若均不为空，则对刚才修改的记录添加 EndTime 的属性值，并且修改师徒用户对应的完成的委托/学习数，从而实现师生互评后自动更新结束时间。后者是在 LEARN_RECORD_Table 中进行更新 EndTime 属性值操作的时候，自动按照评分规则更新 USER_Table 中的师傅/学徒潜力值，从而达到结束学习时自动更新用户潜力值。

3.3.2 数据库 E-R 图

(1) 管理员实体 E-R 图：

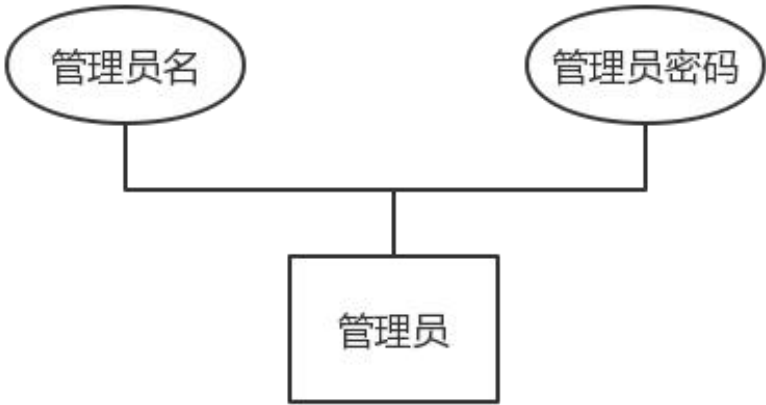


图 3-3 管理员实体及属性

(2) 用户实体 E-R 图：

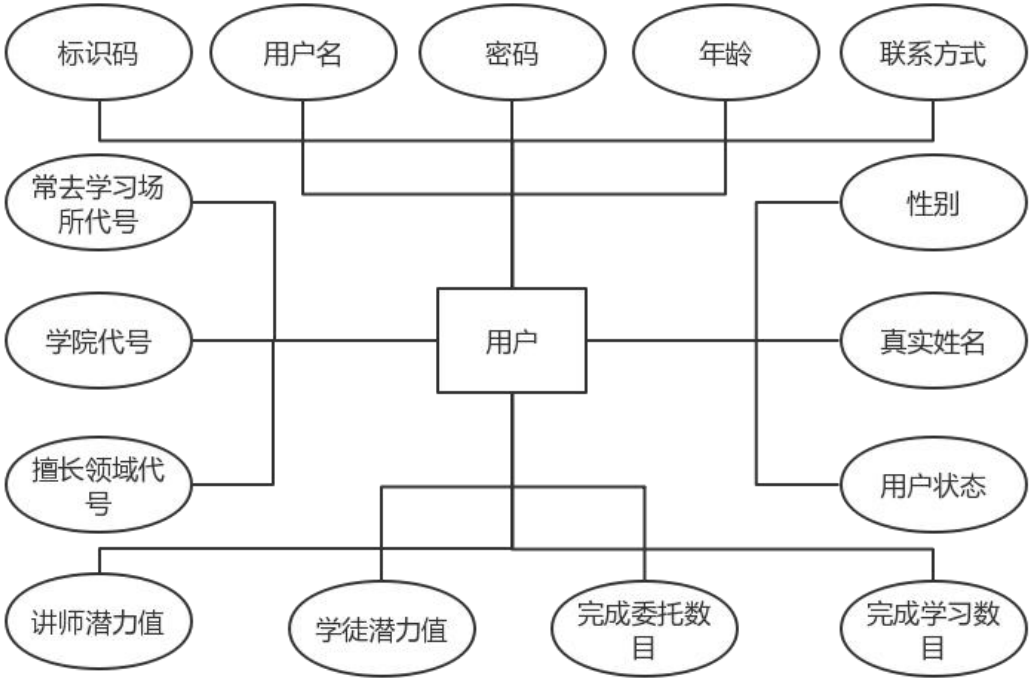


图 3-4 用户实体及属性

(3) 学院实体 E-R 图:

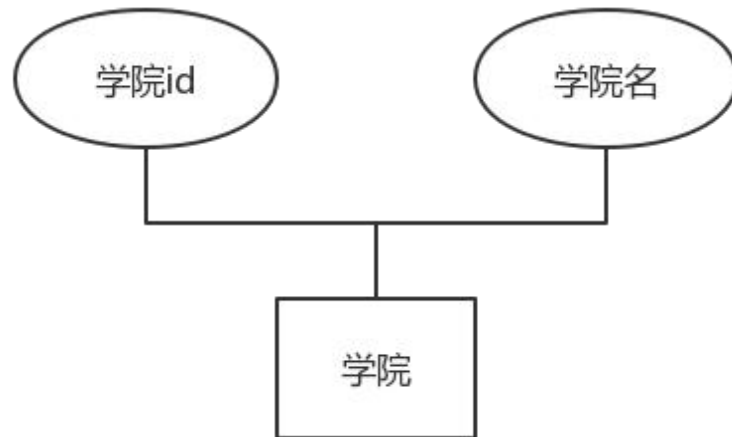


图 3-5 学院实体及属性

(4) 领域实体 E-R 图:

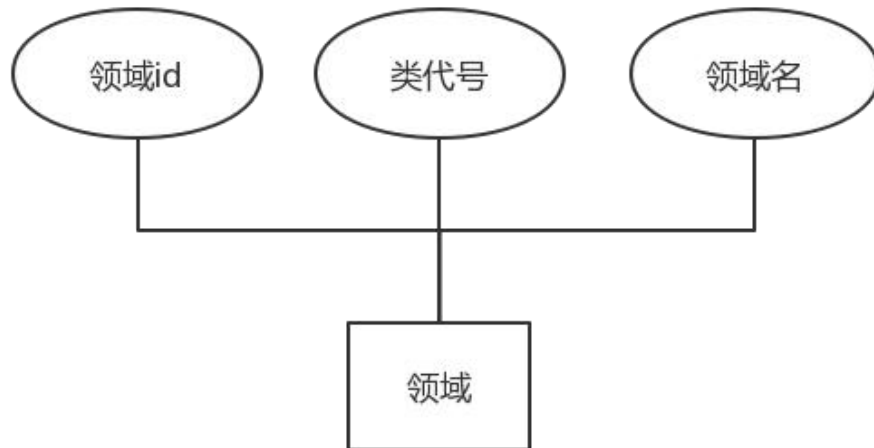


图 3-6 领域实体及属性

(4) 领域分类实体 E-R 图:

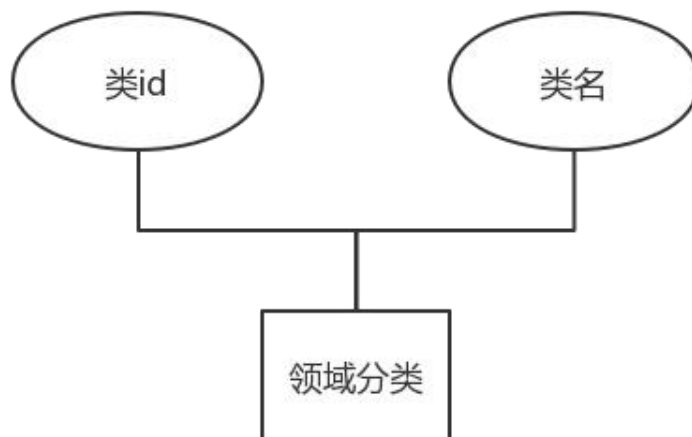


图 3-7 领域分类实体及属性

(6) 学习场所实体 E-R 图:

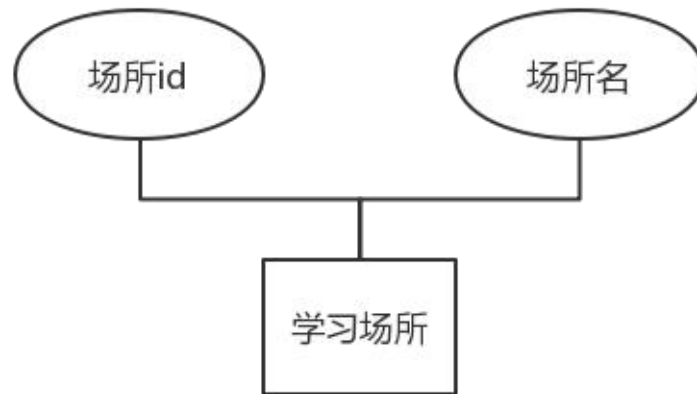


图 3-8 学习场所实体及属性

(7) 问题记录实体 E-R 图:

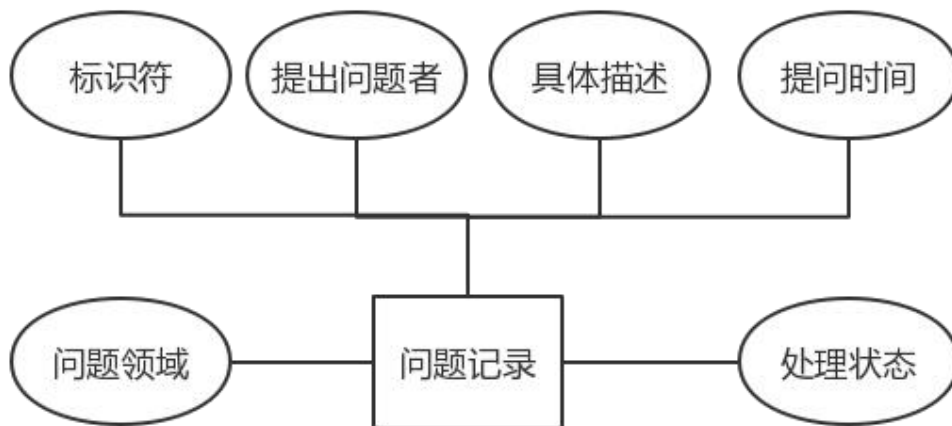


图 3-9 问题记录实体及属性

(8) 拜师记录实体 E-R 图:

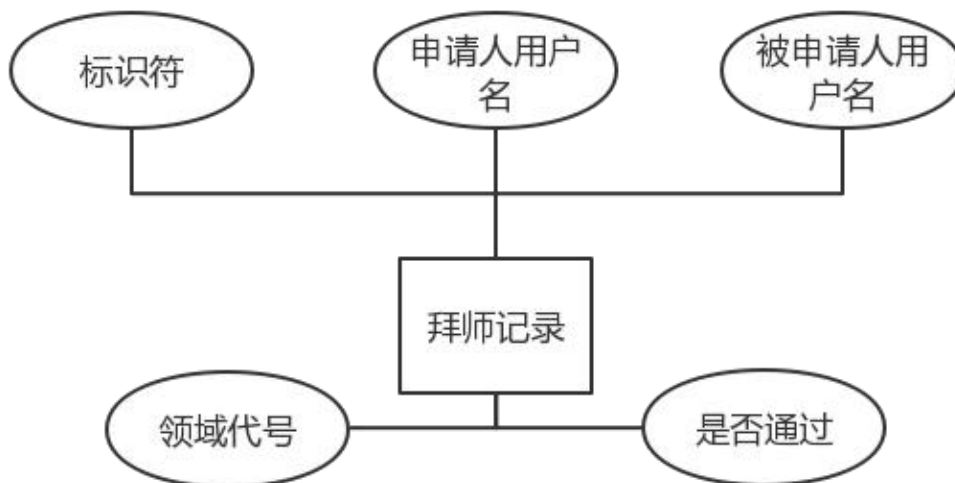


图 3-10 拜师记录实体及属性

(9) 学习记录实体 E-R 图:

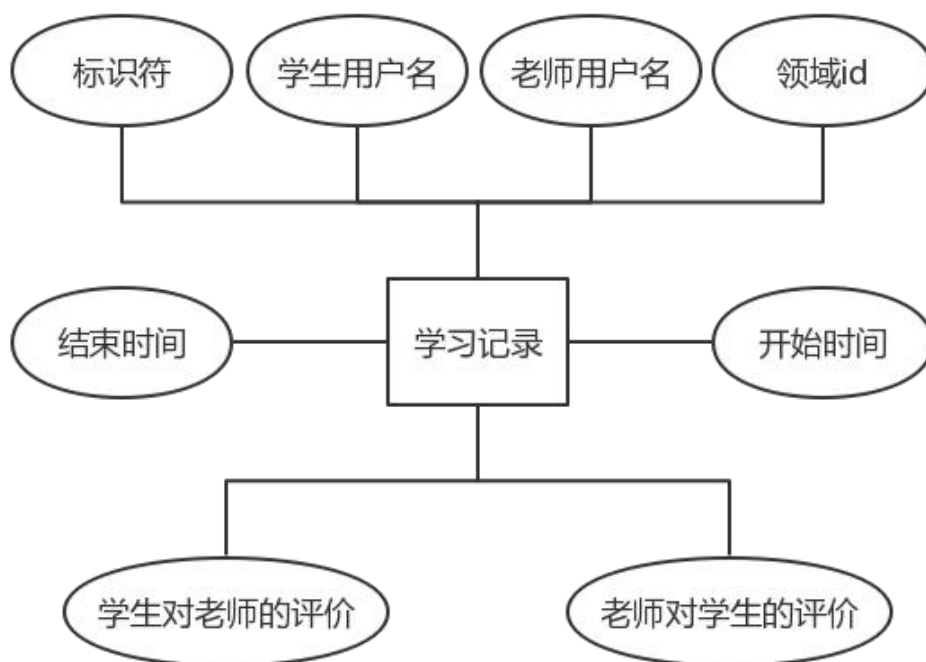


图 3-11 学习记录实体及属性

(10) 登录日志实体 E-R 图:

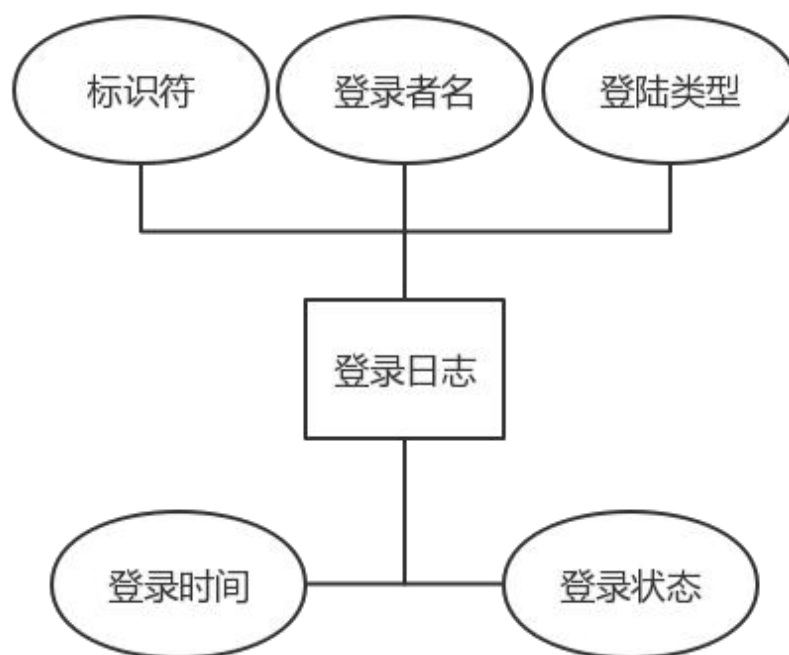


图 3-12 登录日志实体及属性

(11) 系统总 E-R 图:

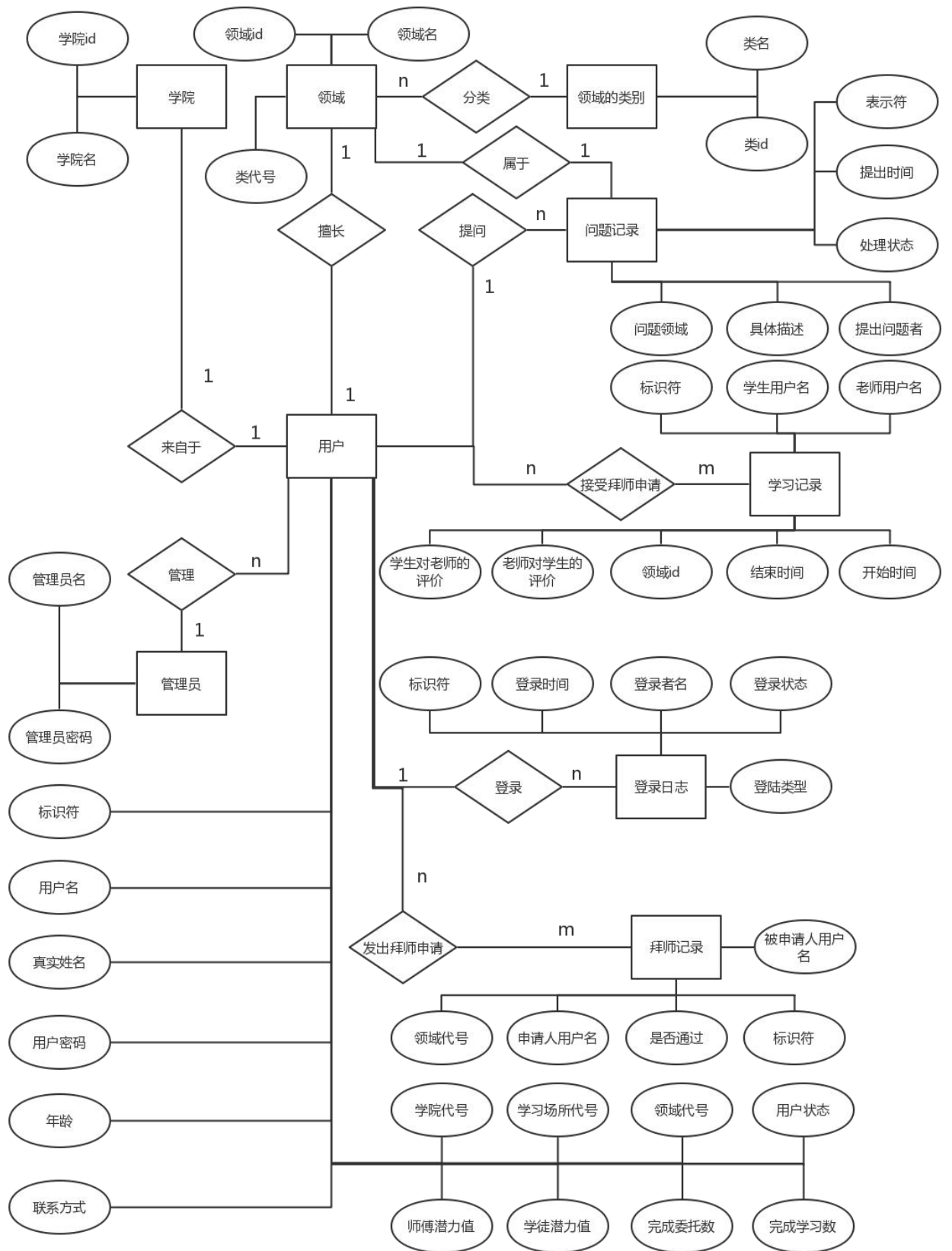


图 3-19 系统总 E-R 图

3.4 系统核心功能设计

3.4.1 核心功能顺序图

UML 通过视图化的表示机制从多个侧面对系统的分析和设计模型进行刻画，其中广泛运用的顺序图是属于行为图的一种，强调了对对象之间发生消息传递的时间顺序。

系统整体数据流动的顺序图如图 3-20:

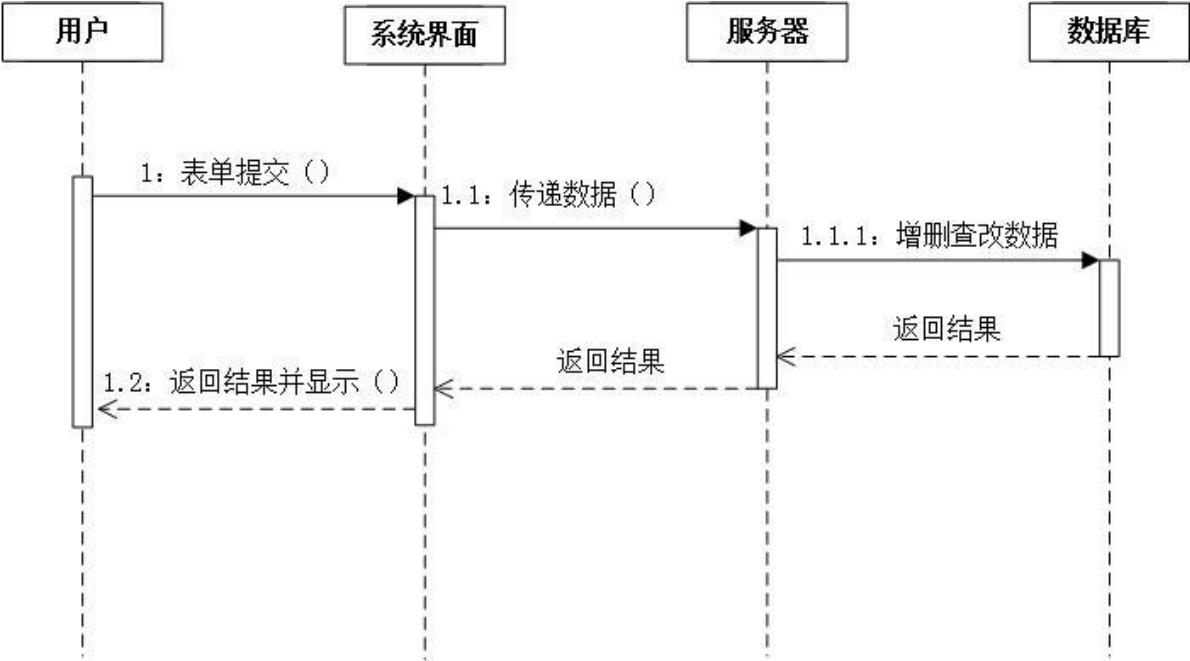


图 3-20 系统顺序图

3.5 逻辑结构设计及物理设计

3.5.1 数据库关系表

表格内容注释:

PK 表示主键，UQ 表示唯一键，FK 表示外键，若有标识量则表示为（标识种子，标识增量），若有外键约束则表示为引用的表。

表 3-1 AMD_Table（管理员表）

字段	数据类型	键	标识/初值	是否可空	Check 约束	外键约束
ADMNo	int	无	(1, 1)	否	无	无
ADMName	nchar(10)	PK	否	否	无	无
ADMCode	nchar(6)	无	否	否	6 位数字	无

其创建代码如下:

```
--管理员表
if exists(select *from sysobjects where name='ADM_Table')
    drop table ADM_Table
go
create table ADM_Table
(
    ADMNo int not null identity(1,1),
    ADMName nchar(10) not null,
    ADMCode nchar(6) not null
)
go
--ADMName 为主键
```

```

if exists(select *from sysobjects where name='PK_ADM_ADMName')
    alter table ADM_Table drop constraint PK_ADM_ADMName
go
alter table ADM_Table
add constraint PK_ADM_ADMName primary key(ADMName)
go
--ADMCode 只能为 6 位数字
if exists(select *from sysobjects where name='CK_ADM_ADMCode')
    alter table ADM_Table drop constraint CK_ADM_ADMCode
go
alter table ADM_Table
add constraint CK_ADM_ADMCode check([ADMCode] like '[0-9][0-9][0-9][0-9][0-9][0-9]')
go

```

表 3-2 USER_Table(用户表)

字段	数据类型	键	标识/初值	是否可空	Check 约束	外键约束
UNo	int	无	(1, 1)	否	无	无
UName	nchar(10)	PK	无	否	无	无
UCode	nchar(6)	无	无	否	6 位数字	无
UTname	nchar(10)	无	无	否	无	无
UGender	bit	无	无	否	无	无
UAge	int	无	无	否	无	无
UPnum	nchar(11)	UQ	无	否	11 位数字	无
UCid	int	FK	无	是	无	COLLEGE_Table
UFid	int	FK	无	是	无	FIELD_Table
UPid	int	FK	无	是	无	PLACE_Table
UTvalue	int	无	60	否	无	无
USvalue	int	无	60	否	无	无
UEntrust	int	无	60	否	无	无
ULearn	int	无	60	否	无	无
UStatus	bit	无	False	否	无	无

其创建代码如下：

```

--用户表
create table USER_Table
(
    UNo int not null identity(1,1),
    UName nchar(10) not null,
    UCode nchar(6) not null,
    UTname nchar(10) not null,
    UGender bit not null,
    UAge int not null,
    UPnum nchar(11) not null,
    UCid int null,
    UFid int null,
    UPid int null,
    UTvalue int not null,

```

```

USvalue int not null,
UEntrust int not null,
ULearn int not null,
UStatus bit not null
)
go
-----USER 表
--UNo 为主键
alter table USER_Table
add constraint PK_USER_UNo primary key(UNo)
go
--UName 唯一键
alter table USER_Table
add constraint UQ_USER_UName unique key(UName)
go
--UPnum 唯一
alter table USER_Table
add constraint UQ_USER_UPnum unique(UPnum)
go
--部分属性的初始值
alter table USER_Table
add constraint DF_USER_UEntrust default(0) for UEntrust
alter table USER_Table
add constraint DF_USER_ULearn default(0) for ULearn
alter table USER_Table
add constraint DF_USER_UTvalue default(60) for UTvalue
alter table USER_Table
add constraint DF_USER_USvalue default(60) for USvalue
alter table USER_Table
add constraint DF_USER_UStatus default('False') for UStatus
go
--部分属性的 check 约束
alter table USER_Table
add constraint CK_USER_UAge check([UAge]>(0) AND [UAge]<=(150))
alter table USER_Table
add constraint CK_USER_UCode check([UCode] like '[0-9][0-9][0-9][0-9][0-9]')
alter table USER_Table
add constraint CK_USER_UPnum check([UPnum] like
'[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]')
go
--添加外键
alter table USER_Table
add constraint FK_USER_UCid_COLLEGE_COid foreign key(UCid) references
COLLEGE_Table(COid)
on update cascade

```

```

on delete set null
alter table USER_Table
add constraint FK_USER_UFid_FIELD_Fid foreign key(UFid) references FIELD_Table(Fid)
on update cascade
on delete set null
alter table USER_Table
add constraint FK_USER_UPid_PLACE_Pid foreign key(UPid) references PLACE_Table(Pid)
on update cascade
on delete set null
go

```

表 3-3 COLLEGE_Table (学院表)

字段	数据类型	键	标识/初值	是否可空	Check 约束	外键约束
COid	int	PK	(1, 1)	否	无	无
COName	nchar(15)	UQ	无	否	无	无

其创建代码如下：

```

--学院表
create table COLLEGE_Table
(
COid int not null identity(1,1),
COName nchar(15) not null
)
Go
--COid 为主键
alter table COLLEGE_Table
add constraint PK_COLLEGE_COid primary key(COid)
--COName 唯一
alter table COLLEGE_Table
add constraint UQ_COLLEGE_COName unique(COid)

```

表 3-4 FIELD_Table (领域表)

字段	数据类型	键	标识/初值	是否可空	Check 约束	外键约束
Fid	int	PK	(1, 1)	否	无	无
FCid	int	FK	无	是	无	FIELDCLASS
FName	nchar(10)	UQ	无	否	无	无

其创建代码如下：

```

--领域表
create table FIELD_Table
(
Fid int not null identity(1,1),
FCid int null,
FName nchar(10) not null
)
go
-----FIELD 表
--Fid 为主键
alter table FIELD_Table

```

```

add constraint PK_FIELD_Fid primary key(Fid)
go
--FName 唯一
alter table FIELD_Table
add constraint UQ_FIELD_FName unique(FName)
go
--FCid 为外键
alter table FIELD_Table
add constraint FK_FIELD_FCid_FIELDCLASS_Cid foreign key(FCid) references
FIELDCLASS_Table(Cid)
on update cascade
on delete set null
go

```

表 3-5 FIELDCLASS_Table（领域分类表）

字段	数据类型	键	标识/初值	是否可空	Check 约束	外键约束
Cid	int	PK	(1, 1)	否	无	无
CName	nchar(10)	UQ	无	否	无	无

其创建代码如下：

```

--领域分类表
create table FIELDCLASS_Table
(
Cid int not null identity(1,1),
CName nchar(10) not null
)
go
-----FIELDCLASS 表
--Cid 为主键
alter table FIELDCLASS_Table
add constraint PK_FIELDCLASS_Cid primary key(Cid)
go
--CName 唯一
alter table FIELDCLASS_Table
add constraint PK_FIELDCLASS_CName unique(CName)
go

```

表 3-6 PLACE_Table（学习场所表）

字段	数据类型	键	标识/初值	是否可空	Check 约束	外键约束
Pid	int	PK	(1, 1)	否	无	无
PName	nchar(10)	UQ	无	否	无	无

其创建代码如下：

```

--学习场所表
create table PLACE_Table
(
Pid int not null identity(1,1),
PName nchar(10) not null
)

```

```

go
-----PLACE 表
--Pid 为主键
alter table PLACE_Table
add constraint PK_PLACE_Pid primary key(Pid)
go
--PName 唯一
alter table PLACE_Table
add constraint UQ_PLACE_PName unique(PName)
go

```

表 3-7 QUESTION_Table (问题记录表)

字段	数据类型	键	标识/初值	是否可空	Check 约束	外键约束
Qno	int	PK	(1, 1)	否	无	无
QName	nchar(10)	FK	无	否	无	USER_Table
QTime	datetime	无	无	否	无	无
QFid	int	FK	无	是	无	FIELD_Table
QDescribe	nchar(50)	无	无	是	无	无
QSolve	bit	无	False	否	无	无

其创建代码如下：

```

--问题记录表
create table QUESTION_Table
(
Qno int not null identity(1,1),
QName nchar(10) not null,
QTime datetime not null,
QFid int null,
QDescribe nchar(50) null,
QSolve bit not null
)
go
-----QUESTION 表
--Qno 为主键
alter table QUESTION_Table
add constraint PK_QUESTION_Qno primary key(Qno)
go
--QSolve 只能为 True 与 False,初值为 False
alter table QUESTION_Table
add constraint CK_QUESTION_QSolve check(QSolve=0 OR QSolve=1)
alter table QUESTION_Table
add constraint DF_QUESTION_QSolve default(0) for QSolve
go
--添加外键
alter table QUESTION_Table
add constraint FK_QUESTION_QFid_FIELD_Fid foreign key(QFid) references FIELD_Table(Fid)
on update cascade

```

```

on delete set null
alter table QUESTION_Table
add constraint FK_QUESTION_QName_USER_UserName foreign key(QName) references
USER_Table(Username)
on update no action
on delete no action
go

```

表 3- 8 SIGNLOG_Table（登录日志表）

字段	数据类型	键	标识/初值	是否可空	Check 约束	外键约束
SLno	int	PK	(1, 1)	否	无	无
SLName	nchar(10)	FK	无	否	无	USER_Table
SLStatus	nchar(15)	无	无	否	User 或 Administrator	无
SLTime	datetime	无	无	否	无	无
SLReturn	nchar(15)	无	无	否	Allowed 或 Refused	无

其创建代码如下：

```

--登录日志表
create table SIGNLOG_Table
(
SLno int not null identity(1,1),
SLName nchar(10) not null,
SLStatus nchar(15) not null,
SLTime datetime not null,
SLReturn nchar(15) not null
)
go
-----SIGNLOG 表
--SLno 为主键
alter table SIGNLOG_Table
add constraint PK_SIGNLOG_SLno primary key(SLno)
go
--约束
alter table SIGNLOG_Table
add constraint CK_SIGNLOG_SLStatus check([SLStatus]='User' OR [SLStatus]='Administrator')
alter table SIGNLOG_Table
add constraint CK_SIGNLOG_SLReturn check([SLReturn]='Allowed' OR [SLReturn]='Refused')
go
--添加外键
alter table SIGNLOG_Table
add constraint FK_SIGNLOG_SLName_USER_UserName foreign key(SLName) references
USER_Table(Username)
on delete no action
on update no action
go

```

表 3-9 LEARN_RECORD_Table (学习记录表)

字段	数据类型	键	标识/初值	是否可空	Check 约束	外键约束
LRno	int	PK	(1, 1)	否	无	无
LTName	nchar(10)	FK	无	否	无	USER_Table
LSName	ncahr(10)	FK	无	否	无	USER_Table
LRFid	int	FK	无	是	无	FIELD_Table
StartTime	datetime	无	无	否	无	无
EndTime	datetime	无	无	是	无	无
SEva	int	无	无	是	[0,100]	无
TSEva	int	无	无	是	[0,100]	无

其创建代码如下：

```

create table LEARN_RECORD_Table
(
  LRno int not null identity(1,1),
  LTName nchar(10) not null,
  LSName nchar(10) not null,
  LRFid int null,
  StartTime datetime not null,
  EndTime datetime null,
  SEva int null,
  TSEva int null
)
go
-----LEARN_RECORD 表
--LRno 为主键
alter table LEARN_RECORD_Table
add constraint PK_LEARN_LRno primary key(LRno)
go
--部分列的 check 约束
alter table LEARN_RECORD_Table
add constraint CK_LEARN_SEva check([SEva]>=(0) AND [SEva]<=(100))
go
alter table LEARN_RECORD_Table
add constraint CK_LEARN_TSEva check([TSEva]>=(0) AND [TSEva]<=(100))
go
--添加外键
alter table LEARN_RECORD_Table
add constraint FK_LEARN_LRFid_FIELD_Fid foreign key(LRFid) references FIELD_Table(Fid)
on update cascade
on delete set null
alter table LEARN_RECORD_Table
add constraint FK_LEARN_LSName_USER_UName foreign key(LSName) references
USER_Table(UName)
on update no action
on delete no action

```



```

go
alter table LEARN_RECORD_Table
add constraint FK_LEARN_LTName_USER_UserName foreign key(LTName) references
USER_Table(UserName)
on update no action
on delete no action
go

```

表 3-10 MASTER_RECORD_Table（拜师申请记录表）

字段	数据类型	键	标识/初值	是否可空	Check 约束	外键约束
MRno	int	PK	(1, 1)	否	无	无
MSName	nchar(10)	FK	无	否	无	USER_Table
MTName	ncahr(10)	FK	无	否	无	USER_Table
MFid	int	FK	无	是	无	FIELD_Table
Flag	bit	无	无	否	无	无

其创建代码如下：

```

--拜师申请记录表
create table MASTER_RECORD_Table
(
MRno int not null identity(1,1),
MSName nchar(10) not null,
MTName nchar(10) not null,
MFid int null,
Flag bit not null
)
go
-----MASTER_RECORD 表
--MRno 为主键
alter table MASTER_RECORD_Table
add constraint PK_MASTER_MRno primary key(MRno)
go
--Flag 只能为 True 或者 False
alter table MASTER_RECORD_Table
add constraint CK_MASTER_Flag check(Flag=1 OR Flag=0)
go
--添加外键
alter table MASTER_RECORD_Table
add constraint FK_MASTER_MFid_FIELD_Fid foreign key(MFid) references FIELD_Table(Fid)
on update cascade
on delete set null
go
alter table MASTER_RECORD_Table
add constraint FK_MASTER_MSName_USER_UserName foreign key(MSName) references
USER_Table(UserName)
on update no action

```

```

on delete no action
go
alter table MASTER_RECORD_Table
add constraint FK_MASTER_MTName_USER_UName foreign key(MTName) references
USER_Table(UName)
on update no action
on delete no action
go

```

3.5.2 数据库视图

表 3-11 FIELD_FIELDCLASS_VIEW (领域-领域类视图)

序号	列	表
1	Fid	FIELD_Table
2	CName	FIELDCLASS_Table
3	FName	FIELD_Table

其创建代码如下:

```

-----FIELD_FIELDCLASS_VIEW
create view FIELD_FIELDCLASS_VIEW
as
select Fid,CName,FName
from FIELD_Table,FIELDCLASS_Table
where Cid=FCid
with check option
go

```

表 3-12 LEARN_MORE_VIEW (学习记录详细视图)

序号	列	表
1	LRno	LEARN_RECORD_Table
2	LTName	LEARN_RECORD_Table
3	LSName	LEARN_RECORD_Table
4	LRFid	LEARN_RECORD_Table
5	FName	FIELD_Table
6	StartTime	LEARN_RECORD_Table
7	EndTime	LEARN_RECORD_Table
8	SEva	LEARN_RECORD_Table
9	TSEva	LEARN_RECORD_Table

其创建代码如下:

```

-----LEARN_MORE_VIEW
create view LEARN_MORE_VIEW
as
select LRno,LTName,LSName,LRFid,FName,StartTime,EndTime,SEva,TSEva
from LEARN_RECORD_Table,FIELD_Table
where LRFid=Fid
with check option
go

```

表 3-13 QUESTION_MORE_VIEW (问题记录详细视图)

序号	列	表
1	Qno	QUESTION_Table
2	QName	QUESTION_Table
3	QTime	QUESTION_Table
4	QFid	QUESTION_Table
5	FName	FIELD_Table
6	QDescribe	QUESTION_Table
7	QSolve	QUESTION_Table

其创建代码如下：

```

-----QUESTION_MORE_VIEW
create view QUESTION_MORE_VIEW
as
select Qno,QName,QTime,QFid,FName,QDescribe,QSolve
from QUESTION_Table,FIELD_Table
where QFid=Fid
with check option
go

```

表 3-14 MASTER_MORE_VIEW (拜师记录详细视图)

序号	列	表
1	MRno	MASTER_RECORD_Table
2	MSName	MASTER_RECORD_Table
3	MTName	MASTER_RECORD_Table
4	MFid	MASTER_RECORD_Table
5	FName	FIELD_Table
6	Flag	MASTER_RECORD_Table

其创建代码如下：

```

-----MASTER_MORE_VIEW
create view MASTER_MORE_VIEW
as
select MRno,MSName,MTName,MFid,FName,Flag
from MASTER_RECORD_Table,FIELD_Table
where MFid=Fid
with check option
go

```

表 3-15 USER_MORE_VIEW (用户信息详细视图)

序号	列	表
1	UName	USER_Table
2	UCode	USER_Table
3	UName	USER_Table
4	UGender	USER_Table
5	UAge	USER_Table
6	UPnum	USER_Table
7	COName	COLLEGE_Table

序号	列	表
8	CName	FIELD_FIELDCLASS_Table
9	FName	FIELD_Table
10	PName	PLACE_Table
11	UTvalue	USER_Table
12	USvalue	USER_Table
13	UEntrust	USER_Table
14	ULearn	USER_Table
15	UStatus	USER_Table

其创建代码如下：

```

-----USER_MORE_VIEW
create view USER_MORE_VIEW
as
select
  UName, UCode, UTname, UGender, UAge, UPnum, COName, CName, FName, PName, UTvalue, USvalue,
  UEntrust, ULearn, UStatus
from USER_Table, COLLEGE_Table, PLACE_Table, FIELD_FIELDCLASS_VIEW
where UFid=Fid and UPid=Pid and UCid=COid
with check option
go

```

3.5.1 数据库触发器

本数据库在表 LEARN_RECORD_Table 上定义了两个触发器，目的是当用户进行师生互评后——即系统对表 LEARN_RECORD_Table 进行了 UPDATE 操作后，触发触发器，在该记录中写入学习结束的时间，并根据师生互评的值修改对应用户的师傅潜力值，学徒潜力值，完成委托的数目，完成学习的数目这四种属性。

其创建代码如下：

```

--触发器 LEARN_EndTime_TRIGGER
create trigger LEARN_EndTime_TRIGGER
on LEARN_RECORD_Table
after update
as
begin
  if update(STEva) or update(TSEva)
  begin
    declare @lrno int
    declare @tname nchar(10)
    declare @sname nchar(10)
    select @lrno=LRno from inserted
    select @tname=LTName from inserted
    select @sname=LSName from inserted
    if exists(select * from inserted,deleted where inserted.LRno=deleted.LRno and inserted.TSEva is
not null and inserted.STEva is not null)
    begin
      update LEARN_RECORD_Table set EndTime=(select GETDATE()) where LRno=@lrno
      update USER_Table set UEntrust+=1 where UName=@tname

```

```

        update USER_Table set ULearn+=1 where UName=@sname
    end;
end;
end;
--触发器 LEARN_Eva_TRIGGER
create trigger LEARN_Eva_TRIGGER
on LEARN_RECORD_Table
after update
as
begin
    if update(EndTime)
    begin
        declare @tname nchar(10)
        declare @sname nchar(10)
        declare @steva int
        declare @tseva int
        select @tname=LTName from inserted
        select @steva=SEva from inserted
        select @sname=LSName from inserted
        select @tseva=TSEva from inserted
        update USER_Table set UTvalue=(UTvalue+@steva)/2 where UName=@tname
        update USER_Table set USvalue=(USvalue+@tseva)/2 where UName=@sname
    end;
end;

```

3.5.1 数据库整体关系图

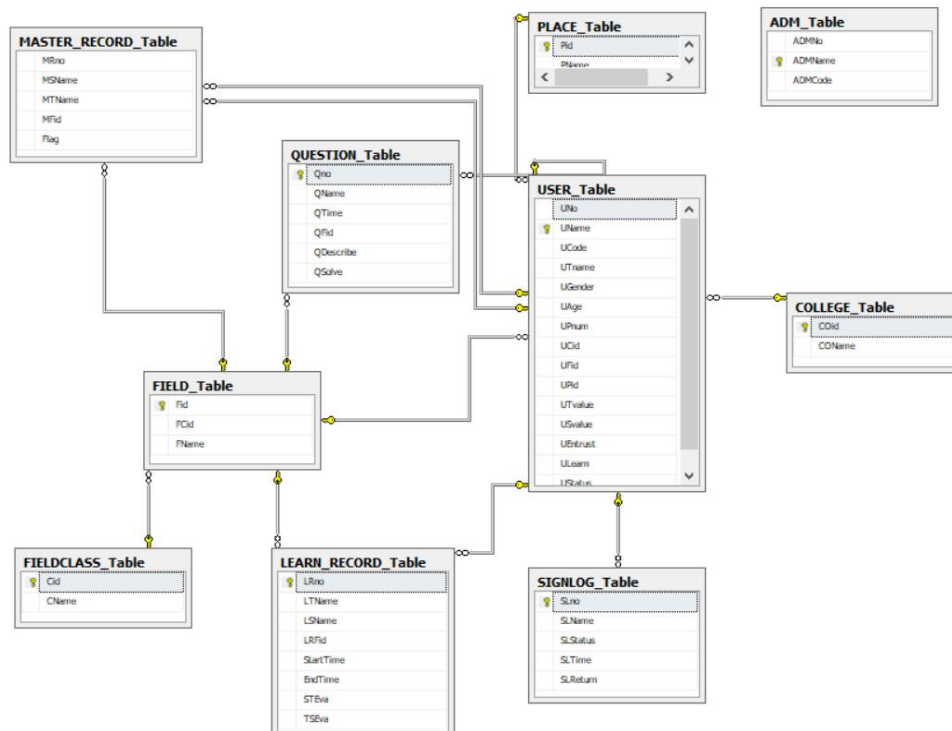


图 3-21 数据库整体关系图

3.6 模块划分

按照功能分析图将系统划分为如下几个模块：

- (1) 管理员登录模块，完成管理员名与密码的验证。
- (2) 管理员管理模块，完成管理员对用户的管理，对系统信息的查看。
- (3) 用户登录模块，完成用户名与密码的验证。
- (4) 用户注册模块，完成新用户的注册。
- (5) 用户信息模块，完成用户信息的修改，系统信息的查询。
- (6) 用户拜师学习模块，完成拜师，开始学习，师生互评，结束学习等一系列核心功能。
- (7) 数据库模块，完成系统依托的数据库设计。

4 开发与实现

4.1 系统核心程序编写

4.1.1 C#访问 SQL Server 数据库

连接数据库服务器，就是客户端向已运行的数据库服务器发出连接请求，成功后就可以对数据库进行访问，执行相应的操作。C#提供了一系列 SQL Server 连接和操作的类，类 `SqlConnection` 的实例用来连接数据库，类 `SqlCommand` 的实例用来对数据库进行操作，类 `SqlDataReader` 的实例用来读取查询操作的结果，其实现代码如下：

```
//连接数据库
SqlConnection con = new SqlConnection();
con.ConnectionString = "server=.;database=NJAUST;uid=sa;pwd=msf97347110lol";
con.Open();
//对数据库进行查询操作
SqlCommand com = new SqlCommand();
com.Connection = con;
com.CommandType = CommandType.Text;
com.CommandText = "SQL 查询语句";
SqlDataReader sr = com.ExecuteReader();
//读取
while(sr.Read())
{变量名 = sr["表的列名"].ToString();}
sr.Close();
//对数据库进行插入等操作
com.CommandText = "SQL 插入，删除，更新语句";
com.ExecuteNonQuery();
//断开连接
con.Close();
```

4.1.2 验证用户登录的流程图

用户登录时除了要匹配用户名与密码之外，还要查看用户的账户是否被冻结，在代码中其实现的流程图如下：

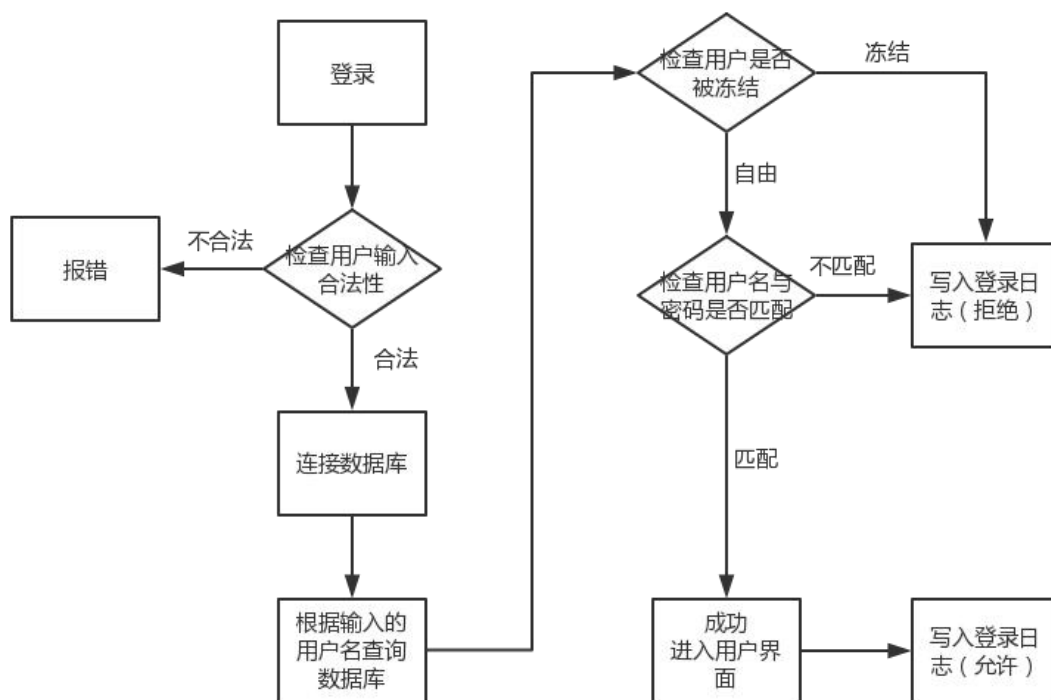


图 3-22 用户登录流程图

4.1.3 实现用户拜师功能的流程图

用户的拜师功能是本系统的核心功能，用户可以在这里发出拜师申请，查看申请，同意申请并开始学习，师生互评，结束学习等多种系统功能，在代码中其实现的流程图如下：

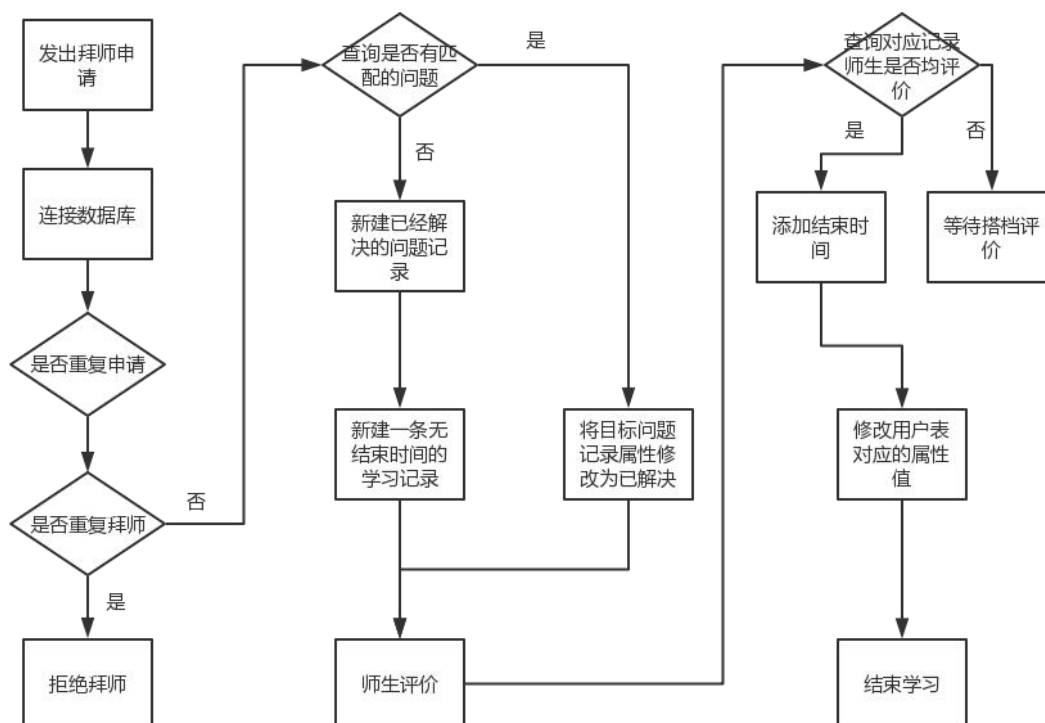


图 3-22 拜师功能流程图

4.2 系统实施

4.2.1 管理员登录模块

本模块由“用户登录界面”的“管理员入口”进入。可以允许系统内记录的管理员登录，进入“管理员界面”。

图 4-1 管理员登录模块

4.2.2 管理员管理模块

管理员登录请求被允许后进入本模块，在本模块中，允许管理员查看系统内存储的全部领域及其分类，全部的学院，全部的学习场所以及历史登录日志，还允许管理员清空登录日志，添加新的管理员，并且查看或检索某用户信息，实现对用户账户的冻结和解冻的管理操作。

图 4-2 管理员管理模块

4.2.3 用户注册模块

本模块由“用户登录界面的”的“注册”进入，在这里可以实现创建一个新用户的功能，但要注意的是，这里的用户名必须与系统存储的用户名不同且不能包含空格，电话号码也不能与系统内存储的不同。



The screenshot shows a Windows-style window titled "Form1_2" with a standard minimize, maximize, and close button set. The main content area has a title bar "用户注册界面" (User Registration Interface). Below the title, there are several input fields and labels: "用户名:" (Username) with a text box and a label "用户名最长9个字符" (Username max 9 characters); "密码:" (Password) with a text box and a label "密码6位数字" (Password 6 digits); "真实姓名:" (Real Name) with a text box and a label "姓名最长9个字符" (Name max 9 characters); "性别:" (Gender) with a dropdown menu; "年龄:" (Age) with a text box and a label "年龄0~150岁" (Age 0~150 years); "电话号码:" (Phone Number) with a text box and a label "11位有效号码" (11-digit valid number); "学院:" (College) with a dropdown menu; "擅长领域:" (Specialty) with a dropdown menu; and "学习场所:" (Learning Place) with a dropdown menu. At the bottom, there are two buttons: "确定" (OK) and "取消" (Cancel). A red note on the right side says "注意 用户名一旦确认 便不允许去修改" (Note: Username once confirmed, not allowed to be modified).

图 4-3 用户注册模块

4.2.4 用户登录模块

本模块为系统启动后弹出的第一个界面，用户在输入用户名和密码后可以点击“登录”，匹配成功后进入“用户信息模块”，当用户忘记密码时，也可以在这里点击“忘记密码”，通过注册时的电话号进行找回。



The screenshot shows a Windows-style window titled "Form1" with a standard minimize, maximize, and close button set. The main content area has a title bar "用户登录界面" (User Login Interface). Below the title, there are two input fields: "用户名:" (Username) and "密码:" (Password), each with a text box. Below the input fields, there are five buttons: "登录" (Login), "忘记密码" (Forgot Password), "注册" (Register), "管理员入口" (Admin Entry), and "关闭" (Close).

图 4-4 用户登录模块

4.2.5 用户信息模块

用户登录成功后进入本模块，在本模块中你可以点击“师生交互”进入“用户拜师学习模块”，同时本模块也集成了“查看个人信息”，“修改个人信息”，“排行榜”，“查看历史记录”等多种功能，方便用户使用。



The interface for the 'User Information Module' (南农师徒系统) displays user details for 'Bob'. On the left, a sidebar contains buttons for '查看个人信息' (View Personal Information), '排行榜' (Ranking), '师生交互' (Teacher-Student Interaction), '修改个人信息' (Modify Personal Information), and '退出系统' (Exit System). The main area shows user data: '用户名: Bob', '注册编号: 1号', '真实姓名: 曹植', '学院: 农学院', '电话号码: 18863059124', '学习场所: 图书馆', '擅长领域: 单机游戏', '性别: 男', and '年龄: 21'. On the right, there are buttons for '查看个人学习记录', '查看个人拜师记录', and '查看个人收徒记录', along with statistics: '师傅潜力值: 93', '学徒潜力值: 97', '完成的委托: 6', and '完成的学习: 12'.

图 4-5 用户信息模块

4.2.6 用户拜师学习模块

本模块是系统的核心模块，这里用户可以“发布问题”，也可以选择筛选方式后，根据筛选条件进行筛选，点击“显示信息”后，系统即可显示你选中的用户的具体信息，你也可以点击“已发布的问题”来查看有谁需要帮助；点击“当前学习状况”开查看或结束当前的学习；点击“当前申请状况”来查看自己的请求是否通过，查看自己同意过的申请，同意其他用户对你发出的申请；点击“发出拜师申请”来向选中的用户发出拜师申请。



The interface for the 'User Mentorship Learning Module' (师生交互) allows users to interact with the system. It features a '当前用户: Bob' header. On the left, there's a '发布问题' (Post Question) button and a text area for '问题描述' (Problem Description) with a red warning: '问题描述不能超过25个字哦要简明确切'. Below this are filters for '选择筛选方式', '要筛选的学院', '要筛选的场所', and '要筛选的领域'. The center section shows a '查询结果' (Search Results) area with a '显示信息' (Show Information) button and a '筛选' (Filter) button. The right side displays a list of user details: '真实姓名', '电话', '擅长领域', '学习场所', '完成委托', '师傅潜力值', '学院', '性别', and '年龄'. At the bottom right, there are buttons for '已发布的问题', '当前学习状况', '当前申请状况', and '发出拜师申请'. A search result summary at the bottom center shows '搜索到的大神 0 人'.

图 4-6 用户拜师学习模块

5 系统测试与应用

5.1 系统测试运行环境

系统运行环境有服务器端、因特网、客户端构成。

5.1.1 服务器端运行环境

服务器端由 Web 服务器、应用程序服务器、数据库服务器等组成。

服务器运行环境基本要求：操作系统为 Windows10，数据库服务器为 SQL Server 2008R2。

5.1.2 客户端运行环境

系统能够用于访问各类 Internet 的计算机，只要计算机能够支持 TCP/IP 协议，操作系统为 Windows 10 即可以支持本系统。

5.2 数据库测试与运行

5.2.1 管理员登录模块

输入管理员名：MSF_1，管理员密码：123456，点击“登录”。



图 5-1 进入管理员登录界面

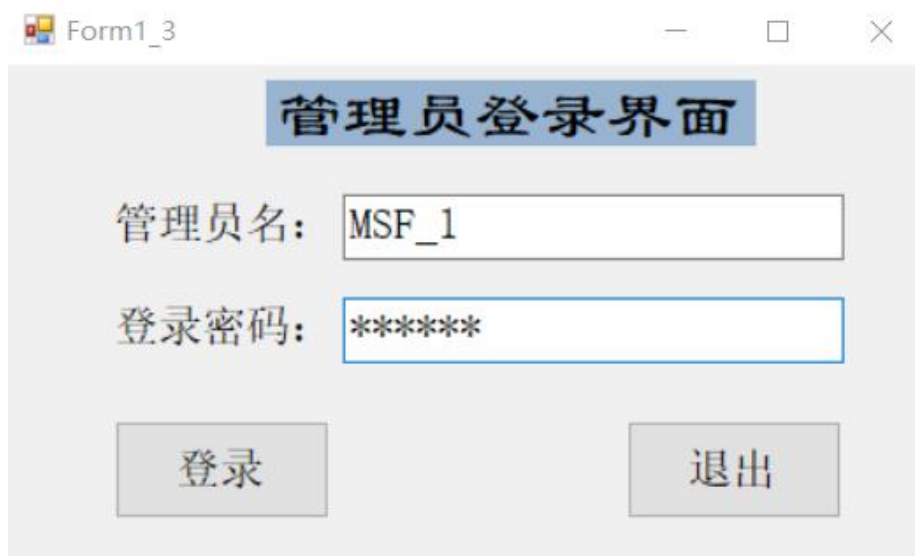


图 5-2 输入信息后的管理员登录界面

5.2.2 管理员管理模块

登录成功后进入“管理员界面”。下面对各个功能进行测试演示。

(1) 管理员界面



图 5-3 管理员界面

(2) 查看登录日志




图 5-4 查看登录日志功能

(3) 清空登录日志

	SLno	SLName	SLStatus	SLTime	SLReturn
*	NULL	NULL	NULL	NULL	NULL

图 5-4 清空登录日志功能（清空后的 SIGNLOG_Table）

(4) 添加管理员



Form3_2

添加管理员

操作的管理员 MSF_1

设置账户名 MSF_5

设置密码 123456

创建 取消

图 5-5 添加管理员功能



Form3

管理员界面

当前管理员 MSF_5

查看登录日志 查看全部领域

清空登录日志 查看全部学院

添加新管理员 查看学习场所

修改用户状态 退出登录

图 5-6 新管理员登录成功

(5) 查看系统基本信息

Form3_3

查看领域列表

当前管理员 MSF_5

31条记录

5条记录

序号	类名	
1	专业课	英语
2	专业课	高数
3	专业课	线性代
4	专业课	概率论
5	专业课	金融
6	专业课	社会
7	专业课	C语言
8	专业课	数据
9	软件	PS

刷新

退出

图 5-7 查看全部领域及其分类功能

Form3_4

查看学院目录

当前管理员 MSF_5

17条记录

序号	代号	学院
1	1	农学院
2	2	植物保护学院
3	3	资源与环境科学学
4	4	园艺学院
5	5	动物科技学院
6	6	动物医学院
7	7	经济管理学院
8	8	公共管理学院

刷新

退出

图 5-8 查看全部学院功能

Form3_5

查看学习场所目录

当前管理员 MSF_5 8条记录

序号	代号	学习场所
1	1	图书馆
2	2	主楼自习室
3	3	教学楼自习室
4	4	教四楼自习室
5	5	宿舍
6	6	一食堂
7	7	二食堂
8	8	三食堂

刷新 退出

图 5-9 查看全部学习场所功能

(6) 查看或修改用户信息

Form3_6

修改用户状态

当前管理员 MSF_5

用户列表

- Bob
- Brown
- Davix
- Garcia
- Jace
- Jack
- Jones
- Smith
- Taylor
- Tom

10条记录

输入用户名

搜索

电话号码 NULL

密码 NULL

账户状态 NULL

解冻账户 冻结账户

刷新 显示细节 退出

图 5-10 查看全部用户名

Form3_6

当前管理员
MSF_5

修改用户状态

用户列表

Bob
Brown
Davix
Garcia
Jace
Jack
Jones
Smith
Taylor
Tom

10条记录

输入用户名

搜索

电话号码 18863059124

密码 123456

账户状态 自由

解冻账户 冻结账户

刷新 显示细节 退出

图 5-11 显示某用户的信息

Form3_6

当前管理员
MSF_5

修改用户状态

用户列表

Jack

1条记录

输入用户名

搜索

电话号码 18865329875

密码 654321

账户状态 自由

解冻账户 冻结账户

刷新 显示细节 退出

图 5-12 搜索某用户

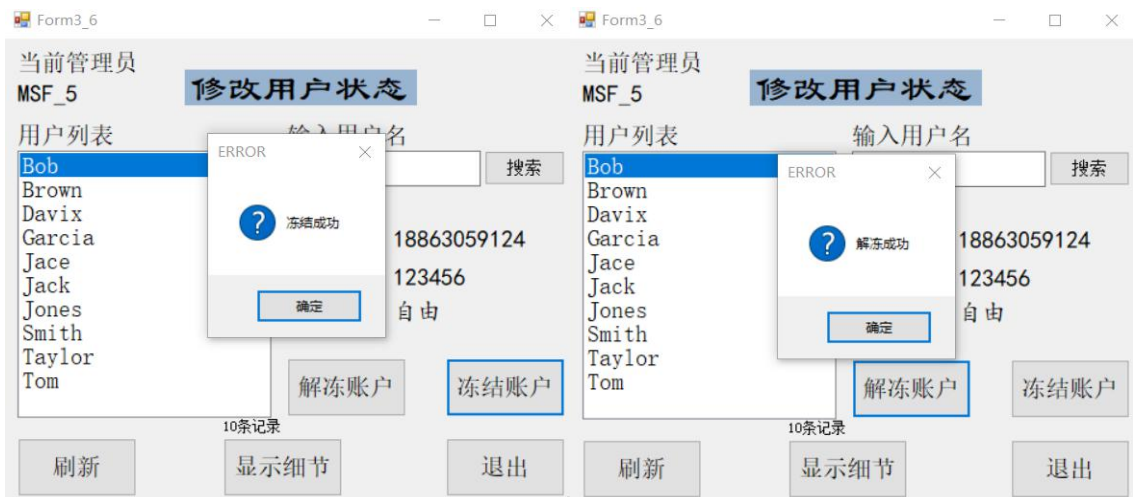


图 5-13 冻结或解冻某用户

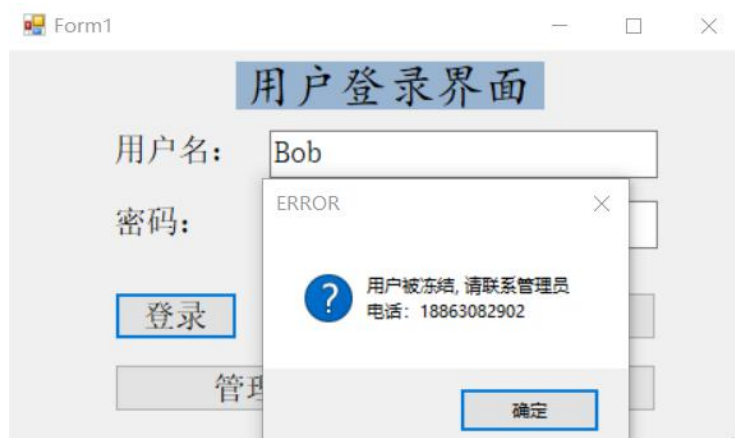


图 5-14 冻结后的用户无法登陆

5.2.3 用户注册模块

用户点击“注册”后，输入相关信息后点击“确定”即可完成注册。

图 5-15 用户注册界面

5.2.4 用户登录模块

我们对 5.2.3 中创建的新用户“Jarry”进行操作，假设我们忘了他的密码，在“用户登录界面”中点击“忘记密码”即可进入密码找回界面，这里我们设置新密码为“654321”。

找回密码

手机号: 18859874562

用户名: Jarry

新密码: 654321

密码为六位数字

确定 取消

图 5-16 找回密码界面

修改密码成功后，我们用新密码登录“Jarry”用户。

欢迎你! Jarry

南农师徒系统

查看个人信息

查看个人学习记录

查看个人拜师记录

查看个人收徒记录

排行榜

师生交互

修改个人信息

退出系统

用户名: Jarry 注册编号: 12号

真实姓名: 刘婵

学院: 信息科技学院

电话号码: 18859874562

学习场所: 图书馆

擅长领域: 数据库

性别: 男

年龄: 18

师傅潜力值: 60

学徒潜力值: 60

完成的委托: 0

完成的学习: 0

图 5-17 用户登录成功后进入系统主界面

5.2.5 用户信息模块

我们使用“Jarry”用户进行前两功能测试，使用“Bob”用户进行后续功能测试。下面进行各种功能演示。

(1) 修改个人信息



你的用户名: Jarry

真实姓名: 刘二婵

学习场所: 主楼自习室

账户密码: 654321

擅长领域: C语言

年龄: 20

电话号码: 18898756542

性别: 女

学院: 信息科技学院

保存设置

取消

图 5-18 修改个人信息界面



欢迎你! Jarry

南农师徒系统

查看个人信息

排行榜

师生交互

修改个人信息

退出系统

用户名: Jarry 注册编号: 12号

真实姓名: 刘二婵

学院: 信息科技学院

电话号码: 18898756542

学习场所: 主楼自习室

擅长领域: C语言

性别: 女

年龄: 20

查看个人学习记录

查看个人拜师记录

查看个人收徒记录

师傅潜力值: 60

学徒潜力值: 60

完成的委托: 0

完成的学习: 0

图 5-19 显示修改后的信息

(2) 查看排行榜



图 5-20 用户基本信息排行

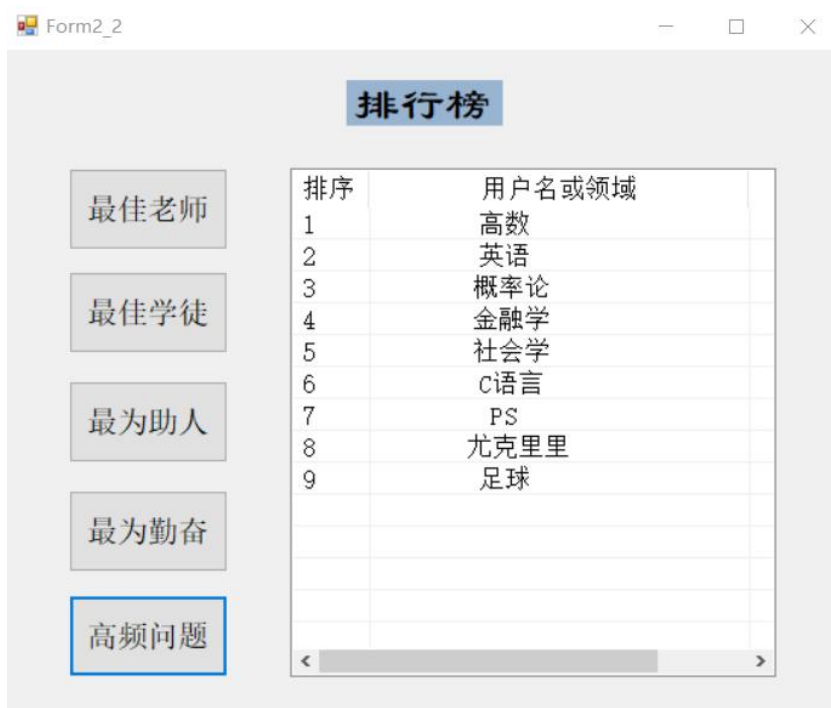


图 5-21 高频问题领域排行

(2) 查看历史记录

Form2_4

个人学习记录

用户 Bob 10条记录

序号	老师	领域	开始时间	结束时间
1	Brown	概率论 ...	2020/5/1 0:00:00	2020/5/2 0:00:00
2	Jack	高数 ...	2020/5/2 0:00:00	2020/6/10 13:14:...
3	Brown	概率论 ...	2020/5/14 0:00:00	
4	Jack	高数 ...	2020/6/1 0:00:00	2020/6/10 13:14:...
5	Jack	高数 ...	2020/6/1 0:00:00	2020/6/12 11:27:...
6	Jack	高数 ...	2020/6/2 0:00:00	2020/6/12 11:27:...
7	Jack	高数 ...	2020/6/2 0:00:00	2020/6/10 13:14:...

刷新 退出

图 5-22 个人学习记录

Form2_5

拜师记录

用户 Bob 3条记录

序号	师傅	领域
1	Smith	数据库
2	Taylor	金融学
3	Tom	英语

刷新 退出

图 5-23 个人拜师记录

Form2_6

收徒记录

用户 Bob 2条记录

序号	徒弟	领域
1	Jones	单机游戏
2	Smith	单机游戏

刷新 退出

图 5-24 个人收徒记录

5.2.6 用户拜师学习模块

用户登录后点击“师生交互”进入“师生交互界面”。

(1) 发布问题

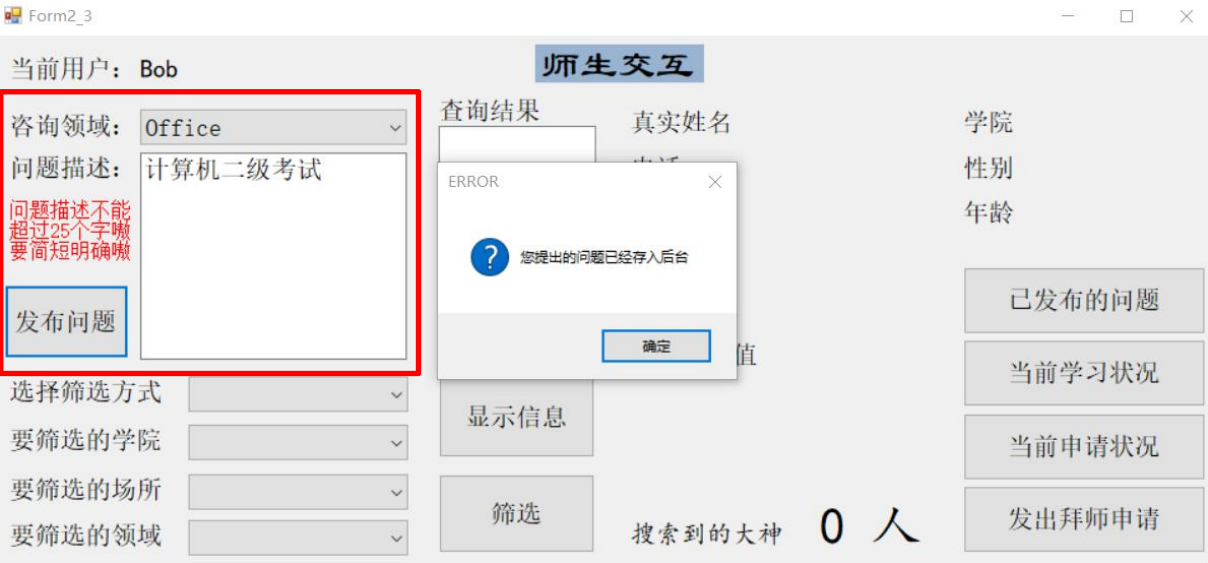


图 5-25 发布一个问题

(2) 查看发布问题

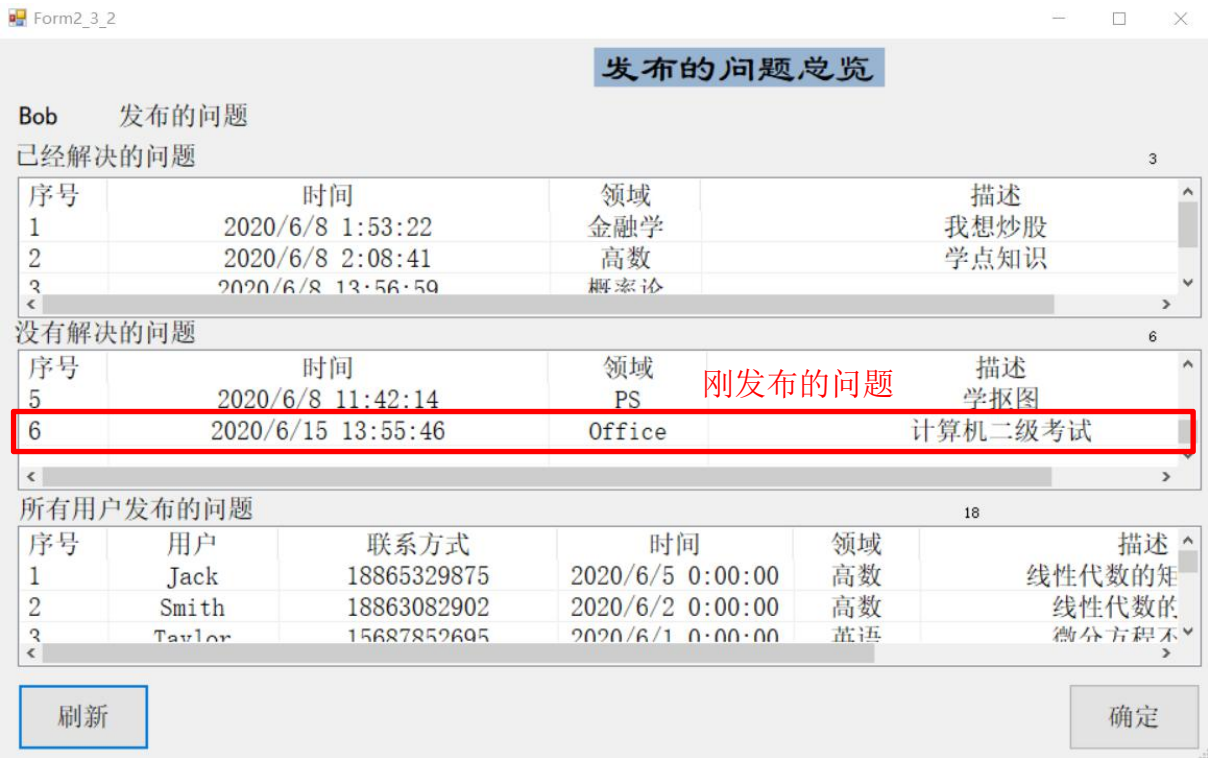


图 5-26 查看发布的问题

(3) 筛选一个用户，并显示他的信息，这里我们筛选一位“C 语言”用户

Form2_3

当前用户: Bob

师生交互

咨询领域: Office

问题描述: 计算机二级考试

问题描述不能超过25个字哦
要简短明确哦

发布问题

选择筛选方式 领域

要筛选的学院

要筛选的场所

要筛选的领域 C语言

查询结果

Jarry

真实姓名 刘二婵

电话 18898756542

擅长领域 C语言

学习场所 主楼自习室

完成委托 0

师傅潜力值 60

学院 信息科技学院

性别 女

年龄 20

已发布的问题

当前学习状况

当前申请状况

发出拜师申请

显示信息

筛选

搜索到的大神 1人

图 5-27 筛选用户

(4) 这里 Bob 用户可以向 Jarry 用户发出拜师申请

Form2_3

当前用户: Bob

师生交互

咨询领域: Office

问题描述: 计算机二级考试

问题描述不能超过25个字哦
要简短明确哦

发布问题

选择筛选方式 领域

要筛选的学院

要筛选的场所

要筛选的领域 C语言

查询结果

Jarry

真实姓名 刘二婵

电话 18898756542

擅长领域 C语言

学习场所 主楼自习室

完成委托 0

师傅潜力值 60

学院 信息科技学院

性别 女

年龄 20

已发布的问题

当前学习状况

当前申请状况

发出拜师申请

显示信息

筛选

搜索到的大神 1人

Form2_3_1

请开始你的拜师

请仔细核对以下信息，一旦确认不可更改

老师用户名 Jarry

老师领域 C语言

你的用户名 Bob

求助领域 C语言

确认拜师

取消拜师

图 5-28 发出拜师申请

(4) 发出申请后，我们可以让 Bob 查看当前申请状况

Form2_3_3

当前申请状态

用户 Bob

你发出的申请

序号	老师	领域	状态
6	Taylor	金融学	通过
7	Tom	英语	通过
8	Jarry	C语言	未通过

你收到的申请(未批准)

真实姓名

电话

求助领域

常去场所

学徒潜力值

完成的学习

0

你批准的申请

序号

学生

单

1	Jones	单
2	Smith	单
3	Smith	单

刷新

确认

图 5-29 当前申请状况

(5) 登录 Jarry 用户，让他同意 Bob 的申请。

你收到的申请(未批准)

Bob

1

你批准的申请

真实姓名

曹植

电话

18863059124

求助领域

C语言

常去场所

图书馆

学徒潜力值

97

完成的学习

12

细节

同意

GOOD

操作成功，请点击刷新检查

确定

图 5-30 同意拜师申请

48

(6) 老师同意后，即建立学习关系，我们可以查看 Bob 的当前学习状况。

图 5-31 当前学习状况

(7) 若想结束学习，则要求师生进行互评，这里我们让 Bob 与 Jarry 进行互评。

图 5-32 师生互评

(8) 数据库后台查看 Bob 与 Jarry 的学习记录，因为 Jarry 是新建用户，所以他的师傅潜力值，学徒潜力值，完成委托数，完成学习数是初值，学习完成后在数据库中查看 Jarry 的个人信息变化。

20	Jarry	Bob	7	2020-06-15 14:...	2020-06-15 14:...	100	100
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

图 5-33 数据库中 Jarry 与 Bob 的测试学习记录

	UName	UTvalue	USvalue	UEntrust	ULearn
1	Jarry	80	60	1	0

图 5-33 数据库中 Jarry 的个人信息变化

6 总结与展望

6.1 总结

本系统在充分分析校园拜师需求的基础之上，实现了如下核心功能：

- (1) 管理员登录；
- (2) 管理员管理用户信息，查看系统基本信息；
- (3) 用户查看修改个人信息；
- (4) 用户查看历史记录；
- (5) 用户查看排行榜；
- (6) 用户进行拜师学习；
- (7) 用户师生互评与结束学习。

目前，本系统的大体核心功能都基本实现，在测试中也体现了初步的实用功能，但是整个系统的应用上来说还是有很多的不足。比如师生互动仅局限在拜师申请与师生互评中，并不能完全适应用户的需求，系统缺少用户主动收徒的功能，管理员不能对系统的一些基本信息进行直接修改。

通过本次数据库课程设计，我熟悉了数据库设计的基本流程，了解了数据库在实际应用中的逻辑结构与物理结构的设计思路，学会了模型由 2NF 逐渐到 3NF 的改良，最重要的是学习了后端与前端之间的连接，用前端去实现系统的方法。

6.2 展望

本系统面向南京农业大学在校学生，希望在之后能够在校园实施，真正实现学生之间的互帮互助，养成良好的学习氛围。并且我也提出如下系统的改进方案：

- (1) 增加师生聊天功能，实现用户之间的在线聊天；
- (2) 修改基本表设计，在每个表中增加一个属性值，用来标明某条记录是否被删除；
- (3) 增加管理员修改系统基本信息的功能，扩大管理员权限；
- (4) 完成网页端与小程序端的开发，处理客户端，网页端，小程序端的登录请求冲突。

参考文献：

- [1] C#教程|菜鸟教程. <https://www.runoob.com/csharp/csharp-tutorial.html>
- [2] 王珊. 数据库系统概论[J]. 高等教育出版社，2014