****

课程设计报告书

**题目：《华工帮》校园助手**

**学 院 计算机科学与工程**

**专 业 网络工程**

**学生姓名 马一宁**

**学生学号 201536551381**

**指导教师 李家春**

**课程编号 145241**

**课程学分 2.0**

**起始日期 2016.4**

|  |  |
| --- | --- |
| 教  师  评  语 | 马一宁同学与组员一起设计实现的校园助手系统除了实现好友聊天、本地助手等功能外，还实现了校园一卡通助手等实用功能，具有较好的实际利用价值。该同学在本次设计中将多线程通信、Mysql数据库、网络编程等知识结合起来，并将服务器部署到了百度云服务器上，使得在不同网段的用户都能与服务器通信，体现了优秀的动手能力，表现出色。  教师签名：  日期： |
| 成  绩  评  定 |  |
| 备  注 |  |

目录

[**一、问题描述** 1](#_Toc453497927)

[**1.1 问题描述** 1](#_Toc453497928)

[**1.2 功能需求** 1](#_Toc453497929)

[**1.3 开发环境** 4](#_Toc453497930)

[**1.4 项目简介** 4](#_Toc453497931)

[**1.5 项目分工** 4](#_Toc453497932)

[**1.6 项目难点** 6](#_Toc453497933)

[**二、系统设计** 6](#_Toc453497934)

[**2.1 系统架构** 6](#_Toc453497935)

[**2.1.1 程序架构** 6](#_Toc453497936)

[**2.1.2 服务器端口分配** 6](#_Toc453497937)

[**2.2 类体系设计** 8](#_Toc453497938)

[**2.2.1 服务器类体系设计** 8](#_Toc453497939)

[**2.2.2 客户端类体系设计** 14](#_Toc453497940)

[**2.2.3 重要头文件** 15](#_Toc453497941)

[**2.3 程序流程图** 20](#_Toc453497942)

[**2.4 数据库设计** 21](#_Toc453497943)

[**2.4.1 数据库表** 21](#_Toc453497944)

[**2.4.2 数据库创建指令** 22](#_Toc453497945)

[**三、程序测试** 24](#_Toc453497946)

[**3.1 服务器开服** 24](#_Toc453497947)

[**3.2 登录界面** 25](#_Toc453497948)

[**3.3 本人完成的功能** 30](#_Toc453497949)

[**3.3.1 好友列表及聊天模块** 31](#_Toc453497950)

[**3.3.2 教师留言模块** 33](#_Toc453497951)

[**3.3.3 修改用户个人信息** 36](#_Toc453497952)

[**3.3.4 计算器功能** 38](#_Toc453497953)

[**3.3.5 教师管理员登陆** 39](#_Toc453497954)

[**四、使用说明** 41](#_Toc453497955)

[**4.1使用流程** 41](#_Toc453497956)

[**4.2使用说明** 41](#_Toc453497957)

[**五、收获体会及建议** 41](#_Toc453497958)

**《华工帮》校园助手**

**一、问题描述**

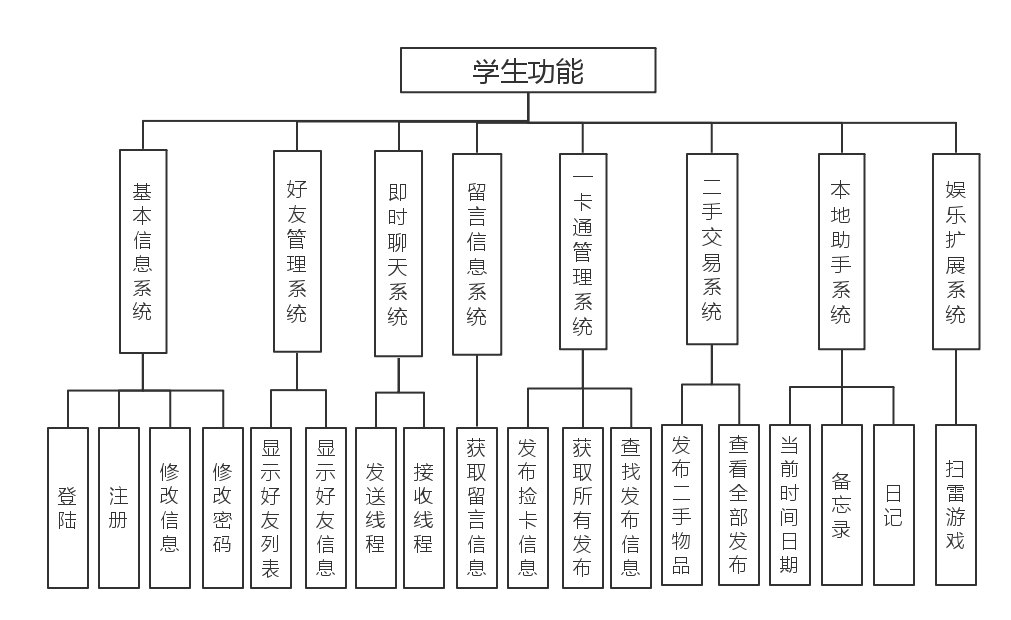
**1.1 问题描述**

时至今日，我们的校园生活已经不在是当年那样的朴素简单，日新月异的科技发展给我们的校园生活增添了许多不一样的精彩。然而，在我们的生活中，许多事物依然不可以得到有效的记录和整理，逐渐演变了许多生活中难以解决的问题。因此，我们将开发一个华工校园助手，为在校的老师、学生们提供不一样的快捷与便利。

本程序使用SQL-Server-Client多线程模式，实现多客户端与服务器通信同时运行。

**1.2 功能需求**

本程序将用户分为管理员（教师）与普通用户（学生）两个角色。

**1.2.1 学生功能**

**图1.1 学生功能展示**

**1.基本信息系统：**

学生提供登陆、注册以及修改个人信息基本功能。

**2.好友管理系统：**

学生之间可以查看好友列表，查看好友信息。

**3.即时聊天系统：**

与好友在线离线即时聊天功能。

**4.留言信息系统：**

学生可以通过服务器获取教师或管理员对其发布的留言。

**5.一卡通管理系统：**

学生可以发布捡到校园一卡通的信息，提供从服务器数据库中访问所有已发布的失卡信息，并可查找自己的失卡信息，展示捡卡人。

**6.二手交易系统：**

学生可以发布自己的二手物品出售信息，提供从服务器数据库中访问所有已发布的二手交易信息，并可查找展示人的个人信息功能。

**7.本地助手系统：**

该系统下含有三个功能，当前时间日历展示功能，备忘录功能，日记功能。当前时间日历展示功能显示当期日期，并实时刷新时间，备忘录功能提供本地保存并自动同步数据，可以新建备忘录获取历史备忘录。日记功能可以输出或读取日记内容。

**8.娱乐扩展系统：**

含有界面的新款扫雷游戏。

**1.2.2 教师功能**

**图1.2 教师功能展示**

**1.教师登陆系统：**

登录功能。

**2.学生信息管理系统：**

教师可以管理学生信息，修改学生的注册信息（学号除外），删除学生。将学生信息展示排序。

**3.留言发布系统：**

教师可以向学生留言，留言将在学生的留言系统中展示。

**1.3 开发环境**

系统环境：Windows10 64位专业版

开发软件：Visual Studio 2010旗舰版

数据库：MySQL 5.7

**1.4 项目简介**

项目名称：华工帮 校园助手

项目类型：Win32窗体应用程序

项目语言： C++ CLR/.NET4.0

项目负责人：马一宁

项目成员：黄家锐、胡雨薇

项目完成时间：2016年6月6日星期一

**1.5 项目分工**

**马一宁：**

完成服务器重要头文件类体系设计

完成服务器端的架构

完成服务器与客户端的网络编程套接字通信

完成数据库的设计和功能使用

完成多线程算法设计

完成所有与数据库的同步功能算法

完成好友列表ListView展示聊天框跳转功能

完成即时聊天功能

完成教师的学生信息管理功能

完成教师管理员的留言功能

使用完成的头文件编写具体功能，实现功能函数

**黄家锐**

完成项目总框架流程设计

完成客户端的架构

完成客户端重要头文件类体系设计

完成个人信息注册修改管理系统

完成一卡通功能并使用头文件实现与服务器通信

使用完成的头文件编写具体功能，完成二手交易功能

完成学生教师登录功能

完成娱乐扩展扫雷功能

**胡雨薇**

设计所有界面

美化界面设计

设计界面动画算法，界面移动最小化关闭

完成本地助手时间显示功能

完成本地助手备忘录功能

完成本地助手日记功能

**1.6 项目难点**

本项目难点为: 数据库正确使用，多线程进程遗留与互斥锁，网络编程，服务器与客户端之间指令的编码与解码，套接字通讯中TCP协议可能出现的BUG修复，服务器心跳检测，服务器循环接收客户端请求，服务器遇到错误链接释放请求，服务器与客户端类体系设计，各功能的结合bug修复，界面设计中控件的正确使用，局部变量全局变量以及命名空间的正确使用，多人framework代码结合问题。

**.**

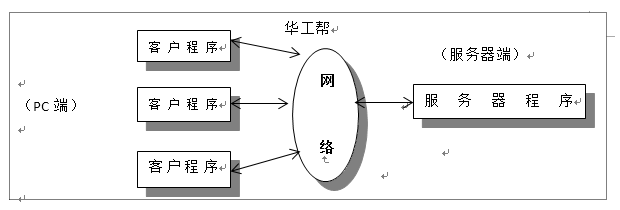
**二、系统设计**

**2.1 系统架构**

**2.1.1 程序架构**

本程序分为三个模块，客户端、服务器与数据库。

客户端可以在网络中向服务器不同端口发布不同命令指令以实现各个功能，服务器处理相关指令后与数据库建立通信，通过网络返回结果给客户端。

具体模式如下图，一台服务器可以连接多个客户端：

**图2.1 程序架构**

**2.1.2 服务器端口分配**

服务器共有8个端口设置，端口从4869至4876，分别对应不同功能：

#define Loginportnum "4869"

#define Registerportnum "4870"

#define GetMessageportnum "4871"

#define SendMessageportnum "4872"

#define ChangePeopleportnum "4873"

#define Cardportnum "4874"

#define Tradeportnum "4875"

#define Adminportnum "4876"



**图2.2 服务器端口设置**

**1. 端口4869 学生登陆服务**

从客户端接收编码后的登陆指令，检查数据库账号密码，返回登陆是否成功。

**2. 端口4870 学生注册服务**

从客户端接收编码后的用户信息指令解码，存入数据库，返回注册是否成功。

**3. 端口4871 聊天信息获取服务**

从客户端接收编码后的用户聊天信息指令解码，存入数据库，返回聊天信息写入是否成功。

**4. 端口4872 聊天信息发送服务**

从客户端接收用户学号，查找数据库，若存在未收取到的消息，向客户端发送该条消息。

**5. 端口4873 学生信息查询修改查询服务**

从客户端接收指令，指令分为两种：接收学号返回编码后的个人信息，接收编码后的个人信息返回是否修改成功。

**6. 端口4874 一卡通丢失查询添加服务**

从客户端接收指令，指令分为三种：收到“A”返回所有一卡通的编码信息，收到“G”后接学号向数据库中更新此人卡片已经找到，收到一条编码信息则解码后写入数据库新的失卡记录。

**7. 端口4875 二手交易查询添加服务**

从客户端接收指令，指令分为两种：收到“A”返回所有二手信息的编码信息，收到一条编码信息则解码后写入数据库新的二手物品发布记录。

**8. 端口4876 教师登陆管理服务**

从客户端接收指令，指令分为四种：收到“A”返回所有学生的编码信息，收到“L”后接用户名密码的编码从数据库查找确认返回登陆结果，收到“D”后接学号从数据库中删除对应学生，返回结果。收到编码信息，解码后在数据库中更新学生信息。

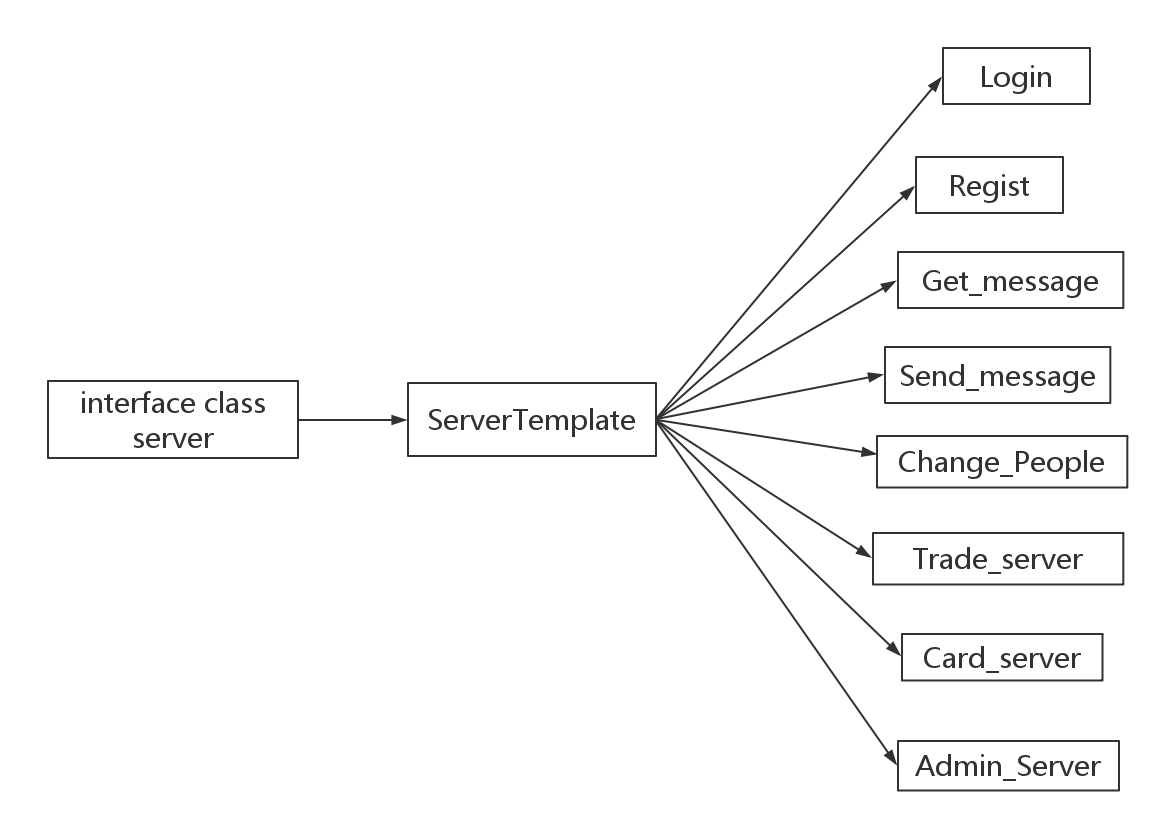
**2.2 类体系设计**

本程序中分为**服务器类体系**设计，**客户端类体系**设计和**重要通用头文件**设计三部分。

**2.2.1 服务器类体系设计**

**2.2.1.1服务器类三层类体系**

服务器类体系如下图：



**图2.3 服务器类体系设计**

**2.2.1.2服务器类说明**

**第一层 接口**

服务器接口类server包含在Server命名空间下，包含抽象出的各个派生类共有的方法，派生来重写这些虚函数:

void run();

该函数为服务开始运行的函数，判断服务是否正在运行，若没有运行启动服务。设置端口和IP信息，启动TCP服务器端等待Socket连接，开启一个新的服务线程进入阻塞状态等待连接。

void stop();

使用Abort()方法终止线程，shutdown()关闭服务，如果服务器还在阻塞状态，则发一个模拟数据包结束阻塞状态后安全关闭。

void heart\_check();

为避免错误连接或恶意攻击，提高服务器可靠性，使用一个心跳检测功能函数，检测当前连接是否依然有效，如果没有2秒内没有回复，则关闭当前连接，服务器可以接受下一个连接请求。

void doit();

这个函数依然为虚函数，为真正各个派生类的服务器功能函数，该函数在建立连接以后由run()函数启动一个新线程执行。

**第二层 服务模板**

ServerTemplate实现接口中的公有功能，并在构造函数中初始化相关变量，在析构函数中安全有效的关闭当前连接。

类中成员变量有：

String^portnow;//存储当前端口号，在预编译头文件中设置端口号。

Socket^Socket\_now;//用于正在处理的请求

Thread^myThread;//用于循环执行的线程

Thread^heartcheck\_Thread;//用于心跳检测线程

TcpListener^TCP\_server;//用于启动监听

Boolean isstart;//修改bug多次按键

Boolean con;//修改bug系统残留进程

int total\_connects;//总共连接数

int check\_num;//连接重复次数

int curnum;//当前连接数

**第三层 服务类**

**1.Login类**

继承以上服务模板，该类功能为学生登录功能。

**virtual void run()override；**

该函数使用AcceptSocket()方法循环接收新的请求，客户端将用户名和学号以“$”+用户名+“$”+ 密码+“$”的方式使用自定义的Scoket类的方法发向服务器，等待服务器调用自定义People.h头文件中的方法验证登陆密码，返回结果，继续下一次AcceptSocket()。

**2.Regist**

继承以上服务模板，该类功能为学生注册功能。

**virtual void run()override；**

该函数使用AcceptSocket()方法循环接收新的请求，客户端将编码好的用户信息以 “$”+用户名+“$”+ 密码+“$”+“”+“”+…的方式使用自定义的Scoket类的方法发向服务器，等待服务器调用自定义People.h头文件中的方法解码后调用自定义People.h头文件中的regist（）方法向数据库中注册新信息，返回结果，继续下一次AcceptSocket()。

**3.Get\_message**

继承以上服务模板，该类功能为新信息获取功能。

**virtual void run()override；**

该函数使用AcceptSocket()方法循环接收新的请求，客户端将聊天信息使用自定义头文件Message.h中的编码方法编码后使用自定义的Scoket类的方法发向服务器，等待服务器调用解码方法解码后调用自定义头文件Message.h中write（）方法向数据库中写入新记录，返回结果，继续下一次AcceptSocket()。

**4.Send\_message**

继承以上服务模板，该类功能为新信息发送功能。

**virtual void run()override；**

该函数使用AcceptSocket()方法循环接收新的请求，客户端将学号使用自定义的Scoket类的方法发向服务器，等待服务器调用头文件Message.h中的read（）方法读取数据库，若存在新消息，编码后调返回结果，在客户端解码后实现功能，继续下一次AcceptSocket()。

**5.Change\_People**

继承以上服务模板，该类功能为学生信息查询修改查询。

**virtual void run()override；**

该函数使用AcceptSocket()方法循环接收新的请求，从客户端接收指令，指令分为两种：客户端将学号使用自定义的Scoket类的方法发向服务器，等待服务器调用头文件People.h中的read（）方法读取数据库编码后返回客户端。客户端将编码后的个人信息发送给服务器，服务器用People.h中与数据库同步的方法修改数据库，返回结果。然后继续下一次AcceptSocket()。

**6.Trade\_server**

继承以上服务模板，该类功能为二手交易查询添加。

**virtual void run()override；**

该函数使用AcceptSocket()方法循环接收新的请求，从客户端接收指令，指令分为两种：收到“A”通过调用Trade类中的readall()方法将所有二手信息的编码信息，收到一条编码信息则通过Trade类中的decode()解码后写入数据库新的二手物品发布记录。然后继续下一次AcceptSocket()。

**7.Card\_server**

继承以上服务模板，该类功能为一卡通查询添加功能。

**virtual void run()override；**

该函数使用AcceptSocket()方法循环接收新的请求，从客户端接收指令，指令分为三种：收到“A”返回调用Card.h中的readall()方法返回所有一卡通的编码信息，由客户端解码后逐条展示，收到“G”后接学号调用Card.h中的getcard()f方法向数据库中更新此人卡片已经找到，返回操作结果，收到一条客户端的card编码信息则调用Card.h中的decode（）解码后调用write（）写入数据库新的失卡记录。然后继续下一次AcceptSocket()。

**8. Admin\_Server**

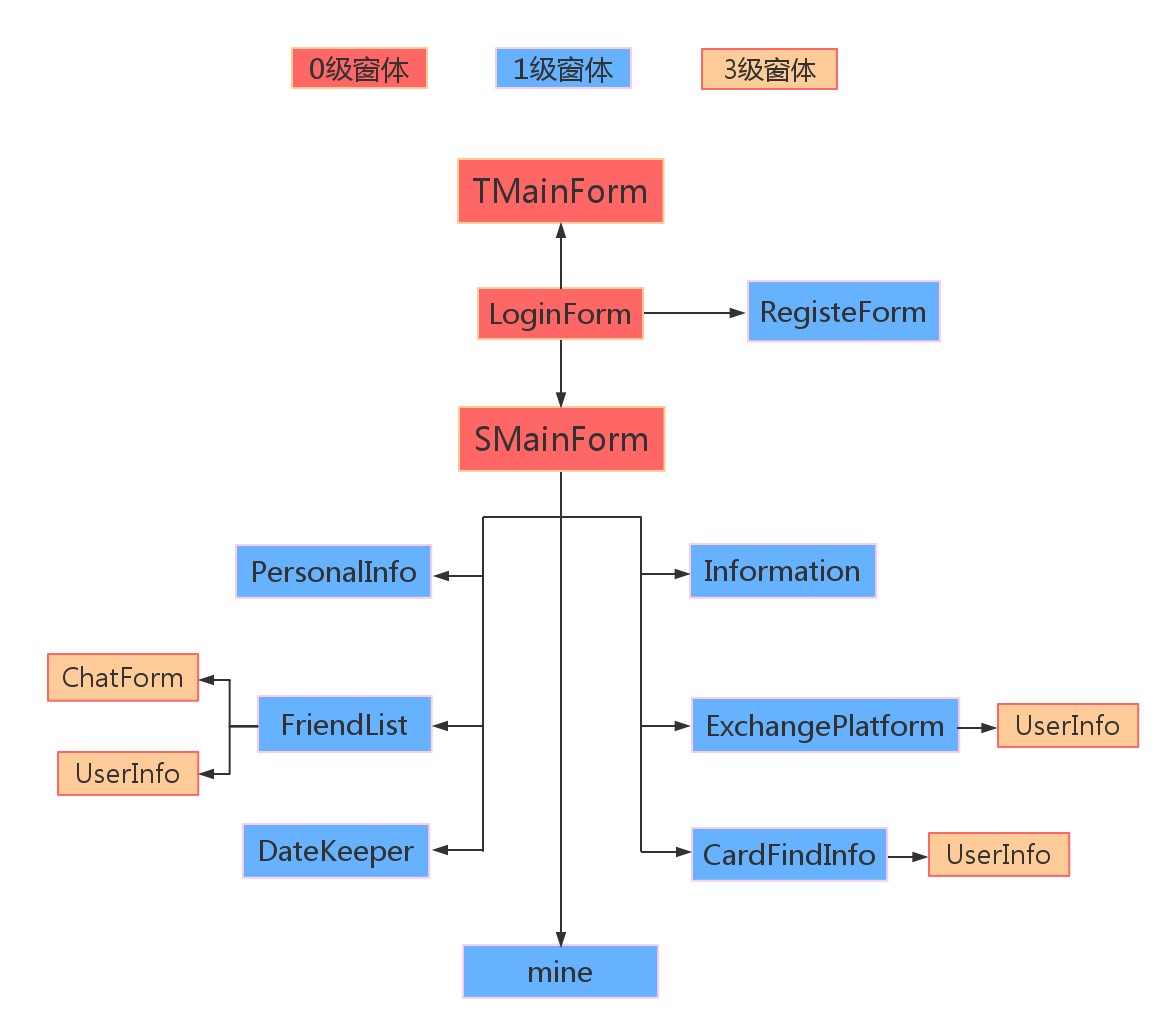
继承以上服务模板，该类功能为登陆管理功能。

**virtual void run()override；**

该函数使用AcceptSocket()方法循环接收新的请求，从客户端接收指令，指令分为四种：收到“A”调用People.h中\_teacher\_do（）类中的方法返回所有学生的编码信息，收到“L”后接用户名密码的编码就调用People.h中\_teacher\_do（）类中的\_search（）方法从数据库查找确认返回登陆结果，收到“D”后接学号则调用People.h中\_teacher\_do（）类中写好的方法从数据库中删除对应学生，返回结果。若收到一条\_teacher\_do（）类的编码信息，则decode()解码后在数据库中更新学生信息，返回结果。然后继续下一次AcceptSocket()。

**2.2.2 客户端类体系设计**

**2.2.2.1客户端Form结构图调用关系如下：**



**图2.4 客户端类体系**

**2.2.2.2客户端类体系说明**

窗体调用描述：

0级窗体相互转跳，不同时存在；

1级窗体为0级窗体的子窗体；

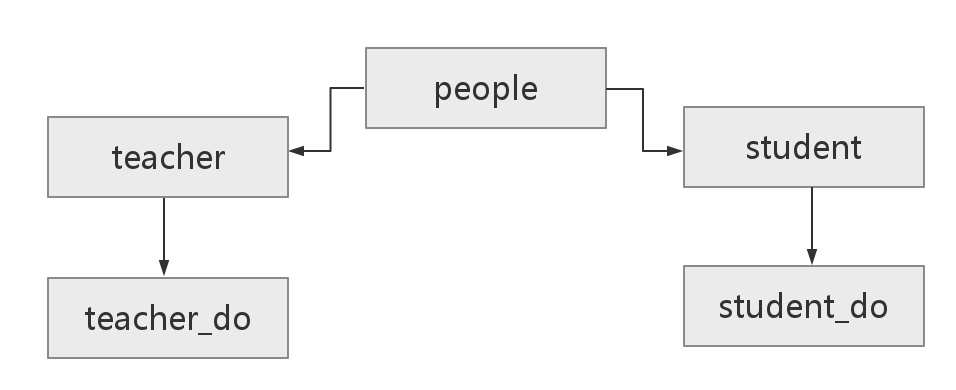
2级窗体为1级窗体对应的子窗体；

同级窗体相互独立，不相互影响。

**2.2.3 重要头文件**

**2.2.3.1 重要头文件类体系**

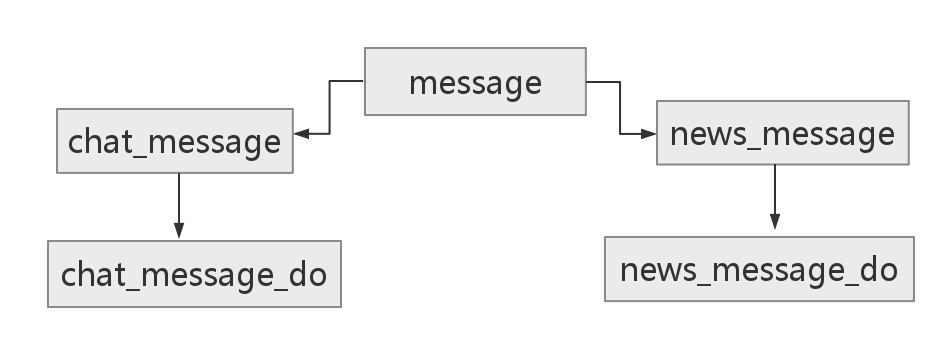
**Ⅰ 用户基本信息功能类**



**图2.5 - people类体系关系图**

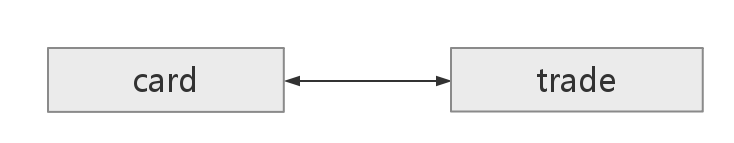
由基类people派生teacher以及student两个类，再分别以teacher和student为基类派生teacher\_do和student\_do两个类；

**Ⅱ 通讯信息功能类**



**图2.6 - message类体系关系图**

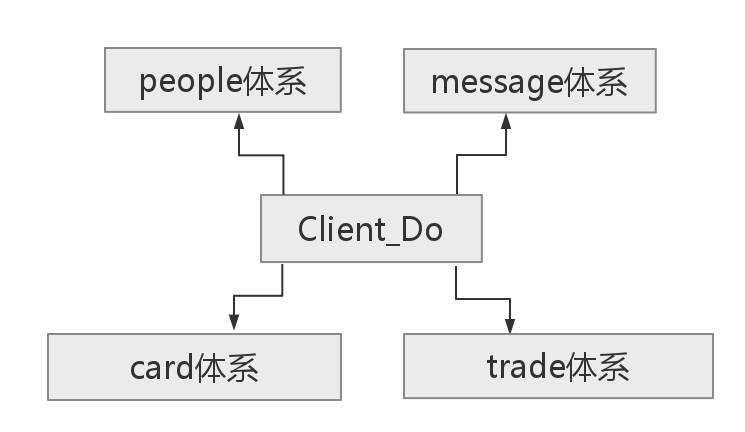
由基类message派生chat\_message以及news\_message两个类，再分别以chat\_message和news\_message为基类派生chat\_message\_do和news\_message\_do两个类；

**Ⅲ 一卡通信息类以及二手交易信息功能类**

**图2.7 –一卡通信息类以及二手交易信息功能类关系图**

card和trade两个功能函数处所封装数据的内容不同外，功能基本一致，两个类为并列关系；

**Ⅳ Socket套接字功能类**

**图2.8 - Socket套接字功能类体系图**

所有功能体系都须调用Socket套接字体系的Client\_Do功能类；

**2.2.3.2 重要头文件类说明**

**a) people类说明**

表1 - people类成员列表

|  |  |  |
| --- | --- | --- |
|  | 成员名称 | 功能描述 |
| 共有数据成员  （String类型） | 学号（SchoolId） | 存放基本信息 |
| 姓名（Name） | 存放基本信息 |
| 登录密码（Password） | 存放基本信息 |
| 性别（Sex） | 存放基本信息 |
| 生日（BirthDate） | 存放基本信息 |
| 电话号码（Phonenum） | 存放基本信息 |
| QQ号码（QQnum） | 存放基本信息 |
| 电子邮件（Email） | 存放基本信息 |
| 个性签名（Signature） | 存放基本信息 |
| 照片地址（photo） | 存放基本信息 |
| 注册时间（RegistDate） | 存放基本信息 |
| student类特有数据成员  （String类型） | 学院（Academy） | 存放特有信息 |
| 专业（Major） | 存放特有信息 |
| 班级（Class） | 存放特有信息 |
| 年级（Grade） | 存放特有信息 |
| teacher类特有数据成员  （String类型） | 用户名称（admin\_name） | 存放管理员信息 |
| 管理员密码（admin\_password） | 存放管理员信息 |
| 共有成员函数 | Set 函数集合 | 将传入的参数赋值给对应变量 |
| Get函数集合 | 返回对象中的对应变量 |
| String code\_ID(); | 将对象中的ID编码为可发送给服务器的指令 |
| void decode\_ID(); | 将接收到的已编码ID指令转为正常账号形式 |
| bool checklogin(); | 检查访问服务器后收到的回复，并给出登录是否成功的结果 |
| student\_do类特有成员函数 | Boolean regist(); | 将账号和密码编码为向服务器发送的指令，请求注册新用户 |
| String code(); | 将用户信息编码成指定格式发送用于发送给服务器 |
| String decode(); | 将从服务器接收到的指令解码存于当前对象中 |
| teacher\_do类特有成员函数 | String^ SearchPeople(String ^ID); | 向服务器发送查找指定ID的学生信息的指令 |
| String^ changePeople(String ^ID); | 向服务器发送修改指定ID的学生信息的指令 |
| String^ deletePeople(String ^ID)’ | 向服务器发送删除指定ID的学生信息的指令 |

附：表中括号内为变量名

**b) message类说明**

表2 - message类成员列表

|  |  |  |
| --- | --- | --- |
|  | 成员名称 | 功能描述 |
| 共有数据成员  （String类型） | 消息时间（time） | 记录该条消息的发送时间 |
| 内容（sendmessage） | 存放消息的内容 |
| 类型（kind） | 本条消息的类型 |
| 发送方（from） | 存放消息的发送方ID信息 |
| 接收方（to） | 存放消息的接收方ID信息 |
| 共有成员函数 | String code(); | 将该条信息编码成指定格式发送用于发送给服务器 |
| void decode(); | 将从服务器接收到的信息指令后解码存于当前对象中 |
| void setmessage(); | 设置该条信息的基本信息 |
| Boolean write(); | 将接收到的消息对象信息写入数据库，成功返回1，否则返回0 |
| String read(); | 读取数据库中的消息，有未读消息返回消息，如失败返回”N” |
| news\_do，  chat\_do  类特有性质 | 不同类型消息虽在数据库中以相同的形式存放，但编码时按照不同的信息对应编码 | |

附：表中括号内为变量名

**c) card类和trade类说明**

表3 – card和trade类成员列表

|  |  |  |
| --- | --- | --- |
|  | 成员名称 | 功能描述 |
| 共有数据成员 | 物主对象（student\_do） | 存放卡主完整信息的用户对象 |
| 物主学号（String） | 存放丢卡人的学号 |
| 发布时间（String） | 存放发布时间 |
| card特有数据成员 | 拾卡人学号（String） | 存放拾卡人的学号 |
| 拾卡对象（student\_do） | 存放拾卡人完整信息的用户对象 |
| 共有成员函数 | bool getpeople(); | 用学号向服务器请求两个用户对象的所有信息，收到则存于card对象中并返回1，否则返回0 |
| void setmsg(String,String); | 设置丢卡人和拾卡人的学号 |
| void setcurdate(); | 记录当前时间 |
| Boolean write(); | 将接收到的消息对象信息写入数据库，成功返回1，否则返回0 |
| String read(); | 读取数据库中的消息，有未读消息返回消息，如失败返回”N” |
| news\_do，  chat\_do  类特有性质 | 不同类型消息虽在数据库中以相同的形式存放，但编码时按照不同的信息对应编码 | |

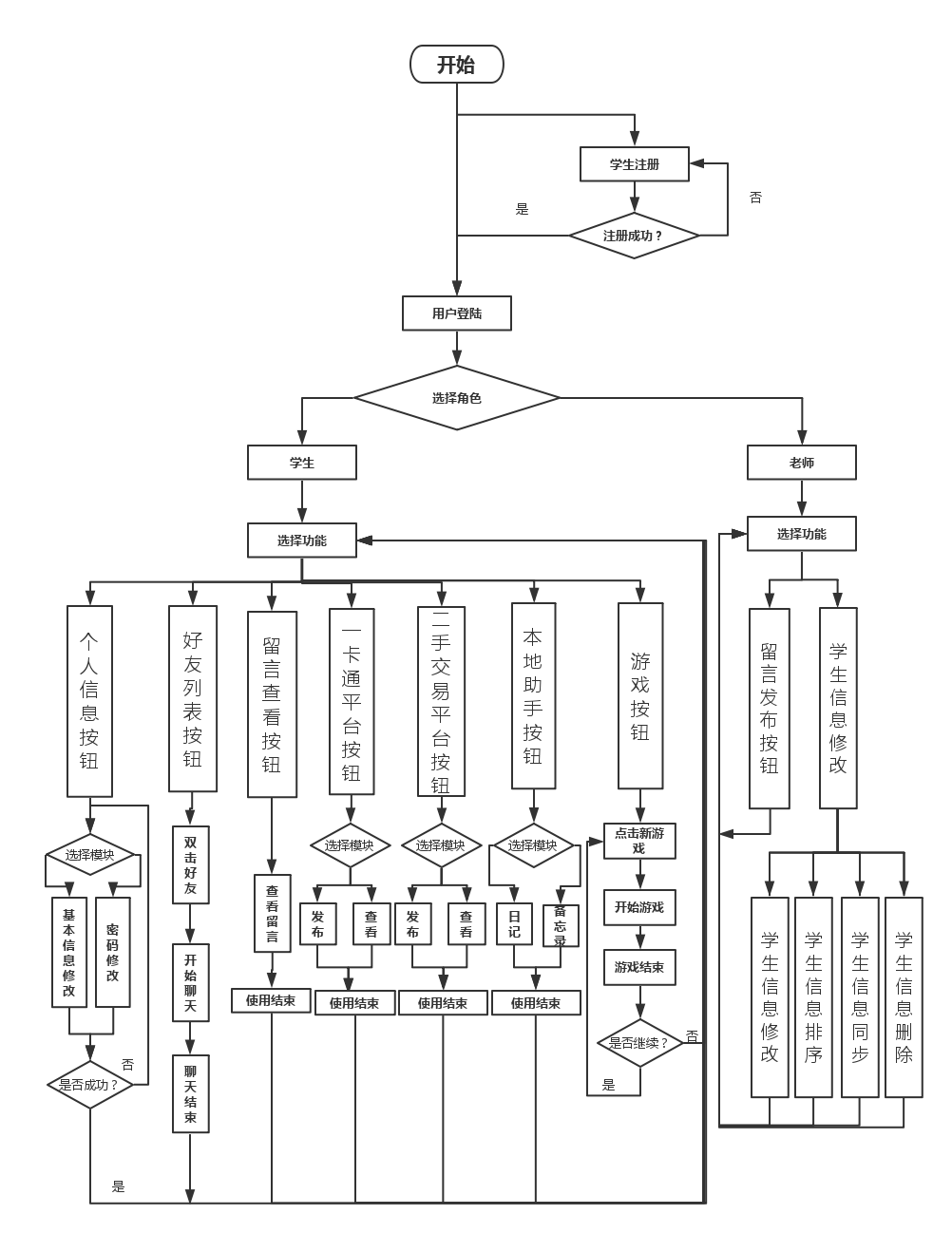
附：表中括号内为对应的数据类型

**d) Socket类说明**

表4 –Socket类成员列表

|  |  |  |
| --- | --- | --- |
|  | 成员名称 | 功能描述 |
| 数据成员 | Socket^socketAtClient | 用于当前连接的Socket |
| IPEndPoint^remoteEP | 用于当前连接的TCP地址对象 |
| 成员函数 | void op\_Setport(Int32 portnum) | 设置当前连接的端口 |
| Boolean op\_Connect\_Server() | 连接端口返回是否成功 |
| Boolean op\_Send(String^msg) | 用于向连接发送信息 |
| String^ op\_Receive() | 用于从连接接受信息 |
| Boolean op\_Shutdown\_Server() | 用于关闭当前连接 |

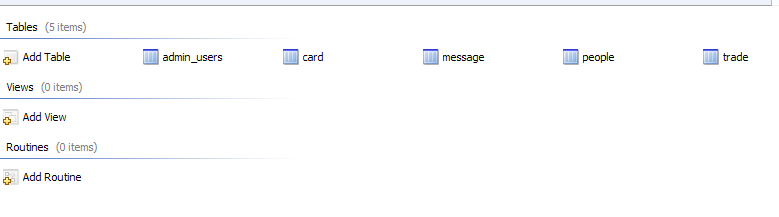
**2.3 程序流程图**

****

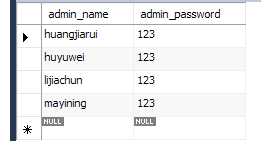
**图2.9 – 程序流程图**

**2.4 数据库设计**

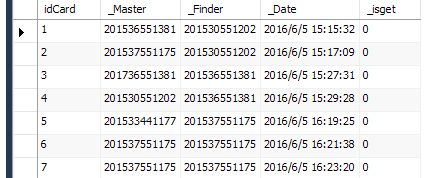
**2.4.1 数据库表**

本程序共使用5个数据表。

**图2.10 – 数据库表**

**1. admin\_users 管理员数据**

**图2.11 – 管理员数据表**

**2. card 一卡通数据**

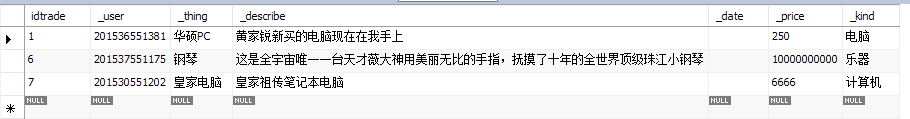
**图2.12 – 一卡通数据表**

**3. message 聊天数据**

**图2.13 – 聊天数据表**

**4. people 用户数据**

**图2.14 – 用户数据表**

**5. trade 二手交易数据**

**图2.15 – 二手交易数据表**

**2.4.2 数据库创建指令**

DROP Database IF EXISTS `scut\_helper`;

create database scut\_helper;

USE scut\_helper;

DROP TABLE IF EXISTS `people`;

CREATE TABLE `people` (

`\_userId` int(11) NOT NULL,

`\_SchoolId` varchar(45) NOT NULL,

`\_Name` varchar(45) NOT NULL,

`\_Password` varchar(45) NOT NULL,

`\_Sex` varchar(45) NOT NULL,

`\_BirthDate` varchar(45) NOT NULL,

`\_Phonenum` varchar(45) DEFAULT NULL,

`\_QQnum` varchar(45) DEFAULT NULL,

`\_Signature` varchar(45) DEFAULT NULL,

`\_Academy` varchar(45) DEFAULT NULL,

`\_Major` varchar(45) DEFAULT NULL,

`\_Class` varchar(45) DEFAULT NULL,

`\_Grade` varchar(45) DEFAULT NULL,

`\_RegistDate` varchar(45) NOT NULL,

`\_Status` varchar(45) DEFAULT NULL,

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

`\_Email` varchar(45) DEFAULT NULL,

`\_photo` varchar(45) DEFAULT NULL,

PRIMARY KEY (`id`,`\_userId`,`\_SchoolId`),

UNIQUE KEY `\_SchoolId\_UNIQUE` (`\_SchoolId`),

UNIQUE KEY `\_userId\_UNIQUE` (`\_userId`)

) ENGINE=InnoDB AUTO\_INCREMENT=40 DEFAULT CHARSET=utf8;

DROP TABLE IF EXISTS `admin\_users`;

CREATE TABLE `admin\_users` (

`admin\_name` varchar(12) NOT NULL DEFAULT 'admin',

`admin\_password` varchar(20) DEFAULT '123456',

PRIMARY KEY (`admin\_name`),

UNIQUE KEY `admin\_name\_UNIQUE` (`admin\_name`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

DROP TABLE IF EXISTS `card`;

CREATE TABLE `card` (

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

`\_Master` varchar(45) NOT NULL,

`\_Finder` varchar(45) NOT NULL,

`\_Date` varchar(45) NOT NULL,

`\_isget` varchar(45) NOT NULL DEFAULT '0',

PRIMARY KEY (`idCard`)

) ENGINE=InnoDB AUTO\_INCREMENT=31 DEFAULT CHARSET=utf8;

DROP TABLE IF EXISTS `message`;

CREATE TABLE `message` (

`\_Time` varchar(45) NOT NULL,

`\_From` varchar(45) NOT NULL,

`\_To` varchar(45) NOT NULL,

`\_Message` varchar(45) NOT NULL,

`\_isreceived` varchar(45) NOT NULL,

`\_kind` varchar(45) NOT NULL,

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

PRIMARY KEY (`id`)

) ENGINE=InnoDB AUTO\_INCREMENT=130 DEFAULT CHARSET=utf8;

DROP TABLE IF EXISTS `trade`;

CREATE TABLE `trade` (

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

`\_user` varchar(45) DEFAULT NULL,

`\_thing` varchar(45) DEFAULT NULL,

`\_describe` varchar(45) DEFAULT NULL,

`\_date` varchar(45) DEFAULT NULL,

`\_price` varchar(45) DEFAULT NULL,

`\_kind` varchar(45) DEFAULT NULL,

PRIMARY KEY (`idtrade`)

) ENGINE=InnoDB AUTO\_INCREMENT=8 DEFAULT CHARSET=utf8;

**三、程序测试**

**3.1 服务器开服**

服务器端打开本软件，然后点击左上角一键开启即可完成开服****

**3.2 登录界面**

1. 已经注册过的用户。

（1）正常情况：

登录界面要求用户选择输入账号与密码，选择学生或者教师这两个登录身份。输入正确的账户密码后，系统显示登陆成功。





1. 异常情况：

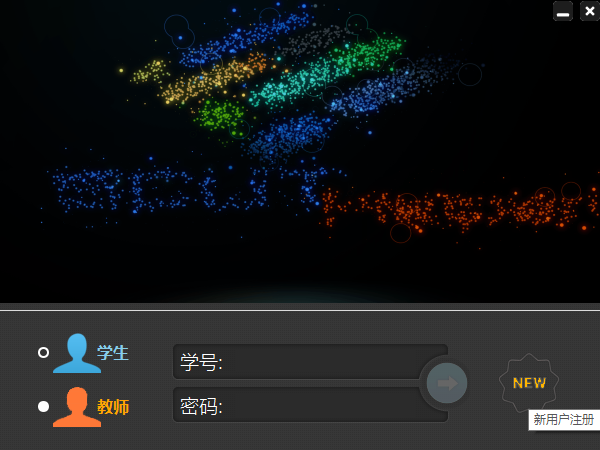
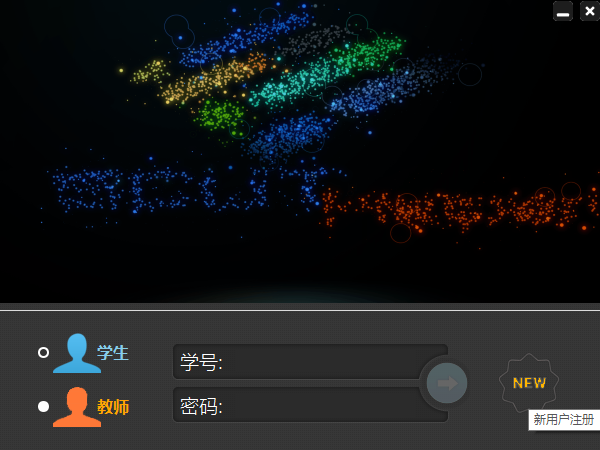
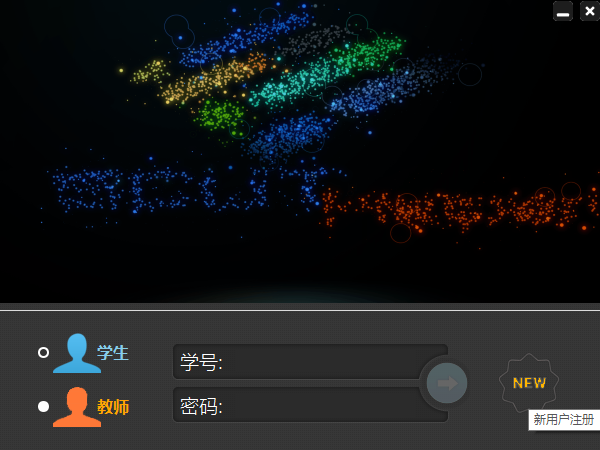
若用户不存在、密码错误或网络连接有问题，系统将提示。





1. 新用户

单击右下角的“new”按钮，进入注册的界面。



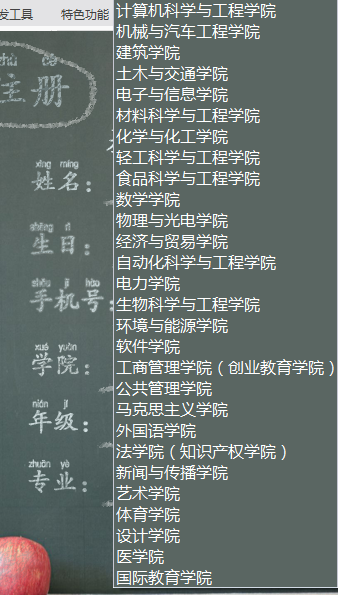
1. 注册界面展示：

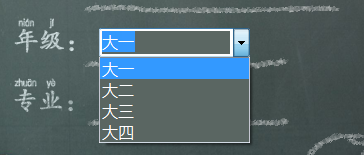
a.填写基本资料

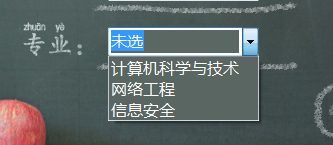


b.选择生日、学院、年级、专业。







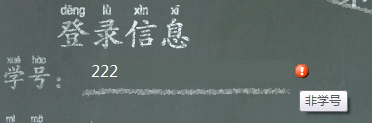


C.用户完成必填信息后，系统提示注册成功

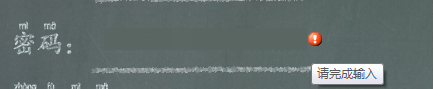


1. 若用户没有按要求输入注册信息的时候，系统将提示，如：

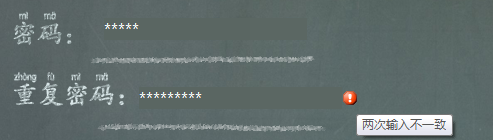
a.不输入学号

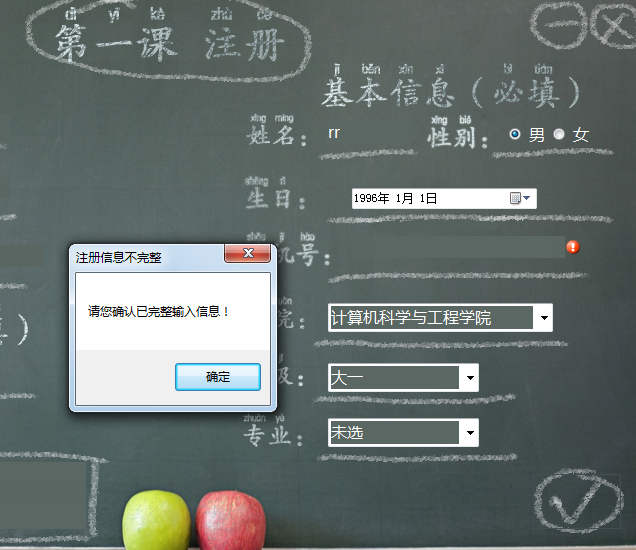


b.不输入密码



c.两次输入的密码不一样。



d.没有输入完整信息就完成注册

**3.3 本人完成的功能**

这就是华工帮软件的主界面。分为六个主要大模块（界面中间的六个图标）和四个小功能（右上人头图标和右下计算器图标，左下音乐和笑脸图标）。



六个模块分别为

1. 好友列表模块
2. 好友聊天模块
3. 重要日程模块
4. 校园一卡通模块
5. 二手交易模块
6. 小游戏模块

三个小功能为

1. 修改个人信息
2. 播放音乐
3. 使用计算器

接下来我们详细介绍这九个功能。

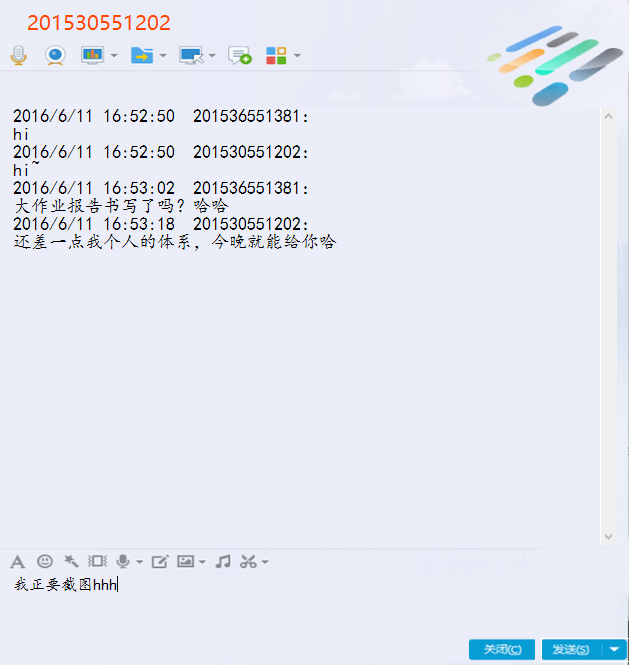
**3.3.1 好友列表及聊天模块**



（1）双击好友，可弹出聊天框。当双方都在线时可以进行聊天



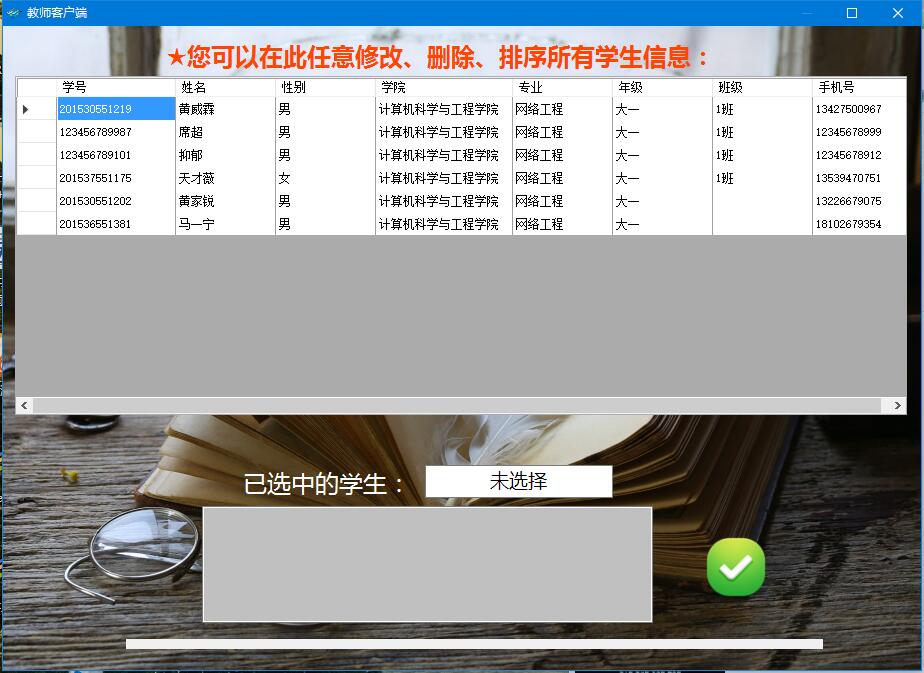
（2）对方可以回复你的消息



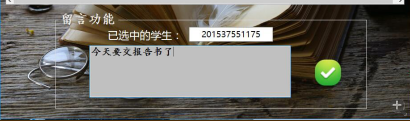
**3.3.2 教师留言模块**



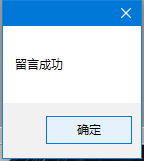
1. 若用教师的身份登录使用该功能
2. 主界面显示学生用户的信息，并且每一栏都可以选择升序或降序排序。



B.单击选中某学生，给她留言

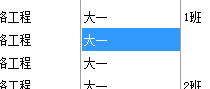


C.按发送留言，将显示留言成功。学生将收到留言。

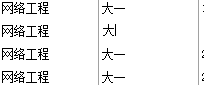


D.教师可以对某个学生的信息进行修改。比如将某同学的年级从大一改为大二。

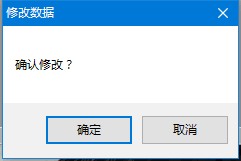
双击该信息



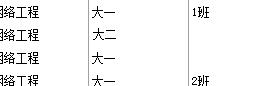
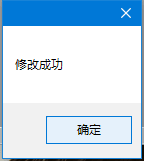
直接修改后按回车键



系统提示是否确认修改



点击确认，之后修改成功



（2）用学生的身份登录后使用该功能，将显示教师发给你的留言。



**3.3.3 修改用户个人信息**



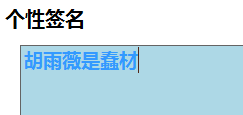
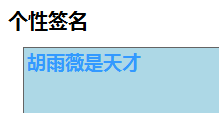
1. 主界面



（2）进行信息的修改



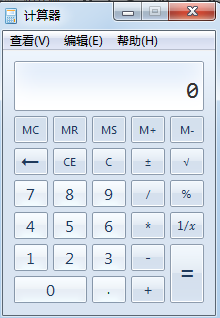
修改前： 修改后：



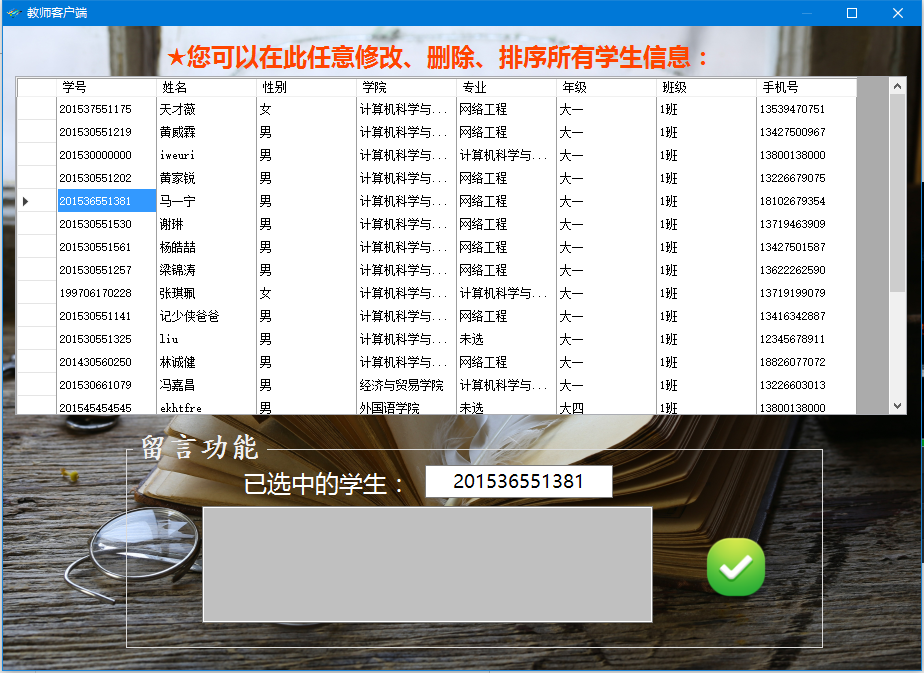
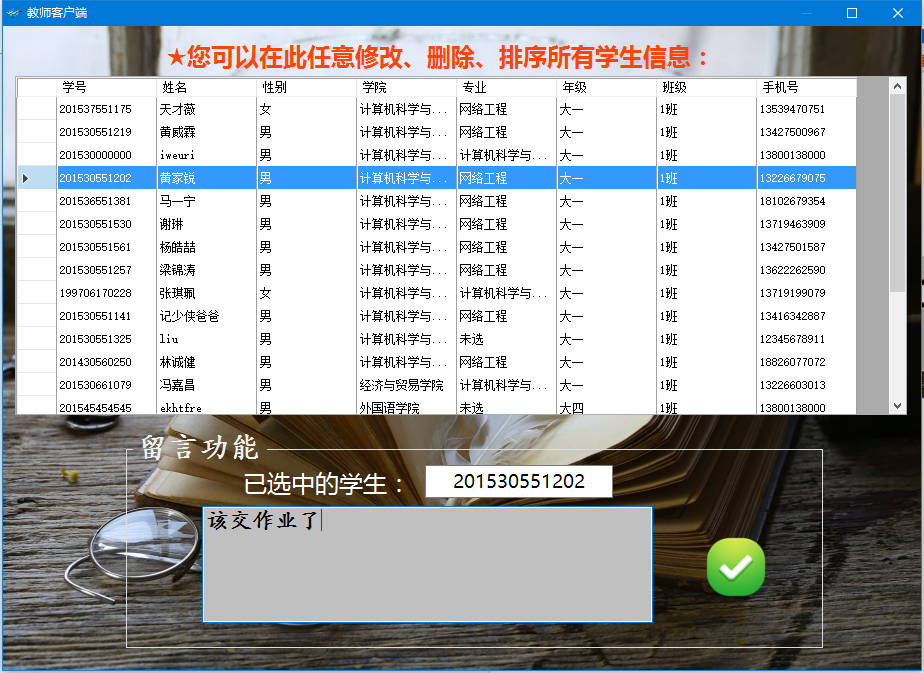
点击。修改成功。

**3.3.4 计算器功能**





**3.3.5 教师管理员登陆**



老师可以修改注册的用户的信息，进行修改删除，排序学生信息。

老师可以对选中的学生留言。

**四、使用说明**

**4.1使用流程**

软件提供方：

1.配置MySQL环境，并按照SQL语句创建数据库

2.修改程序中的IP地址为服务器内网和外网地址重新编译

3.启动服务器端（一键开启）

使用方：

打开客户端注册使用即可。

**4.2使用说明**

本程序使用微软.Net4.0编写，需使用Windows7及其以上版本系统方可使用。

**五、收获体会及建议**

这次的大作业设计从构思到真正实现共经历了三个月的时间，这三个月里我收获到的不只是真真切切的知识还有的是完整的程序设计的体验和宝贵的人生经历。

大作业刚开始打的前几周，我们便分析了这个程序需要掌握的哪些新知识，脑海中大致对这些新的领域有了初步的认识，然后就是开始分工合作。我用了一周的时间学习c++下册内容，了解了.NET的存在，学习了更高级一点的CLR语法以及托管代码的好处，在这个过程中我真正体验到了编程语言的发展和革新，也对以后的学习充满了憧憬。

学习过了新语法之后，我便进入了界面的学习，这是我第一次设计程序接界面UI。怀着激动的心情，我打了书上的第一个例子，这也标志值我对界面设计的初步认识。 后来，我渐进通过课本和网上的博客学习界面知识，了解了MFC和其他组件的用法，在我们大作业的进程中其实不时地就会有各种问题，比如想实现一种控件的效果却不知道如何设置编写，其实，解决问的过程中才是我们进步的时刻，每一个问题的解决都会学到很多。渐渐地，进度条的使用原理，界面托管素材的连接，界面类的原理，界面的跳转，界面属性的设置……随着这些原理意义浮出水面，我对界面的理解一次次加深，现在已经基本理解各个软件界面的原理和布局方法。

接下来，我们程序由于使用的是网络客户端和服务器的架构，所以学习网络编程势在必行。我通过网上的博客进行学习，完全零基础接触TCP这些协议的原理，在大致了解了TCP、UDP、IP这些协议后，我开始了Socket协议的学习。通过一些例子和自己的摸索，一次次尝试后我编写了第一个客户端和服务器的简易通讯程序，这个程序的设计就很简单，一个给另一个发送一条消息，另一个而接受后再发送回执，然后这样循环等待。终于，成功的那一刻我是多么开心，现在我第一次做到了两个程序之间的通信！虽然说只是本地局域网内，没有向后来这样可以连入互联网，而且接收和发送只能轮流进行不能同时多线程运行，但这成功的一小步却让我信心满满。在完全了解了网络编程的基础上，我为了可以进行多线程套接字通讯，开始学习多线程知识。

学习多线程知识就显得比较简单，因为我以前接触过Java的多线程，在c++里面也是大同小异，很快我便成功做到了多线程的网络通讯。

接下来的学习重点转移到了数据库上。也是因为之前完全没有接触过数据库，零基础的我去买了一本《数据库原理》然后开始从简单的概论开始认识数据库。一开始我打算用的是嵌入式数据库SOLite，在经过了各种权衡之后还是放弃了，因为我想既然是真正的想去完成一个完整模式的应用程序，就应该考虑以后的扩展性，所以我还是使用了市面上几乎小型软件都在使用的数据库MySQL在网上各种教程的帮助下，我安装了MySQL对VS的补丁，了解到了一些常用操作的语法，又是一次次的尝试，那时候几乎每天都会有新的功能被理解使用，从数据库的创建到数据库的更新、获取和删除，真的体会了每天进步一点点积少成多的乐趣。

至此，基本的学习任务就告一段落。我作为组长和组员一起经过多次讨论，在不同地点开会后分配了每个人的任务。然后就开始了分工合作。这也是我第一次和别人的代码一起合作编写，虽然一开始由于没有经验，在代码的耦合上面出现了各种难以预料的问题，随着我们问题的一一解决，合作显得越来越得心应手，合作的效率优势也很好的体现出来了。在后期，我们几乎可以做到每天实现一个新功能，我作为基本类的设计者和每一个新功能服务器的设计者，家锐作为客户端的设计者，胡雨薇作为界面控件的设计者，基本同时进行的话，很快就可以产生一个新的功能。

那么，在我自己的工作中，在服务器类体系设计中，我也收获了很多。在编写前期，第一个服务产生的时候，我基本是用了一周的时间去调试bug和完善服务器的使用质量，在总结了各种服务器和客户端通信过程中可能出现的连接错误的情况下要做到服务器不可以瘫痪，而且要可以做到连续的为客户端提供服务，这些不得不让我查找很多资料，学习很多诸如心跳检测的优化方法。

在第一版服务打好的情况下，我开始构思如何扩展我的服务，也就是开始构思服务器的类体系设计。我从第一版服务中开始抽象出每一个服务必备的功能然后把这些放在基类里面，为后面代码重用做足准备。现在的我发现，继承虚函数等这些代码重用技术真的可以在实际项目开发中节约很多时间，而且设计思路会很明确，在真正完美使用这些代码重用技术后，我真的很难想象如果没有用的话，工作量会加大多少倍。

最后的最后，当我们打完所有代码和功能后，大家都是发自内心的开心和激动。我们在网上购买了第三方服务器，然后成功的将程序服务器和数据库部署在了上面，也成功的让程序脱离了局域网的束缚，可以自由使用，这种成就感是从未有过的。现在我们的程序，可以随时登录使用，和服务器通讯，用户之间可以聊天通讯，一起都是成功实现。

这次的程序设计大作业就告一段落了，总结一下，真的学到了无数宝贵的开发经验和真实的知识，接触了从未接触过的界面设计、网络编程、数据库和服务器，也体验了真正软件在开发工程中的步骤和精神，积累了丰富的经验。

在这里，我想感谢老师给我们这样美好的机会，感谢我的组员和我一起成长一起分享这段欢乐，感谢给予我帮助的老师同学，感谢使用我们软件内侧和公测的好朋友们，谢谢你们！