暨南大學
JINAN UNIVERSITY

本科课程设计

课 程 名 称：C++程序设计

课 程 编 号：08060226

学 生 姓 名：柯瑞凯　陈彦亨　柯悦

学　　号：2020101601　2020101602　2020101599

学　　院：信息科学技术学院

系：计算机科学系

专　　业：网络工程

指 导 教 师：张晓刚

教 师 单 位：信息科学技术学院

开 课 时 间：2020～2021 学年度第二学期

暨南大学教务处

2021　年 6 月 23 日

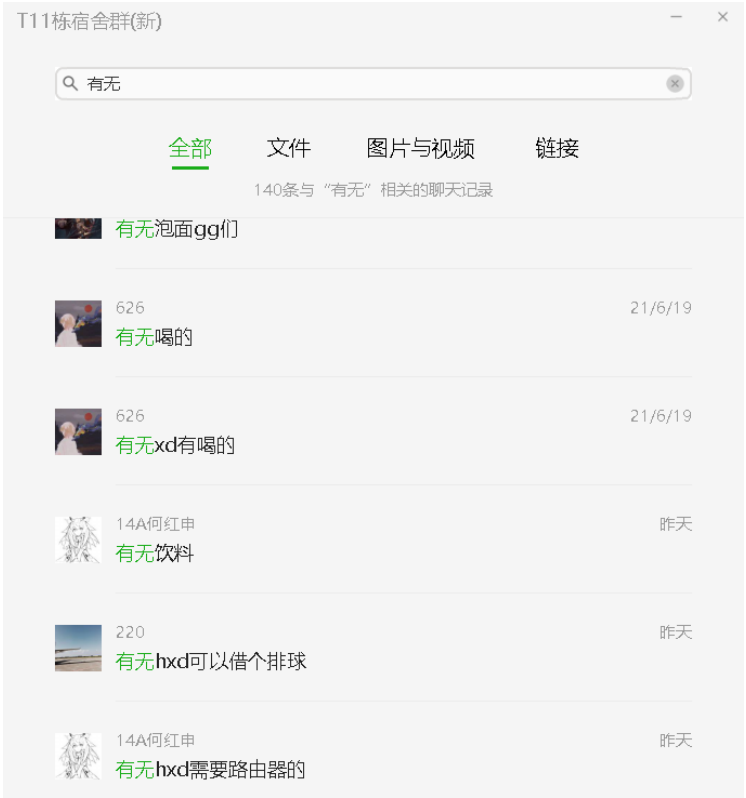
目录

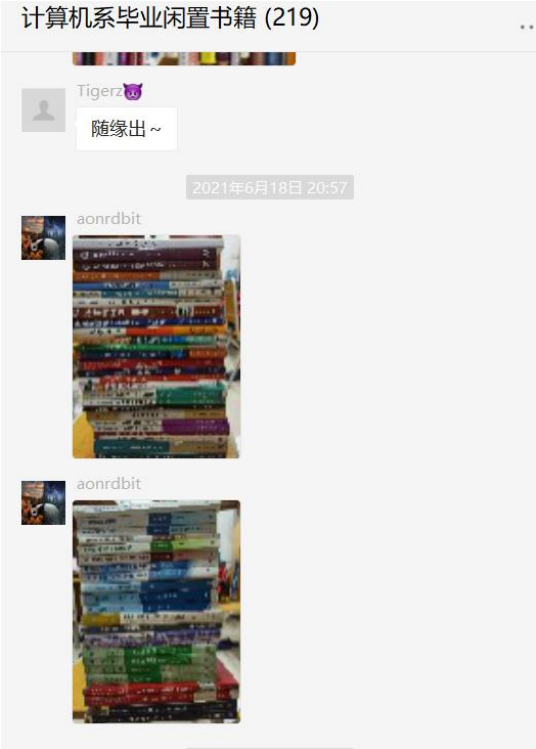
一、题目及需求分析

1．选做此课题的背景，目的与意义

1)背景分析

在我们进入学校的这一年来，我们意识到日常生活中同学们有着许多的需求，例如有些同学因为课程时间问题无法及时取出快递，需要求助他人帮忙代拿，有些同学需要买卖或借用某些物品，却找不到可以交易的同学。在众多的群聊中（如下图），我们注意到学校有着庞大的二手交易需求和服务需求，而却缺乏能将提供需求和接受需求的双方高效对接的平台或渠道。





2)目的

为了便利同学们的需求，我们制作这样的一个学生需求服务系统，为同学们提供一个平台解决一些交易需求和服务需求。有时间和精力的同学可以通过这样的一个平台为有需要的人提供服务从而获得报酬，有需求的同学也可以通过这个平台寻找其他人的帮助。

2．系统的功能需求分析

我们的学生需求服务系统能提供主要包括：商品交易，快递代拿，资料分享和交友平台这四个功能。

1)商品交易需求分析

二手商品的交易在校园里面具有巨大的潜力，同学们生活中总会产生一些不再需要的物品，如书籍，食物，生活用品等，但是直接扔垃圾桶又觉得可惜；又或者是大四即将毕业的学长学姐没有办法将一些物品带离学校，他们往往会将商品挂在二手交易的微信群里面，但是微信群里面又不是特别的方便，这时候如果他们使用我们的学生需求服务系统，只需注册一个账号，便能利用极短的时间将自己所有的需要转卖的东西信息填上，而后只需等待他人与自己联系，手上的商品便能比较轻松地卖出，大大节省了时间。

2)快递代拿需求分析

在如今网上购物蓬勃发展的时代，校园里面的快递站几乎是天天爆满，每天都有许多人往返于菜鸟驿站和宿舍楼之间。而到了 618，双 11 这样的购物狂欢节，许多人想必每天都有许多快递要取，但对于那些力气小并且不想晒太阳的女生来说简直是一场噩梦，这时候如果这些有代拿需求的人将快递取件码等信息发布到我们的学生需求服务系统，并支付一定的酬劳，彼时在快递点取快递的人就可以接下这个需求，回宿舍时顺手帮他们把快递取回来，这样子又促成了双方互赢的局面。

3)资料分享需求分析

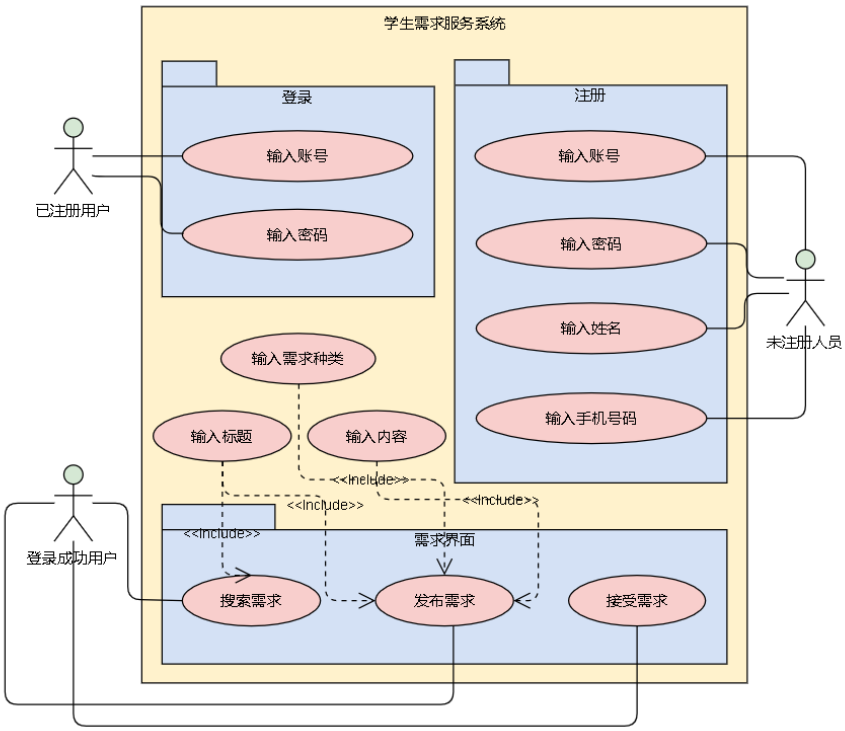
现在已经到了期末考复习的阶段，但说到考试资料，肯定是同学们十分头疼的问题，获得去年的期末考试试题或者是学长学姐精心准备的复习资料，能够帮助同学们更精准地把握考点，大大地减少复习的时间和提高复习的效率，但最大的问题是，由于自己跟往届的师兄师姐不熟，任课老师又不没有发布往年的考试题，自己难以获得对口的考试资料或是期末试题，复习仿佛就是大海捞针。这个时候如果你实用我们的学生需求服务系统发布自己的对考试资料的需求，你的一些好心的学长学姐看到了肯定会满足你的需求，把资料发放给你，而你的学长学姐们也可以从更高一级的师兄师姐获取考试资料，这样就形成了一个考试资料分享的良性循环，也促进了同专业不同届同学之间的交流。

4)交友平台需求分析

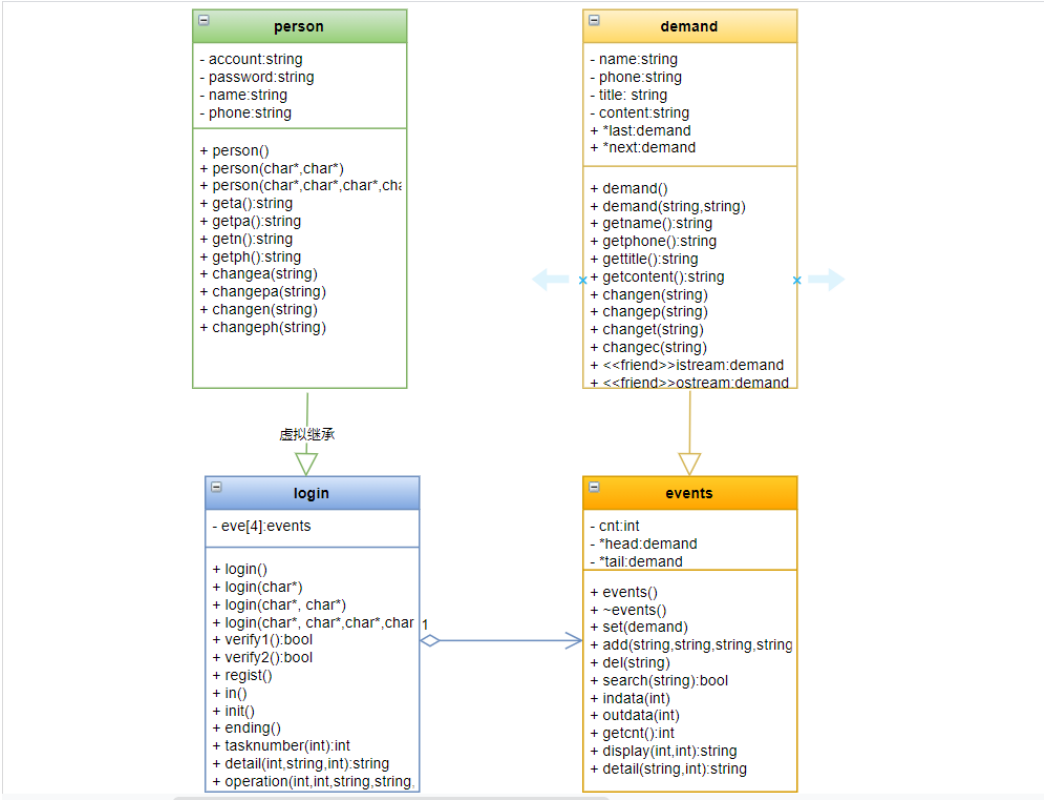
现代社会追求人的全面发展，每个人从小就能接触到大千世界不同的东西，从而培养了广泛的兴趣爱好，而在生活的校园里面，却难以寻找到意气相投的伙伴，未免有些可惜。如今，通过我们的学生需求服务系统，能够在交友平台发布自己想要交什么样的朋友，无论是能一起开黑的游戏伙伴，或者是能一起打球的球友，能一起玩桌游的小伙伴，抑或是爱好比较文艺清新如摄影，绘画，旅游等等的朋友，相信都能在我们的平台上想遇。世上最难受的事情莫过于找不到和自己有相同爱好的人，相信我们的系统能够帮助大家找到意气相投的小伙伴，让生活更有趣。

二、系统设计

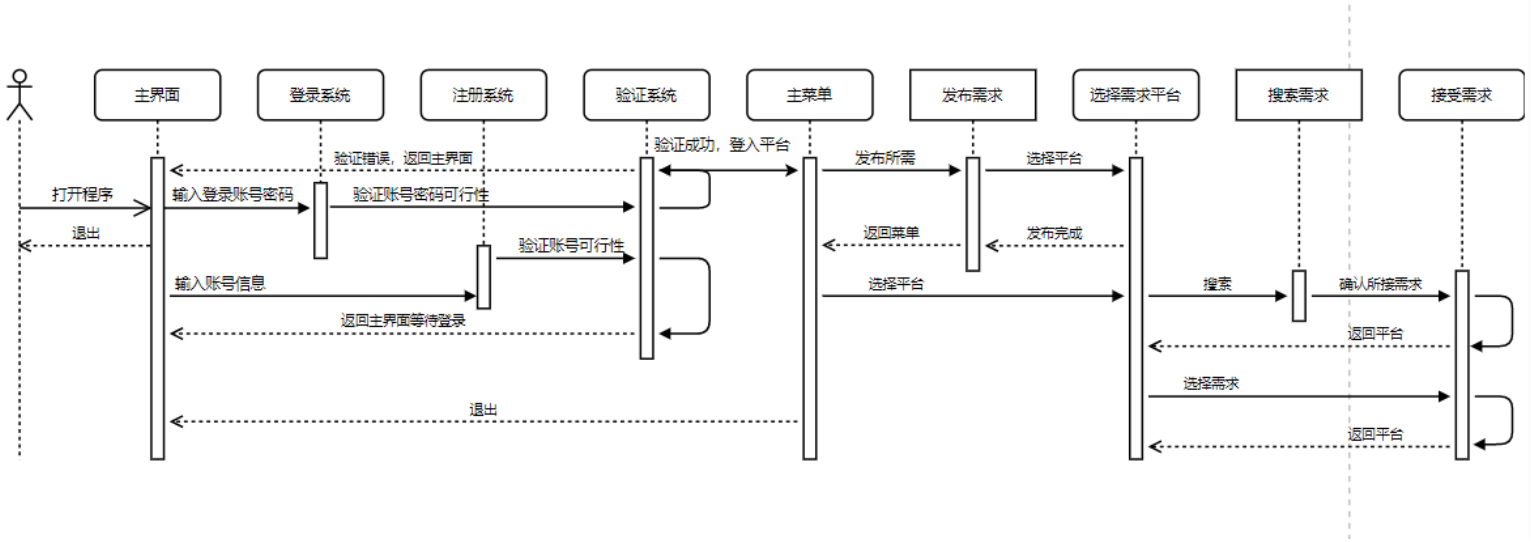
1.根据所选题目，通过用例图描述系统总体功能



2.使用类图描述系统总体设计



3.使用时序图描述系统的主要业务操作



三、系统运行及调试

1.系统运行及调试

1） 运行所需软件：visual studio(vs)

2）运行前的准备：

该源程序引入了 graphics.h 头文件，所以在运行之前，需要将 Easyx 安装到 visual studio 中。

3）运行程序：对压缩包解压，可以将压缩包中的源程序在 vs 中直接运行（需要完成步骤 1，2 ）。也可以通过解压后文件夹内附带的应用程序进行系统操作。

4）运行过程：按照窗口界面提示进行操作。

5） 结束：请勿直接强制中断程序地进行，按照界面提示退出程序,防止数据丢失。

2.系统问题及其解决方案

1） 在实现可视化编程时，单线程无法同时实现鼠标和键盘消息的读取，故在登录界面输入账号密码时，只检测键盘信息，若检测到回车键，则由账号输入框进入到密码输入框，密码输入结束后按下回车键再进行鼠标信息的读取。

2）由于数据的储存需要，在暂未学习数据库的情况下，无法将数据在单次运行后保存。故采用文件流输入输出的方式，将每次运行完后的的数据保存在文件中，当后面运行时可再次把文件里的数据读出来，实现数据的持久化。

3）在系统运行过程中存在对数据的实时操作，对文件中储存的数据进行直接操作，一是难以达到目标操作的结果，二是对文件的实时操作有导致文件无法关闭的风险，造成文件内数据的大量积累，影响系统。故选择一种合适的数据结构——链表，来解决这个问题。通过构建多条链表，将文件的信息读取出来，再在链表里对数据进行操作，在更方便修改数据情况下也避免数据溢出。最后只要将原文件清空，将链表里的数据输出即可。

4） 使用链表时，对链表的析构显得非常重要，这会防止内存泄漏。然后对链表中出现的各种指针，错误的析构可能会导致野指针的频繁出现，以致程序卡退。故在析构时，需谨慎对虚构对象进行判断，防止悬挂指针。

3.可扩展功能及设计实现构想

1） 增设查看个人所发布的所有需求和已接受的所有需求。（通过读取需求文件，输出所有与当前登陆账号相同实名的需求）

2）增设任务截至日期，超过日期后自动清除相关需求的显示。（每次进入需求显示界面时获取当前时间，将需求文件中的设置的截至日期进行对比，只将未超过日期的需求写入链表，后将链表内容重新覆盖需求文件）

3）增设搜索用户功能，输入用户账号后可以得到其可现实信息，如历史接受的需求数量、历史发布的需求数量和当前待接受的需求数量。（在每次用户发布需求和接受需求后在用户的账户文件进行数量的更新，搜索账号后则显示该账号的相关数量信息）

四、设计总结

该系统根据需求定义了四个类，分别是 person，login，events 和 demand。首先是 person 类，用以实现简单的账号密码等数据的获取；login 类是在 person 类上派生出来的，继承了其数据获取的功能，并实现了注册时判断账号是否已被使用、判断账号密码是否匹配、以及导入文件信息、导出文件信息等用户操作功能，events 类则对 demand 类进行聚合。demand 类作为基类拥有数据单元结构，events 类因此可以实现对数据的存储。同时，在 events 类中添加了多个成员函数，实现对所储存数据的实时修改，达到在线化数据的功能。同时，基于 demand 重载的输入输出，可以方便地实现将数据保存进文件中，达到数据可持久化的效果，最终构建出较为完整的系统平台。完成了面向对象的编程后，我们通过图形库函数以及部分 windows 编程函数的使用，将代码辅以 UI 设计，避免了枯燥的控制台操作，实现了可视化。然而目前由于知识储备不足，无法真正让该系统的需求数据在不同电脑之间进行交互。未来我们将会自学数据库相关知识，尝试将数据的交互真正实现，让系统功能不再停留在单个电脑的“自娱自乐”，让项目真正落地。

五、参考文献

技术分享 - 使用 mciSendString 函数实现循环播放音频文件 (write-bug.com)

char 与 TCHAR 相互转化_B_H_L 的专栏-CSDN 博客_char 转 tchar

EasyX 文档 - Basic introduction

六、附录

源程序：

```
#include <stdlib.h>
#include <graphics.h>
#include <conio.h>
#include <windows.h>
#include <mmsystem.h>
#pragma comment (lib, "winmm.lib")
#include"login.h"
#include"demand.h"

const int LEN=15;//长度限制
const int HEIGHT = 990;//界面长度
const int WIDTH = 556.5;//界面宽度
TCHAR account[LEN], password[LEN], phone[LEN], name[LEN];//账号、密码、手机、姓名
TCHAR diversity[2], title[LEN], content[50];//diversity[0]储存需求种类、需求标题、需求内容
TCHAR ss[] = _T("w");//char 转为 TCHAR 的媒介数组，调用函数后 ss 为参数的 TCHAR 形式
string number[1000];//需求编号
string element[10] = { "0","1","2","3","4","5","6","7","8","9" };//组成需求编号的元素
login usernow;//当前菜单界面的登录账号

char* TCHARTOCHAR(LPWSTR);//将 TCHAR 类型的数组转换为 char 类型数组
void CHARTOTCHAR(const char*);//将 char 类型的数组转换为 TCHAR 类型数组
void Begin();//起始界面
void Register();//注册流程
void Login();//登录界面
void Login_Word();//登录界面文字显示
void Login_Change();//登录界面登录取消按钮变色
void Login_Inputbox();//登录界面绘制输入框
void Login_Cursor1();//登录界面账号栏光标闪烁
void Login_Cursor2();//登录界面密码栏光标闪烁
void Login_Input1();//登录界面账号输入
void Login_Input2();//登录界面密码输入
void Menu_Initial();//进入菜单界面
void Menu_Need(int,int);//菜单界面需求展示
void Menu_Publish(int);//菜单界面发布需求

struct btNode {
    int x, y;
    wchar_t text[20];//内容
    int status;//默认为 0，按下按钮时为 1
};//按钮
btNode Node[5] = {
    {30,50,L"资料分享",0},{30,50,L"资料分享",0},
    {30,150,L"快递代拿",0},
    {30,250,L"商品交易",0},
    {30,350,L"交友平台",0},
};//主界面按钮信息
void DrawBtn(int k) {
    btNode t = Node[k];
    settextcolor(BLACK);
    settextstyle(45, 0, _T("迷你简中特广告"));
    setlinecolor(RGB(50,50,75));
    setlinestyle(PS_SOLID, 2, NULL, 0);//设置文字、边框的格式及颜色
    if (t.status == 0) {
        roundrect(t.x, t.y - 8, t.x + 193, t.y + 55, 20, 20);//起始为无填充按钮
    }
    else {
        setfillcolor(RGB(130,127,133));
        fillroundrect(t.x, t.y - 8, t.x + 193, t.y + 55, 20, 20);//点击时 status=1，转换为
填充按钮
    }
    setbkmode(TRANSPARENT);
    outtextxy(t.x + 13, t.y + 2, t.text);//输出按钮文字
}//绘制按钮
```

```
inline char* TCHARTOCHAR(LPWSTR lpwszStrIn){
     LPSTR pszOut = NULL;
     if (lpwszStrIn != NULL){
          int nInputStrLen = wcslen(lpwszStrIn);
          int nOutputStrLen = WideCharToMultiByte(CP_ACP, 0, lpwszStrIn, nInputStrLen, NULL,
0, 0, 0) + 2;
          pszOut = new char[nOutputStrLen];
          if (pszOut){
               memset(pszOut, 0x00, nOutputStrLen);
               WideCharToMultiByte(CP_ACP, 0, lpwszStrIn, nInputStrLen, pszOut,
nOutputStrLen, 0, 0);
          }
     }
     return pszOut;
}
inline void CHARTOTCHAR(const char* s) {
     int iLength;
     iLength = MultiByteToWideChar(CP_ACP, 0, s, -1, NULL, 0);
     MultiByteToWideChar(CP_ACP, 0, s, -1, ss, iLength);
}
void Begin() {
     initgraph(HEIGHT, WIDTH);
     setbkmode(TRANSPARENT);
     loadimage(NULL, _T("0.jpg"), HEIGHT, WIDTH);//初始化界面
     settextcolor(BLACK);
     settextstyle(30 * 1.5, 0, _T("华文彩云"), 0, 0, 750, 0, 0, 0);//设置文字格式
     TCHAR s2[] = _T("登录");
     outtextxy(295 * 1.5, 151 * 1.5, s2);//输出文字
     TCHAR s3[] = _T("注册");
     outtextxy(295 * 1.5, 225 * 1.5, s3);
     TCHAR s4[] = _T("退出");
     outtextxy(295 * 1.5, 299 * 1.5, s4);
     settextcolor(RED);
     settextstyle(40 * 1.5, 0, _T("楷体"), 0, 0, 1000, 0, 0, 0);//设置文字格式
     TCHAR s1[] = _T("学生需求服务系统");
     outtextxy(160 * 1.5, 70 * 1.5, s1);
     settextstyle(15 * 1.5, 0, _T("方正少儿简体"), 0, 0, 0, 0, 0, 0);
     TCHAR s0[] = _T("By: 第41小组");
     outtextxy(568 * 1.5, 350 * 1.5, s0);
     FlushMouseMsgBuffer();//清空鼠标操作缓存区
     MOUSEMSG m;
     while (1)
     {
          m = GetMouseMsg();//获取鼠标信息
          if (m.uMsg == WM_LBUTTONDOWN) {//单击左键后判断各个按钮的坐标进入不同界面
               if (m.x <= 358 * 1.5 && m.x >= 295 * 1.5 && m.y <= 181 * 1.5 && m.y >= 151 *
1.5) {
                    Login();
               }
               if (m.x <= 358 * 1.5 && m.x >= 295 * 1.5 && m.y <= 255 * 1.5 && m.y >= 225 *
1.5) {
                    Register();
               }
               if (m.x <= 358 * 1.5 && m.x >= 295 * 1.5 && m.y <= 329 * 1.5 && m.y >= 299 *
1.5) {
                    exit(0);
               }
          }
     }
     closegraph();//关闭窗口
}
void Register() {
     login user;//声明一个未初始化的对象，用于后续储存注册人员信息
     int i = InputBox(account, LEN, L"请输入账号(不支持特殊符号或中文)", L"注册", 0, 0, 0,
0);//弹出对话框
     user.changea(TCHARTOCHAR(account));//储存账号
     if (i == 0) {//点击了右上角关闭键或选择了取消
          MessageBox(GetForegroundWindow(), L"注册失败", L"提示", 0);//弹出消息，提示注册失
败
```

```
            Begin();//回到主页面
        }
        else {
            while (user.verify2()) {//判断账号是否已存在，不存在则跳出循环
                InputBox(account, LEN, L"账号已被注册，请重新输入", L"提示", 0, 0, 0, 1);
                user.changea(TCHARTOCHAR(account));//存入账号
            }
            int len1;//用于判断账号长度
            len1 = lstrlen(account);
            while (len1 < 6) {//当账号长度大于等于6时跳出循环
                InputBox(account, LEN, L"账号长度需大于等于六位，请重新输入", L"提示", 0, 0,
0, 1);
                len1 = lstrlen(account);
            }
            user.changea(TCHARTOCHAR(account));//更新账号
            int j = InputBox(password, LEN, L"请输入密码(不支持特殊符号或中文)", L"注册", 0,
0, 0, 0);
            int len2 = lstrlen(password);//判断密码长度
            if (j == 0) {//输入密码时点击取消或关闭窗口则终止注册
                MessageBox(GetForegroundWindow(), L"注册失败", L"提示", 0);
                Begin();
            }
            else {//当密码长度大于等于6时跳出循环
                while (len2 < 6) {
                    InputBox(password, LEN, L"密码长度需大于等于六位，请重新输入", L"提示",
0, 0, 0, 1);
                    len2 = lstrlen(password);
                }
                user.changepa(TCHARTOCHAR(password));//更新密码
                MessageBox(GetForegroundWindow(), L"注册成功", L"提示", 0);//注册成功
            }
        }
        MessageBox(GetForegroundWindow(), L"请开始完善个人信息", L"提示", 0);
        InputBox(name, LEN, L"请输入姓名", L"个人信息", 0, 0, 0, 1);
        user.changen(TCHARTOCHAR(name));//将输入的姓名储存
        InputBox(phone, LEN, L"请输入手机号码", L"个人信息", 0, 0, 0, 1);
        int len3 = lstrlen(phone);//判断手机号码长度
        while (len3 != 11) {//当手机长度为11时跳出循环
            InputBox(phone, LEN, L"请输入正确的手机号码(11位)", L"个人信息", 0, 0, 0, 1);
            len3 = lstrlen(phone);
        }
        user.changeph(TCHARTOCHAR(phone));//更新手机号码
        MessageBox(GetForegroundWindow(), L"个人信息填写成功", L"提示", 0);
        user.regist();//将个人信息全部写入文件
        Login();//成功注册后进入登录界面
}
void Login() {
        IMAGE LOGIN;
        initgraph(HEIGHT, WIDTH);
        loadimage(&LOGIN, _T("0.jpg"), HEIGHT, WIDTH);
        SetWorkingImage(&LOGIN);////设定当前的绘图设备为LOGIN对象
        login user;//声明一个对象用于记录登录时输入的账号密码
        Login_Word();//将登录界面的文本导入
        Login_Inputbox();//将登录界面的输入框导入
        SetWorkingImage(); //设定当前的绘图设备为默认绘图窗口
        putimage(0, 0, &LOGIN);//在当前设备绘制指定图像，即在默认绘图窗口绘制LOGIN对象
        settextstyle(18 * 1.5, 0, L"宋体", 0, 0, 0, 0, 0, 0);
        setbkmode(OPAQUE); //设置当前设备图案填充和文字输出时的背景模式，OPAQUE表示背景用当前背
景色填充
        setbkcolor(RGB(255, 255, 255));//设置默认背景颜色
        MOUSEMSG m;
        while (1) {
            m = GetMouseMsg();
            if (m.uMsg == WM_LBUTTONDOWN) {
                if (m.x <= 426 * 1.5 && m.x >= 353 * 1.5 && m.y <= 290 * 1.5 && m.y >= 255 *
1.5) {
                    Begin();//点击取消则回到初始界面
                    break;
                }
```

```
            else if (m.x <= 298 * 1.5 && m.x >= 225 * 1.5 && m.y <= 290 * 1.5 && m.y >=
255 * 1.5) {
                MessageBox(GetForegroundWindow(), L"请输入账号", L"提示", 0);//点击登录则
提示先输入账号
            }
            else if (m.x <= 425 * 1.5 && m.x >= 275 * 1.5 && m.y <= 191 * 1.5 && m.y >=
161 * 1.5) {//点击输入框
                while (_kbhit()) {//清空键盘消息缓冲区，避免输入前的键盘消息直接被导入输
入框
                    _getch();
                }
                Login_Cursor1();//控制光标闪烁，检测到键盘输入后结束闪烁
                break;//结束闪烁后退出鼠标判断
            }
        }
    }
    Login_Input1();//根据键盘消息在账号输入框显示账号，点击回车键或到达指定输入位数后结束
    user.changea(TCHARTOCHAR(account));//将账号输入 login
    Login_Cursor2();//密码输入框闪烁，检测到键盘输入后结束闪烁
    Login_Input2();//根据键盘消息在密码输入框显示密码（*号显示），点击回车键或到达指定输入位
数后结束
    user.changepa(TCHARTOCHAR(password));//将密码输入 login
    Login_Change();//填充登录、取消键位
    FlushMouseMsgBuffer();//清空鼠标消息缓冲区
    while (_kbhit()) {//清空键盘消息缓冲区
        _getch();
    }
    while (1) {
        m = GetMouseMsg();
        if (m.uMsg == WM_MOUSEMOVE) {
            if (m.x >= 225 * 1.5 && m.x <= 298 * 1.5 && m.y >= 255 * 1.5 && m.y <= 290 *
1.5) {//鼠标移动到登录键时，登录二字变为红色
                settextcolor(RGB(255, 0, 0));
                outtextxy(236 * 1.5, 260 * 1.5, _T("登录"));
            }
            else if (m.x >= 353 * 1.5 && m.x <= 426 * 1.5 && m.y >= 255 * 1.5 && m.y <=
290 * 1.5) {//鼠标移动到取消键时，取消二字变为红色
                settextcolor(RGB(255, 0, 0));
                outtextxy(364 * 1.5, 260 * 1.5, _T("取消"));
            }
            else {
                settextcolor(BLACK);//在其他位置则两个键位文字都为黑色
                outtextxy(236 * 1.5, 260 * 1.5, _T("登录"));
                outtextxy(364 * 1.5, 260 * 1.5, _T("取消"));
            }
        }
        else if (m.uMsg == WM_LBUTTONDOWN) {
            if (m.x <= 298 * 1.5 && m.x >= 225 * 1.5 && m.y <= 290 * 1.5 && m.y >= 255 *
1.5) {//左键单击登录键
                if (user.verify1()) {//账号密码匹配
                    usernow = user;//将当前 user 的账号密码赋值给全局对象 usernow 以便在需
求界面的使用
                    usernow.in();//查到该账号对应的手机和姓名并赋值给 usernow
                    Menu_Initial();//进入需求界面
                    break;
                }
                MessageBox(GetForegroundWindow(), L"密码错误或账号不存在！", L"提示",
0);//账号密码不匹配则进行提示
                Login();//刷新登录界面重新登录
                break;
            }
            else if (m.x <= 426 * 1.5 && m.x >= 353 * 1.5 && m.y <= 290 * 1.5 && m.y >=
255 * 1.5) {
                Begin();//点击取消则回到初始界面
                break;
            }
        }
    }
}
```

```
void Login_Word() {
    setbkmode(TRANSPARENT);//设置当前设备图案填充和文字输出时的背景模式为透明
    settextstyle(40 * 1.5, 0, L"楷体", 0, 0, 1000, 0, 0, 0);
    settextcolor(RED);
    TCHAR s1[] = _T("用户登录");
    outtextxy(245 * 1.5, 55 * 1.5, s1);
    setlinecolor(BLACK);
    rectangle(175 * 1.5, 125 * 1.5, 475 * 1.5, 315 * 1.5);//外围矩形框
    setfillcolor(RGB(190, 190, 190));
    rectangle(225 * 1.5, 255 * 1.5, 298 * 1.5, 290 * 1.5);//登录框
    rectangle(353 * 1.5, 255 * 1.5, 426 * 1.5, 290 * 1.5);//取消框
    settextstyle(25 * 1.5, 0, L"宋体", 0, 0, 0, 0, 0, 0);
    settextcolor(BLACK);
    outtextxy(215 * 1.5, 162 * 1.5, _T("账号"));
    outtextxy(215 * 1.5, 205 * 1.5, _T("密码"));
    settextstyle(25 * 1.5, 0, L"宋体", 0, 0, 500, 0, 0, 0);
    outtextxy(236 * 1.5, 260 * 1.5, _T("登录"));
    outtextxy(364 * 1.5, 260 * 1.5, _T("取消"));
}
void Login_Change() {
    setbkmode(TRANSPARENT);
    setlinestyle(PS_SOLID, 1, NULL, 0);//设置边框格式
    setlinecolor(BLACK);
    setfillcolor(RGB(190, 190, 190));
    fillrectangle(225 * 1.5, 255 * 1.5, 298 * 1.5, 290 * 1.5);
    fillrectangle(353 * 1.5, 255 * 1.5, 426 * 1.5, 290 * 1.5);
    settextstyle(25 * 1.5, 0, L"宋体", 0, 0, 800, 0, 0, 0);
    settextcolor(BLACK);
    outtextxy(236 * 1.5, 260 * 1.5, _T("登录"));
    outtextxy(364 * 1.5, 260 * 1.5, _T("取消"));
}
void Login_Inputbox()
{
    setlinestyle(PS_SOLID, 1, NULL, 0);
    setfillcolor(WHITE);
    int i;
    for (i = 0; i < 2; i++) {
        bar(275 * 1.5, (161 + 41 * i) * 1.5, 425 * 1.5, (191 + 41 * i) * 1.5);
        setcolor(RGB(0, 0, 0));
        rectangle(275 * 1.5, (161 + 41 * i) * 1.5, 425 * 1.5, (191 + 41 * i) * 1.5);
    }
}
void Login_Cursor1()//闪烁光标
{
    while (1) {//如果键盘没有进行输入这循环闪烁
        Login_Inputbox();//画账号密码框
        if (_kbhit()) {//当键盘有反应时_kbhit()会返回一个非零值
            break;//退出闪烁
        }
        Sleep(500);
        setlinestyle(PS_SOLID, 2, NULL, 0);//设置线的样式为 PS_SOLID, 宽度为 2
        line(278 * 1.5, 165 * 1.5, 279 * 1.5, 185 * 1.5);//光标的描绘
        Sleep(500);//延时
    }
}
void Login_Cursor2()//闪烁光标
{
    while (1) {//如果键盘没有进行输入这循环闪烁
        setlinestyle(PS_SOLID, 1, NULL, 0);
        bar(275 * 1.5, 202 * 1.5, 425 * 1.5, 232 * 1.5);
        rectangle(275 * 1.5, 202 * 1.5, 425 * 1.5, 232 * 1.5);
        if (_kbhit()) {//当键盘有反应时_kbhit()会返回一个非零值
            break;//退出闪烁
        }
        Sleep(500);
        setlinestyle(PS_SOLID, 2, NULL, 0);//设置线的样式为 PS_SOLID, 宽度为 2
        line(278 * 1.5, 206 * 1.5, 279 * 1.5, 226 * 1.5);//光标的描绘
        Sleep(500);//延时
    }
```

```
}
void Login_Input1() {
    for (int i = 0; i < LEN; i++) {
        account[i] = _getch();
        outtextxy((277 + 8 * i) * 1.5, 167 * 1.5, account[i]);
        if (account[i] == 8) {//使用 delete 键
            account[i] = 0;
            outtextxy((277 + 8 * i) * 1.5, 167 * 1.5, L"  ");//输出空白挡住 delete 键
            if (i > 0) outtextxy((277 + 8 * (i - 1)) * 1.5, 167 * 1.5, L"  ");//输出空白挡
住前一个字符以示删除
            i -= 2;
            if (i < 0) {
                i = -1;
            }
        }
        else if (account[i] == 13) {//使用回车键
            outtextxy((277 + 8 * i) * 1.5, 167 * 1.5, L"  ");
            account[i] = '\0';
            break;
        }
    }
}
void Login_Input2() {
    int j = 0;
    for (int j = 0; j < LEN; j++) {
        password[j] = _getch();
        outtextxy((277 + 8 * j) * 1.5, 208 * 1.5, L'*');//输出*号
        if (password[j] == 8) {//使用 delete 键
            password[j] = 0;
            outtextxy((277 + 8 * j) * 1.5, 208 * 1.5, L"  ");//输出空白挡住 delete 键
            if(j>0) outtextxy((277 + 8 * (j - 1)) * 1.5, 208 * 1.5, L"  ");//输出空白挡住
前一个字符以示删除
            j -= 2;
            if (j < 0) {
                j = -1;
            }
        }
        else if (password[j] == 13) {//使用回车键
            outtextxy((277 + 8 * j) * 1.5, 208 * 1.5, L"  ");
            password[j] = '\0';
            break;
        }
    }
}
void Menu_Initial() {
    usernow.init();//将需求文件全部导入链表
    initgraph(HEIGHT, WIDTH);//绘制需求界面
    Menu_Need(1, 0);//调用第一个需求的第一页
}
void Menu_Need(int op,int n) {//op 表示第几个需求，n 表示第 n+1 页
    cleardevice();//用当前背景色清空绘图设备，并将当前点移至 (0, 0)
    setbkmode(TRANSPARENT);
    loadimage(NULL, _T("0.jpg"), HEIGHT, WIDTH, 1);
    setlinestyle(PS_DASH, 2, NULL, 0);
    setlinecolor(BLACK);
    circle(75, 493, 30);//发布按钮
    circle(173, 493, 30);//退出按钮
    for (int j = 1; j < 5; j++) {
        if (j == op) Node[j].status = 1;//改界面展示第 op 个需求，则将该按钮的状态更改
        else Node[j].status = 0;
        DrawBtn(j);//画出所有按钮
    }
    settextcolor(BLACK);
    settextstyle(30, 0, _T("方正少儿简体"), 0, 0, 0, 0, 0, 0);
    outtextxy(50, 480, _T("发布"));//发布文字
    outtextxy(148, 480, _T("退出"));//退出文字
    setlinestyle(PS_DASHDOT, 2, NULL, 0);
    line(253, 0, 253, WIDTH);
    line(250, 0, 250, WIDTH);//设置分界线
```

```
    int cnt = 5 * n;//到目前为止共有需求数量
    int goucnt = 0;//表示该页的需求需求数量-1
    for (int i = 0; i < 5 ; i++, cnt++) {//展示需求
        if (cnt == usernow.tasknumber(op-1)) break;//到目前为止的需求等于该链表拥有的所有需求则结束展示
        settextcolor(RED);//编号颜色
        number[cnt + 1] = element[(int)((cnt + 1) / 10)] + element[(int)(cnt + 1) % 10];//生成当前页第 i+1 个需求的编号
        CHARTOTCHAR(number[cnt+1].data());//将编号转为 TCHAR 型
        outtextxy(307, 57 + 100 * i, ss);//输出编号
        settextcolor(BLACK);
        setlinecolor(BLACK);
        setlinestyle(PS_SOLID, 2, NULL, 0);
        setfillcolor(RGB(130, 127, 133));
        circle(848, 73 + 100 * i, 25);
        outtextxy(834, 60 + 100 * i, _T("√"));//画出接受需求的按钮
        setlinestyle(PS_DASHDOTDOT, 2, NULL, 0);
        rectangle(365, 48 + 100 * i, 800, 98 + 100 * i);//展示需求的边框
        CHARTOTCHAR(usernow.operation(op-1, 3, " ", " ", cnt, 3).data());//将标题转为 TCHAR 型
        outtextxy(374, 59 + 100 * i, ss);//输出需求的标题
        goucnt = i;//更新该页到目前为止的需求数量-1
    }
    for (int i = 0; i < 3; i++) {//画出上下页和搜索的按钮边框
        setlinecolor(RGB(51,56,75));
        setlinestyle(PS_SOLID, 1, NULL, 0);
        setfillcolor(RGB(130, 127, 133));
        fillrectangle(915, 68 + 160 * i, 955, 158 + 160 * i);
    }
    outtextxy(922, 70, _T("上"));
    outtextxy(922, 98, _T("一"));
    outtextxy(922, 126, _T("页"));
    outtextxy(922, 240, _T("搜"));
    outtextxy(922, 280, _T("索"));
    outtextxy(922, 392, _T("下"));
    outtextxy(922, 420, _T("一"));
    outtextxy(922, 448, _T("页"));
    MOUSEMSG m;
    while (1) {
        m = GetMouseMsg();//获取鼠标信息
        if (m.uMsg == WM_LBUTTONDOWN) {
            for (int i = 1; i < 5; i++) {
                if (m.x >= Node[i].x && m.x <= Node[i].x + 193 && m.y >= Node[i].y - 8 && m.y <= Node[i].y + 55) {//点击到第 i 个需求按钮
                    Menu_Need(i, 0);//转跳至第 i 个需求展示
                    break;
                }
            }
            for (int i = 0; i <= goucnt; i++) {//点击到接受需求按钮
                if (m.x >= 823 && m.x <= 873 && m.y >= 48 + 100 * i && m.y <= 98 + 100 * i) {
                    string a = "这是", b = "发布的消息，是否接受？", c = "发布人的电话为";
                    string d = usernow.operation(op - 1, 3, "", "", n * 5 + i, 1);
                    string e = usernow.operation(op - 1, 3, "", "", n * 5 + i, 2);
                    string f = usernow.operation(op - 1, 3, "", "", n * 5 + i, 3);
                    string g = a + d + b;
                    string h = c + e;
                    CHARTOTCHAR(g.data());
                    int flag = MessageBox(GetForegroundWindow(), ss, L"提示", 1);//提示这是某个人发布的需求
                    if (flag==1) {
                        MessageBox(GetForegroundWindow(), L"接受成功！", L"提示", 0);
                        usernow.operation(op - 1, 2, f, "", n * 5 + i, 2);//删除该需求
                        CHARTOTCHAR(h.data());
                        MessageBox(GetForegroundWindow(), ss, L"提示", 0);//展示改需求发布人的的电话
                        Menu_Need(op, 0);//刷新该界面
                    }
```

```
                    }
               }
               for (int i = 0; i <= goucnt; i++) {
                    if (m.x >= 365 && m.x <= 800 && m.y >= 48 + 100 * i && m.y <= 98 + 100 *
i) {
                         CHARTOTCHAR(usernow.operation(op - 1, 3, "", "", n * 5 + i,
4).data());//点击每个需求展示框
                         MessageBox(GetForegroundWindow(), ss, L"详情", 0);//展示需求的具体内
容
                    }
               }
               if (m.x >= 45 && m.x <= 105 && m.y >= 463 && m.y <= 523) {//点击发布需求按钮
                    Menu_Publish(op);//进入发布函数
                    Menu_Need(op, 0);//刷新界面
               }
               else if (m.x >= 143 && m.x <= 203 && m.y >= 463 && m.y <= 523) {//点击退出按钮
                    usernow.ending();//将当前链表数据读入文件
                    Begin();//回到主界面
               }
               else if (m.x >= 915 && m.x <= 955 && m.y >= 228 && m.y <= 318) {//点击搜索按钮
                    TCHAR temp[LEN];
                    InputBox(temp, LEN, L"请输入要搜索的标题", L"搜索", 0, 0, 0, 1);//输入所
搜索需求的标题
                    if (usernow.operation(op - 1, 4, TCHARTOCHAR(temp), "", 0, 0) == "a") {//
判断是否有该需求
                         MessageBox(GetForegroundWindow(), L"搜索成功", L"提示", 0);
                         string a = usernow.detail(op - 1, TCHARTOCHAR(temp), 1);
                         string b = usernow.detail(op - 1, TCHARTOCHAR(temp), 2);
                         string c = usernow.detail(op - 1, TCHARTOCHAR(temp), 3);
                         string d = usernow.detail(op - 1, TCHARTOCHAR(temp), 4);
                         string e = "发布人姓名:" + a + "\n" + "发布人手机号码:" + b + "\n" +
"消息标题:" + c + "\n" + "消息内容:" + d + "\n";
                         int iLength;//读入需求相关的所有信息
                         TCHAR tmp[] = _T("w");
                         iLength = MultiByteToWideChar(CP_ACP, 0, e.data(), -1, NULL, 0);
                         MultiByteToWideChar(CP_ACP, 0, e.data(), -1, tmp, iLength);//将所有
信息转化为 TCHAR 类型
                         MessageBox(GetForegroundWindow(), tmp, L"具体信息", 0);//展示所有信
息
                         int flag = MessageBox(GetForegroundWindow(), L"是否接受此条消息", L"
提示", 1);
                         if (flag == 1) {
                              usernow.operation(op - 1, 2, TCHARTOCHAR(temp), "", 0, 0);//如果
接受信息则删除该需求
                              Menu_Need(op, 0);//并刷新界面
                         }
                         Menu_Need(op, n);//不接受则不进行操作
                    }
                    else {
                         MessageBox(GetForegroundWindow(), L"该专栏未找到相关消息", L"提示",
0);//搜索不到则进行提示
                    }
               }
               else if (m.x >= 915 && m.x <= 955 && m.y >= 68 && m.y <= 158) {//点击上一页
                    if (n == 0) {//n=0 代表首页
                         MessageBox(GetForegroundWindow(), L"当前已是首页", L"提示", 0);//提
示无上一页
                    }
                    else {
                         Menu_Need(op, n - 1);//否则进入上一页
                    }
               }
               else if (m.x >= 915 && m.x <= 955 && m.y >= 388 && m.y <= 478) {
                    if (cnt == usernow.tasknumber(op - 1)) {//若到该界面的需求已经等于该需求
链表的所有需求
                         MessageBox(GetForegroundWindow(), L"当前已是末页", L"提示", 0);//则
说明已经到达末页
                    }
                    else {
```

```
                    Menu_Need(op, n + 1);//否则进入下一页
                }
            }
        }
    }
}
void Menu_Publish(int op) {//发布第 op 个类型的需求
    bool flag = InputBox(diversity, 2, L"请选择你想发布的内容\n1-资料分享;2-快递代拿;3-商品交易;4-交友平台", L"发布", 0, 0, 0, 0);
    if (!flag) {//点击取消或关闭对话框
        Menu_Need(op, 0);
        return;
    }
    while (TCHARTOCHAR(diversity)[0] < '1' || TCHARTOCHAR(diversity)[0] > '4') {//输入非规定数字
        flag = InputBox(diversity, 2, L"请输入 1-4 内的数字", L"提示", 0, 0, 0, 1);
        if (!flag) {//点击取消或关闭对话框
            Menu_Need(op, 0);
            return;
        }
    }
    InputBox(title, LEN, L"请输入发布内容的标题", L"发布", 0, 0, 0, 1);
    InputBox(content, 50, L"输入想发布的内容", L"发布", 0, 0, 0, 1);
    usernow.operation((TCHARTOCHAR(diversity)[0] - 49), 1, TCHARTOCHAR(title), TCHARTOCHAR(content), 0, 0);//发布需求
    MessageBox(GetForegroundWindow(), L"发布成功", L"提示", 0);
}
int main() {
    mciSendString(L"open bgm.mp3 alias start", 0, 0, 0);
    mciSendString(L"play start repeat", NULL, 0, NULL);
    Begin();
}
```

person 类：

```
#ifndef PERSON_H
#define PERSON_H

#include<iostream>
#include<algorithm>
#include<cstring>
#include<fstream>
using namespace std;

class person{
    public:
        person() = default;
        person(char*);
        person(char*,char*);//登录初始化
        person(char*,char*,char*,char*);//注册初始化
        string geta();//返回账号
        string getpa();//返回密码
        string getn();//返回姓名
        string getph();//返回电话号
        void changea(string);//初始化写入账号
        void changepa(string);//初始化写入密码
        void changen(string);//初始化写入姓名
        void changeph(string);//初始化写入电话号
    private:
        string account, password, name, phone;
};

#endif

#include "person.h"

person::person(char* a) {
    account = a;
}

person::person(char* a,char* b){
```

```
	account = a;
	password = b;
}

person::person(char* a,char* b,char* c,char* d){
	account = a;
	password = b;
	name = c;
	phone = d;
}

string person::geta(){
	return account;
}

string person::getpa(){
	return password;
}

string person::getn(){
	return name;
}

string person::getph(){
	return phone;
}

void person::changea(string a){
	account = a;
}

void person::changepa(string a){
	password = a;
}

void person::changen(string a){
	name = a;
}

void person::changeph(string a){
	phone = a;
}
```

login 类：

```
#ifndef LOGIN_H
#define LOGIN_H

#include<iostream>
#include<algorithm>
#include<cstring>
#include<fstream>
#include "person.h"
#include "events.h"
using namespace std;

class login:virtual public person{
	public:
		login() = default;
		login(char*);
		login(char*,char*);//登录初始化
		login(char*,char*,char*,char*);//注册初始化
		bool verify1();//登录验证密码是否匹配
		bool verify2();//注册验证账号是否存在
		void regist();//注册
		void in();//将账号文件中的姓名和电话赋给对象
		void init();//读取 file 中所有需求
		void ending();//将链表中所有数据更新到文件夹中
		int tasknumber(int);//获得链表中现有的需求数量
		string detail(int, string, int);//获得需求的具体信息
		string operation(int, int, string, string, int, int);//对需求进行各种操作
```

```
	private:
		events eve[4];
};

#endif

#include "login.h"
#include<graphics.h>
using namespace std;

const string adr="account\\";
const int mod1 = 247;
const int mod2 = 233;

login::login(char*a):person(a){}

login::login(char* a,char* b):person(a,b){}

login::login(char* a,char* b,char* c,char* d):person(a,b,c,d){}

bool login::verify1(){
	string sadr=adr+geta()+".txt";
	string spassword="";
	ifstream in(sadr.data(),ios::in|ios::binary);
	if(!in){
		in.close();
		return false;
	}
	in>>spassword;
	in.close();
	string s1 = getpa();
	for (int i = 1; i < s1.length(); i++) {
		s1[i] = (s1[i - 1] * mod1 + s1[i]) % mod2;
	}
	if(s1==spassword){
		return true;
	}
	else return false;
}

bool login::verify2() {
	string sadr = adr + geta() + ".txt";
	ifstream ins(sadr.data(), ios::in);
	if (ins) {
		return 1;
	}
	return 0;
}

void login::regist(){
	string sadr=adr+geta()+".txt";
	ofstream out(sadr.data(),ios::out|ios::trunc|ios::binary);
	string s1 = getpa();
	for (int i = 1; i < s1.length(); i++) {
		s1[i] = (s1[i - 1] * mod1 + s1[i]) % mod2;
	}
	out<<s1 <<endl
	   <<getn()   <<endl
	   <<getph() <<endl;
	out.close();
}

void login::in() {
	string sadr = adr + geta() + ".txt";
	ifstream ins(sadr.data(), ios::in);
	string temp1, temp2, temp3;
	ins >> temp1 >> temp2 >> temp3;
	ins.close();
	changen(temp2);
```

```
	changeph(temp3);
}

void login::init() {
	for (int i = 0; i < 4; i++) {
		eve[i].indata(i);
	}
}

void login::ending() {
	for (int i = 0; i < 4; i++) {
		eve[i].outdata(i);
	}
}

string login::operation(int diversity, int oper, string title, string content,int i,int j) {//i 表示第几个需求，j 表示需要第几个数据
		if (oper == 1) {//发布需求
			eve[diversity].add(getn(), getph(), title, content);
			return "";
		}
		else if (oper == 2) {//接受需求
			eve[diversity].del(title);
			return "";
		}
		else if (oper == 3) {//展示需求
			return eve[diversity].display(i, j);
		}
		else if (oper == 4) {//搜索需求
			if (eve[diversity].search(title)) return "a";
			else return"b";
		}
}

int login::tasknumber(int i) {
	return eve[i].getcnt();
}

string login::detail(int diversity, string title,int cho) {
	return eve[diversity].detail(title, cho);
}
```

event 类：

```
#ifndef events_H
#define events_H

#include<iostream>
#include<cstring>
#include<algorithm>
#include<fstream>
#include "demand.h"
using namespace std;

class events{
	public:
		events();
		~events();
		void set(demand);//将需求放至链表
		void add(string,string,string,string);//封装新需求
		void del(string);//接受需求后从链表删除
		bool search(string);//寻找需求
		void indata(int);//将某条链表数据存入文件夹
		void outdata(int);//将某个文件夹数据读入链表
		int getcnt();//得到链表中的需求数量
		string display(int, int);//将需求展示到界面
		string detail(string,int);//获得某个需求的其他信息
	private:
		int cnt;
		demand *head,*tail;
};
```

```
#endif

#include "events.h"

const string adr[4]={"file\\knowledge.txt",
                     "file\\delivery.txt" ,
                     "file\\trade.txt"       ,
                     "file\\friend.txt"     };

events::events(){
    cnt=0;
    head=NULL;
    tail=NULL;
}

events::~events(){
    demand *p=head;
    if (!p)return;
    demand *p1=p->next;
    do{
        delete p;
        p=p1;
        if(p1!=NULL)p1=p1->next;
    }while(p1!=NULL);
    if(p)delete p;
}

void events::set(demand ap){
    cnt++;
    demand *p=new demand(ap);
    if(tail)tail->next=p;
    p->last=tail;
    tail=p;
    tail->next=NULL;
    if(cnt==1)head=tail;
}

void events::add(string name,string phone,string title,string content){
    demand p;
    p.changen(name);
    p.changep(phone);
    p.changet(title);
    p.changec(content);
    set(p);
}

bool events::search(string title){
    if(!cnt){
        return 0;
    }
    demand *p=head;
    do{
        if(p->gettitle()==title){
            return 1;
        }
        p=p->next;

    }while(p!=NULL);
    return 0;
}

void events::del(string title){
    demand* p = head;
    if (!p) return;
    do {
        if (p->gettitle() == title) {
            cnt--;
            demand* p1 = p->last;
```

```
                demand* p2 = p->next;
                if (p1) {
                        p1->next = p2;
                        if (p2)p2->last = p1;
                        else tail = p1;
                }
                else {
                        head = p->next;
                        if (p2)p2->last = NULL;
                }
                if (p)delete p;
                return;
            }
            p = p->next;
        } while (p != NULL);
        return;
}

void events::indata(int k){
        ifstream ins(adr[k].data(),ios::in|ios::binary);
        while(!ins.eof()&&ins){
                demand a;
                ins>>a;
                if (a.getname() != "") {
                        set(a);
                }
        }
        ins.close();
}

string events::display(int cnt, int j) {
        demand *p=head;
        for (int i = 1; i <= cnt; i++) {
                        p = p->next;
        }
        if (j == 1) return p->getname();
        if (j == 2) return p->getphone();
        if (j == 3) return p->gettitle();
        if (j == 4) return p->getcontent();
}

void events::outdata(int k){
        ofstream out(adr[k].data(),ios::out|ios::trunc|ios::binary);
        demand *p=head;
        while(p){
                out<<*p;
                p=p->next;
        }
        out.close();
}

int events::getcnt() {
        return cnt;
}

string events::detail(string title, int cho) {
        demand* p = head;
        do {
                if (p->gettitle() == title) {
                        if (cho == 1) return p->getname();
                        else if (cho == 2) return p->getphone();
                        else if (cho == 3) return p->gettitle();
                        else if (cho == 4) return p->getcontent();
                }
                p = p->next;
        } while (p != NULL);
        return NULL;
}
```

demand 类：

```
#ifndef DEMAND_H
#define DEMAND_H

#include<iostream>
#include<algorithm>
#include<cstring>
#include<fstream>
using namespace std;

class demand {
public:
	demand() {}
	demand(string, string);
	demand* last=NULL, * next=NULL;//链表指针
	string getname();
	string getphone();
	string gettitle();
	string getcontent();
	void changen(string);
	void changep(string);
	void changet(string);
	void changec(string);
	friend istream& operator>>(istream&, demand&);//重载输入运算符
	friend ostream& operator<<(ostream&, demand&);//重载输出运算符
private:
	string name, phone, title, content;
};

#endif

#include "demand.h"

demand::demand(string a, string b) :name(a), phone(b) {
	last = next = NULL;
}

string demand::getname() {
	return name;
}

string demand::getphone() {
	return phone;
}

string demand::gettitle() {
	return title;
}

string demand::getcontent() {
	return content;
}

void demand::changen(string s) {
	name = s;
}

void demand::changep(string s) {
	phone = s;
}

void demand::changet(string s) {
	title = s;
}

void demand::changec(string s) {
	content = s;
}

istream& operator>>(istream& in, demand& a) {
```

```
	in >> a.name >> a.phone >> a.title>> a.content;
	return in;
}

ostream& operator<<(ostream& os, demand& a) {
	os << a.name << endl
		<< a.phone << endl
		<< a.title << endl
		<< a.content << endl;
	return os;
}
```