目录

[1需求分析 2](#_Toc486076869)

[1.1功能性需求 2](#_Toc486076870)

[1.2非功能性需求 2](#_Toc486076871)

[1.3约束需求 2](#_Toc486076872)

[2系统设计 3](#_Toc486076873)

[2.1流程图 3](#_Toc486076874)

[2.2框架结构 4](#_Toc486076875)

[2.3数据库 4](#_Toc486076876)

[3系统实现 8](#_Toc486076877)

[3.1核心代码 8](#_Toc486076878)

[3.2程序截图 11](#_Toc486076879)

[4总结 15](#_Toc486076880)

[参考文献 15](#_Toc486076881)

# 1需求分析

## 1.1功能性需求

这是一个答题系统。管理员可以管理题库，增删改题，普通用户登录后可以答题，也可以查看历史答题记录。

## 1.2非功能性需求

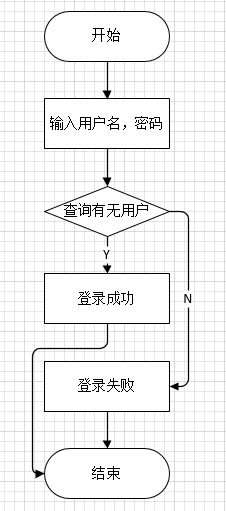
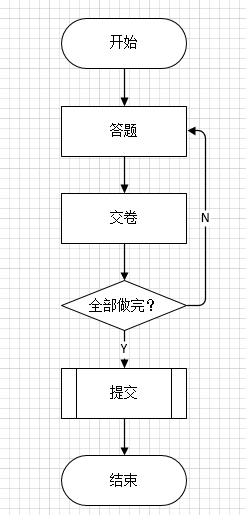
界面美观大方

## 1.3约束需求

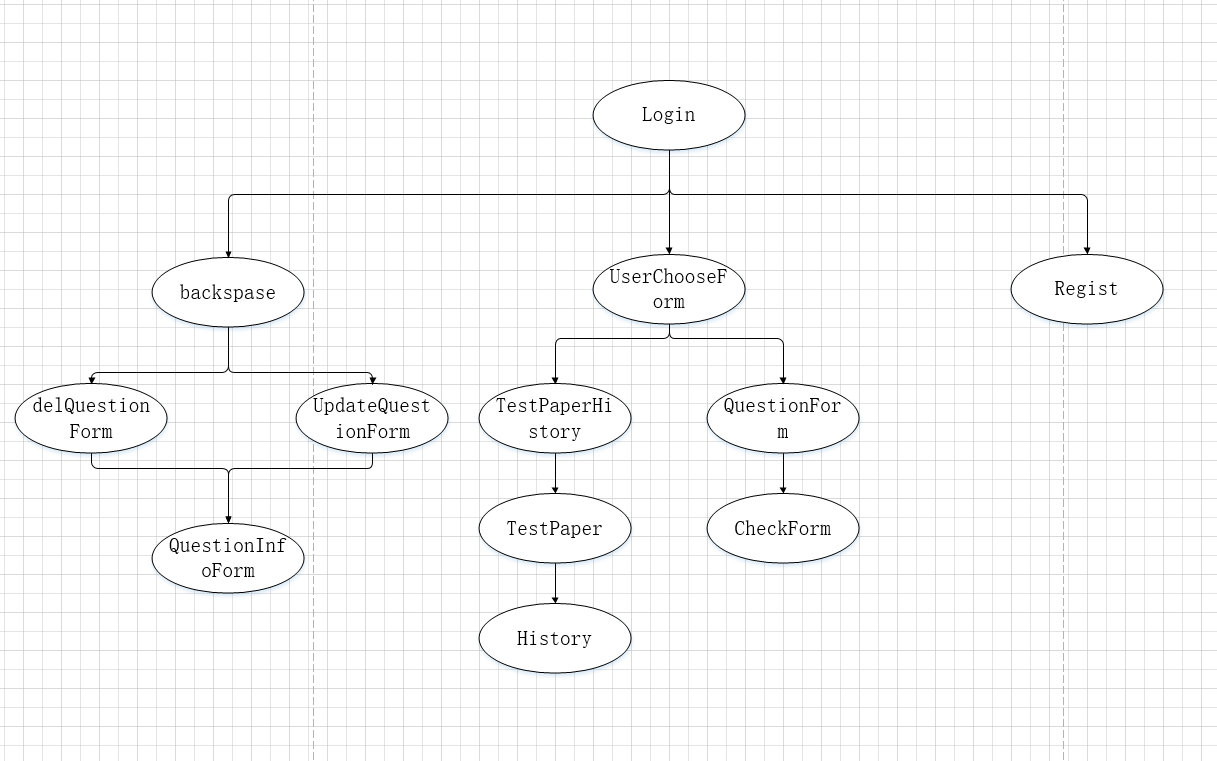
此系统需要运行在.NETFramework4.5.2平台及以上，使用mysql数据库。

# 2系统设计

## 2.1流程图



## 2.2框架结构



一共有13个Form，主窗体是LoginForm用于用户登录，学生登录成功，会进入UserChooseForm，管理员登录成功会进入backspaseForm，还可以点击注册进入RegistFrom

进入UserChooseForm后会有两个选择，1做题，2查看记录，选择了做题会进入，QuestionForm

选择了查看历史记录会进入TestPaperHistoryForm。QuestionFome可以调出CheckForm用于检查试卷。TestPaperHistoryForm会列出所有做过的试卷TestPapser，点击试卷会进入HistoryForm。

backspaceForm中可以对数据库进行增删改操作。

## 2.3数据库

user 用户表

列名 属性 描述

username varchar(50) primarykey notnull

password varchar(50) not null

sex tinyint 1 男 2女

class varchar(50) 班级

question varchar(50) 密保问题

answer varchar(50) 答案

admin 管理员表

列名 属性 描述

adminname varchar(50) primarykey not null

password varchar(50) not null

question 题目表

列名 属性 描述

questionid int 自增 主键 非空

q\_name varchar(1000) not null

c\_a varchar(50) notnull 选项A

c\_b varchar(50) notnull 选项B

c\_c varchar(50) notnull 选项C

c\_d varchar(50) notnull 选项D

hardlevel int 1 2 3 难度级别

score int not null 分数

answer varchar(2) notnull 答案

history 历史记录表

列名 属性 描述

historyid int 自增 主键 非空

username varchar(50) 外键

questionid int 外键

useranswer varchar(1000) 用户答案

rightorwrong int 1 对 0 错

tpid int 试卷id

score 分数表

列名 属性 描述

scoreid int 自增 主键

username varchar(50) 外键

tpid int 外键

score int 分数

tpcount 试卷数量表

列名 属性 描述

countid int 1

count int 计数器

**数据库sql语句**

CRETAT DATABASE question;

DROP TABLE IF EXISTS `admin`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `admin` (

`adminname` varchar(50) NOT NULL,

`password` varchar(50) NOT NULL,

PRIMARY KEY (`adminname`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Dumping data for table `admin`

--

LOCK TABLES `admin` WRITE;

/\*!40000 ALTER TABLE `admin` DISABLE KEYS \*/;

INSERT INTO `admin` VALUES ('admin','123');

/\*!40000 ALTER TABLE `admin` ENABLE KEYS \*/;

UNLOCK TABLES;

DROP TABLE IF EXISTS `history`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `history` (

`historyid` int(11) NOT NULL AUTO\_INCREMENT,

`username` varchar(50) NOT NULL,

`questionid` int(11) NOT NULL,

`useranswer` varchar(10) NOT NULL,

`rightorwrong` int(11) NOT NULL,

`tpid` int(11) NOT NULL,

PRIMARY KEY (`historyid`),

KEY `username` (`username`),

KEY `questionid` (`questionid`),

CONSTRAINT `history\_ibfk\_1` FOREIGN KEY (`username`) REFERENCES `user` (`username`) ON DELETE CASCADE ON UPDATE CASCADE,

CONSTRAINT `history\_ibfk\_2` FOREIGN KEY (`questionid`) REFERENCES `question` (`questionid`) ON DELETE CASCADE ON UPDATE CASCADE

) ENGINE=InnoDB AUTO\_INCREMENT=21 DEFAULT CHARSET=utf8;

DROP TABLE IF EXISTS `question`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `question` (

`questionid` int(11) NOT NULL AUTO\_INCREMENT,

`q\_name` varchar(1000) NOT NULL,

`c\_a` varchar(50) NOT NULL,

`c\_b` varchar(50) NOT NULL,

`c\_c` varchar(50) NOT NULL,

`c\_d` varchar(50) NOT NULL,

`hardlevel` int(11) NOT NULL,

`score` int(11) NOT NULL,

`answer` varchar(2) NOT NULL,

PRIMARY KEY (`questionid`)

) ENGINE=InnoDB AUTO\_INCREMENT=33 DEFAULT CHARSET=utf8;

DROP TABLE IF EXISTS `score`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `score` (

`scoreid` int(11) NOT NULL AUTO\_INCREMENT,

`username` varchar(50) NOT NULL,

`tpid` int(11) NOT NULL,

`score` int(11) NOT NULL,

PRIMARY KEY (`scoreid`),

KEY `username` (`username`),

CONSTRAINT `score\_ibfk\_1` FOREIGN KEY (`username`) REFERENCES `user` (`username`) ON DELETE CASCADE ON UPDATE CASCADE

) ENGINE=InnoDB AUTO\_INCREMENT=3 DEFAULT CHARSET=utf8;

DROP TABLE IF EXISTS `tpcount`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `tpcount` (

`countid` int(11) NOT NULL,

`count` int(11) NOT NULL,

PRIMARY KEY (`countid`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

LOCK TABLES `tpcount` WRITE;

/\*!40000 ALTER TABLE `tpcount` DISABLE KEYS \*/;

INSERT INTO `tpcount` VALUES (1,0);

/\*!40000 ALTER TABLE `tpcount` ENABLE KEYS \*/;

UNLOCK TABLES;

DROP TABLE IF EXISTS `user`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `user` (

`username` varchar(50) NOT NULL,

`password` varchar(50) NOT NULL,

`sex` int(11) NOT NULL,

`class` varchar(50) NOT NULL,

`question` varchar(50) NOT NULL,

`answer` varchar(50) NOT NULL,

PRIMARY KEY (`username`)

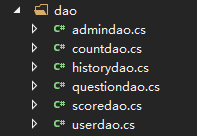
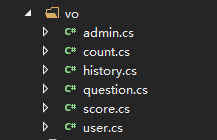
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

# 3系统实现

## 3.1核心代码

数据库后台操作，一共写了12个类来进行对数据的操作

分别对应个个数据库表，vo是数据库表的类，dao是对数据库表的增删改查操作



并且使用了一个MysqlHelper类来封装对数据库的直接操作。

数据库插入：

public bool addUser(user u) {

string sql = "INSERT INTO user(username,password,sex,class,question,answer) VALUES('"+u.username+"','"+u.password+"','"+u.sex+"','"+u.Class+"','"+u.question+"','"+u.answer+"');";

int i = MySqlHelper.ExecuteNonQuery(MySqlHelper.Conn, System.Data.CommandType.Text, sql, null);

if (i > 0) {

return true;

} else{

return false;

}

}

数据库更新：

public bool UpdateQuetion(Question q) {

string sql = "UPDATE question SET q\_name='" + q.score + "',c\_a='" + q.c\_a + "',c\_b='" + q.c\_b + "',c\_c='" + q.c\_c + "',c\_d='" + q.c\_d + "',hardlevel='" + q.hardlevel + "',score='" + q.score + "',answer='" + q.answer + "' WHERE questionid='" + q.questionid + "';";

int i = MySqlHelper.ExecuteNonQuery(MySqlHelper.Conn, CommandType.Text, sql, null);

if (i > 0) {

return true;

}else{

return false;

}

}

数据库删除：

public bool DeleteQuestion(int questionid) {

string sql = "DELETE FROM question WHERE questionid='" + questionid + "';";

int i = MySqlHelper.ExecuteNonQuery(MySqlHelper.Conn, CommandType.Text, sql, null);

if (i > 0) {

return true;

}else{

return false;

}

}

导入试题

public void LoadQuestion(){

this.QuestionText.Text = q[index].q\_name;

this.AText.Text = q[index].c\_a;

this.BText.Text = q[index].c\_b;

this.CText.Text = q[index].c\_c;

this.DText.Text = q[index].c\_d;

this.Level1.BackgroundImage = Properties.Resources.课后作业\_难度1\_\_2\_;

this.Level2.BackgroundImage = Properties.Resources.课后作业\_难度2\_\_2\_;

this.Level3.BackgroundImage = Properties.Resources.课后作业\_难度3\_\_2\_;

this.Level4.BackgroundImage = Properties.Resources.课后作业\_难度4\_\_2\_;

this.Level5.BackgroundImage = Properties.Resources.课后作业\_难度5\_\_2\_;

switch (q[index].hardlevel) {

case 1: this.Level1.BackgroundImage = Properties.Resources.课后作业\_难度1; break;

case 2: this.Level2.BackgroundImage = Properties.Resources.课后作业\_难度2; break;

case 3: this.Level3.BackgroundImage = Properties.Resources.课后作业\_难度3; break;

case 4: this.Level4.BackgroundImage = Properties.Resources.课后作业\_难度4; break;

case 5: this.Level5.BackgroundImage = Properties.Resources.课后作业\_难度5; break;

default: this.Level1.BackgroundImage = Properties.Resources.课后作业\_难度1; break;

}

if (answer[index] != null) {

if (answer[index] == "A") {

ABut.Select();

}

if (answer[index] == "B"){

BBut.Select();

}

if (answer[index] == "C") {

CBut.Select();

}

if (answer[index] == "D"){

DBut.Select();

}}else{

ABut.Checked = false;

BBut.Checked = false;

CBut.Checked = false;

DBut.Checked = false;

}

if (index == 9)

{

this.NextBut2.BackgroundImage = Properties.Resources.icon\_下一题\_\_1\_;

this.NextBut3.ForeColor = Color.FromArgb(138, 138, 138);

this.NextBut1.Cursor = Cursors.Default;

this.NextBut2.Cursor = Cursors.Default;

this.NextBut3.Cursor = Cursors.Default;

choosenextflag = true;

}

else if (index == 0)

{

this.LastBut2.BackgroundImage = Properties.Resources.icon\_上一题\_\_1\_;

this.LastBut3.ForeColor = Color.FromArgb(138, 138, 138);

this.LastBut1.Cursor = Cursors.Default;

this.LastBut2.Cursor = Cursors.Default;

this.LastBut3.Cursor = Cursors.Default;

chooselastflag = true;

}

else

{

if (choosenextflag)

{

this.NextBut2.BackgroundImage = Properties.Resources.icon\_下一题;

this.NextBut3.ForeColor = Color.White;

this.NextBut1.Cursor = Cursors.Hand;

this.NextBut2.Cursor = Cursors.Hand;

this.NextBut3.Cursor = Cursors.Hand;

}

if (chooselastflag)

{

this.LastBut2.BackgroundImage = Properties.Resources.icon\_上一题;

this.LastBut3.ForeColor = Color.White;

this.LastBut1.Cursor = Cursors.Hand;

this.LastBut2.Cursor = Cursors.Hand;

this.LastBut3.Cursor = Cursors.Hand;

}

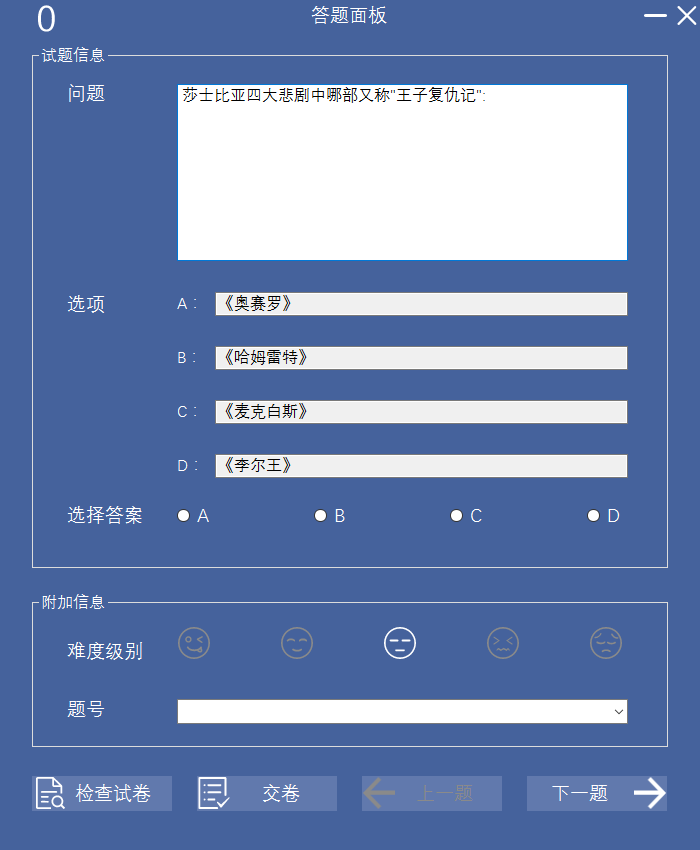
}

changeNum();

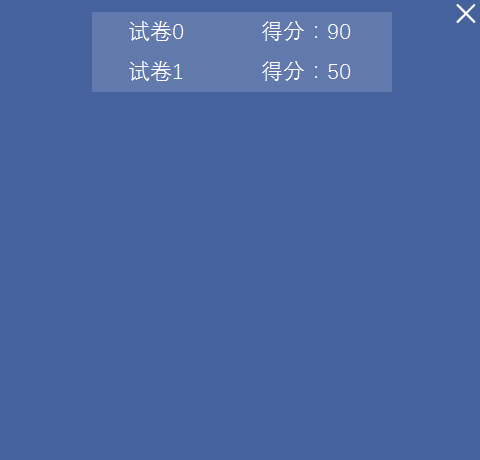
}

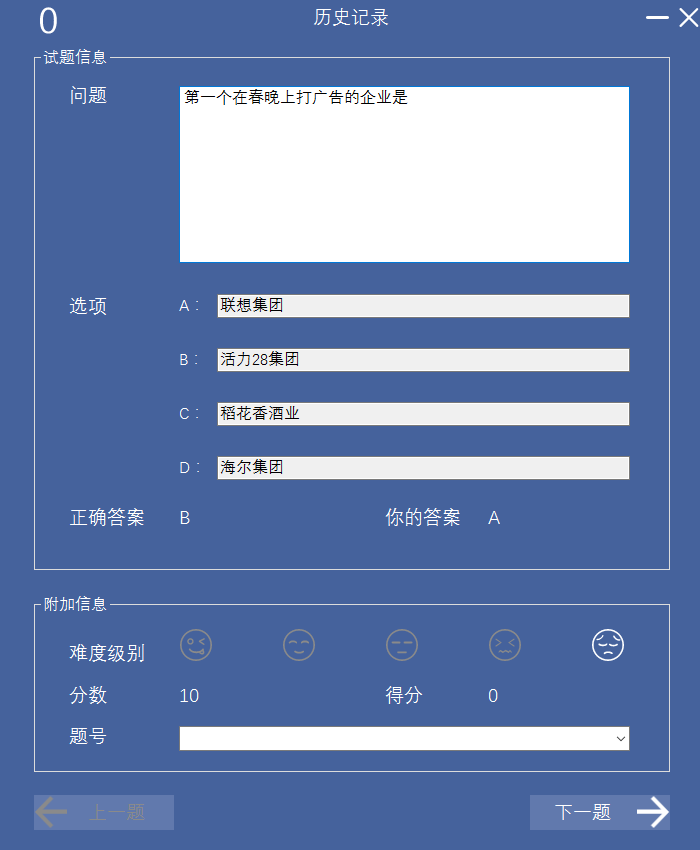
## 3.2程序截图

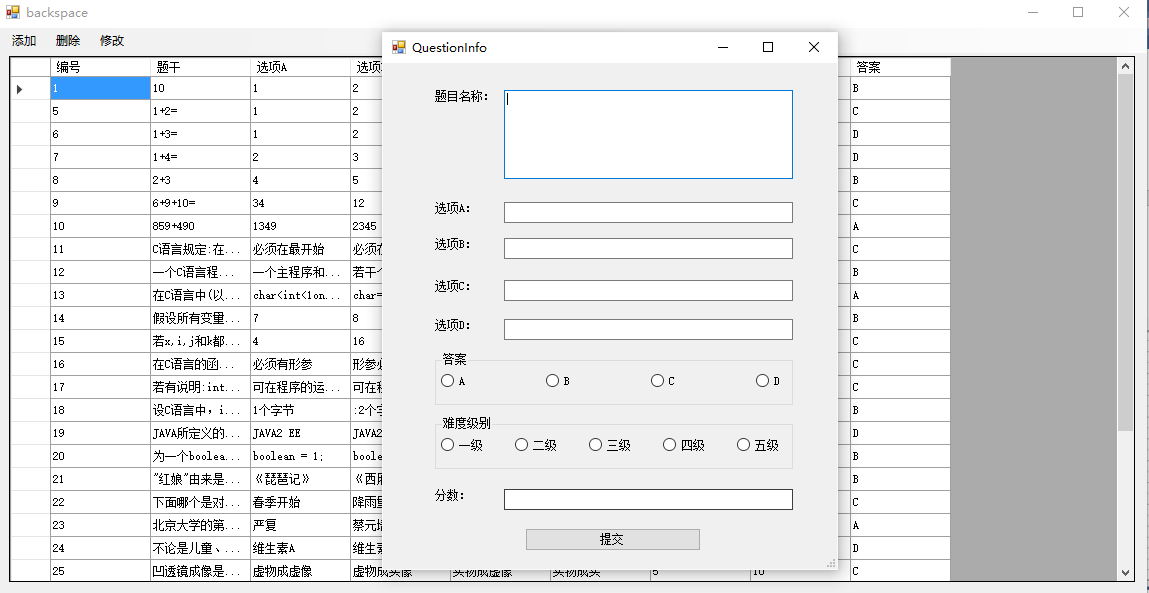














# 4总结

随着现代社会的高速发展，人和人、企业与企业之间的交流逐渐频繁。作为交流的主 要媒介，通讯系统显得尤为重要，所以通讯系统需要不断地完善。C#作为一款简单、快捷的编程软件对承担完善类似通讯系统这样的程序是再合适不过。

通过学习和写程序，我们着重练习使用了windows应用程序的使用方法和常用windows控件的使用。还通过运用类方面的知识，更扎实的掌握了其中的内容。通过连接数据库对以前学习的数据库课程得到了很好的发挥和利用。在制作过程中，我们能够发现问题，并且能够及时向周围的同学和老师请教，弥补不足。

# 参考文献

[1] 郭常圳．《C#网络应用开发例学与实践》．清华大学出版社，2006 11

[2] 杨晓光．《Visual C#.NET程序设计》．北京：清华大学出版社、北京交通大学出版社，200701

[3] 游祖元．《C#案例教程》．北京：电子工业出版社，200701