人工智能与自动化学院
School of Artificial Intelligence and Automation,HUST

C 语言课程设计

验收报告

某市新冠疫情通报系统

刘子恒 [redacted]

韩海若 [redacted]

指导老师：周纯杰、何顶新、彭刚、周凯波、桑农、左峥嵘、高常鑫、汪国有、陈忠

上交时间：2020年10月31日星期六

目录

1. 前言

编写背景

2020 年，新型冠状病毒(Covid-19)引发了全球大流行疫情，截至 2020 年 9 月 14 日，全球已有 188 个国家和地区累计报告逾 2,900.6 万名确诊病例，其中至少约 1,962.5 万人已康复、至少约 92.4 万人死亡。疫情扩散对全球航空、旅游、娱乐、体育、石油市场、金融市场等方面造成巨大影响。

该传染病传染性极强，目前尚无针对新型冠状病毒的预防疫苗及治疗方法。由于该病传染性强且潜伏期为 14-28 天，初步判断外来人员的被感染风险显得极为重要。目前各国都已建立了一套完备的自下而上的疫情通报系统：由基层社区、医院等逐层上报，汇总到数据中心进行处理，计算出新增，累计，死亡与治愈等病例数据；结合大数据技术，制作出涵盖各级行政区的“疫情地图”，并进行可视化处理，从而方便公众及时了解疫情信息。目前较为典型的有中国疾控中心疫情分布系统(http://2019ncov.chinacdc.cn/2019-nCoV/)、WHO 疫情地图(https://covid19.who.int/)、约翰·霍普金斯大学疫情地图(https://coronavirus.jhu.edu/map.html)。

笔者设计了这个新冠疫情通报系统，旨在使居民能够了解本地区疫情状况，以便能够获取最新疫情动态、判断被感染风险、及时储存物资。对于医疗工作者而言，他们可以利用该系统向上通报所在单位当日疫情新增、治愈与死亡人数。对于普通用户而言，可以了解到该地区新增、累计病例，并能从可视化地图上了解该地区风险等级，保护自己与他人的健康。

编写目的

该项报告对于整个“某市新冠疫情通报系统”进行了全面的用户需求与功能的分析。包括可行性分析、需求分析、系统功能设计、代码实现、集成测试等等。本报告明确了本软件系统架构设计，软件结构与数据结构设计，各模块之间的接口和调用，系统界面设计，系统功能设计（函数罗列）。同时，该项报告也明确了两位开发者的分工。

参考资料

1. 王士元. C 高级实用程序设计. 北京: 清华大学出版社. 1996 年
2. 周纯杰，何顶新等. 程序设计与应用（用 C/C++编程）. 北京: 机械工业出版社.2008 年
3. [美] Prata. C Primer Plus（第六版）北京：人民邮电出版社. 2016 年
4. 严蔚敏，吴伟民编著. 数据结构（C 语言版）北京：清华大学出版社. 2018 年

2. 需求分析

2.1 目标与受众

1. 本软件旨在模拟疫情通报系统基本使用（对用户而言）和管理（对于管理员而言，例如相关疾控部门）过程，将疫情通报系统以模拟化的方式展现在使用者面前，起到普及的作用；
2. 通过软件的开发过程，了解掌握软件开发的方法、技巧和基本流程，培养团队合作的能力；
3. 本软件面向所有人群，软件设计独立，拥有友好的图形化界面，简洁易懂的交互系统，可以较好地实现对疫情通报系统的模拟。
4. 本软件面向的人群大致有：
   a) 想要了解疫情通报系统的人群；
   b) 操作机维护人员：具有良好的 C 语言编程技术，能够充分了解软件模块，并且熟练掌握软件的数据结构。

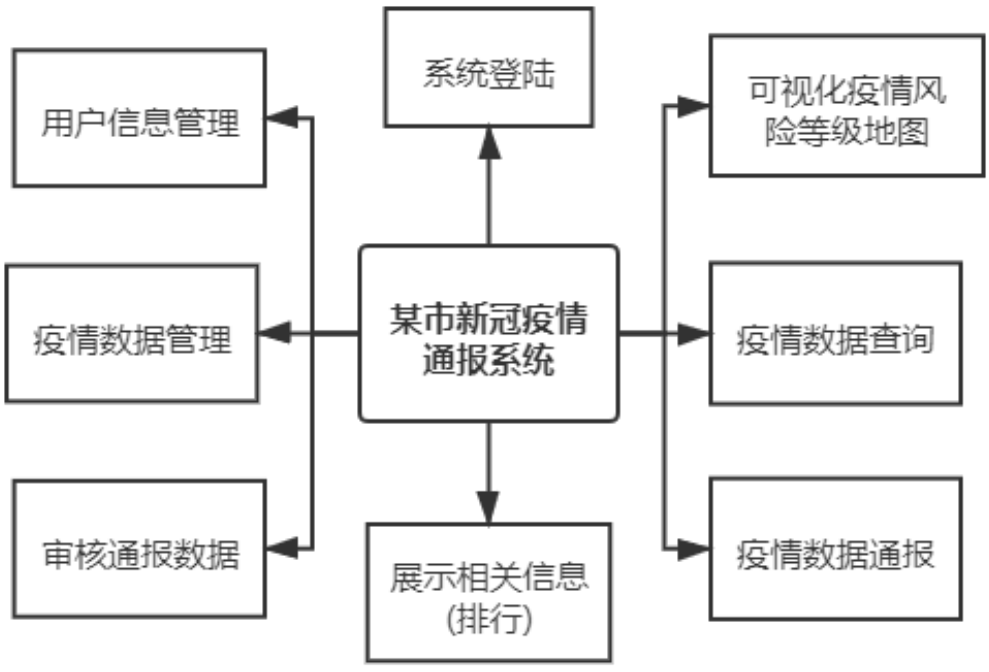
2.2 假定和约束

1. 假定疫情发生的起始日期为 2020 年 1 月 1 日，截止至今日；
2. 由于软件图形化功能有限，因此将小区个数设为 8 个，社区、区个数设为 4 个、2 个。
3. 时间方面的限制：软件开发的时间有限，对软件功能的实现有所限制。
4. 由于所有数据均以“小区”为底层进行逐层上报，因此对于疫情数据的通报及修改均只能处理“小区”对应的数据内容。

2.3 功能需求

对于一个简易的新冠疫情通报系统，需要实现以下几种功能：

1. **系统登录**：用户可输入用户名、用户密码和身份证号进行登陆；
2. **可视化风险等级地图**：将该地疫情风险信息转换为不同颜色在地图上显示；用户可点击相关区域并查看选中区域名；
3. **疫情数据查询**：查询当日或累计疫情数据信息；系统读入用户在地图中选中的地区并显示相关数据；
4. **通报当日数据**：用户上传当日或其他日期的疫情数据信息；
5. **展示疫情相关信息**：展示各地疫情数据排行，包括总确诊、总治愈和总死亡人数；
6. **用户信息管理**：查看用户信息，冻结或解冻用户；
7. **疫情数据管理**：选择日期和地点并修改疫情信息；
8. **审核通报数据**：选择是否采纳用户通报的疫情数据。



3. 总体设计

3.1 目标功能

该新冠疫情通报系统可以实现疫情数据通报查询软件的基本功能。在进入欢迎界面后，系统用户可以选择管理员、用户的登录、注册等操作。

用户可以了解到市级，区级，社区，小区不同级别的疫情数据信息。疫情数据分为当日（累计）新增、当日（累计）治愈、当日（累计）死亡，所有数据都由最基本的“小区”一级逐层上报，并在高一级汇总。用户可以查看疫情数据的曲线图，从而了解到一段时间内的疫情变化趋势；可以查询某日的疫情数据；可以查看该地所有区域疫情数据的排行。该系统还设计了风险等级可视化界面，用户可以直观的了解到截至当日各区域的确诊人数对应的等级，分为高风险、中高风险、中风险与低风险。在数据通报界面，用户还可以对当日该地区疫情状况进行通报。并由管理员审核后发布。

管理员可以使用该系统对后台疫情数据进行修改；并且可以管理注册用户；还可以对用户通报的数据进行审核，成功后将存入数据库。

3.2 编写规范

1. **命名规范：**

变量命名，涉及用户以及疫情数据信息的，应该尽量用英文表达其准确定义。其他类型变量名应给出详细注释以说明其主要功能。函数命名应该用英文表达其确定含义文件命名都用小写，并且表达出该文件所包含函数的主要功能。涉及数据结构的命名应参考《数据结构（C 语言版）》，并进行适当修改。

2. **注释：**

函数功能都要在函数原型后注明；部分令测试者比较难以理解的算法和流程应该给出相应的注释。

4. 运行环境和配置

4.1 硬件接口：

1、硬件接口

处理器：Intel Pentium 166 MX 或以上

硬盘：空间 500MB 以上

屏幕适配器：SVGA 接口

系统运行内存：要求 32MB 以上

4.2 软件接口：

开发软件工具：Borland C 3.1

文字编辑工具：Notepad++；DevC++

数据库：文本存储（.txt）

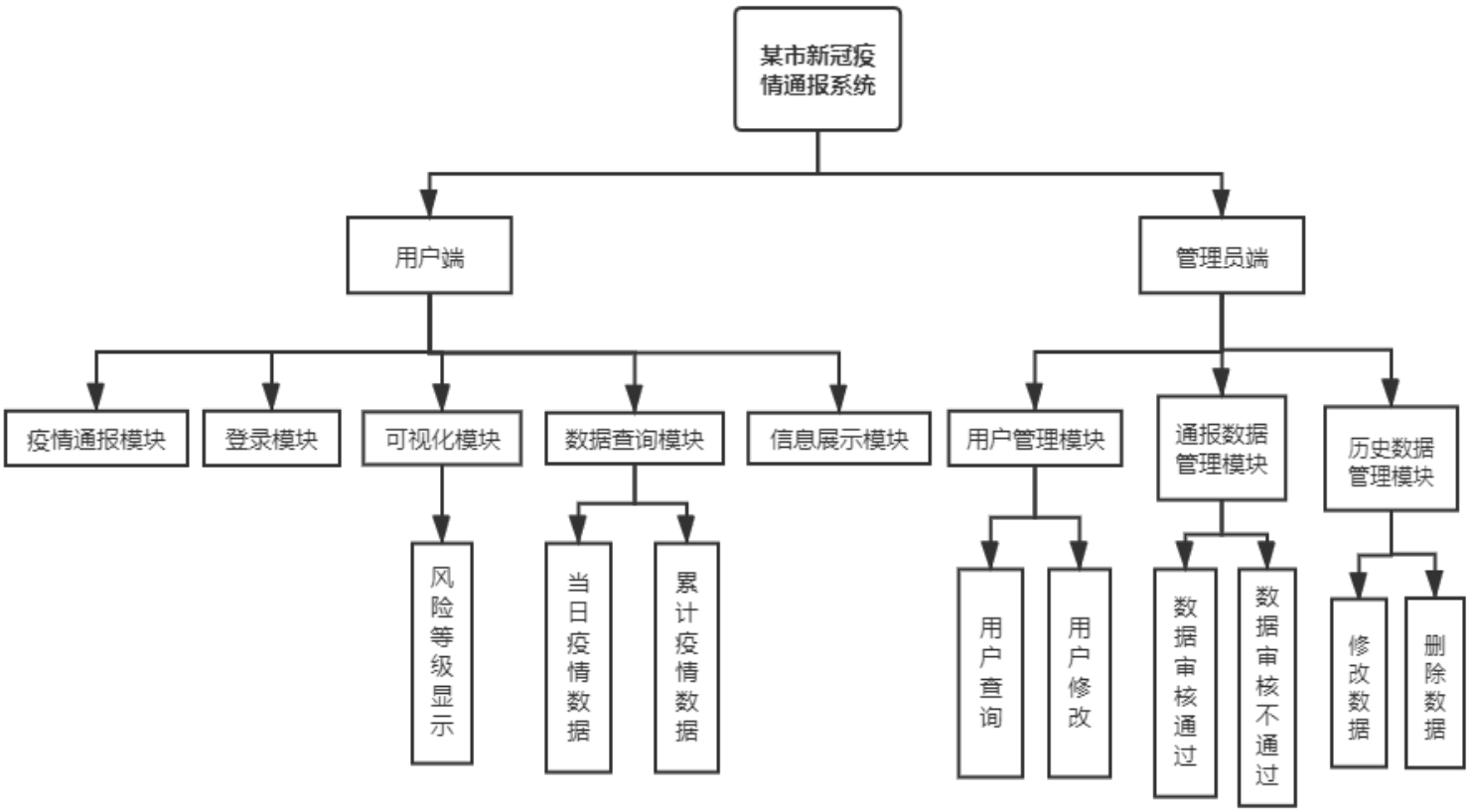
操作系统：DOS WINDOWS 9X/XP/WINDOWS 7/WINDOWS 10

4.3 人机交互：

该系统通过鼠标与键盘直接进行控制。用户将鼠标移至需要操作的功能区进行点击，同时通过键盘来完成登陆、注册的输入功能。操作完毕后在菜单点击相应位置退出系统。通过中断技术与相应函数来获取鼠标的位置与键盘的输入功能。

5. 系统设计

5.1 系统架构设计：



5.2 软件结构设计

登录模块：

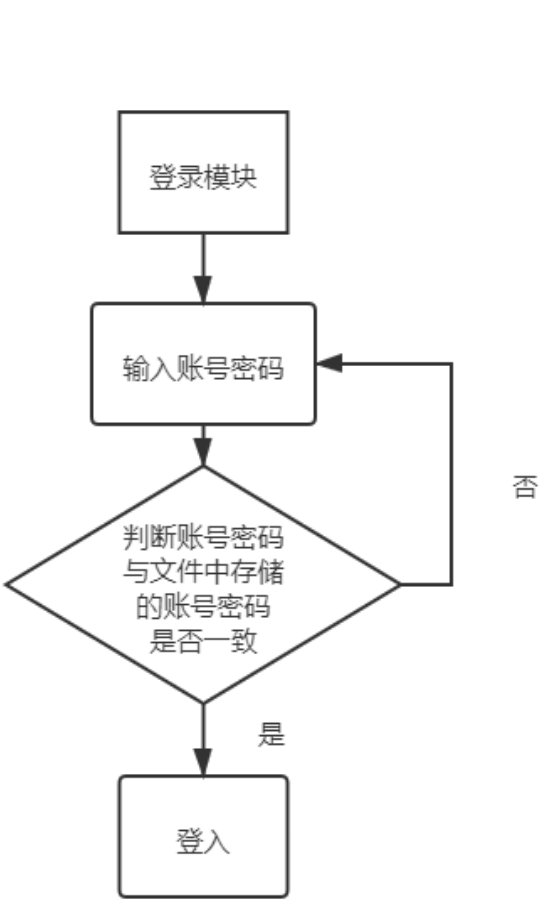
1. 用户输入账号密码；

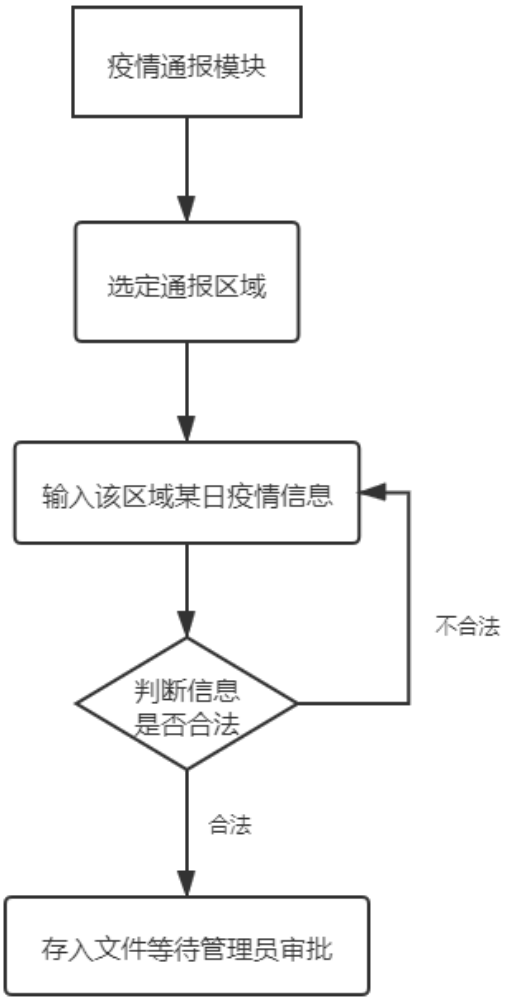
2. 系统判断账号密码与文件中储存的是否一致：一致则登录，不一致则提示重新输入。

疫情通报模块：

1. 输入通报时间；输入当日新增确诊病例；输入当日死亡病例；输入当日治愈病例；

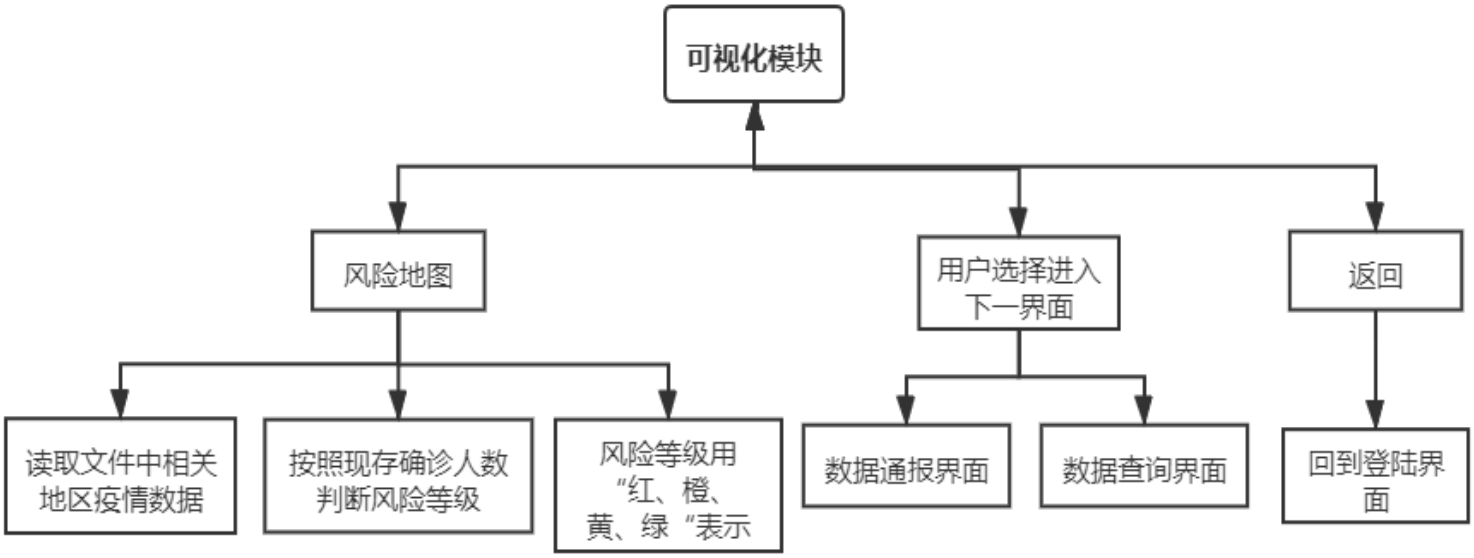
2. 检查输入信息是否合法，将信息存入相应文件；不合法则提示重新输入信息。





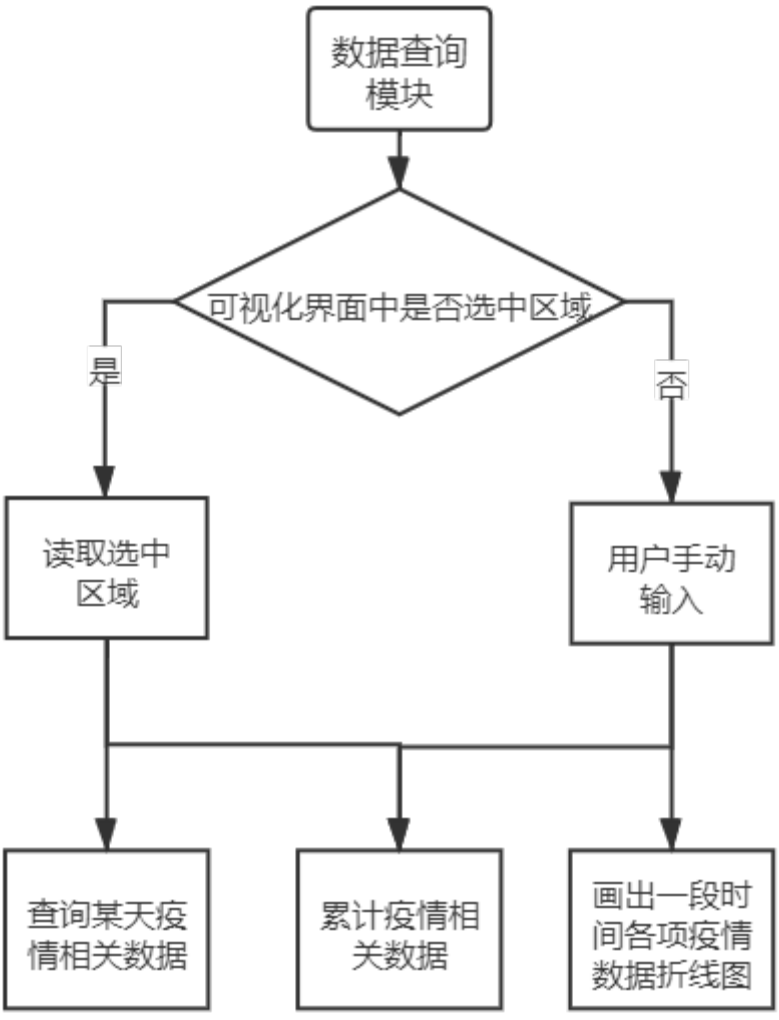
可视化模块（风险等级可视化显示）：

1. 风险等级由该地区现存确诊病例决定；
2. 进入界面后，显示各地区风险等级：高、中、低对应红、橙、黄、绿；
3. 点击相关区域，显示该地名称，并将用户的选择传入数据通报或数据查询界面。



数据查询模块：

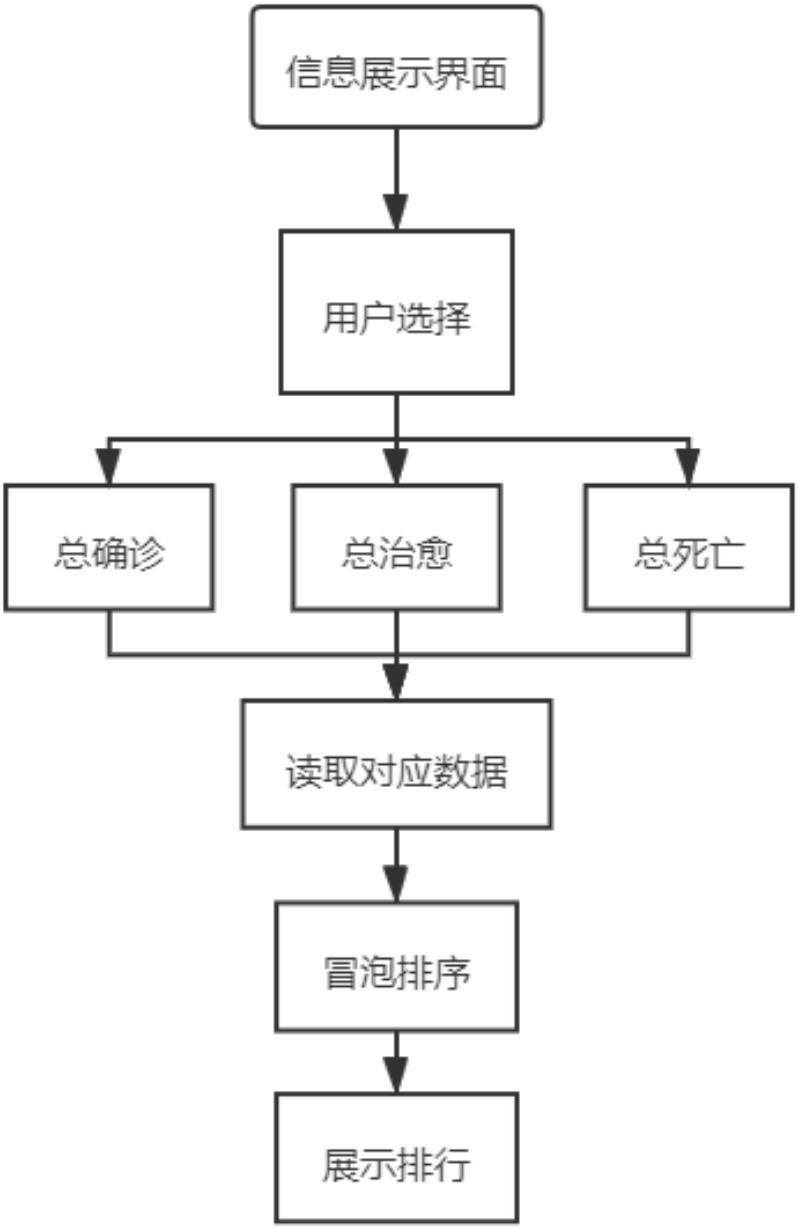
1. 用户输入要查询地区和时间；
2. 查询该日和累计疫情相关数据
3. 画出一段时间各项疫情数据折线图



信息展示模块：

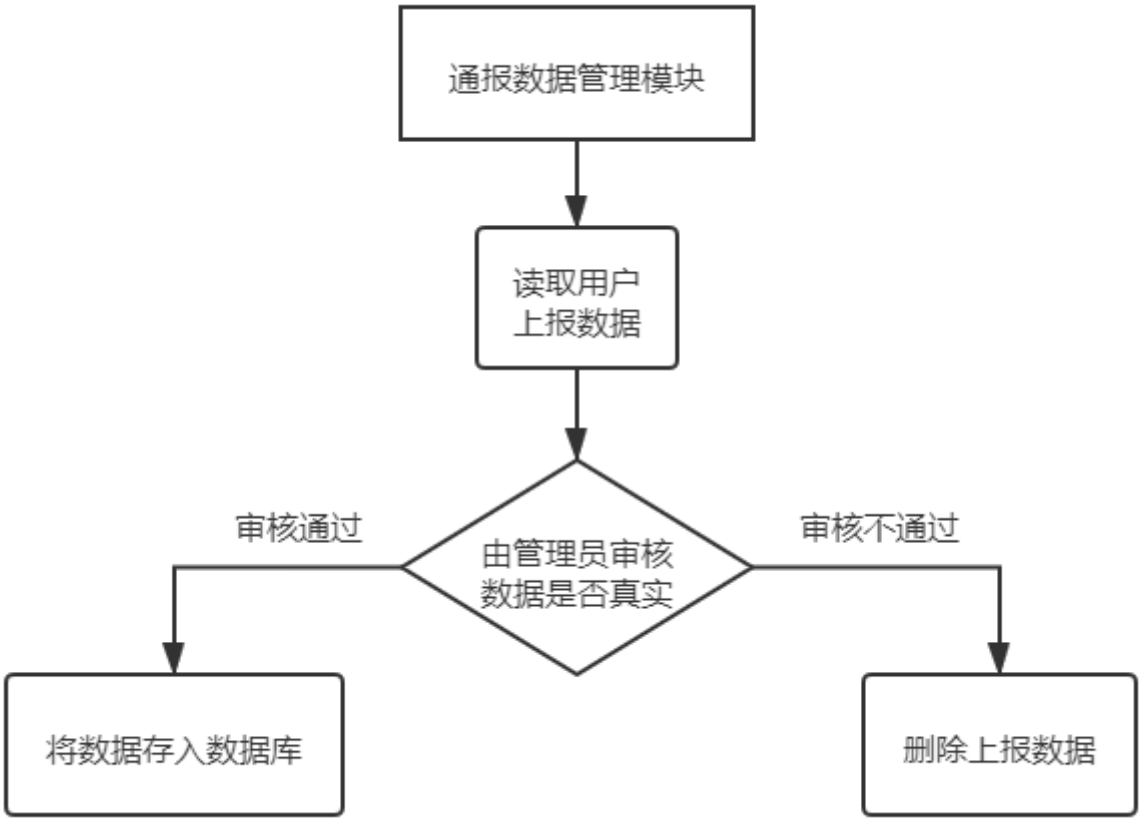
1. 在界面上显示各地区“总确诊”的数据排行；

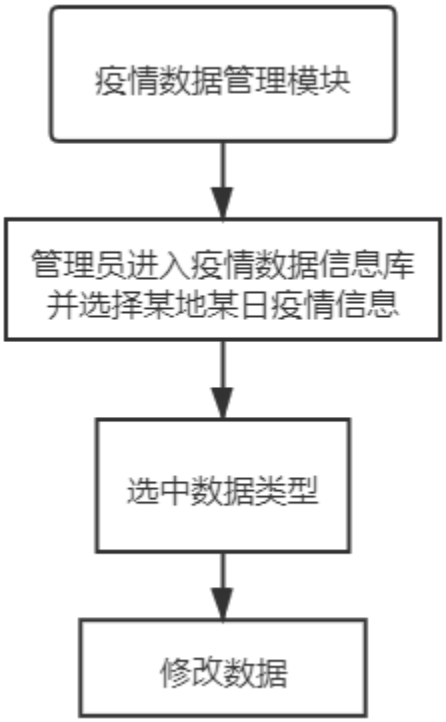
2. 用户还可选择“总治愈”、“总死亡”人数并查看相应地区排行。

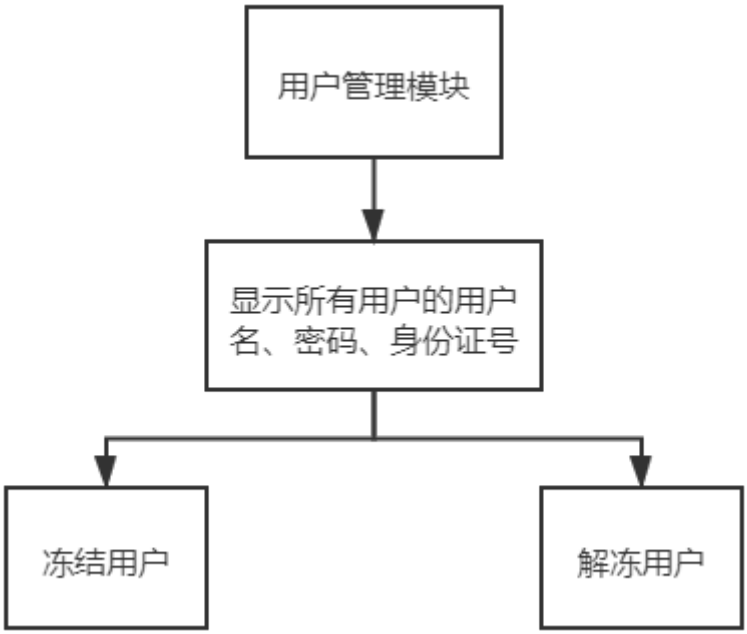


管理员：

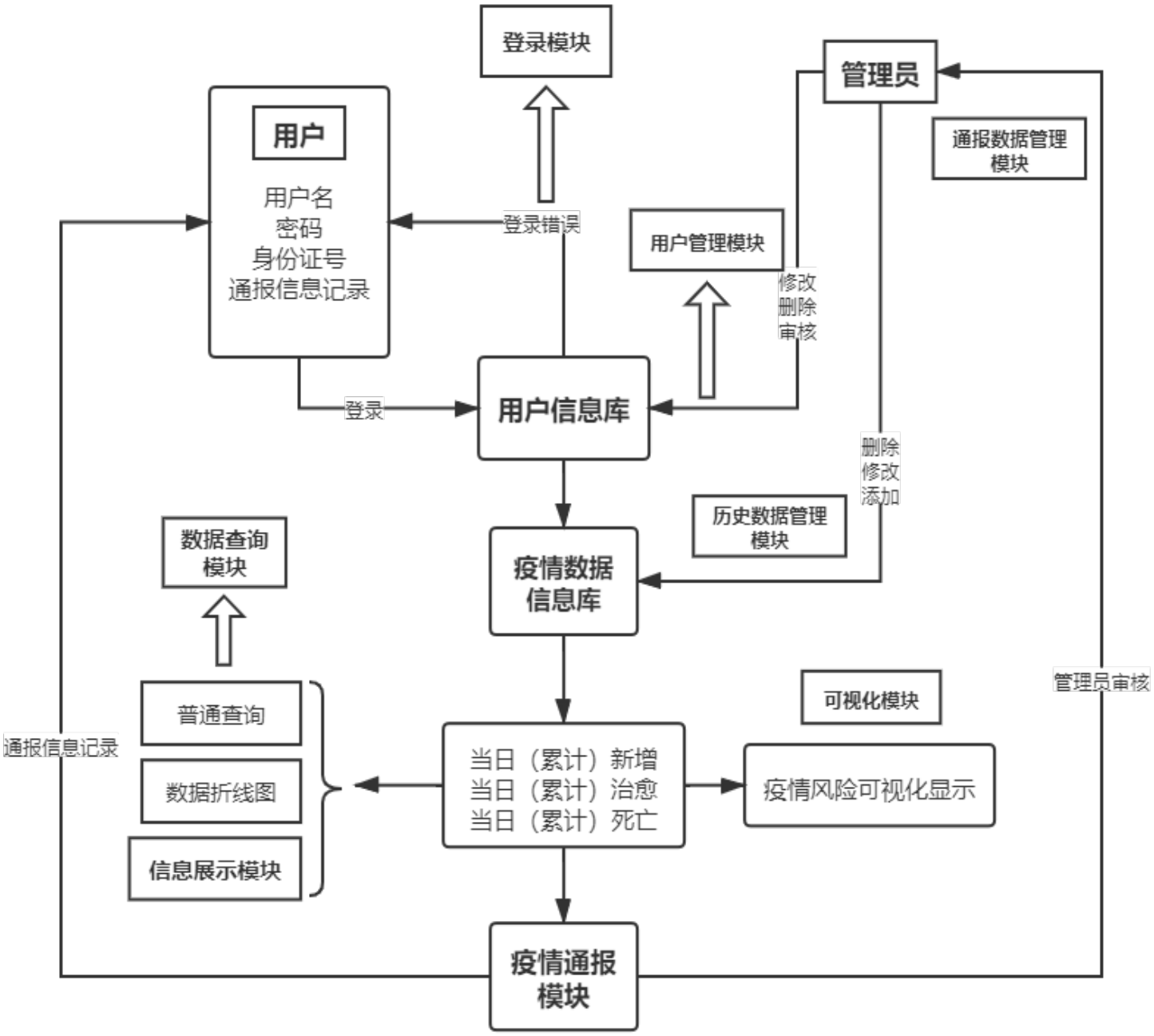
1. 管理已注册用户：查看用户信息、冻结或解冻用户；
2. 查看并审核用户通报的数据
3. 修改疫情数据




疫情数据管理模块
管理员进入疫情数据信息库
并选择某地某日疫情信息
选中数据类型
修改数据


用户管理模块
显示所有用户的用户
名、密码、身份证号
冻结用户
解冻用户

5.2 各模块的调用与接口设计



6. 界面设计

欢迎界面：

欢迎界面中采用了绿色背景，为用户营造一种健康、卫生的感觉，有助于产生积极的情绪；黄色标题给则给用户带来温暖的感觉。考虑到本软件用户（医疗工作者、普通居民等）的计算机操作水平有差异，故本欢迎界面力求简洁，方便用户使用。该界面可以跳转至登陆界面

登录界面：

账号密码输入采用简洁而明显的输入框，在输入错误时下方还会浮现出提示字样。用户可以选择登录、注册或忘记密码等操作。

用户注册界面：

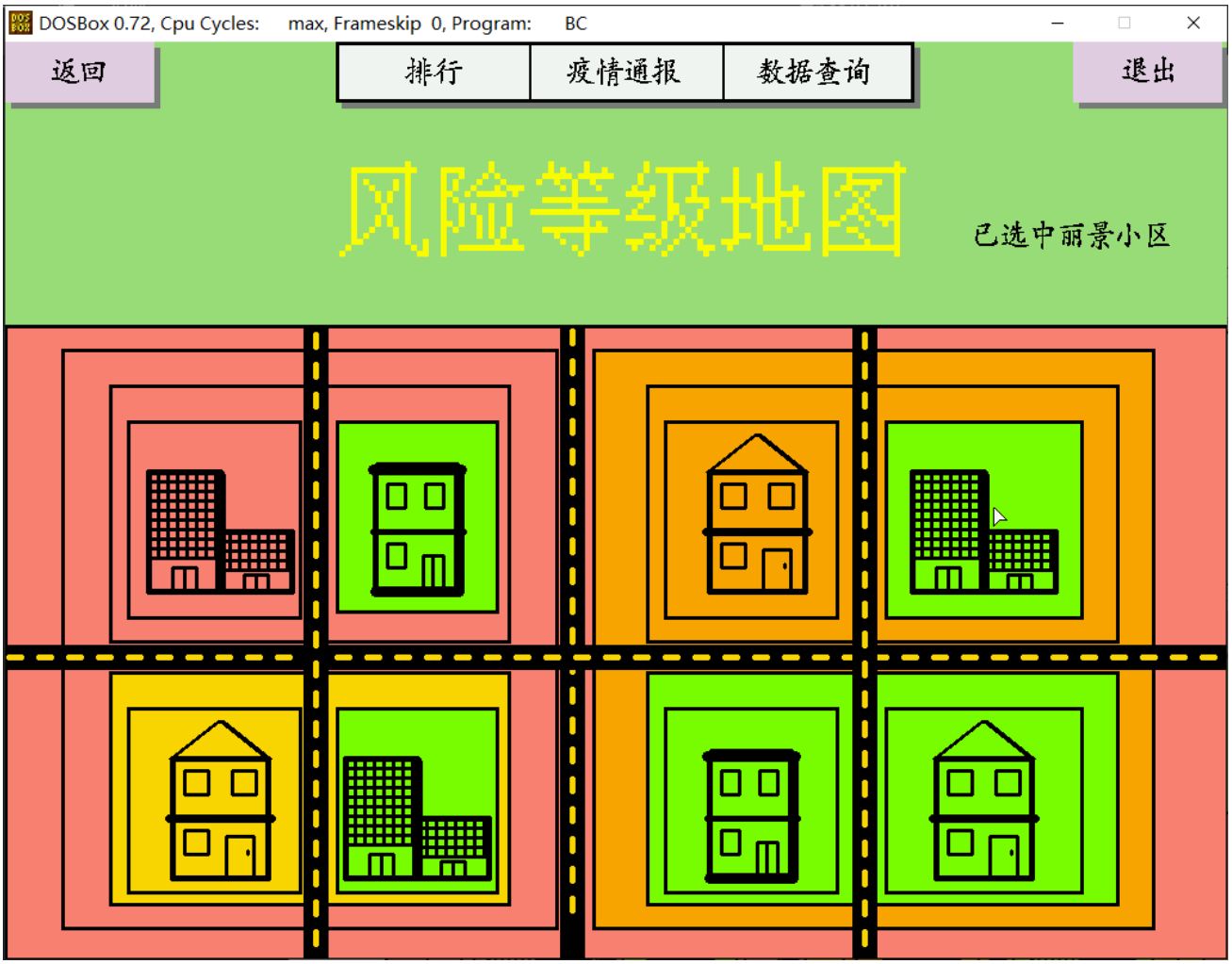
用户选择“注册”后则跳转至注册界面。。注册界面也采用简洁而明显的输入框，方便用户输入。输入时还会自动判断位数是否合法，防止密码过于简单或身份证错误。

忘记密码界面

用户选择“忘记密码”后则跳转至忘记密码界面。在此界面用户可以重设密码恢复登录状态。界面采用了网络上主流“忘记密码”界面，包括重设密码与确认密码，防止用户误输入。

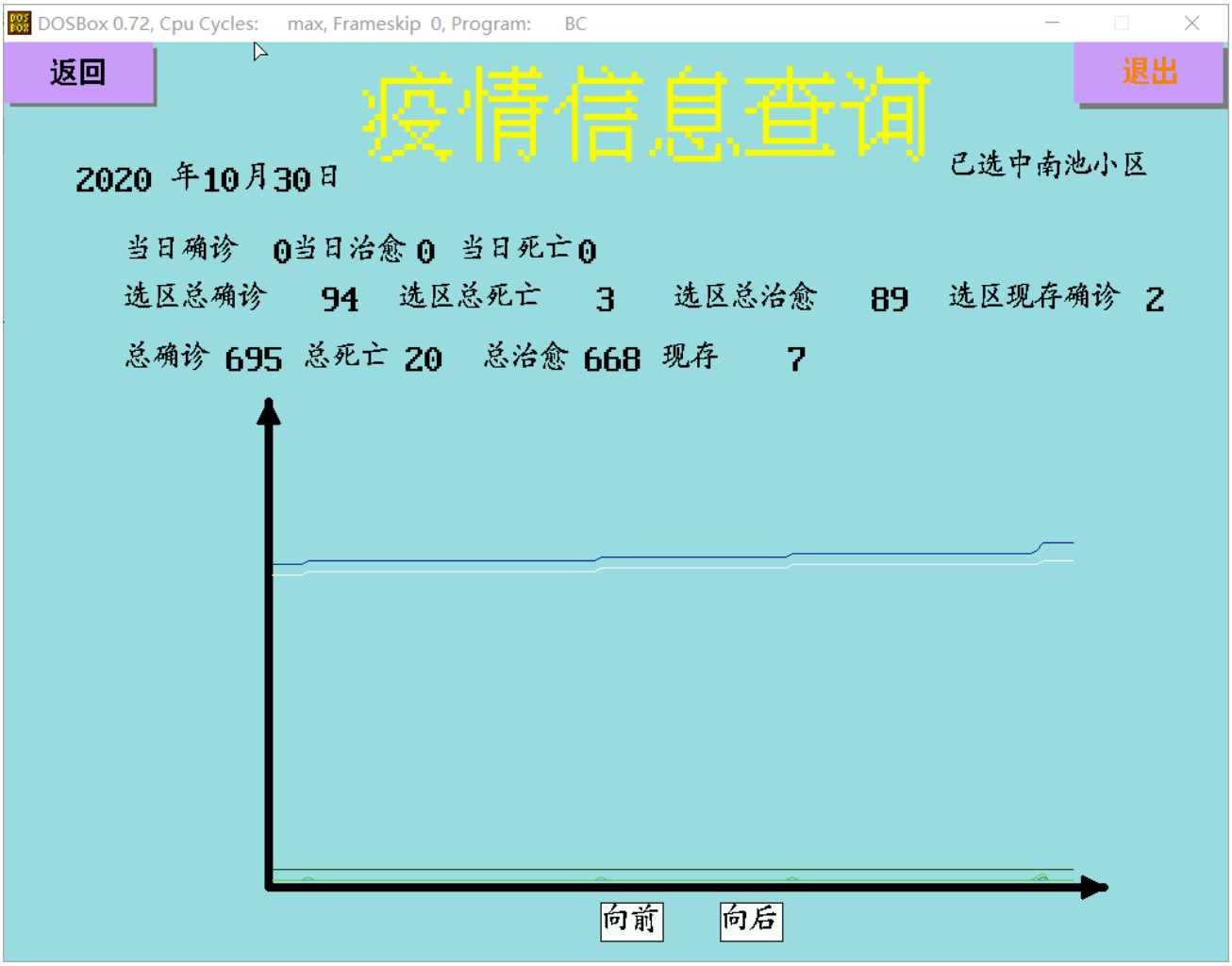
可视化界面：

可视化界面的背景采用了前几页面的主题绿色，用于强调下方的可视化区域，且更为亲切。上方设置跳转栏方便用户选择进入其他界面。下方采用了小区、社区、区、市分级表示。颜色采用对比度较高且普遍适用的绿、黄、橙、红作为风险等级标志。给用户较为醒目的观感，一目了然。用户在点击相关区域后还会显示区域名。



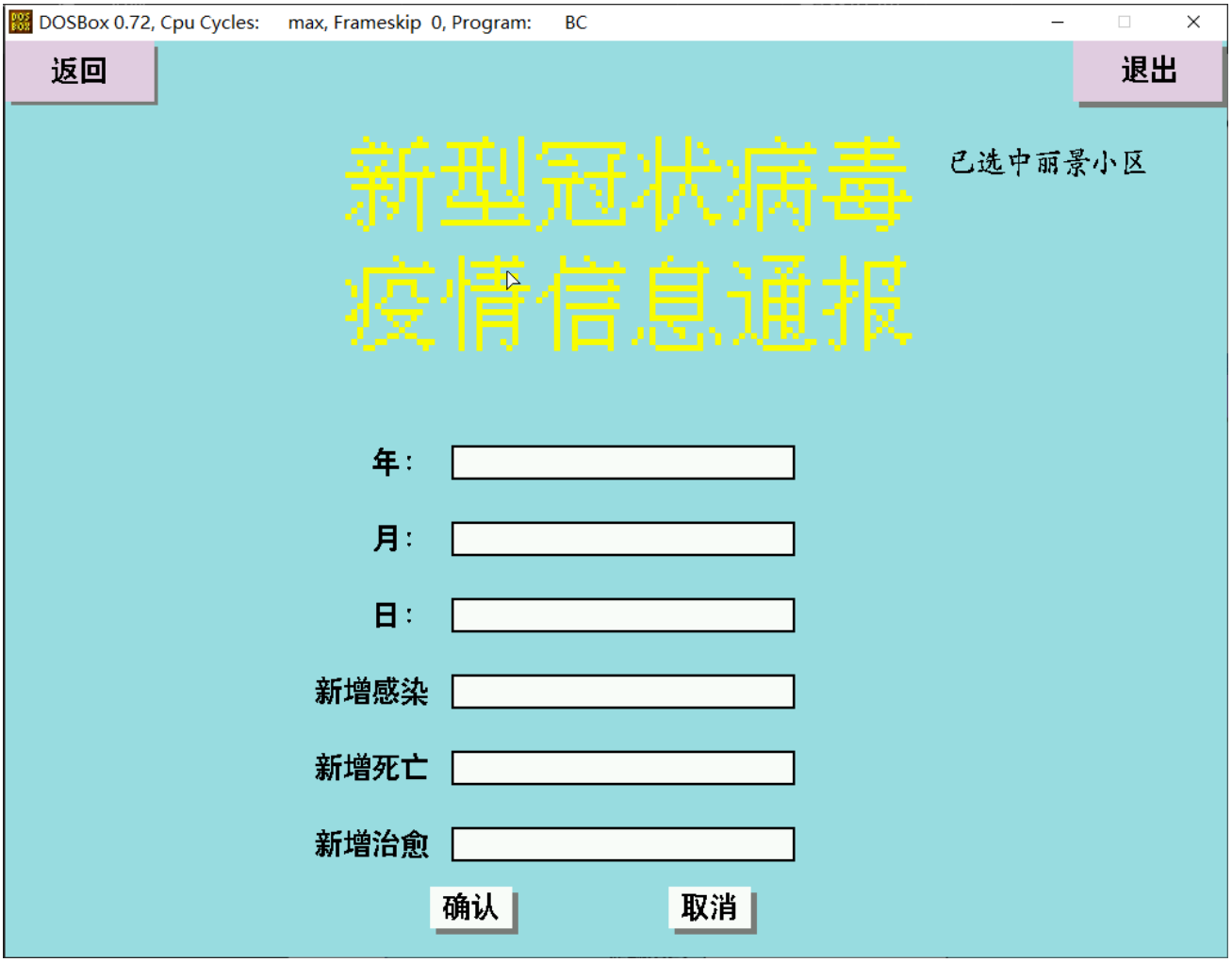
数据查询界面：

数据查询界面右侧显示用户选择的区域；下方分种类显示各疫情相关数据，使用户能够便捷准确的找到所选区域的相关信息。还显示了数据所对应的曲线图，方便用户了解疫情趋势。



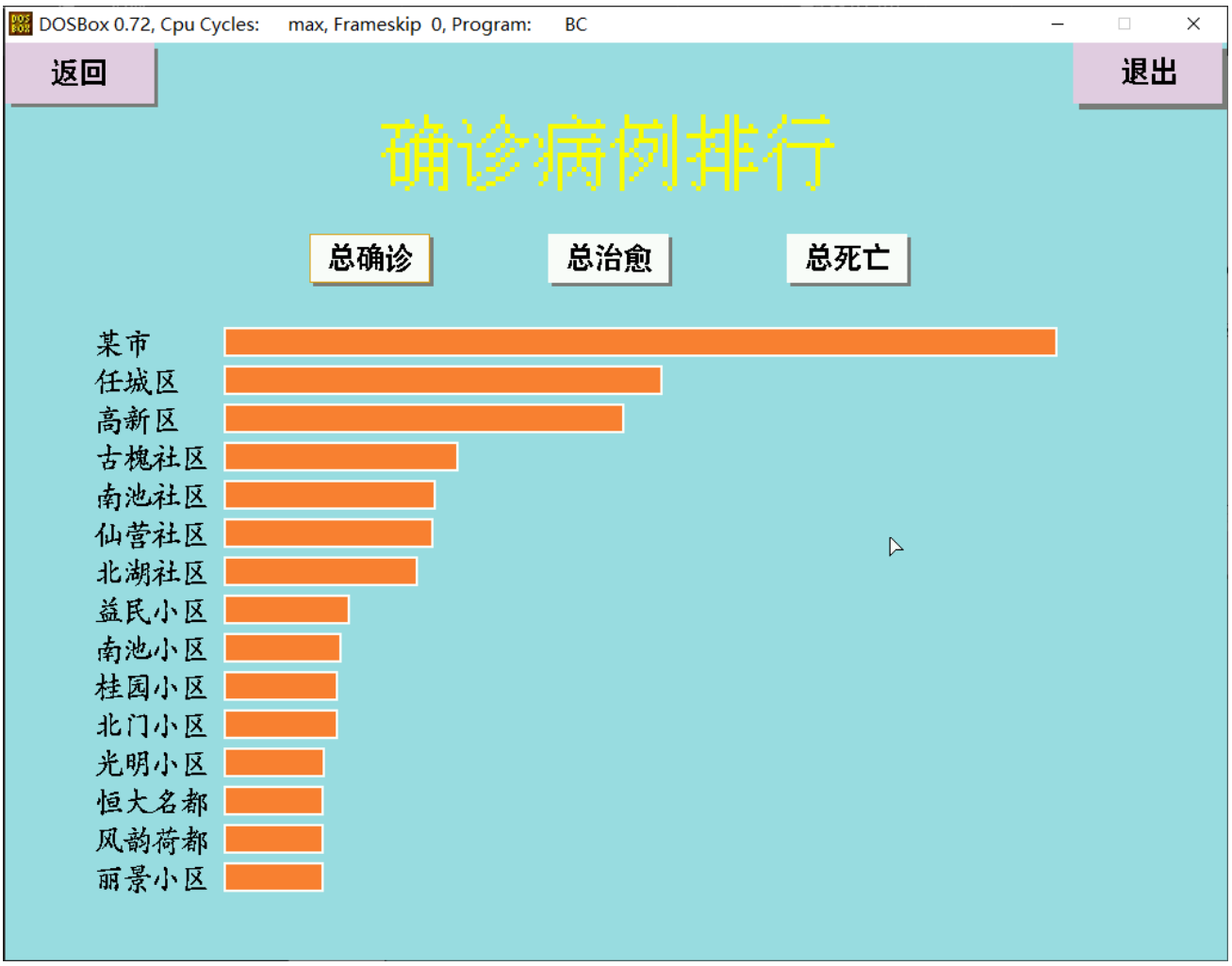
疫情信息通报界面：

信息通报界面使用简洁的输入框进行输入，页面右侧标明了用户所选区域，同时在输入框前有提示字符，以减少用户误输入对后续审核带来的不便。用户点击年、月、日后会自动显示今日时间，用户也可进行更改。



相关信息展示界面：

该界面展示各地区的总确诊、总感染与总治愈排行。条形图用较为醒目的橙色表示，上方显示选择键并伴有点击反馈。



管理员登录界面：

管理员界面采用前界面主题，管理员输入密码后进入管理员主界面。

管理员主界面：

本界面采用不同配色以凸显管理员的职能。界面左侧有提示字符方便管理员进行操作；上方是一个简单绘制的“管理员图标”。下方是管理员的三个功能选择键，同时伴有点击反馈。

管理员审核用户通报界面：

若无待审核数据，则提示并返回上一界面；

若有待审核数据，则在屏幕上高亮显示；管理员可在下方选择采纳或删除。操作后出现提示并返回上一界面。

无数据待审核：

有数据待审核：

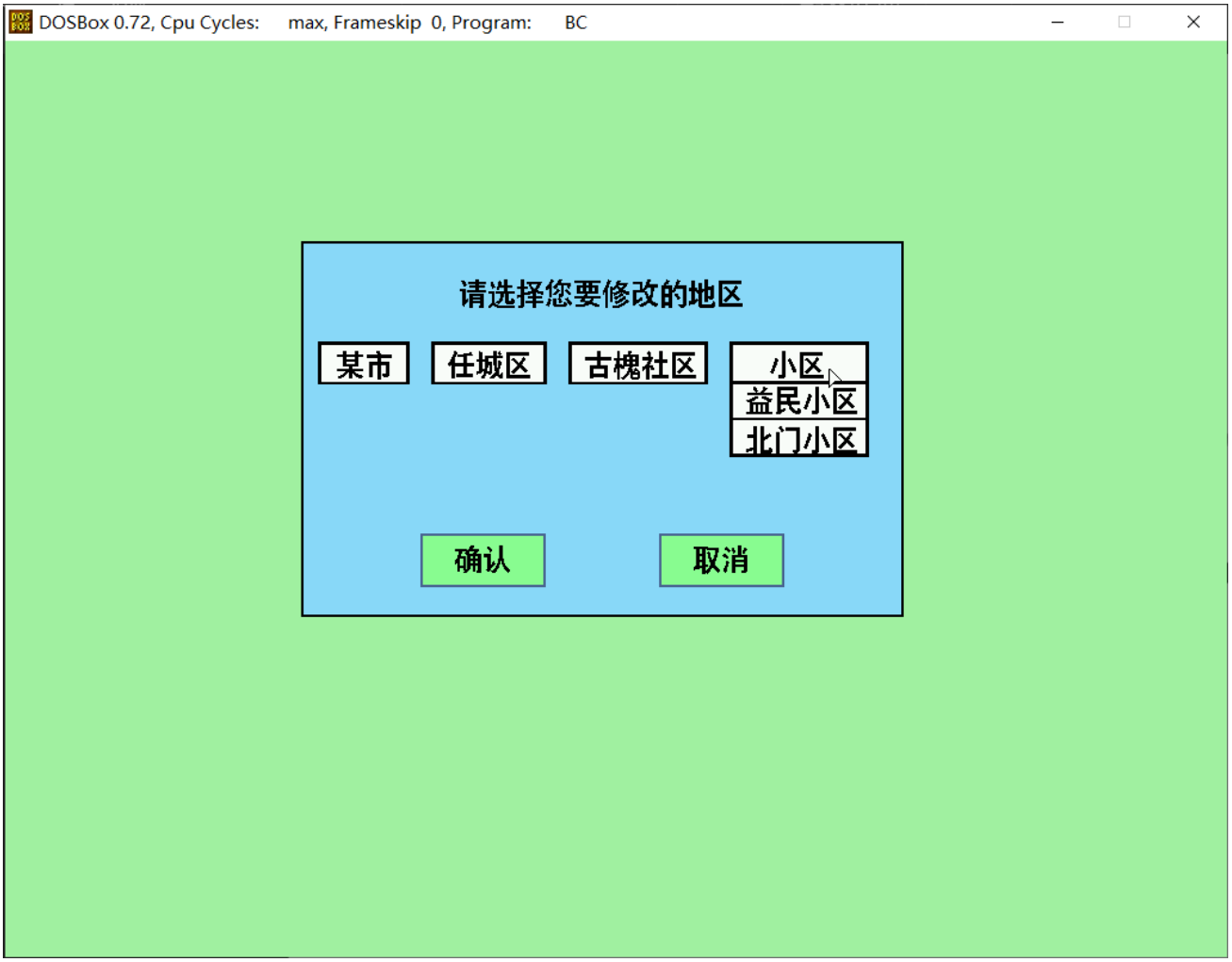

DOSBox 0.72, Cpu Cycles: max, Frameskip 0, Program: BC
用户名 年 月 日 新增确诊 新增死亡 新增治愈 数据区域
1111111 2020 4 15 1 0 0 南池小区
返回
采纳数据
删除数据
退出

审核成功：


DOSBox 0.72, Cpu Cycles: max, Frameskip 0, Program: BC
用户名 年 月 日 新增确诊 新增死亡 新增治愈 数据区域
1111111 2020 4 15 1 0 0 南池小区
返回
采纳数据
审核通过
删除数据
退出

管理员修改数据界面：

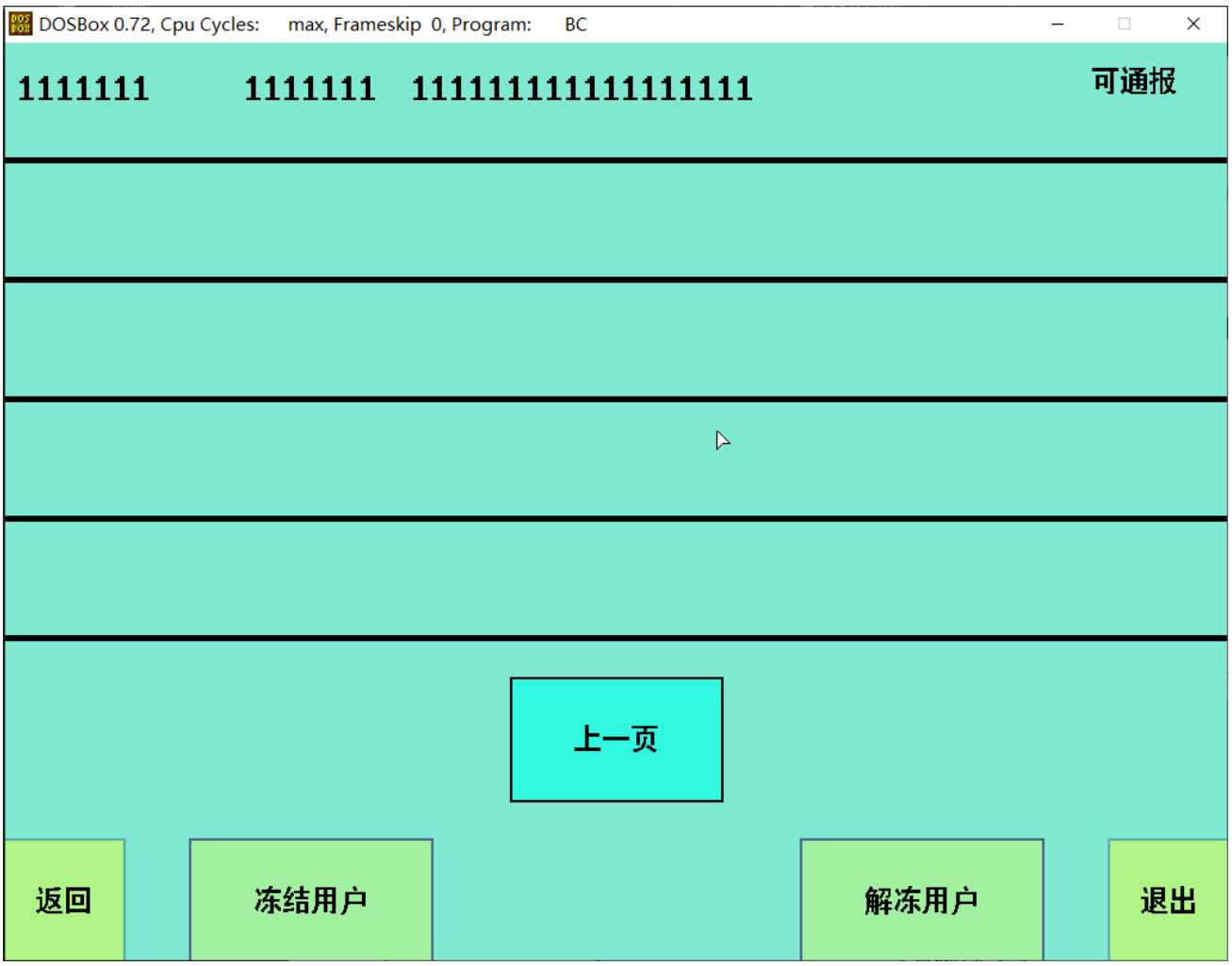
管理员进入该页面后可选择小区并输入时间




DOSBox 0.72, Cpu Cycles: max, Frameskip 0, Program: BC
当日确诊
累计确诊
1
15
当日治愈
累计治愈
1
8
当日死亡
累计死亡
0
0
返回
修改数据
退出

管理员的用户信息修改界面：

管理员可查看用户名与身份证，并查看用户状态，分为“可通报”和冻结；管理员可在下方选择冻结用户或解冻用户。



7. 数据结构

```
typedef struct UseInfo  //用户数据
{
    char UserName[12 + 1]; //用户名
    char password[12 + 1]; //密码
    char IDnumber[18 + 1]; //身份证号
    char UserState;        //用户是否被冻结  0：未被冻结，1：被冻结
    char n[2];             //存储'\n\r' 使得数据存储时换行

} USER;

typedef struct DayInfo //用于储存用户通报数据，等待管理员审核
{
    char area[5 + 1];  //通报区域
    char year[4 + 1];  //年
    char month[2 + 1]; //月
    char day[2 + 1];   //日
    char infestor[6];  //该日新增感染
    char death[6];     //该日新增死亡
    char heal[6];      //该日治愈
    char username[13]; //上报者
    char infostate;    //数据状态 0 为未审核，1为审核通过，2为审核不通过
    char n[2];         //存储'\n\r' 使得数据存储时换行
} DAY;

typedef struct Save_Dayinfo //储存信息
{
    char year[4 + 1];  //年
    char month[2 + 1]; //月
    char day[2 + 1];   //日
    char infestor[6];  //该日新增感染
    char death[6];     //该日新增死亡
    char heal[6];      //该日治愈
    char allinf[7];    //到该日为止 总感染数
    char alldea[7];    //到该日为止 总治愈数
    char allhea[7];    //到该日为止 总死亡数
    char n[2];         //存储'\n\r' 使得数据存储时换行
} SAVEDAY;
```

```
typedef struct Chartinfo //储存信息
{
    int infestor; //该日新增感染
    int death;    //该日新增死亡
    int heal;     //该日治愈
    int allinf;   //到该日为止 总感染数
    int alldea;   //到该日为止 总治愈数
    int allhea;   //到该日为止 总死亡数

} CHARTDATA;
```

8. 时间安排

日期	任务
暑假	研究课题任务，进行需求分析并设计报告， 学习文件读写、界面绘制知识、BC 的使用等。
第二周	完成鼠标、汉字库，初步设计绘制界面和主要框架
第三周	学习注册登录代码，初步完成登录模块。
第四周	完成疫情通报、数据查询模块
第五周	完成满足基础要求的测试程序，整理中期报告，准备中期验收。
第六周	绘制数据可视化界面框架。
第七周	完成信息展示模块，数据可视化模块，制作附加内容。
第八周	完成用户管理、通报管理、数据管理模块。
第九周	界面美化，优化代码与函数，增加数据量，完善模拟效果； 程序调试，修复 bug，整理报告，准备终期验收。

9. 源代码:

WELCOME.C

```
/*********************************************
FileName: welcome.c
Author:   韩海若
Date:    2020/10/28
Description: 绘制欢迎界面
Function list:
       int welcome() //欢迎界面的图形绘制
*********************************************/
#include "common.h"
#include "welcome.h"

/**********************
Function: int welcome()
Description: 欢迎界面的图形绘制
Calls:  无
Called by: main()
Input:  无
Output: 无
Return: 1   -> 用户登录
       2   -> 管理员登录
      -1   -> 退出
**********************/
int welcome(void)
{
    int flg = 0;

    Bar1(0, 0, 1024, 768, 0x9eef); //0x9eef

    /*画出按钮阴影*/
    Bar1(403, 450, 590, 525, 0x7bef); //用户键
    Bar1(403, 560, 590, 635, 0x7bef); //管理员键

    /*画出按钮*/
    Bar1(400, 445, 585, 520, 0xde5d); //用户键
    Bar1(400, 555, 585, 630, 0xde5d); //管理员键

    Circlefill(420, 482, 10, 0xffff); //用户登录圆形标志
    Circlefill(420, 592, 10, 0xffff); //管理员登录圆形标志
    Bar2(400, 445, 585, 520, 0xffff); //用户登录键
    Bar2(400, 555, 585, 630, 0xffff); //管理员键
```

```
    /*画相关信息*/
    prt_hz16_size(285, 150, 5, 5, "新型冠状病毒", 0xffe0, "HZK\\Hzk16s");
    prt_hz16_size(285, 250, 5, 5, "疫情通报系统", 0xffe0, "HZK\\Hzk16s");
//0xffe0 亮黄色
    prt_hz24(455, 472, "用户登录", 0x0000, "HZK\\Hzk24s");
    prt_hz24(455, 582, "管理员", 0x0000, "HZK\\Hzk24s");

    /*绘制退出键*/
    Bar1(900, 5, 1024, 55, 0x7bef); //退出键阴影
    Bar1(895, 0, 1019, 50, 59004);  //退出键实体
    prt_hz24(935, 12, "退出", 0x0000, "HZK\\Hzk24h");

    /*交互设计*/
    Mouse_Init();
    while (1)
    {
        MouseShow(&mouse);
        if (MousePress(400, 445, 585, 520)) //点击用户键
        {
            return 1; //进入用户登录界面
        }
        if (MousePress(400, 555, 585, 630)) //点击管理员
        {
            return 2; //进入管理员界面
        }
        if (MousePress(895, 0, 1019, 50)) //点击退出
        {
            return -1;
        }
        if (flg == 0 && MouseIn(895, 0, 1019, 50)) //退出键反馈
        {
            MouseOff(&mouse);
            Bar1(895, 0, 1019, 50, 56603);
            prt_hz24d(935, 12, "退出", 0x0000, "HZK\\Hzk24h");
            MouseOn(mouse);
            flg = 1;
        }
        if (flg == 0 && MouseIn(400, 445, 585, 520)) //用户键反馈
        {
            MouseOff(&mouse);
            Bar2(400, 445, 585, 520, 0x429a);
            Circlefill(420, 482, 10, 0x429a);
            MouseOn(mouse);
            flg = 1;
```

```
        }
        if (flg == 0 && MouseIn(400, 555, 585, 630)) //管理员反馈
        {
            MouseOff(&mouse);
            Circlefill(420, 592, 10, 0x429a);
            Bar2(400, 555, 585, 630, 0x429a);
            MouseOn(mouse);
            flg = 1;
        }
        else if (flg == 1 && !(MouseIn(400, 445, 585, 520)) && !(MouseIn(400,
555, 585, 630)) && !(MouseIn(900, 0, 1024, 50)))
        {
            MouseOff(&mouse);
            Circlefill(420, 482, 10, 0xffff); //用户登录圆形标志
            Circlefill(420, 592, 10, 0xffff); //管理员登录圆形标志
            Bar2(400, 445, 585, 520, 0xffff); //用户登录键
            Bar2(400, 555, 585, 630, 0xffff); //管理员键
            Bar1(895, 0, 1019, 50, 59004);
            prt_hz24d(935, 12, "退出", 0x0000, "HZK\\Hzk24h");
            MouseOn(mouse);
            flg = 0;
        }
    }
}
```

LOGIN.C

```
/*****************************************
FileName: login.c
Author:   韩海若
Date:    2020/10/28
Description: 该程序文件用于用户登录
Others: 无
Function List:
        login_user()   //用户登录函数
*****************************************/
#include "common.h"
#include "lgstate.h"
#include "login.h"

/*********************************
Function: login_user()
Description: 用户登录函数
Calls:  wr_user        //将登录的账号写入文件
```

```
       judge_userfrozen //判断用户是否冻结
       right_password  //判断密码是否正确
Called by: main()
Table Accessed: userinfo.DAT
Table Updated:  usernow.DAT
Return:   3 -> 进入注册界面
       5 -> 登录成功
       4 -> 忘记密码
      -1 -> 退出
       0 —> 返回
*********************************/
int login_user(void)
{
    //初始化
    int flag = 0;
    int state = 1;
    char username[12 + 1] = "\0";
    char key[12 + 1] = "\0";

    Bar1(0, 0, 1024, 768, 0x9eef);

    //退出返回键
    Bar1(900, 5, 1024, 55, 0x7bef); //退出框阴影
    Bar1(895, 0, 1019, 50, 59004);  //退出框实体
    prt_hz24(935, 12, "退出", 0x0000, "HZK\\Hzk24h");
    Bar1(5, 5, 127, 53, 0x7bef);
    Bar1(0, 0, 124, 50, 59004); //返回键
    prt_hz24(38, 13, "返回", 0x0000, "HZK\\Hzk24h");

    //标题
    prt_hz16_size(285, 150, 5, 5, "新型冠状病毒", 0xffe0, "HZK\\Hzk16s");
    prt_hz16_size(285, 250, 5, 5, "疫情通报系统", 0xffe0, "HZK\\Hzk16s");

    //账号密码框
    Bar1(374, 420, 659, 449, 0);
    Bar1(376, 422, 657, 447, 0xffff);
    Bar1(374, 484, 659, 513, 0);
    Bar1(376, 486, 657, 511, 0xffff);

    //登录注册键
    Bar1(405, 560, 473, 594, 0x7bef); //阴影
    Bar1(400, 555, 468, 589, 0xffff); //实体
    Bar1(545, 560, 613, 594, 0x7bef); //阴影
    Bar1(540, 555, 608, 589, 0xffff); //实体
```

```
//忘记密码键
Bar1(700, 484 + 76, 815, 513 + 76, 0x7bef); //阴影
Bar1(695, 479 + 76, 810, 508 + 76, 0xffff); //实体

prt_hz24d(307, 423, "账号", 0, "HZK\\Hzk24h");
prt_hz24d(307, 487, "密码", 0, "HZK\\Hzk24h");
prt_hz24d(410, 560, "登录", 0, "HZK\\Hzk24h");
prt_hz24d(550, 560, "注册", 0, "HZK\\Hzk24h");
prt_hz24d(705, 484 + 76, "忘记密码", 0, "HZK\\Hzk24h");

Mouse_Init();
while (1)
{
    char key_0 = 0; //按下登录键后的状态
                    //1 表示全部输入，2 表示账号未输入，3 表示密码未输入

    char key_1 = 2; //按下登录键后的状态
                    //0 表示已注册，1 表示尚未注册

    char key_2 = 0; //按下登录键后的状态，账户密码匹配
    char key_3 = 0; //0 表示不正确，1 表示正确

    MouseShow(&mouse);

    if (flag == 0 && MouseIn(895, 0, 1019, 50)) //退出框反馈
    {
        MouseOff(&mouse);
        Bar1(895, 0, 1019, 50, 56603);
        prt_hz24d(935, 12, "退出", 0x0000, "HZK\\Hzk24h");
        MouseOn(mouse);
        flag = 1;
    }
    if (flag == 0 && MouseIn(0, 0, 124, 50)) //返回框反馈
    {
        MouseOff(&mouse);
        Bar1(0, 0, 124, 50, 56603);
        prt_hz24d(38, 13, "返回", 0x0000, "HZK\\Hzk24h");
        MouseOn(mouse);
        flag = 1;
    }
    if (flag == 0 && MouseIn(400, 555, 468, 589)) //登录键反馈
    {
        MouseOff(&mouse);
```

```
    Bar1(405, 560, 473, 594, 0x7bef);
    Bar1(400, 555, 468, 589, 57083);
    prt_hz24d(410, 560, "登录", 0, "HZK\\Hzk24h");
    MouseOn(mouse);
    flag = 1;
}
if (flag == 0 && MouseIn(540, 555, 608, 589)) //注册键反馈
{
    MouseOff(&mouse);
    Bar1(545, 560, 613, 594, 0x7bef);
    Bar1(540, 555, 608, 589, 57083); //注册框
    prt_hz24d(550, 560, "注册", 0, "HZK\\Hzk24h");
    MouseOn(mouse);
    flag = 1;
}
if (flag == 0 && MouseIn(695, 479 + 76, 810, 508 + 76)) //忘记密码键反馈
{
    MouseOff(&mouse);
    Bar1(700, 484 + 76, 815, 513 + 76, 0x7bef);
    Bar1(695, 479 + 76, 810, 508 + 76, 57083); //忘记密码框
    prt_hz24d(705, 484 + 76, "忘记密码", 0, "HZK\\Hzk24h");
    MouseOn(mouse);
    flag = 1;
}
else if (flag == 1 && !(MouseIn(895, 0, 1019, 50))
                   && !(MouseIn(540, 555, 608, 589))
                   && !(MouseIn(400, 555, 468, 589))
                   && !(MouseIn(0, 0, 124, 50))
                   && !(MouseIn(695, 479 + 76, 810, 508 + 76)))
{
    MouseOff(&mouse);
    Bar1(895, 0, 1019, 50, 59004);
    Bar1(0, 0, 124, 50, 59004);
    Bar1(400, 555, 468, 589, 0xffff);
    Bar1(540, 555, 608, 589, 0xffff);
    Bar1(695, 479 + 76, 810, 508 + 76, 0xffff);
    prt_hz24d(935, 12, "退出", 0x0000, "HZK\\Hzk24h");
    prt_hz24d(38, 13, "返回", 0x0000, "HZK\\Hzk24h");
    prt_hz24d(410, 560, "登录", 0, "HZK\\Hzk24h");
    prt_hz24d(550, 560, "注册", 0, "HZK\\Hzk24h");
    prt_hz24d(705, 484 + 76, "忘记密码", 0, "HZK\\Hzk24h");
    MouseOn(mouse);
    flag = 0;
}
```

```
else if (MousePress(376, 422, 657, 447)) //点击账号框
{
    prt_hz24d(700, 454, "（回车键结束输入）", 0, "HZK\\Hzk24h");
    MouseOff(&mouse);
    Bar1(376, 422, 657, 447, 0xffff);
    Getinfo(380, 424, username, 9); //得到账号
    MouseOn(mouse);
    Bar1(700, 454, 910, 454 + 24, 0x9eef);
}
else if (MousePress(376, 486, 657, 511)) //点击密码框
{
    MouseOff(&mouse);
    Bar1(376, 486, 657, 511, 0xffff);
    Getcode(380, 488, key, 9); //得到密码
    MouseOn(mouse);
    Bar1(700, 486, 900, 486 + 28, 0x9eef);
}
else if (MousePress(540, 555, 608, 589)) //点击注册
{
    return 3;
}
if (MousePress(400, 555, 468, 589)) //登录确认键
{
    /*判断是否输入完全*/
    state = 1;
    key_0 = complete_login(username[0], key[0]);   //1 表示全部输入
                                                   //2 表示账号未输入
                                                   //3 表示密码未输入
    {
        if (key_0 == 2)
        {
            prt_hz24d(700, 410, "帐户未输入", 0, "HZK\\Hzk24h");
            state = 0;
        }
        else if (key_0 == 3)
        {
            prt_hz24d(700, 490, "密码未输入", 0, "HZK\\Hzk24h");
            state = 0;
        }
    }

    /*判断登录账号是否已经注册*/
    if (key_0 == 1)
    {
```

```
        key_1 = judge_same_name(username); //0 表示已注册，1 表示尚未注册
        if (key_1 == 1)
        {

            prt_hz24d(650, 660, "帐户未注册", 0, "HZK\\Hzk24h");
            delay(1000);
            Bar1(650, 660, 850, 660 + 35, 0x9eef);
            state = 0;
        }
    }

    /*判断账号密码是否匹配*/
    if (key_1 == 0) //若已经注册，则看账户密码是否匹配
    {
        key_2 = right_password(username, key); //0 表示不正确，1 表示正确
        if (key_2 == 0)
        {

            prt_hz24d(650, 660, "密码不正确", 0, "HZK\\Hzk24h");
            delay(1000);
            Bar1(650, 660, 850, 660 + 35, 0x9eef);
            state = 0;
        }
    }
    if (key_2 == 1)
    {
        key_3 = judge_userfrozen(username);
        if (key_3 == 1)
        {
            prt_hz24d(650, 660, "帐户已冻结", 0, "HZK\\Hzk24h");
            delay(500);
            Bar1(650, 660, 850, 660 + 35, 0x9eef);
            state = 0;
        }
    }
    if (state == 1) //将登录的账号写入文件
    {
        wr_user(username);
        return 5; //进入系统
    }
}
if (MousePress(695, 479 + 76, 810, 508 + 76)) //点击忘记密码
{
    return 4;
```

```
        }
        if (MousePress(895, 0, 1019, 50)) //点击退出
        {
            return -1;
        }
        else if (MousePress(0, 0, 124, 50)) //点击返回
        {
            return 0;
        }
    }
}
```

VISUSAL.C

```
/*****************************************
FileName: visual.c
Author:   韩海若
Date:    2020/10/28
Description: 该程序文件用于实现疫情风险等级可视化
Others: 无
Function List:
        int visual page()   //底层界面绘制
        void visual_basic() //可视化区域风险表示
        int risk_level()    //判断风险等级
*****************************************/
#include "common.h"
#include "dayinfo.h"
#include "draw.h"
#include "visual.h"

/*****************************
Function: int visual_page(void)
Description: 可视化背景界面绘制
Calls:    void visual_basic()
Called by: main()
Table Accessed: areanow.DAT
Table Updated:  areanow.DAT
Input:  无
Output: 无
Return: 1   -> 返回
        6   -> 通报界面
        7   -> 查询界面
       12   -> 注销
*****************************/
```

```
int visual_page(void)
{
    int flg = 0;
    int choose = -1;
    Bar1(0, 0, 1024, 768, 0x9eef); //主窗口

    //退出键
    Bar1(900, 5, 1024, 55, 0x7bef); //退出框阴影
    Bar1(895, 0, 1019, 50, 59004);  //退出框实体
    prt_hz24(935, 12, "退出", 0x0000, "HZK\\Hzk24k");

    //返回键
    Bar1(5, 5, 129, 55, 0x7bef); //0x7bef 灰色
    Bar1(0, 0, 124, 50, 59004);
    prt_hz24(38, 13, "返回", 0x0000, "HZK\\Hzk24k");

    //跳转选择控制台
    Bar1(282, 5, 766, 55, 0x7bef); //灰色阴影
    Bar1(277, 0, 761, 50, 0x0000); //黑色底色
    Bar1(280, 3, 438, 47, 63422);  //注销
    Bar1(441, 3, 600, 47, 63422);  //通报
    Bar1(603, 3, 758, 47, 63422);  //查询

    //标题
    prt_hz24(335, 13, "排行", 0x0000, "HZK\\Hzk24k");
    prt_hz24(470, 13, "疫情通报", 0x0000, "HZK\\Hzk24k");
    prt_hz24(630, 13, "数据查询", 0x0000, "HZK\\Hzk24k");
    prt_hz16_size(280, 100, 5, 5, "风险等级地图", 0xffe0, "HZK\\Hzk16s");

    visual_basic(); //绘制区域基本框架

    Mouse_Init();
    while (1)
    {
        MouseShow(&mouse);
        if (flg == 0 && MouseIn(895, 0, 1019, 50)) //退出框反馈
        {
            MouseOff(&mouse);
            Bar1(895, 0, 1019, 50, 56603);
            prt_hz24d(935, 12, "退出", 0, "HZK\\Hzk24k");
            MouseOn(mouse);
            flg = 1;
        }
        else if (flg == 0 && MouseIn(0, 0, 124, 50)) //返回框反馈
```

```
{
    MouseOff(&mouse);
    Bar1(0, 0, 124, 50, 56603);
    prt_hz24d(38, 13, "返回", 0, "HZK\\Hzk24k");
    MouseOn(mouse);
    flg = 1;
}
else if (flg == 0 && MouseIn(280, 3, 438, 47)) //注销框反馈
{
    MouseOff(&mouse);
    Bar2(280, 3, 438, 47, 7327);
    MouseOn(mouse);
    flg = 1;
}
else if (flg == 0 && MouseIn(441, 3, 600, 47)) //通报框反馈
{
    MouseOff(&mouse);
    Bar2(441, 3, 600, 47, 7327);
    MouseOn(mouse);
    flg = 1;
}
if (flg == 0 && MouseIn(603, 3, 758, 47)) //查询框反馈
{
    MouseOff(&mouse);
    Bar2(603, 3, 758, 47, 7327);
    MouseOn(mouse);
    flg = 1;
}
else if (flg == 1 && MouseIn(277, 0, 761, 50)
                  && !(MouseIn(280, 3, 438, 47))
                  && !(MouseIn(441, 3, 600, 47))
                  && !(MouseIn(603, 3, 758, 47))
                  && !(MouseIn(805, 140, 865, 180)))
{
    MouseOff(&mouse);
    Bar2(280, 3, 438, 47, 63422); //注销
    Bar2(441, 3, 600, 47, 63422); //通报
    Bar2(603, 3, 758, 47, 63422); //查询
    MouseOn(mouse);
    flg = 0;
}
else if (flg == 1 && !(MouseIn(277, 0, 761, 50))
                 && !(MouseIn(805, 140, 865, 180))
                 && !(MouseIn(0, 0, 124, 50))
```

```
                && !(MouseIn(895, 0, 1019, 50)))
{
    //还原界面
    MouseOff(&mouse);
    Bar2(280, 3, 438, 47, 63422);     //注销
    Bar2(441, 3, 600, 47, 63422);     //通报
    Bar2(603, 3, 758, 47, 63422);     //查询
    Bar1(895, 0, 1019, 50, 59004);    //退出
    Bar1(0, 0, 124, 50, 59004);       //返回
    prt_hz24(38, 13, "返回", 0x0000, "HZK\\Hzk24k");
    prt_hz24(935, 12, "退出", 0x0000, "HZK\\Hzk24k");
    MouseOn(mouse);
    flg = 0;
}
if (MousePress(895, 0, 1019, 50)) //点击退出
{
    CloseSVGA();
    delay(1000);
    exit(1);
}
else if (MousePress(0, 0, 124, 50)) //点击返回
{
    return 1;
}
else if (MousePress(280, 3, 438, 47)) //点击注销
{
    return 12;
}
else if (MousePress(603, 3, 758, 47)) //点击查询
{
    if (choose == -1)
    {
        prt_hz24(810, 150, "您未选中区域", 0x0000, "HZK\\Hzk24k");
        delay(1000);
        Bar1(810, 150, 1000, 200, 0x9eef);
    }
    else
    {
        return 7;
    }
}
else if (MousePress(441, 3, 600, 47)) //点击通报
{
    if (choose == 0)
```

```
    {
        return 6;
    }
    else if (choose == 1)
    {
        Bar1(810, 150, 1000, 200, 0x9eef);
        prt_hz24(810, 150, "该区域无法通报", 0x0000, "HZK\\Hzk24k");
        delay(1000);
        Bar1(810, 150, 1000, 200, 0x9eef);
    }
    else if (choose == -1)
    {
        prt_hz24(810, 150, "您未选中区域", 0x0000, "HZK\\Hzk24k");
        delay(1000);
        Bar1(810, 150, 1000, 200, 0x9eef);
    }
}
/******************
判断条件由小区->社区->区->市
优先级由高到低
可避免重叠部分的错误
******************/
if (MousePress(105, 320, 245, 480)) //益民小区
{
    Bar1(805, 140, 805 + 200, 140 + 45, 0x9eef);
    prt_hz24(810, 150, "已选中益民小区", 0x0000, "HZK\\Hzk24k");
    wr_area(0);
    choose = 0;
    continue;
}
else if (MousePress(280, 320, 410, 475)) //北门小区
{
    Bar1(805, 140, 805 + 200, 140 + 45, 0x9eef);
    prt_hz24(810, 150, "已选中北门小区", 0x0000, "HZK\\Hzk24k");
    wr_area(1);
    choose = 0;
    continue;
}
else if (MousePress(105, 560, 245, 710)) //光明小区
{
    Bar1(805, 140, 805 + 200, 140 + 45, 0x9eef);
    prt_hz24(810, 150, "已选中光明小区", 0x0000, "HZK\\Hzk24k");
    wr_area(2);
    choose = 0;
```

```
    continue;
}
else if (MousePress(280, 560, 410, 710)) //桂园小区
{
    Bar1(805, 140, 805 + 200, 140 + 45, 0x9eef);
    prt_hz24(810, 150, "已选中桂园小区", 0x0000, "HZK\\Hzk24k");
    wr_area(3);
    choose = 0;
    continue;
}
else if (MousePress(555, 320, 695, 480)) //南池小区
{
    Bar1(805, 140, 805 + 200, 140 + 45, 0x9eef);
    prt_hz24(810, 150, "已选中南池小区", 0x0000, "HZK\\Hzk24k");
    wr_area(4);
    choose = 0;
    continue;
}
else if (MousePress(740, 320, 920, 480)) //丽景小区
{
    Bar1(805, 140, 805 + 200, 140 + 45, 0x9eef);
    prt_hz24(810, 150, "已选中丽景小区", 0x0000, "HZK\\Hzk24k");
    wr_area(5);
    choose = 0;
    continue;
}
else if (MousePress(555, 560, 695, 710)) //风韵荷都
{
    Bar1(805, 140, 805 + 200, 140 + 45, 0x9eef);
    prt_hz24(810, 150, "已选中风韵荷都", 0x0000, "HZK\\Hzk24k");
    wr_area(6);
    continue;
}
else if (MousePress(740, 560, 900, 710)) //恒大名都
{
    Bar1(805, 140, 805 + 200, 140 + 45, 0x9eef);
    prt_hz24(810, 150, "已选中恒大名都", 0x0000, "HZK\\Hzk24k");
    wr_area(7);
    choose = 0;
    continue;
}
else if (MousePress(90, 290, 420, 500)
          && !MousePress(105, 320, 245, 480)
          && !MousePress(280, 320, 410, 475)) //古槐社区
```

```
{
    Bar1(805, 140, 805 + 200, 140 + 45, 0x9eef);
    prt_hz24(810, 150, "已选中古槐社区", 0x0000, "HZK\\Hzk24k");
    wr_area(8);
    choose = 1;
    continue;
}
else if (MousePress(90, 530, 420, 720)
          && !MousePress(105, 560, 245, 710)
          && !MousePress(280, 560, 410, 710)) //仙营社区
{
    Bar1(805, 140, 805 + 200, 140 + 45, 0x9eef);
    prt_hz24(810, 150, "已选中仙营社区", 0x0000, "HZK\\Hzk24k");
    wr_area(9);
    choose = 1;
    continue;
}
else if (MousePress(540, 290, 930, 500)
          && !MousePress(555, 320, 695, 480)
          && !MousePress(740, 320, 920, 480)) //南池社区
{
    Bar1(805, 140, 805 + 200, 140 + 45, 0x9eef);
    prt_hz24(810, 150, "已选中南池社区", 0x0000, "HZK\\Hzk24k");
    wr_area(10);
    choose = 1;
    continue;
}
else if (MousePress(540, 530, 930, 720)
          && !MousePress(555, 560, 695, 710)
          && !MousePress(740, 560, 900, 710)) //北湖社区
{
    Bar1(805, 140, 805 + 200, 140 + 45, 0x9eef);
    prt_hz24(810, 150, "已选中北湖社区", 0x0000, "HZK\\Hzk24k");
    wr_area(11);
    choose = 1;
    continue;
}
else if (MousePress(50, 260, 460, 740)
          && !MousePress(90, 290, 420, 500)
          && !MousePress(90, 530, 420, 720)) //任城区
{
    Bar1(805, 140, 805 + 200, 140 + 45, 0x9eef);
    prt_hz24(810, 150, "已选中任城区", 0x0000, "HZK\\Hzk24k");
    wr_area(12);
```

```
            choose = 1;
            continue;
        }
        else if (MousePress(495, 260, 960, 740)
                  && !MousePress(540, 290, 930, 500)
                  && !MousePress(540, 530, 930, 720)) //高新区
        {
            Bar1(805, 140, 805 + 200, 140 + 45, 0x9eef);
            prt_hz24(810, 150, "已选中高新区", 0x0000, "HZK\\Hzk24k");
            wr_area(13);
            choose = 1;
            continue;
        }
        else if (MousePress(2, 240, 1022, 766)
                  && !MousePress(495, 260, 960, 740)
                  && !MousePress(50, 260, 460, 740)) //市
        {
            Bar1(805, 140, 805 + 200, 140 + 45, 0x9eef);
            prt_hz24(810, 150, "已选中市", 0x0000, "HZK\\Hzk24k");
            wr_area(14);
            choose = 1;
            continue;
        }
    }
}

/*****************************
Function: void visual_basic(void)
Description: 可视化背景界面绘制
Calls:    risk_level()
         draw_floor1()
Called by: visual_page()
Input:  无
Output: 无
Return: 1   -> 返回
        6   -> 通报界面
        7   -> 查询界面
       12   -> 注销
*****************************/
void visual_basic(void)
{
    //把每个地区的数据存在数组里
    int data[15] = {95, 85, 66, 23, 69, 47, 52, 96, 32, 20, 12, 8, 3, 24, 7};
    int temp1, temp2, temp3, temp4 = 10;
```

```
    int i = 0;
    for (i = 14; i >= 0; i--)
    {
        countareacases(i, &temp4, &temp1, &temp3, &temp2);
        data[14 - i] = temp2;
    }

    Bar1(0, 237, 1024, 768, 0x0000);            //黑边框，主体范围
    Bar1(2, 240, 1022, 766, risk_level(data[0])); //最外围(市级)(2,240,1022,766)

    Bar1(47, 257, 463, 743, 0x0000);
    Bar1(50, 260, 460, 740, risk_level(data[2])); //左侧任城区 50,260,460,740

    //画两个区之间的马路
    Bar1(465, 240, 485, 768, 0);
    for (i = 0; i < 20; i++)
    {
        Line3(475, 242 + 25 * i, 475, 242 + 25 * i + 10, 2, 65184);
    }

    Bar1(492, 257, 963, 743, 0x0000);
    Bar1(495, 260, 960, 740, risk_level(data[1])); //右侧高新区 495,260,960,740

    Bar1(87, 287, 423, 503, 0x0000);
    Bar1(90, 290, 420, 500, risk_level(data[6])); //左侧古槐社区 90,290,420,500

    Bar1(87, 527, 423, 723, 0x0000);
    Bar1(90, 530, 420, 720, risk_level(data[5])); //左侧仙营社区 90,530,420,720

    //画上下社区之间的马路
    Bar1(0, 505, 1024, 525, 0);
    for (i = 0; i < 46; i++)
    {
        Line3(3 + 25 * i, 515, 3 + 25 * i + 10, 515, 2, 65184);
    }

    Bar1(537, 287, 933, 503, 0x0000);
    Bar1(540,  290,  930,  500,  risk_level(data[4]));     // 右侧南池社区
540,290,930,500

    Bar1(537, 527, 933, 723, 0x0000);
    Bar1(540,  530,  930,  720,  risk_level(data[3]));     // 右侧北湖社区
540,530,930,720
```

```
    Bar1(102, 317, 248, 483, 0x0000);
    Bar1(105,  320,  245,  480,  risk_level(data[14]));  // 左侧益民小区
105,320,245,480
    draw_floor1(120, 360);

    Bar1(277, 317, 413, 478, 0x0000);
    Bar1(280,  320,  410,  475,  risk_level(data[13]));  // 左侧北门小区
280,320,410,475
    draw_floor2(310, 360);

    //画左侧小区之间的马路
    Bar1(250, 240, 270, 766, 0);
    for (i = 0; i < 46; i++)
    {
        Line3(260, 245 + 25 * i, 260, 245 + 25 * i + 10, 2, 65184);
    }

    Bar1(102, 557, 248, 713, 0x0000);
    Bar1(105,  560,  245,  710,  risk_level(data[12]));  // 左侧光明小区
105,560,245,710
    draw_floor3(140, 600);

    Bar1(277, 557, 413, 713, 0x0000);
    Bar1(280,  560,  410,  710,  risk_level(data[11]));  // 左侧桂园小区
280,560,410,710
    draw_floor1(285, 600);

    Bar1(552, 317, 698, 483, 0x0000);
    Bar1(555,  320,  695,  480,  risk_level(data[10]));  // 右侧南池小区
555,320,695,480
    draw_floor3(590, 360);

    Bar1(737, 317, 903, 483, 0x0000);
    Bar1(740, 320, 900, 480, risk_level(data[9])); //右侧丽景小区 740,320,920,480
    draw_floor1(760, 360);

    //画右侧小区之间的马路
    Bar1(710, 240, 730, 766, 0);
    for (i = 0; i < 46; i++)
    {
        Line3(720, 245 + 25 * i, 720, 245 + 25 * i + 10, 2, 65184);
    }

    Bar1(552, 557, 698, 713, 0x0000);
```

```
    Bar1(555, 560, 695, 710, risk_level(data[8])); //右侧风韵荷都 555,560,695,710
    draw_floor2(590, 600);

    Bar1(737, 557, 903, 713, 0x0000);
    Bar1(740, 560, 900, 710, risk_level(data[7])); //右侧恒大名都 740,560,900,710
    draw_floor3(780, 600);
}

/*********************
Function: int risk_level(int cases)
Description: 判断风险等级函数
Calls:  无
Called by: visual_basic()
Input:  int cases 为对应地区确诊人数
Output: 无
Return: 不同风险对应的颜色 RGB 值
Others:
    *高风险*地区的分级标准为累计确诊病例    *超过 50 例*
    *中高风险*地区的分级标准为累计确诊病例  *超过 30 例*
    *中风险*地区的分级标准为累计确诊病例    *超过 10 例*
    *低风险*地区的分级标准为累计确诊病例    *不超过 10 例*
**********************/
int risk_level(int cases)
{
    if (cases == 0)
    {
        return 32736; //低风险绿色
    }
    else if (cases == 1)
    {
        return 65184; //中风险黄色
    }
    else if (cases == 2)
    {
        return 64800; //中高风险橙色
    }
    else if (cases > 2)
    {
        return 64526; //高风险红色
    }
}
```

SHOWINFO.C

```
/************************************************
FileName: showinfo.h
Author:   韩海若
Date:    2020/10/28
Description: 该程序文件用于实现疫情数据排行图
Others: 无
Function List:
       int showinfo_main(void); //参数初始化, 调用其他函数绘图
       void drawbar(int key); //绘制选择键
       void bubble_Sort(int arr[], int eq_loc[], int len); //冒泡排序
       void swap(int *a, int *b); //交换两数
************************************************/
#include "common.h"
#include "dayinfo.h"
#include "draw.h"
#include "showinfo.h"
/************************************************
Function: int showinfo_main(void)
Description: 参数初始化, 调用其他函数绘图
Calls:    void drawbar(int key); //绘制选择键
         void bubble_Sort(int arr[], int eq_loc[], int len); //冒泡排序
         void swap(int *a, int *b); //交换两数
         countareacases(i, &tmp1, &tmp2, &tmp3, &tmp4) //读取总数据

Called by: main()
Table Accessed: i.DAT(不同地区对应的数据文件)
Table Updated:  无
Input:  无
Output: 无
Return: 5   -> 可视化界面
      -1   -> 退出
************************************************/
int showinfo_main(void)
{
    //参数初始化
    int flg = 0, key = 0;
    int data[15] = {0};
    int tmp1, tmp2, tmp3, tmp4 = 10;
    int i, k;

    //初始化地名数组与匹配数组
    int eq_loc[15] = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14};
```

```
char loc[15][20] = {"\0"};
char loc_name[15][20] =
{"某市", "高新区", "任城区", "北湖社区", "南池社区",
"仙营社区", "古槐社区", "恒大名都", "风韵荷都", "丽景小区",
"南池小区", "桂园小区", "光明小区", "北门小区", "益民小区"};

//绘制选择键
drawbar(key);

//标题
prt_hz16_size(315, 60, 4, 4, "确诊病例排行", 0xffe0, "HZK\\Hzk16s");

//读入确诊数据
for (i = 14; i >= 0; i--)
{
   countareacases(i, &tmp1, &tmp2, &tmp3, &tmp4);
   data[14 - i] = tmp1;
}

//对确诊数据进行冒泡排序
bubble_Sort(data, eq_loc, 15);

//地名匹配
for (i = 0, k = 0; i < 15; i++)
{
   stpcpy(loc[i], loc_name[eq_loc[k++]]);
}

//输出地名与条形图
for (i = 0; i < 15; i++)
{
   prt_hz24(75, 240 + 32 * i, loc[i], 0x0000, "HZK\\Hzk24k");
   draw_column(185, 240 + 32 * i, data[i]);
}

Mouse_Init();
while (1)
{
   MouseShow(&mouse);

   //每次循环为 eq_loc 重新正序赋值
   for (i = 0; i < 15; i++)
   {
      eq_loc[i] = i;
```

```
}

if (MousePress(895, 0, 1019, 50)) //点击退出
{
    return -1;
}
if (MousePress(0, 0, 124, 50)) //点击返回
{
    return 5; //返回可视化界面
}
if (MousePress(255, 160, 355, 200) && key != 0) //点击总确诊
{
    for (i = 14; i >= 0; i--)
    {
        countareacases(i, &tmp1, &tmp2, &tmp3, &tmp4);
        data[14 - i] = tmp1;
    }
    key = 0;
    drawbar(key);
    prt_hz16_size(315, 60, 4, 4, "确诊病例排行", 0xffe0, "HZK\\Hzk16s");

    //对确诊数据进行冒泡排序
    bubble_Sort(data, eq_loc, 15);

    //地名匹配
    for (i = 0, k = 0; i < 15; i++)
    {
        stpcpy(loc[i], loc_name[eq_loc[k++]]);
    }

    //绘制条形图
    for (i = 0; i < 15; i++)
    {
        prt_hz24(70, 240 + 32 * i, loc[i], 0x0000, "HZK\\Hzk24k");
        draw_column(180, 240 + 32 * i, data[i]);
    }
}
if (MousePress(255 + 200, 160, 355 + 200, 200) && key != 1) //点击总治愈
{
    for (i = 14; i >= 0; i--)
    {
        countareacases(i, &tmp1, &tmp2, &tmp3, &tmp4);
        data[14 - i] = tmp3;
    }
```

```
            key = 1;
            drawbar(key);
            prt_hz16_size(315, 60, 4, 4, "治愈病例排行", 0xffe0, "HZK\\Hzk16s");

            //对治愈数据进行冒泡排序
            bubble_Sort(data, eq_loc, 15);

            //地名匹配
            for (i = 0, k = 0; i < 15; i++)
            {
                stpcpy(loc[i], loc_name[eq_loc[k++]]);
            }

            //绘制条形图
            for (i = 0; i < 15; i++)
            {
                prt_hz24(70, 240 + 32 * i, loc[i], 0x0000, "HZK\\Hzk24k");
                draw_column(180, 240 + 32 * i, data[i]);
            }
        }
        if (MousePress(255 + 200 + 200, 160, 355 + 200 + 200, 200) && key != 2)
//点击总死亡
        {
            for (i = 14; i >= 0; i--)
            {
                countareacases(i, &tmp1, &tmp2, &tmp3, &tmp4);
                data[14 - i] = tmp2;
            }
            key = 2;
            drawbar(key);
            prt_hz16_size(315, 60, 4, 4, "死亡病例排行", 0xffe0, "HZK\\Hzk16s");
            bubble_Sort(data, eq_loc, 15);

            //地名匹配
            for (i = 0, k = 0; i < 15; i++)
            {
                stpcpy(loc[i], loc_name[eq_loc[k++]]);
            }

            for (i = 0; i < 15; i++)
            {
                prt_hz24(70, 240 + 32 * i, loc[i], 0x0000, "HZK\\Hzk24k");
                draw_column(180, 240 + 32 * i, data[i] * 10);
            }
```

```
        }

        if (flg == 0 && MouseIn(895, 0, 1019, 50)) //退出框反馈
        {
            MouseOff(&mouse);
            Bar1(895, 0, 1019, 50, 56603);
            prt_hz24d(935, 12, "退出", 0x0000, "HZK\\Hzk24h");
            MouseOn(mouse);
            flg = 1;
        }
        if (flg == 0 && MouseIn(0, 0, 124, 50)) //返回框反馈
        {
            MouseOff(&mouse);
            Bar1(0, 0, 124, 50, 56603);
            prt_hz24d(38, 13, "返回", 0x0000, "HZK\\Hzk24h");
            MouseOn(mouse);
            flg = 1;
        }

        else if (flg == 1 && !(MouseIn(0, 0, 124, 50)) && !(MouseIn(900, 0, 1024,
50)))
        {
            MouseOff(&mouse);
            Bar1(895, 0, 1019, 50, 59004); //退出框实体
            prt_hz24(935, 12, "退出", 0x0000, "HZK\\Hzk24h");

            Bar1(0, 0, 124, 50, 59004);
            prt_hz24(38, 13, "返回", 0x0000, "HZK\\Hzk24h");

            MouseOn(mouse);
            flg = 0;
        }
    }
}

/*****************************************************
Function: void drawbar(int key)
Description: 绘制选择键
Called by: showinfo_main(void)
Input:  int key 选择类型参数
        0 -> 总确诊
        1 -> 总治愈
        2 -> 总死亡
*****************************************************/
```

```
void drawbar(int key)
{
    Bar1(0, 0, 1024, 768, 0x9efc);

    //绘制退出框
    Bar1(900, 5, 1024, 55, 0x7bef); //退出框阴影
    Bar1(895, 0, 1019, 50, 59004);  //退出框实体
    prt_hz24(935, 12, "退出", 0x0000, "HZK\\Hzk24h");

    //返回键
    Bar1(5, 5, 127, 53, 0x7bef); //0x7bef 灰色
    Bar1(0, 0, 124, 50, 59004);
    prt_hz24(38, 13, "返回", 0x0000, "HZK\\Hzk24h");

    Bar1(258, 163, 358, 203, 0x7bef);
    Bar1(255, 160, 355, 200, 0xffff);
    prt_hz24(270, 168, "总确诊", 0x0000, "HZK\\Hzk24h");

    Bar1(255 + 203, 163, 355 + 203, 203, 0x7bef);
    Bar1(255 + 200, 160, 355 + 200, 200, 0xffff);
    prt_hz24(270 + 200, 168, "总治愈", 0x0000, "HZK\\Hzk24h");

    Bar1(255 + 200 + 203, 163, 355 + 200 + 203, 203, 0x7bef);
    Bar1(255 + 200 + 200, 160, 355 + 200 + 200, 200, 0xffff);
    prt_hz24(270 + 200 + 200, 168, "总死亡", 0x0000, "HZK\\Hzk24h");

    if (key == 0)
    {
        Bar2(255, 160, 355, 200, 56612);
    }
    if (key == 1)
    {
        Bar2(255 + 200, 160, 355 + 200, 200, 56612);
    }
    if (key == 2)
    {
        Bar2(255 + 200 + 200, 160, 355 + 200 + 200, 200, 56612);
    }
}

//交换两数
void swap(int *a, int *b)
{
    int tmp = *a;
```

```
    *a = *b;
    *b = tmp;
}

//冒泡排序
void bubble_Sort(int arr[], int eq_loc[], int len)
{
    int i, j;
    for (i = 0; i < len - 1; i++)
        for (j = 0; j < len - 1 - i; j++)
            if (arr[j] < arr[j + 1])
            {
                swap(&arr[j], &arr[j + 1]);
                swap(&eq_loc[j], &eq_loc[j + 1]);
            }
    return;
}
```

ADMIN.C

```
/*********************************************
FileName: admin.c
Author:   韩海若
Date:    2020/10/29
Description: 该文件主要用于管理员登录注册功能
Function List:
       login_user()   //用户登录函数
*********************************************/
#include "admin.h"
#include "common.h"
#include "lgstate.h"

/*********************************
Function: login_admin()
Description: 管理员登录函数
Called by: main()
Return:   8 -> 进入管理员主界面
       0 —> 返回欢迎界面
*********************************/
int login_admin(void)
{
    int flag = 0, key = 0; //flag 鼠标反馈 key  判断密码是否正确
    char code[5] = "";     //储存密码并进行比对
```

```
Bar1(0, 0, 1024, 768, 0x9eef); //初始化界面

//绘制退出返回键
Bar1(900, 5, 1024, 55, 0x7bef);
Bar1(895, 0, 1019, 50, 59004);
prt_hz24(935, 12, "退出", 0x0000, "HZK\\Hzk24h");
Bar1(5, 5, 127, 53, 0x7bef);
Bar1(0, 0, 124, 50, 59004);
prt_hz24(38, 13, "返回", 0x0000, "HZK\\Hzk24h");

//标题
prt_hz16_size(285, 150, 5, 5, "新型冠状病毒", 0xffe0, "HZK\\Hzk16s");
prt_hz16_size(285, 250, 5, 5, "疫情通报系统", 0xffe0, "HZK\\Hzk16s");
prt_hz16_size(404, 450, 2, 2, "输入管理员密码", 0x5091, "HZK\\Hzk16s");

//密码框
Bar1(382, 500, 682, 530, 0);
Bar1(384, 502, 680, 528, 0xffff);

//确认框
Bar1(477, 585, 577, 635, 0);
Bar1(472, 580, 572, 630, 0xffff);
prt_hz24d(320, 500, "密码：", 0, "HZK\\Hzk24h");
prt_hz24d(497, 592, "确认", 0, "HZK\\Hzk24h");

Mouse_Init();
while (1)
{
    MouseShow(&mouse);
    if (flag == 0 && MouseIn(895, 0, 1019, 50)) //退出框反馈
    {
        MouseOff(&mouse);
        Bar1(895, 0, 1019, 50, 56603);
        prt_hz24d(935, 12, "退出", 0x0000, "HZK\\Hzk24h");
        MouseOn(mouse);
        flag = 1;
    }
    if (flag == 0 && MouseIn(0, 0, 124, 50)) //返回框反馈
    {
        MouseOff(&mouse);
        Bar1(0, 0, 124, 50, 56603);
        prt_hz24d(38, 13, "返回", 0x0000, "HZK\\Hzk24h");
        MouseOn(mouse);
        flag = 1;
```

```
}
if (flag == 0 && MouseIn(472, 580, 572, 630)) //确认键反馈
{
    MouseOff(&mouse);
    Bar1(472, 580, 572, 630, 57083);
    prt_hz24d(497, 592, "确认", 0, "HZK\\Hzk24h");
    MouseOn(mouse);
    flag = 1;
}

else if (flag == 1 && !(MouseIn(0, 0, 124, 50))
                   && !(MouseIn(895, 0, 1019, 50))
                   && !(MouseIn(472, 580, 572, 630)))
{
    MouseOff(&mouse);
    Bar1(895, 0, 1019, 50, 59004);
    Bar1(0, 0, 124, 50, 59004);
    Bar1(472, 580, 572, 630, 0xffff);

    prt_hz24d(497, 592, "确认", 0, "HZK\\Hzk24h");
    prt_hz24d(38, 13, "返回", 0x0000, "HZK\\Hzk24h");
    prt_hz24d(935, 12, "退出", 0x0000, "HZK\\Hzk24h");

    MouseOn(mouse);
    flag = 0;
}
else if (MousePress(384, 502, 680, 528)) //点击密码输入框
{
    MouseOff(&mouse);
    Getcode(384, 502, code, 10);
    MouseOn(mouse);
}
if (MousePress(472, 580, 572, 630)) //点击确认键
{
    if (code[0] == '\0')
    {
        prt_hz24d(450, 664, "请输入密码！", 0, "HZK\\Hzk24h");
        delay(1000);
        Bar1(450, 664, 660, 688, 0x9eef);
    }
    else
    {
        key = admin_password(code);
        if (key == 1)
```

```
            {
                prt_hz24d(480, 664, "登录成功! ", 0, "HZK\\Hzk24h");
                delay(1000);
                return 8; //进入管理员主菜单界面
            }
            if (key != 1)
            {
                prt_hz24d(450, 664, "密码错误! ", 0, "HZK\\Hzk24h");
                delay(1000);
                Bar1(450, 664, 560, 688, 0x9eef);
            }
        }
    }
    if (MousePress(0, 0, 150, 55)) //点击返回
    {
        return 0;
    }
    else if (MousePress(874, 0, 1024, 55)) //点击退出
    {
        CloseSVGA();
        delay(1000);
        exit(1);
    }
  }
}
```

AREVIEW.C

```
/*******************************************
FileName: areview.c
Author:   韩海若
Date:    2020/10/29
Description: 该程序文件用于通报数据审核
Function List:
       admin_review(void); //通报数据审核函数
       judge_redata(void); //判断是否有待审核通报
*******************************************/
#include "areview.h"
#include "common.h"

/*********************************
Function: admin_review()
Description: 通报数据审核函数
Calls:  judge_redata
```

```
       wr_database
Called by: main()
Table Accessed: userepo.DAT
Table Updated:  userepo.DAT
*******************************/
int admin_review(void)
{
    int flg = 0;
    FILE *fp;
    DAY *d = NULL;
    int i, j;
    int length;
    char key = 0;

    Bar1(0, 0, 1024, 768, 0xa794);

    if ((fp = fopen("userepo.dat", "rb+")) == NULL)
    { //打开 userepo.dat

        printf("\nUserInfo.dat error file");
        delay(3000);
        exit(1); //打开失败，退出程序
    }

    if ((d = (DAY *)malloc(sizeof(DAY))) == NULL)
    { //初始化当日信息结构体

        printf("memory error JudgeSameName");
        delay(3000);
        exit(1); //分配空间不足，退出程序
    }

    fseek(fp, 0, SEEK_END); //移至文件末尾
    length = ftell(fp) / sizeof(DAY); //计算通报个数

    for (i = 0; i < length; i++)
    {
        memset(d, '\0', sizeof(DAY));   //初始化结构体
        fseek(fp, i * sizeof(DAY), SEEK_SET);   //移至当前通报数据
        fread(d, sizeof(DAY), 1, fp);   //将数据写进结构体
        if (d->infostate == '0')       //如果未审核则跳出
        {
            break;
        }
```

```
}

if (fclose(fp) != 0) //关闭文件
{
    printf("\nError in closing file .");
    delay(3000);
    exit(1);
}

key = judge_redata();   //判断是否有待审核通报

if (key == '1')
{
    prt_hz16_size(420, 300, 2, 2, "无数据待审核", 0, "HZK\\Hzk16s");
    prt_hz16_size(375, 350, 2, 2, "将自动返回上一界面", 0, "HZK\\Hzk16s");
    delay(2000);
    if (d != NULL)
    {
        free(d);
        d = NULL;
    }
    return 8;
}

//标题
prt_hz24d(40, 50, "用户名", 0, "HZK\\Hzk24h");
prt_hz24d(250, 50, "年", 0, "HZK\\Hzk24h");
prt_hz24d(320, 50, "月", 0, "HZK\\Hzk24h");
prt_hz24d(400, 50, "日", 0, "HZK\\Hzk24h");
prt_hz24d(500, 50, "新增确诊", 0, "HZK\\Hzk24h");
prt_hz24d(650, 50, "新增死亡", 0, "HZK\\Hzk24h");
prt_hz24d(800, 50, "新增治愈", 0, "HZK\\Hzk24h");
prt_hz24d(920, 50, "数据区域", 0, "HZK\\Hzk24h");

Line3(0, 98, 1024, 98, 2, 0);
Bar1(0, 100, 1024, 198, 14137); //突出第一行

Line3(0, 198, 1024, 198, 2, 0);
Line3(0, 298, 1024, 298, 2, 0);
Line3(0, 398, 1024, 398, 2, 0);
Line3(0, 498, 1024, 498, 2, 0);

Bar1(0, 666, 100, 768, 0x5b4f);
Bar1(0, 668, 98, 768, 0x37dc);
```

```
   prt_hz24d(25, 706, "返回", 0, "HZK\\Hzk24h");
   Bar1(924, 666, 1024, 768, 0x5b4f);
   Bar1(926, 668, 1024, 768, 0x37dc);
   prt_hz24d(951, 706, "退出", 0, "HZK\\Hzk24h");
   Bar1(204, 666, 408, 768, 0x5b4f);
   Bar1(206, 668, 406, 768, 0xfd84);
   prt_hz24d(258, 706, "采纳数据", 0, "HZK\\Hzk24h");
   Bar1(616, 666, 820, 768, 0x5b4f);
   Bar1(618, 668, 818, 768, 0xfd84);
   prt_hz24d(670, 706, "删除数据", 0, "HZK\\Hzk24h");

   prt_hz16_asc16_size(20, 20 + 100, 2, 2, d->username, 0, "HZK\\Hzk16k");
   prt_hz16_asc16_size(40 + 200, 20 + 100, 2, 2, d->year, 0, "HZK\\Hzk16k");
   prt_hz16_asc16_size(40 + 200 + 100, 20 + 100, 2, 2, d->month, 0,
"HZK\\Hzk16k");
   prt_hz16_asc16_size(40 + 200 + 100 + 60, 20 + 100, 2, 2, d->day, 0,
"HZK\\Hzk16k");
   prt_hz16_asc16_size(520, 20 + 100, 2, 2, d->infestor, 0, "HZK\\Hzk16k");
   prt_hz16_asc16_size(670, 20 + 100, 2, 2, d->death, 0, "HZK\\Hzk16k");
   prt_hz16_asc16_size(820, 20 + 100, 2, 2, d->heal, 0, "HZK\\Hzk16k");
   if (d->area[0] == '0')
   {
      prt_hz24(920, 20 + 100, "益民小区", 0x0000, "HZK\\Hzk24k");
   }
   if (d->area[0] == '1')
   {
      prt_hz24(920, 20 + 100, "北门小区", 0x0000, "HZK\\Hzk24k");
   }
   if (d->area[0] == '2')
   {
      prt_hz24(920, 20 + 100, "光明小区", 0x0000, "HZK\\Hzk24k");
   }
   if (d->area[0] == '3')
   {
      prt_hz24(920, 20 + 100, "桂园小区", 0x0000, "HZK\\Hzk24k");
   }
   if (d->area[0] == '4')
   {
      prt_hz24(920, 20 + 100, "南池小区", 0x0000, "HZK\\Hzk24k");
   }
   if (d->area[0] == '5')
   {
      prt_hz24(920, 20 + 100, "丽景小区", 0x0000, "HZK\\Hzk24k");
   }
```

```
if (d->area[0] == '6')
{
    prt_hz24(920, 20 + 100, "风韵荷都", 0x0000, "HZK\\Hzk24k");
}
if (d->area[0] == '7')
{
    prt_hz24(920, 20 + 100, "恒大名都", 0x0000, "HZK\\Hzk24k");
}

if (d != NULL)
{
    free(d);
    d = NULL;
}

Mouse_Init();
while (1)
{
    MouseShow(&mouse);
    if (flg == 0 && MouseIn(926, 668, 1024, 768)) //退出框反馈
    {
        MouseOff(&mouse);
        Bar1(926, 668, 1024, 768, 0x9ea0);
        prt_hz24d(951, 706, "退出", 0x0000, "HZK\\Hzk24h");
        MouseOn(mouse);
        flg = 1;
    }
    if (flg == 0 && MouseIn(0, 668, 98, 768)) //返回框反馈
    {
        MouseOff(&mouse);
        Bar1(0, 668, 98, 768, 0x9ea0);
        prt_hz24d(25, 706, "返回", 0x0000, "HZK\\Hzk24h");
        MouseOn(mouse);
        flg = 1;
    }
    if (flg == 0 && MouseIn(206, 668, 406, 768)) //采纳数据键反馈
    {
        MouseOff(&mouse);
        Bar1(206, 668, 406, 768, 0x2f44);
        prt_hz24d(258, 706, "采纳数据", 0, "HZK\\Hzk24h");
        MouseOn(mouse);
        flg = 1;
    }
    if (flg == 0 && MouseIn(618, 668, 818, 768)) //删除键反馈
```

```
{
    MouseOff(&mouse);
    Bar1(618, 668, 818, 768, 0x2f44);
    prt_hz24d(670, 706, "删除数据", 0, "HZK\\Hzk24h");
    MouseOn(mouse);
    flg = 1;
}

else if (flg == 1 && !(MouseIn(926, 668, 1024, 768))
                  && !(MouseIn(0, 668, 98, 768))
                  && !(MouseIn(206, 668, 406, 768))
                  && !(MouseIn(618, 668, 818, 768)))
{
    MouseOff(&mouse);
    Bar1(926, 668, 1024, 768, 0xccff);
    Bar1(0, 668, 98, 768, 0xccff);
    Bar1(206, 668, 406, 768, 0xffff);
    Bar1(618, 668, 818, 768, 0xffff);
    prt_hz24d(951, 706, "退出", 0x0000, "HZK\\Hzk24h");
    prt_hz24d(25, 706, "返回", 0x0000, "HZK\\Hzk24h");
    prt_hz24d(258, 706, "采纳数据", 0, "HZK\\Hzk24h");
    prt_hz24d(670, 706, "删除数据", 0, "HZK\\Hzk24h");
    MouseOn(mouse);
    flg = 0;
}
else if (MousePress(206, 668, 406, 768)) //点击采纳数据
{
    if ((fp = fopen("userepo.dat", "rb+")) == NULL)
    {
        printf("\nUserInfo.dat error file");
        delay(3000);
        exit(1); //打开失败，退出程序
    }

    if ((d = (DAY *)malloc(sizeof(DAY))) == NULL)
    {
        printf("memory error JudgeSameName");
        delay(3000);
        exit(1); //分配空间不足，退出程序
    }

    fseek(fp, 0, SEEK_END);
    length = ftell(fp) / sizeof(DAY);
```

```
    for (i = 0; i < length; i++)
    {
        memset(d, '\0', sizeof(DAY));
        fseek(fp, i * sizeof(DAY), SEEK_SET);
        fread(d, sizeof(DAY), 1, fp);
        if (d->infostate == '0')
        {
            d->infostate = '1'; //审核通过将参数置为1
            break;
        }
    }

    fseek(fp, i * sizeof(DAY), SEEK_SET);

    fwrite(d, sizeof(DAY), 1, fp);  //在文件中重设用户通报状态

    wr_database(d); //将通报的数据写入文件

    prt_hz24d(458, 706, "审核通过", 0, "HZK\\Hzk24h");
    delay(1000);

    if (d != NULL)
    {
        free(d);
        d = NULL;
    }
    if (fclose(fp) != 0) //关闭文件
    {
        printf("\nError in closing file .");
        delay(3000);
        exit(1);
    }
    return 10;
}
else if (MousePress(618, 668, 818, 768)) //点击删除数据
{
    if ((fp = fopen("userepo.dat", "rb+")) == NULL)
    {
        printf("\nUserInfo.dat error file");
        delay(3000);
        exit(1); //打开失败，退出程序
    }
    if ((d = (DAY *)malloc(sizeof(DAY))) == NULL)
    {
```

```
        printf("memory error JudgeSameName");
        delay(3000);
        exit(1); //分配空间不足，退出程序
    }

    fseek(fp, 0, SEEK_END);
    length = ftell(fp) / sizeof(DAY);

    for (i = 0; i < length; i++)
    {
        memset(d, '\0', sizeof(DAY));
        fseek(fp, i * sizeof(DAY), SEEK_SET);
        fread(d, sizeof(DAY), 1, fp);
        if (d->infostate == '0')
        {
            d->infostate = '2';
            break;
        }
    }

    fseek(fp, i * sizeof(DAY), SEEK_SET);
    fwrite(d, sizeof(DAY), 1, fp); //在文件中重设用户通报状态

    prt_hz24d(458, 706, "数据有误", 0, "HZK\\Hzk24h");
    delay(1000);

    if (d != NULL)
    {
        free(d);
        d = NULL;
    }
    if (fclose(fp) != 0) //关闭文件
    {
        printf("\nError in closing file .");
        delay(3000);
        exit(1);
    }
    return 10;
}

if (MousePress(926, 668, 1024, 768)) //点击退出
{
    CloseSVGA();
    delay(1000);
```

```
            exit(1);
        }
        else if (MousePress(0, 668, 98, 768)) //点击返回
        {
            if (d != NULL)
            {
                free(d);
                d = NULL;
            }
            return 8;
        }
    }
}

/*********************************
Function: judge_redata()
Description: 通报数据审核函数
Called by: admin_review()
Table Accessed: userepo.DAT
*********************************/
char judge_redata(void)
{
    char flg = '1';
    FILE *fp;
    DAY *d = NULL; //记得 free
    int i;
    int length;

    if ((fp = fopen("userepo.dat", "rb+")) == NULL)
    {
        printf("\nUserInfo.dat error file");
        delay(3000);
        exit(1); //打开失败，退出程序
    }

    if ((d = (DAY *)malloc(sizeof(DAY))) == NULL)
    {
        printf("memory error JudgeSameName");
        delay(3000);
        exit(1); //分配空间不足，退出程序
    }

    memset(d, '\0', sizeof(DAY));   //初始化结构体
    fseek(fp, 0, SEEK_END); //移至文件末尾
```

```
    length = ftell(fp) / sizeof(DAY);   //计算通报个数

    for (i = 0; i < length; i++)
    {
        fseek(fp, i * sizeof(DAY), SEEK_SET);
        fread(d, sizeof(DAY), 1, fp);
        if (d->infostate == '0')    //如果有未审核通报
        {
            flg = '0';
            break;  //退出循环
        }
    }

    if (d != NULL)
    {
        free(d);
        d = NULL;
    }

    if (fclose(fp) != 0) //关闭文件
    {
        printf("\nError in closing file .");
        delay(3000);
        exit(1);
    }
    return flg;
}
```

ADMOD.C

```
/*******************************************
FileName: admod.c
Author:   韩海若
Date:    2020/10/29
Description: 该程序文件用于管理员的数据修改
Function List:
       admin_modify()  //数据修改主控制函数
       pop_window1     //地点与时间弹窗
       pop_window2     //输入修改数据弹窗
       click_Data_Block//点击数据框高亮
       Unhighlight     //取消高亮
       modify_MainPage //主页面绘制
       mouse_feedback_main //鼠标反馈
Others:
```

```
各地区名称序号:
    M 市 1
        任城区 2
            古槐社区 3
                益民小区 4
                北门小区 5
            仙营社区 6
                光明小区 7
                桂园小区 8
        高新区 9
            南池社区 10
                南池小区 11
                丽景小区 12
            北湖社区 13
                风韵荷都 14
                恒大名都 15
*****************************************/
#include "admod.h"
#include "choice.h"
#include "common.h"

/*********************************
Function: admin_modify()
Description: 数据修改主控制函数
Calls:  pop_window1 //地点与时间弹窗
       modify_MainPage //主页面绘制
Called by: main()
Return:   8 返回管理员主界面
*********************************/
int admin_modify(void)
{ //数据修改主函数

    int key;
    char area;

    //选择区域弹窗
    Bar1(0, 0, 1024, 768, 0xa794);
    if (pop_window1(&key, &area))//调用弹窗 1
    {
        return 8;   //点击取消则返回管理员主界面
    }

    Bar1(0, 0, 1024, 768, 0xa794);
```

```
    //显示主页面
    modify_MainPage(&key, &area);

    return 8;
}

/*********************************
Function: pop_window1()
Description: 操作选择区域和时间的两个弹窗
Calls:  choose_district //地点选择菜单
        choose_date    //时间选择
Called by: admin_modify()
Return:   0 返回 admin_modify
        1 返回管理员主界面
*********************************/
int pop_window1(int *key, char *area)
{ //选择区域和时间的两个弹窗
    Bar1(248, 168, 752, 482, 0);
    Bar1(250, 170, 750, 480, 0x8edf); //弹窗底色

    prt_hz24d(380, 200, "请选择您要修改的地区", 0, "HZK\\Hzk24h");

    Bar1(348, 413, 452, 457, 0x4b32);
    Bar1(350, 415, 450, 455, 0x8ff2);
    prt_hz24d(376, 423, "确认", 0, "HZK\\Hzk24h"); //确认键
    Bar1(548, 413, 652, 457, 0x4b32);
    Bar1(550, 415, 650, 455, 0x8ff2);
    prt_hz24d(576, 423, "取消", 0, "HZK\\Hzk24h"); //取消键

    //打开区域选择弹窗
    if (choose_district(265, 255, area)) //点击确定，进入下一弹窗
    {
        Bar1(248, 168, 752, 482, 0);
        Bar1(250, 170, 750, 480, 0x8edf); //弹窗底色
        prt_hz24d(380, 200, "请选择您要修改的日期", 0, "HZK\\Hzk24h");
        Bar1(348, 413, 452, 457, 0x4b32);
        Bar1(350, 415, 450, 455, 0x8ff2);
        prt_hz24d(376, 423, "确认", 0, "HZK\\Hzk24h"); //确认键
        Bar1(548, 413, 652, 457, 0x4b32);
        Bar1(550, 415, 650, 455, 0x8ff2);
        prt_hz24d(576, 423, "取消", 0, "HZK\\Hzk24h"); //取消键

        //打开时间选择窗口
        if (choose_date(345, 255, key))
```

```
            return 0;
        else    //取消选择时间，返回地点选择弹窗
            pop_window1(key, area);
    }
    else
        return 1; //回到管理员主界面
}

/*********************************
Function: modify_MainPage()
Description: 修改数据功能主页面
Calls:  pop_window2     //输入修改数据弹窗
        click_Data_Block  //点击数据框高亮
Called by: admin_modify()
Table Accessed: i.DAT   (选中地区的数据文件)
*********************************/
void modify_MainPage(int *key, char *area)
{ //修改数据功能主页面

    int flg = 0, datKey = -1;
    int block_flg = 1;
    char filename[8];
    const char t[4] = {".DAT"};
    SAVEDAY *d = NULL;
    FILE *fp;

    filename[0] = *area;
    filename[1] = '\0';
    strcat(filename, t);
    filename[6] = '\0';
    if ((d = (SAVEDAY *)malloc(sizeof(SAVEDAY))) == NULL)
    {//为结构体分配内存

        delay(3000);
        exit(1); //分配空间不足，退出程序
    }
    if ((fp = fopen(filename, "rb+")) == NULL)
    {//打开文件
        put_asc16_size(185 + 150 + 80 + 80, 600, 2, 2, filename, 0x0000);
        delay(3000);
        exit(1); //打开失败，退出程序
    }

    //移至选中的日期
```

```
fseek(fp, (*key) * sizeof(SAVEDAY), SEEK_SET);

//将数据读进结构体
fread(d, sizeof(SAVEDAY), 1, fp);

//表格
Line3(0, 98, 1024, 98, 2, 0);
Line3(0, 198, 1024, 198, 2, 0);
Line3(0, 298, 1024, 298, 2, 0);
Line3(0, 398, 1024, 398, 2, 0);
Line3(0, 498, 1024, 498, 2, 0);
Line3(0, 598, 1024, 598, 2, 0);
Line3(512, 0, 512, 598, 2, 0); //竖线
Bar1(0, 666, 100, 768, 0x5b4f);
Bar1(0, 668, 98, 768, 0x37dc);
prt_hz24d(25, 706, "返回", 0, "HZK\\Hzk24h");
Bar1(924, 666, 1024, 768, 0x5b4f);
Bar1(926, 668, 1024, 768, 0x37dc);
prt_hz24d(951, 706, "退出", 0, "HZK\\Hzk24h");
Bar1(410, 666, 614, 768, 0x5b4f);
Bar1(412, 668, 612, 768, 65523);
prt_hz24d(462, 706, "修改数据", 0, "HZK\\Hzk24h");

prt_hz24d(160, 50, "当日确诊", 0, "HZK\\Hzk24h");
put_asc16_size(160, 50 + 100, 2, 2, d->infestor, 0x0000);
prt_hz24d(160, 250, "当日治愈", 0, "HZK\\Hzk24h");
put_asc16_size(160, 250 + 100, 2, 2, d->heal, 0x0000);
prt_hz24d(160, 450, "当日死亡", 0, "HZK\\Hzk24h");
put_asc16_size(160, 450 + 100, 2, 2, d->death, 0x0000);
prt_hz24d(512 + 160, 50, "累计确诊", 0, "HZK\\Hzk24h");
put_asc16_size(512 + 160, 50 + 100, 2, 2, d->allinf, 0x0000);
prt_hz24d(512 + 160, 250, "累计治愈", 0, "HZK\\Hzk24h");
put_asc16_size(512 + 160, 250 + 100, 2, 2, d->allhea, 0x0000);
prt_hz24d(512 + 160, 450, "累计死亡", 0, "HZK\\Hzk24h");
put_asc16_size(512 + 160, 450 + 100, 2, 2, d->alldea, 0x0000);

Mouse_Init();
while (1)
{
    MouseShow(&mouse);
    if (flg == 0 && MouseIn(926, 668, 1024, 768)) //退出框反馈
    {
        MouseOff(&mouse);
        Bar1(926, 668, 1024, 768, 14137);
```

```
    prt_hz24d(951, 706, "退出", 0x0000, "HZK\\Hzk24h");
    MouseOn(mouse);
    flg = 1;
}
if (flg == 0 && MouseIn(0, 668, 98, 768)) //返回框反馈
{
    MouseOff(&mouse);
    Bar1(0, 668, 98, 768, 14137);
    prt_hz24d(25, 706, "返回", 0x0000, "HZK\\Hzk24h");
    MouseOn(mouse);
    flg = 1;
}
if (flg == 0 && MouseIn(412, 668, 612, 768)) //修改数据键反馈
{
    MouseOff(&mouse);
    Bar1(412, 668, 612, 768, 65184);
    prt_hz24d(462, 706, "修改数据", 0, "HZK\\Hzk24h");
    MouseOn(mouse);
    flg = 1;
}
else if (flg == 1 && !(MouseIn(926, 668, 1024, 768))
                  && !(MouseIn(0, 668, 98, 768))
                  && !(MouseIn(412, 668, 612, 768)))
{
    MouseOff(&mouse);
    Bar1(926, 668, 1024, 768, 0x37dc);
    Bar1(0, 668, 98, 768, 0x37dc);
    Bar1(412, 668, 612, 768, 65523);
    prt_hz24d(462, 706, "修改数据", 0, "HZK\\Hzk24h");
    prt_hz24d(951, 706, "退出", 0x0000, "HZK\\Hzk24h");
    prt_hz24d(25, 706, "返回", 0x0000, "HZK\\Hzk24h");
    MouseOn(mouse);
    flg = 0;
}

//点击"修改数据"
else if (MousePress(412, 668, 612, 768) && datKey != -1)
{
    block_flg = 0;
    MouseOff(&mouse);
    pop_window2(datKey, d, area);
    Bar1(0, 0, 1024, 768, 0xa794);
    modify_MainPage(key, area); //重新打开界面
    MouseOn(mouse);
```

```
            block_flg = 1;
        }

        else if (MousePress(926, 668, 1024, 768)) //点击退出
        {
            CloseSVGA();
            delay(1000);
            exit(1);
        }

        else if (MousePress(0, 668, 98, 768)) //点击返回
        {
            if (fclose(fp) != 0) //关闭文件
            {
                printf("\nError in closing file .");
                delay(3000);
                exit(1);
            }
            free(d);
            return;
        }
        if (block_flg) //点击数据块反馈
            click_Data_Block(&datKey);
    }
}

/*********************************
Function: pop_window2()
Description: 输入修改数据弹窗
Calls:  modify_database //修改数据
Called by: modify_MainPage()
*********************************/
void pop_window2(int datKey, SAVEDAY *d, char *area)
{ //输入数据弹窗
    char modify_number[5 + 1] = "\0";
    int flg = 0;

    Bar1(248, 168, 752, 482, 0);
    Bar1(250, 170, 750, 480, 0x8edf); //弹窗底色

    prt_hz24d(390, 200, "请输入修改后数据", 0, "HZK\\Hzk24h");
    Bar1(348, 413, 452, 457, 0x4b32);
    Bar1(350, 415, 450, 455, 0x8ff2);
```

```
    prt_hz24d(376, 423, "确认", 0, "HZK\\Hzk24h"); //确认键
    Bar1(548, 413, 652, 457, 0x4b32);
    Bar1(550, 415, 650, 455, 0x8ff2);
    prt_hz24d(576, 423, "取消", 0, "HZK\\Hzk24h"); //取消键

    Bar1(424, 258, 559, 292, 0);
    Bar1(426, 260, 557, 290, 0xffff); //输入框

    Mouse_Init();
    while (1)
    {
        MouseShow(&mouse);

        if (flg == 0 && MouseIn(350, 415, 450, 455)) //确认框反馈
        {
            MouseOff(&mouse);
            Bar2(350, 415, 450, 455, 800);
            prt_hz24d(376, 423, "确认", 0, "HZK\\Hzk24h"); //确认键
            MouseOn(mouse);
            flg = 1;
        }

        else if (flg == 0 && MouseIn(550, 415, 650, 455)) //取消框反馈
        {
            MouseOff(&mouse);
            Bar2(550, 415, 650, 455, 800);
            prt_hz24d(576, 423, "取消", 0, "HZK\\Hzk24h"); //取消
            MouseOn(mouse);
            flg = 1;
        }

        else if (flg == 1 && !(MouseIn(350, 415, 450, 455)) && !(MouseIn(550,
415, 650, 455)))
        {
            MouseOff(&mouse);
            Bar1(350, 415, 450, 455, 0x8ff2);
            prt_hz24d(376, 423, "确认", 0, "HZK\\Hzk24h"); //确认键
            Bar1(550, 415, 650, 455, 0x8ff2);
            prt_hz24d(576, 423, "取消", 0, "HZK\\Hzk24h"); //取消键
            MouseOn(mouse);

            flg = 0;
        }
```

```
        else if (MousePress(550, 415, 650, 455)) //点击取消键
        {
            return;
        }

        else if (MousePress(350, 415, 450, 455)) //点击确定键
        {
            modify_database(d, area);
            return;
        }

        if (MousePress(426, 260, 557, 290)) //点击输入框
        {
            Bar1(426, 260, 557, 290, 0xffff);
            MouseOff(&mouse);
            Getinfo(430, 260, modify_number, 5); //得到数据
            if (datKey == 1)
                strcpy(d->infestor, modify_number);
            if (datKey == 2)
                strcpy(d->heal, modify_number);
            if (datKey == 3)
                strcpy(d->death, modify_number);
            MouseOn(mouse);
        }
    }
}

/*********************************
Function: click_Data_Block()
Description: 点击高亮选择区域块
*********************************/
void click_Data_Block(int *datKey)
{                               //点击高亮选择区域块
                                //点击数据块
    if (MousePress(0, 0, 510, 196)) //点击“当日确诊”
    {
        Unhighlight();
        Bar2(0, 0, 510, 196, 7327);
        *datKey = 1;
    }

    else if (MousePress(0, 200, 510, 396)) //点击“当日治愈”
    {
        Unhighlight();
```

```
        Bar2(0, 200, 510, 396, 7327);
        *datKey = 2;
    }

    else if (MousePress(0, 400, 510, 596)) //点击“当日死亡”
    {
        Unhighlight();
        Bar2(0, 400, 510, 596, 7327);
        *datKey = 3;
    }
}

/*********************************
Function: Unhighlight()
Description: 取消数据块高亮
*********************************/
void Unhighlight(void)
{ //取消数据块高亮

    Bar2(0, 0, 510, 196, 0xa794);
    Bar2(0, 200, 510, 396, 0xa794);
    Bar2(0, 400, 510, 596, 0xa794);
    Bar2(514, 2, 1024, 196, 0xa794);
    Bar2(514, 200, 1024, 396, 0xa794);
    Bar2(514, 400, 1024, 596, 0xa794);
}

/*********************************
Function: mouse_feedback_main()
Description: 鼠标反馈
*********************************/
void mouse_feedback_main(void)
{ //鼠标反馈
    int flg = 0;

    if (flg == 0 && MouseIn(926, 668, 1024, 768)) //退出框反馈
    {
        MouseOff(&mouse);
        Bar1(926, 668, 1024, 768, 14137);
        prt_hz24d(951, 706, "退出", 0x0000, "HZK\\Hzk24h");
        MouseOn(mouse);
        flg = 1;
    }
    if (flg == 0 && MouseIn(0, 668, 98, 768)) //返回框反馈
```

```
    {
        MouseOff(&mouse);
        Bar1(0, 668, 98, 768, 14137);
        prt_hz24d(25, 706, "返回", 0x0000, "HZK\\Hzk24h");
        MouseOn(mouse);
        flg = 1;
    }
    if (flg == 0 && MouseIn(412, 668, 612, 768)) //修改数据键反馈
    {
        MouseOff(&mouse);
        Bar1(412, 668, 612, 768, 65184);
        prt_hz24d(462, 706, "修改数据", 0, "HZK\\Hzk24h");
        MouseOn(mouse);
        flg = 1;
    }
    else if (flg == 1 && !(MouseIn(926, 668, 1024, 768))
                 && !(MouseIn(0, 668, 98, 768))
                 && !(MouseIn(412, 668, 612, 768)))
    {
        MouseOff(&mouse);
        Bar1(926, 668, 1024, 768, 0x37dc);
        Bar1(0, 668, 98, 768, 0x37dc);
        Bar1(412, 668, 612, 768, 65523);
        prt_hz24d(462, 706, "修改数据", 0, "HZK\\Hzk24h");
        prt_hz24d(951, 706, "退出", 0x0000, "HZK\\Hzk24h");
        prt_hz24d(25, 706, "返回", 0x0000, "HZK\\Hzk24h");
        MouseOn(mouse);
        flg = 0;
    }
}
```

DRAW.C

```
/*******************************************
FileName: draw.h
Author:   韩海若
Date:    2020/10/28
Description: 该程序文件用于绘制可视化界面的建筑
           和排行中的条形图
Function List:
       void draw_floor1(int x, int y); //绘制 visual_basic 中的建筑
       void draw_floor2(int x, int y);
       void draw_floor3(int x, int y);
       void draw_axis(int x, int y); //坐标轴
```

```
       void draw_column(int x, int y, int len); //条形图
*****************************************/
#include "draw.h"

/************************
Function: draw_floor1
Description: 绘制建筑
Called by: visual_basic()
Input: (x, y)为图形左上角位置
*************************/
void draw_floor1(int x, int y)
{
    int i;
    Line3(x,y,x+60,y,2,0);
    Line3(x,y,x,y+100,2,0);
    Line3(x+60,y+3,x+60,y+97,4,0);
    Line3(x,y+100,x+60,y+100,2,0);
    for(i = 8; i<80; i+=8){
        Line3(x,y+i,x+60,y+i,1,0);
    }
    for(i = 8; i<60; i+=8){
        Line3(x+i,y,x+i,y+72,1,0);
    }
    Line3(x+20,y+80,x+30,y+80,1,0);
    Line3(x+20,y+80,x+20,y+100,1,0);
    Line3(x+30,y+80,x+30,y+100,1,0);

    Line3(x+30,y+80,x+40,y+80,1,0);
    Line3(x+30,y+80,x+30,y+100,1,0);
    Line3(x+40,y+80,x+40,y+100,1,0);

    Line3(x+60,y+50,x+120,y+50,2,0);
    Line3(x+60,y+100,x+120,y+100,2,0);
    Line3(x+120,y+50,x+120,y+100,2,0);

    for(i = 68; i<120; i+=8){
        Line3(x+i,y+50,x+i,y+82,1,0);
    }
    for(i = 58; i<90; i+=8){
        Line3(x+60,y+i,x+120,y+i,1,0);
    }

    Line3(x+80,y+86,x+90,y+86,1,0);
    Line3(x+80,y+86,x+80,y+100,1,0);
```

```
    Line3(x+90,y+86,x+90,y+100,1,0);

    Line3(x+90,y+86,x+100,y+86,1,0);
    Line3(x+90,y+86,x+90,y+100,1,0);
    Line3(x+100,y+86,x+100,y+100,1,0);
}

/************************
Function: draw_floor2
Description: 绘制建筑
Called by: visual_basic()
Input: (x, y)为图形左上角位置
*************************/
void draw_floor2(int x, int y){

    Line3(x-2,y-2,x+72,y,4,0);
    Line3(x,y,x,y+100,2,0);
    Line3(x+70,y,x+70,y+97,4,0);
    Line3(x,y+100,x+70,y+100,4,0);

    Line3(x+2,y-5,x+68,y-5,3,0);

    Line3(x-2,y+50,x+74,y+50,2,0);

    Line3(x+10,y+10,x+10,y+30,1,0);
    Line3(x+25,y+10,x+25,y+30,1,0);
    Line3(x+10,y+10,x+25,y+10,1,0);
    Line3(x+10,y+30,x+25,y+30,1,0);

    Line3(x+42,y+10,x+42,y+30,1,0);
    Line3(x+57,y+10,x+57,y+30,1,0);
    Line3(x+42,y+10,x+57,y+10,1,0);
    Line3(x+42,y+30,x+57,y+30,1,0);

    Line3(x+10,y+60,x+10,y+80,1,0);
    Line3(x+25,y+60,x+25,y+80,1,0);
    Line3(x+10,y+60,x+25,y+60,1,0);
    Line3(x+10,y+80,x+25,y+80,1,0);

    Line3(x+40,y+70,x+40,y+100,1,0);
    Line3(x+57,y+70,x+57,y+100,1,0);
    Line3(x+40,y+70,x+57,y+70,1,0);
    Line3(x+48,y+70,x+48,y+100,1,0);
}
```

```
/************************
Function: draw_floor3
Description: 绘制建筑
Called by: visual_basic()
Input: (x, y)为图形左上角位置
*************************/
void draw_floor3(int x, int y)
{
    int i;
    Line3(x, y, x+80, y, 2, 0);
    Line3(x, y, x, y+100, 2, 0);
    Line3(x+80, y, x+80, y+100, 2, 0);
    Line3(x, y+100, x+80, y+100, 2, 0);

    for(i = 0; i < 4; i++)
    {
        Line2(x-i,y,x+41-i,y-32,0);
    }

    for(i = 0; i < 6; i++)
    {
        Line2(x+38+i,y-32,x+78+i,y,0);
    }

    Line3(x+10,y+10,x+10,y+30,1,0);
    Line3(x+30,y+10,x+30,y+30,1,0);
    Line3(x+10,y+10,x+30,y+10,1,0);
    Line3(x+10,y+30,x+30,y+30,1,0);

    Line3(x+50,y+10,x+50,y+30,1,0);
    Line3(x+70,y+10,x+70,y+30,1,0);
    Line3(x+50,y+10,x+70,y+10,1,0);
    Line3(x+50,y+30,x+70,y+30,1,0);

    Line3(x-3, y+50,x+83, y+50, 3, 0);

    Line3(x+10,y+60,x+10,y+80,1,0);
    Line3(x+30,y+60,x+30,y+80,1,0);
    Line3(x+10,y+60,x+30,y+60,1,0);
    Line3(x+10,y+80,x+30,y+80,1,0);

    Line3(x+35+10,y+65,x+35+10,y+100,1,0);
    Line3(x+58+10,y+65,x+58+10,y+100,1,0);
```

```
    Line3(x+35+10,y+65,x+58+10,y+65,1,0);
    Line3(x+63,y+77,x+63,y+80,1,0);
}

/************************
Function: draw_axis
Description: 绘制坐标轴
Input: (x, y)为图形原点位置
*************************/
void draw_axis(int x, int y){
    Line3(x, y-330, x, y, 2, 0);
    Line3(x, y, x+675, y, 2, 0);

    Line2(x-15, y-310, x, y-332,0);
    Line2(x-15, y-309, x, y-331,0);
    Line2(x-15, y-308, x, y-330,0);

    Line2(x, y-332, x+15, y-310,0);
    Line2(x, y-331, x+15, y-309,0);
    Line2(x, y-330, x+15, y-308,0);

    Line2(x+655, y-15, x+675, y, 0);
    Line2(x+656, y-15, x+676, y, 0);
    Line2(x+657, y-15, x+677, y, 0);

    Line2(x+655, y+15, x+675, y, 0);
    Line2(x+656, y+15, x+676, y, 0);
    Line2(x+657, y+15, x+677, y, 0);
}

/************************
Function: draw_column
Description: 绘制条形图
Called by: showinfo_main()
Input: (x, y)为图形左上角位置
*************************/
void draw_column(int x, int y, int len)
{
    Bar1(x-2, y-2, x+len+2,y+20+2, 0xffff);
    Bar1(x, y, x+len,y+20, 64518);
}
```

CHOICE.C

```
/*****************************************
FileName: choice.c
Author:   韩海若
Date:    2020/10/28
Description: 该程序文件用于实现地区分类下拉菜单
           与输入时间功能
Function List:
       choose_district //下拉控制函数
       choose_date     //输入要查询的日期
       dis_bar        //下拉菜单的选择框
*****************************************/
#include "choice.h"
#include "common.h"
#include "drawc.h" //使用 findday 函数

/***********************************
Function: choose_district
Description: 下拉控制函数
           参数 area 为地区对应序号
Called by: pop_window1()
***********************************/
int choose_district(int x, int y, char *area) //(x,y)为左上角坐标
{
    int flg = 0;
    int choice = 0;     //选项标记
    int pre_flg = 0;    //下拉控制参数
    int quit_flg = 0;   //退出参数
    int choose_flg = 0; //判断是否已选择区域

    //四个区域对应的选择框
    Bar1(x - 3, y - 3, x + 73, y + 33, 0);
    Bar1(x, y, x + 70, y + 30, 0xffff);
    prt_hz24d(x + 20, y + 5, "市", 0, "HZK\\Hzk24h");

    Bar1(x + 92, y - 3, x + 188, y + 33, 0);
    Bar1(x + 95, y, x + 185, y + 30, 0xffff);
    prt_hz24d(x + 125, y + 5, "区", 0, "HZK\\Hzk24h");

    Bar1(x + 207, y - 3, x + 323, y + 33, 0);
    Bar1(x + 210, y, x + 320, y + 30, 0xffff);
    prt_hz24d(x + 240, y + 5, "社区", 0, "HZK\\Hzk24h");
```

```
Bar1(x + 342, y - 3, x + 458, y + 33, 0);
Bar1(x + 345, y, x + 455, y + 30, 0xffff);
prt_hz24d(x + 375, y + 5, "小区", 0, "HZK\\Hzk24h");

Mouse_Init();
while (1)
{
    MouseShow(&mouse);
    if (flg == 0 && MouseIn(350, 415, 450, 455)) //确认框反馈
    {
        MouseOff(&mouse);
        Bar2(350, 415, 450, 455, 800);
        prt_hz24d(376, 423, "确认", 0, "HZK\\Hzk24h"); //确认键
        MouseOn(mouse);
        flg = 1;
    }

    if (flg == 0 && MouseIn(550, 415, 650, 455)) //取消框反馈
    {
        MouseOff(&mouse);
        Bar2(550, 415, 650, 455, 800);
        prt_hz24d(576, 423, "取消", 0, "HZK\\Hzk24h"); //取消
        MouseOn(mouse);
        flg = 1;
    }

    else if (flg == 1 && !(MouseIn(350, 415, 450, 455))
                    && !(MouseIn(550, 415, 650, 455)))
    {
        MouseOff(&mouse);
        Bar1(350, 415, 450, 455, 0x8ff2);
        prt_hz24d(376, 423, "确认", 0, "HZK\\Hzk24h"); //确认键
        Bar1(550, 415, 650, 455, 0x8ff2);
        prt_hz24d(576, 423, "取消", 0, "HZK\\Hzk24h"); //取消键
        MouseOn(mouse);
        flg = 0;
    }

    else if (MousePress(350, 415, 450, 455) && choose_flg) //点击确定键
    {
        return 1;
    }

    else if (MousePress(550, 415, 650, 455)) //点击取消键
```

```
{
    Bar1(x, y, x + 70, y + 30, 0xffff);
    prt_hz24d(x + 20, y + 5, "市", 0, "HZK\\Hzk24h");
    Bar1(x + 95, y, x + 185, y + 30, 0xffff);
    prt_hz24d(x + 125, y + 5, "区", 0, "HZK\\Hzk24h");
    Bar1(x + 210, y, x + 320, y + 30, 0xffff);
    prt_hz24d(x + 240, y + 5, "社区", 0, "HZK\\Hzk24h");
    Bar1(x + 345, y, x + 455, y + 30, 0xffff);
    prt_hz24d(x + 375, y + 5, "小区", 0, "HZK\\Hzk24h");

    choose_flg = 0;
    if (quit_flg)
    {
        return 0;
    }
    quit_flg = 1; //进入管理员主界面
}

/**********************
下拉框选择判断语句
***********************/

else if (MousePress(x, y, x + 70, y + 30) && choice == 0)
{//点击市，出现下拉框
    MouseOff(&mouse);
    dis_bar(x, y + 30, 70, 1);
    prt_hz24d(x + 12, y + 35, "某市", 0, "HZK\\Hzk24h");
    MouseOn(mouse);

    pre_flg = 1;
    quit_flg = 0;
}

else if (MousePress(x, y + 30, x + 70, y + 60) && pre_flg == 1)
{//选择"某市"
    MouseOff(&mouse);
    Bar1(x, y, x + 70, y + 30, 0xffff);
    prt_hz24d(x + 12, y + 5, "某市", 0, "HZK\\Hzk24h");
    Bar1(x - 3, y + 33, x + 73, y + 63, 0x8edf);

    choice = 1;
    pre_flg = 0;
    choose_flg = 1;
    MouseOn(mouse);
```

```
}

else if (MousePress(x + 95, y, x + 185, y + 30) && choice == 1)
{//点击区，出现下拉框
    MouseOff(&mouse);
    dis_bar(x + 95, y + 30, 90, 2);
    prt_hz24d(x + 105, y + 35, "任城区", 0, "HZK\\Hzk24h");
    prt_hz24d(x + 105, y + 68, "高新区", 0, "HZK\\Hzk24h");
    MouseOn(mouse);
    pre_flg = 2;
}

else if (MousePress(x + 95, y + 30, x + 185, y + 60) && pre_flg == 2)
{//选择"任城区"
    MouseOff(&mouse);
    Bar1(x + 95, y, x + 185, y + 30, 0xffff);
    prt_hz24d(x + 105, y + 5, "任城区", 0, "HZK\\Hzk24h");
    Bar1(x + 92, y + 33, x + 188, y + 100, 0x8edf);
    choice = 2;
    pre_flg = 0;
    MouseOn(mouse);
}

else if (MousePress(x + 210, y, x + 320, y + 30) && choice == 2)
{//点击社区，出现下拉框
    MouseOff(&mouse);
    dis_bar(x + 210, y + 30, 110, 2);
    prt_hz24d(x + 220, y + 35, "古槐社区", 0, "HZK\\Hzk24h");
    prt_hz24d(x + 220, y + 68, "仙营社区", 0, "HZK\\Hzk24h");
    MouseOn(mouse);
    pre_flg = 3;
}

else if (MousePress(x + 210, y + 30, x + 320, y + 60) && pre_flg == 3)
{//选择"古槐社区"
    MouseOff(&mouse);
    Bar1(x + 210, y, x + 320, y + 30, 0xffff);
    prt_hz24d(x + 220, y + 5, "古槐社区", 0, "HZK\\Hzk24h");
    Bar1(x + 207, y + 33, x + 323, y + 93, 0x8edf);
    choice = 3;
    pre_flg = 0;
    MouseOn(mouse);
}
```

```
else if (MousePress(x + 345, y, x + 455, y + 30) && choice == 3)
{//点击小区，出现下拉框
    MouseOff(&mouse);
    dis_bar(x + 345, y + 30, 110, 2);
    prt_hz24d(x + 355, y + 35, "益民小区", 0, "HZK\\Hzk24h");
    prt_hz24d(x + 355, y + 68, "北门小区", 0, "HZK\\Hzk24h");
    MouseOn(mouse);
    pre_flg = 4;
}

else if (MousePress(x + 345, y + 30, x + 455, y + 60) && pre_flg == 4)
{//选择"益民小区"
    MouseOff(&mouse);
    Bar1(x + 345, y, x + 455, y + 30, 0xffff);
    prt_hz24d(x + 355, y + 5, "益民小区", 0, "HZK\\Hzk24h");
    Bar1(x + 342, y + 33, x + 458, y + 93, 0x8edf);
    choice = 4;
    *area = '0';
    pre_flg = 0;
    MouseOn(mouse);
}

else if (MousePress(x + 345, y + 60, x + 455, y + 90) && pre_flg == 4)
{//选择"北门小区"
    MouseOff(&mouse);
    Bar1(x + 345, y, x + 455, y + 30, 0xffff);
    prt_hz24d(x + 355, y + 5, "北门小区", 0, "HZK\\Hzk24h");
    Bar1(x + 342, y + 33, x + 458, y + 93, 0x8edf);
    choice = 5;
    *area = '1';
    pre_flg = 0;
    MouseOn(mouse);
}

else if (MousePress(x + 210, y + 60, x + 320, y + 90) && pre_flg == 3)
{//选择"仙营社区"
    MouseOff(&mouse);
    Bar1(x + 210, y, x + 320, y + 30, 0xffff);
    prt_hz24d(x + 220, y + 5, "仙营社区", 0, "HZK\\Hzk24h");
    Bar1(x + 207, y + 33, x + 323, y + 93, 0x8edf);
    choice = 6;
    pre_flg = 0;
    MouseOn(mouse);
}
```

```
else if (MousePress(x + 345, y, x + 455, y + 30) && choice == 6)
{//点击小区，出现下拉框
    MouseOff(&mouse);
    dis_bar(x + 345, y + 30, 110, 2);
    prt_hz24d(x + 355, y + 35, "光明小区", 0, "HZK\\Hzk24h");
    prt_hz24d(x + 355, y + 68, "桂园小区", 0, "HZK\\Hzk24h");
    MouseOn(mouse);

    pre_flg = 4;
}

else if (MousePress(x + 345, y + 30, x + 455, y + 60) && pre_flg == 4)
{//选择"光明小区"
    MouseOff(&mouse);
    Bar1(x + 345, y, x + 455, y + 30, 0xffff);
    prt_hz24d(x + 355, y + 5, "光明小区", 0, "HZK\\Hzk24h");
    Bar1(x + 342, y + 33, x + 458, y + 93, 0x8edf);
    choice = 7;
    *area = '2';
    pre_flg = 0;
    MouseOn(mouse);
}

else if (MousePress(x + 345, y + 60, x + 455, y + 90) && pre_flg == 4)
{//选择"桂园小区"
    MouseOff(&mouse);
    Bar1(x + 345, y, x + 455, y + 30, 0xffff);
    prt_hz24d(x + 355, y + 5, "桂园小区", 0, "HZK\\Hzk24h");
    Bar1(x + 342, y + 33, x + 458, y + 93, 0x8edf);
    choice = 8;
    *area = '3';
    pre_flg = 0;
    MouseOn(mouse);
}

else if (MousePress(x + 95, y + 60, x + 185, y + 90) && pre_flg == 2)
{//选择"高新区"
    MouseOff(&mouse);
    Bar1(x + 95, y, x + 185, y + 30, 0xffff);
    prt_hz24d(x + 105, y + 5, "高新区", 0, "HZK\\Hzk24h");
    Bar1(x + 92, y + 33, x + 188, y + 100, 0x8edf);
    choice = 9;
    pre_flg = 0;
```

```
    MouseOn(mouse);
}

else if (MousePress(x + 210, y, x + 320, y + 30) && choice == 9)
{//点击社区，出现下拉框
    MouseOff(&mouse);
    dis_bar(x + 210, y + 30, 110, 2);
    prt_hz24d(x + 220, y + 35, "南池社区", 0, "HZK\\Hzk24h");
    prt_hz24d(x + 220, y + 68, "北湖社区", 0, "HZK\\Hzk24h");
    pre_flg = 5;
    MouseOn(mouse);
}

else if (MousePress(x + 210, y + 30, x + 320, y + 60) && pre_flg == 5)
{//选择"南池社区"
    MouseOff(&mouse);
    Bar1(x + 210, y, x + 320, y + 30, 0xffff);
    prt_hz24d(x + 220, y + 5, "南池社区", 0, "HZK\\Hzk24h");
    Bar1(x + 207, y + 33, x + 323, y + 93, 0x8edf);
    choice = 10;
    pre_flg = 0;
    MouseOn(mouse);
}

else if (MousePress(x + 345, y, x + 455, y + 30) && choice == 10)
{//点击小区，出现下拉框
    MouseOff(&mouse);
    dis_bar(x + 345, y + 30, 110, 2);
    prt_hz24d(x + 355, y + 35, "南池小区", 0, "HZK\\Hzk24h");
    prt_hz24d(x + 355, y + 68, "丽景小区", 0, "HZK\\Hzk24h");
    pre_flg = 7;
    MouseOn(mouse);
}

else if (MousePress(x + 345, y + 30, x + 455, y + 60) && pre_flg == 7)
{//选择"南池小区"
    MouseOff(&mouse);
    Bar1(x + 345, y, x + 455, y + 30, 0xffff);
    prt_hz24d(x + 355, y + 5, "南池小区", 0, "HZK\\Hzk24h");
    Bar1(x + 342, y + 33, x + 458, y + 93, 0x8edf);
    choice = 11;
    *area = '4';
    pre_flg = 0;
    MouseOn(mouse);
```

```
}

else if (MousePress(x + 345, y + 60, x + 455, y + 90) && pre_flg == 7)
{//选择"丽景小区"
    MouseOff(&mouse);
    Bar1(x + 345, y, x + 455, y + 30, 0xffff);
    prt_hz24d(x + 355, y + 5, "丽景小区", 0, "HZK\\Hzk24h");
    Bar1(x + 342, y + 33, x + 458, y + 93, 0x8edf);
    choice = 12;
    *area = '5';
    pre_flg = 0;
    MouseOn(mouse);
}

else if (MousePress(x + 210, y + 60, x + 320, y + 90) && pre_flg == 5)
{//选择"北湖社区"
    MouseOff(&mouse);
    Bar1(x + 210, y, x + 320, y + 30, 0xffff);
    prt_hz24d(x + 220, y + 5, "北湖社区", 0, "HZK\\Hzk24h");
    Bar1(x + 207, y + 33, x + 323, y + 93, 0x8edf);
    choice = 13;
    pre_flg = 0;
    MouseOn(mouse);
}

else if (MousePress(x + 345, y, x + 455, y + 30) && choice == 13)
{//点击小区，出现下拉框
    MouseOff(&mouse);
    dis_bar(x + 345, y + 30, 110, 2);
    prt_hz24d(x + 355, y + 35, "风韵荷都", 0, "HZK\\Hzk24h");
    prt_hz24d(x + 355, y + 68, "恒大名都", 0, "HZK\\Hzk24h");
    pre_flg = 8;
    MouseOn(mouse);
}

else if (MousePress(x + 345, y + 30, x + 455, y + 60) && pre_flg == 8)
{//选择"风韵荷都"
    MouseOff(&mouse);
    Bar1(x + 345, y, x + 455, y + 30, 0xffff);
    prt_hz24d(x + 355, y + 5, "风韵荷都", 0, "HZK\\Hzk24h");
    Bar1(x + 342, y + 33, x + 458, y + 93, 0x8edf);
    choice = 14;
    *area = '6';
    pre_flg = 0;
```

```
            MouseOn(mouse);
        }

        else if (MousePress(x + 345, y + 60, x + 455, y + 90) && pre_flg == 8)
        {//选择"恒大名都"
            MouseOff(&mouse);
            Bar1(x + 345, y, x + 455, y + 30, 0xffff);
            prt_hz24d(x + 355, y + 5, "恒大名都", 0, "HZK\\Hzk24h");
            Bar1(x + 342, y + 33, x + 458, y + 93, 0x8edf);
            choice = 15;
            *area = '7';
            pre_flg = 0;
            MouseOn(mouse);
        }
    }
}

/***********************************
Function: choose_district
Description: 输入要查询的日期
Calls:  findday   //找到某日
Called by: pop_window1()
***********************************/
int choose_date(int x, int y, int *key) //key 为输入时间对应日期信息在文件相应的位置
{

    char year[4 + 1] = "\0";
    char month[2 + 1] = "\0";
    char day[2 + 1] = "\0";

    int quit_flg = 0;
    int flg = 0;
    int choose_flg = 0;

    Bar1(x - 3, y - 3, x + 83, y + 33, 0);
    Bar1(x, y, x + 80, y + 30, 0xffff);
    prt_hz24d(x + 25, y + 5, "年", 0, "HZK\\Hzk24h");
    Bar1(x + 107, y - 3, x + 183, y + 33, 0);
    Bar1(x + 110, y, x + 180, y + 30, 0xffff);
    prt_hz24d(x + 130, y + 5, "月", 0, "HZK\\Hzk24h");
    Bar1(x + 207, y - 3, x + 293, y + 33, 0);
    Bar1(x + 210, y, x + 290, y + 30, 0xffff);
    prt_hz24d(x + 235, y + 5, "日", 0, "HZK\\Hzk24h");
```

```
Mouse_Init();
while (1)
{
    MouseShow(&mouse);
    if (flg == 0 && MouseIn(350, 415, 450, 455))
    {//确认框反馈
        MouseOff(&mouse);
        Bar2(350, 415, 450, 455, 800);
        prt_hz24d(376, 423, "确认", 0, "HZK\\Hzk24h");
        MouseOn(mouse);
        flg = 1;
    }

    if (flg == 0 && MouseIn(550, 415, 650, 455))
    { //取消框反馈
        MouseOff(&mouse);
        Bar2(550, 415, 650, 455, 800);
        prt_hz24d(576, 423, "取消", 0, "HZK\\Hzk24h");
        MouseOn(mouse);
        flg = 1;
    }

    else if (flg == 1 && !(MouseIn(350, 415, 450, 455))
                    && !(MouseIn(550, 415, 650, 455)))
    {
        MouseOff(&mouse);
        Bar1(350, 415, 450, 455, 0x8ff2);
        prt_hz24d(376, 423, "确认", 0, "HZK\\Hzk24h");
        Bar1(550, 415, 650, 455, 0x8ff2);
        prt_hz24d(576, 423, "取消", 0, "HZK\\Hzk24h");
        MouseOn(mouse);
        flg = 0;
    }

    else if (MousePress(350, 415, 450, 455) && choose_flg)
    {//点击确定键
        return 1;
    }

    else if (MousePress(550, 415, 650, 455))
    {//点击取消键
        Bar1(x - 3, y - 3, x + 83, y + 33, 0);
        Bar1(x, y, x + 80, y + 30, 0xffff);
        prt_hz24d(x + 25, y + 5, "年", 0, "HZK\\Hzk24h");
```

```
    Bar1(x + 107, y - 3, x + 183, y + 33, 0);
    Bar1(x + 110, y, x + 180, y + 30, 0xffff);
    prt_hz24d(x + 130, y + 5, "月", 0, "HZK\\Hzk24h");
    Bar1(x + 207, y - 3, x + 293, y + 33, 0);
    Bar1(x + 210, y, x + 290, y + 30, 0xffff);
    prt_hz24d(x + 235, y + 5, "日", 0, "HZK\\Hzk24h");

    if (quit_flg)
    {
        return 0;
    }
    quit_flg = 1;
}

if (MousePress(x, y, x + 80, y + 30)) //点击*年*
{
    MouseOff(&mouse);
    Bar1(x, y, x + 80, y + 30, 0xffff);
    Getinfo(x + 10, y + 5, year, 4); //得到数据
    MouseOn(mouse);
}
else if (MousePress(x + 110, y, x + 180, y + 30)) //点击*月*
{
    MouseOff(&mouse);
    Bar1(x + 110, y, x + 180, y + 30, 0xffff);
    Getinfo(x + 120, y + 5, month, 2); //得到数据
    MouseOn(mouse);
}
else if (MousePress(x + 210, y, x + 290, y + 30)) //点击*日*
{
    MouseOff(&mouse);
    Bar1(x + 210, y, x + 290, y + 30, 0xffff);
    Getinfo(x + 220, y + 5, day, 2); //得到数据
    MouseOn(mouse);
    if (*key = findday(atoi(year), atoi(month), atoi(day)))
    {
        choose_flg = 1;
    }
    else
    {
        prt_hz24d(x+100,y+80,"找不到日期",0,"HZK\\Hzk24h");
        delay(1000);
        Bar1(x+98, y+78, x+350, y+120, 0x8edf);
    }
```

```
        }
    }
}

/**********************************
Function: dis_bar
Description: 绘制下拉方框
Called by: choose_district()
**********************************/
void dis_bar(int x, int y, int wid, int n)
{
    int i;
    Bar1(x - 3, y, x + wid + 3, y + 30 * n + 3, 0);
    for (i = 0; i < n; i++)
    {
        Bar1(x, y + i * 30 + 3, x + wid, y + i * 30 + 30, 0xffff);
    }
}
```

REGISTER.C

```
/*************************
该文件用于新用户注册
*************************/
#include "common.h"
#include "lgstate.h"
#include "register.h"

int register_user(void) //注册界面
{
    int flag = 0; //反馈标志变量
    char state = 1;
    char username[12 + 1] = {'\0'}; //username
    char key1[12 + 1] = {'\0'};    //password
    char key2[12 + 1] = {'\0'};    //ensure password
    char id[18 + 1] = {'\0'};      //IDnumber
    Bar1(0, 0, 1024, 768, 0x9eef);  //重设界面

    //标题
    prt_hz16_size(285, 150, 5, 5, "新型冠状病毒", 0xffe0, "HZK\\Hzk16s");
    prt_hz16_size(285, 250, 5, 5, "疫情通报系统", 0xffe0, "HZK\\Hzk16s");

    /*交互界面*/
    //退出键
```

```
    Bar1(900, 5, 1024, 55, 0x7bef); //退出框阴影
    Bar1(895, 0, 1019, 50, 59004);  //退出框实体
    prt_hz24(935, 12, "退出", 0x0000, "HZK\\Hzk24h");

    //返回键
    Bar1(5, 5, 127, 53, 0x7bef); //0x7bef 灰色
    Bar1(0, 0, 124, 50, 59004);
    prt_hz24(38, 13, "返回", 0x0000, "HZK\\Hzk24h");

    Bar1(374, 423, 661, 451, 0);
    Bar1(376, 425, 659, 449, 0xffff); //账号框

    Bar1(374, 487, 661, 515, 0);
    Bar1(376, 489, 659, 513, 0xffff); //密码框

    Bar1(374, 551, 661, 579, 0);
    Bar1(376, 553, 659, 577, 0xffff); //确认密码框

    Bar1(374, 551 + 64, 661, 579 + 64, 0);
    Bar1(376, 553 + 64, 659, 577 + 64, 0xffff); //身份证框

    Bar1(361, 614 + 100, 429, 648 + 100, 0x7bef);
    Bar1(356, 609 + 100, 424, 643 + 100, 0xffff); //确认框

    Bar1(561, 614 + 100, 629, 648 + 100, 0x7bef);
    Bar1(556, 609 + 100, 624, 643 + 100, 0xffff); //取消框

    prt_hz24(307, 425, "账号：", 0, "HZK\\Hzk24h");
    prt_hz24(307, 489, "密码：", 0, "HZK\\Hzk24h");
    prt_hz24(259, 553, "确认密码：", 0, "HZK\\Hzk24h");
    prt_hz24(259, 553 + 64, "身份证：", 0, "HZK\\Hzk24h");
    prt_hz24(366, 614 + 100, "确认", 0, "HZK\\Hzk24h");
    prt_hz24(566, 614 + 100, "取消", 0, "HZK\\Hzk24h");

    Mouse_Init(); //鼠标初始化
    while (1)
    {
        char key_0 = 0;           //按下注册键后的状态，成功 1，账号未输入 2，密码未输入
3，密码未确认 4，身份证未输入 5
        char key_1 = 0;          //按下注册键后的状态，账号，重名 0，未重名 1
        char key_2 = 0;          //按下注册键后的状态，密码格式，0 表示长度不合乎要求，1
表示合乎要求
        char key_3 = 0;          //按下注册键后的状态，两次密码，0 表示不相同，1 表示相同
        char key_4 = 0;          //按下注册键后的状态，身份证号码格式
```

```
                                                    //0 表示长度不合乎要求，1 表示内容不合乎要求，2 表示合乎要求

        MouseShow(&mouse);       //鼠标循环
        if (flag == 0 && MouseIn(895, 0, 1019, 50)) //退出框反馈
        {
            MouseOff(&mouse);
            Bar1(895, 0, 1019, 50, 56603);
            prt_hz24d(935, 12, "退出", 0x0000, "HZK\\Hzk24h");
            MouseOn(mouse);
            flag = 1;
        }
        if (flag == 0 && MouseIn(0, 0, 124, 50)) //返回框反馈
        {
            MouseOff(&mouse);
            Bar1(0, 0, 124, 50, 56603);
            prt_hz24d(38, 13, "返回", 0x0000, "HZK\\Hzk24h");
            MouseOn(mouse);
            flag = 1;
        }
        if (flag == 0 && MouseIn(366, 614 + 100, 414, 638 + 100)) //确认框反馈
        {
            MouseOff(&mouse);
            Bar1(356, 609 + 100, 424, 643 + 100, 0xeeee);
            prt_hz24d(366, 614 + 100, "确认", 0, "HZK\\Hzk24h");
            MouseOn(mouse);
            flag = 1;
        }
        if (flag == 0 && MouseIn(566, 614 + 100, 614, 638 + 100)) //取消框反馈
        {
            MouseOff(&mouse);
            Bar1(556, 609 + 100, 624, 643 + 100, 0xeeee);
            prt_hz24d(566, 614 + 100, "取消", 0, "HZK\\Hzk24h");
            MouseOn(mouse);
            flag = 1;
        }

        else if (flag == 1 && !(MouseIn(900, 0, 1024, 50)) && !(MouseIn(0, 0,
124, 50)) && !(MouseIn(366, 614 + 100, 414, 638 + 100)) && !(MouseIn(566, 614
+ 100, 614, 638 + 100)))
        {
            MouseOff(&mouse);
            Bar1(895, 0, 1019, 50, 59004);
            Bar1(356, 609 + 100, 424, 643 + 100, 0xffff);
            Bar1(556, 609 + 100, 624, 643 + 100, 0xffff);
```

```
    Bar1(0, 0, 124, 50, 59004);
    prt_hz24d(38, 13, "返回", 0, "HZK\\Hzk24h");
    prt_hz24d(935, 12, "退出", 0, "HZK\\Hzk24h");
    prt_hz24d(366, 614 + 100, "确认", 0, "HZK\\Hzk24h");
    prt_hz24d(566, 614 + 100, "取消", 0, "HZK\\Hzk24h");
    MouseOn(mouse);
    flag = 0;
}
else if (MousePress(378, 425, 659, 449)) //输入账号
{
    prt_hz24d(700, 454, "（回车键结束输入）", 0, "HZK\\Hzk24h");
    MouseOff(&mouse);
    Bar1(378, 425, 659, 449, 0xffff);
    Getinfo(380, 427, username, 12); //gets.h
    MouseOn(mouse);
    Bar1(670, 425, 670 + 200, 425 + 28, 0x9eef);
}
else if (MousePress(378, 489, 659, 513)) //输入密码
{
    MouseOff(&mouse);
    Bar1(378, 489, 659, 513, 0xffff);
    Getcode(380, 491, key1, 12);
    MouseOn(mouse);
    Bar1(670, 425 + 64, 670 + 200, 425 + 28 + 64, 0x9eef);
}
else if (MousePress(378, 553, 659, 577)) //重新输入密码
{
    MouseOff(&mouse);
    Bar1(378, 553, 659, 577, 0xffff);
    Getcode(380, 555, key2, 12);
    MouseOn(mouse);
    Bar1(670, 425 + 64 * 2, 670 + 200, 425 + 28 + 64 * 2, 0x9eef);
}
else if (MousePress(378, 553 + 64, 659, 577 + 64)) //输入身份证
{
    MouseOff(&mouse);
    Bar1(378, 553 + 64, 659, 577 + 64, 0xffff);
    Getinfo(380, 555 + 64, id, 18);
    MouseOn(mouse);
    Bar1(670, 425 + 64 * 3, 670 + 200, 425 + 28 + 64 * 3, 0x9eef);
}
if (MousePress(366, 614 + 100, 414, 638 + 100)) //点击确认键
{
    state = 1;
```

```
        /*判断是否输入完全*/
        key_0 = complete_register(username[0], key1[0], key2[0], id[0]); //
成功 1，账号未输入 2，密码未输入 3，密码未确认 4，身份证未输入 5
        switch (key_0)
        {
        case 1:
            break;
        case 2:
            prt_hz24(670, 425, "账号未输入", 0, "HZK\\Hzk24h");
            state = 0;
            break;
        case 3:
            prt_hz24(670, 489, "密码未输入", 0, "HZK\\Hzk24h");
            state = 0;
            break;
        case 4:
            prt_hz24(670, 553, "请确认密码", 0, "HZK\\Hzk24h");
            state = 0;
            break;
        case 5:
            prt_hz24(670, 553 + 64, "身份证号码未输入", 0, "HZK\\Hzk24h");
            state = 0;
            break;
        default:
            prt_hz24(670, 553, "错误", 0, "HZK\\Hzk24h");
            break;
        }
        /*判断是否重名*/
        if (key_0 == 1)
        {
            key_1 = judge_same_name(username); //0 表示重名，1 表示不重名
            if (key_1 == 0)
            {
                prt_hz24(670, 553, "用户名已被使用", 0, "HZK\\Hzk24h");
                state = 0;
            }
        }
        /*判断密码格式*/
        if (key_1 == 1)
        {
            key_2 = judge_password(key1); //0 表示长度不合乎要求，1 表示合乎要求
            if (key_2 == 0)
            {
                prt_hz24(670, 553, "密码过短", 0, "HZK\\Hzk24h");
```

```
                state = 0;
            }
        }
        /*判断两次密码是否相同*/
        if (key_2 == 1)
        {
            key_3 = compare_password(key1, key2); //0 表示不相同，1 表示相同
            if (key_3 == 0)
            {
                prt_hz24(670, 553, "密码不一致", 0, "HZK\\Hzk24h");
                state = 0;
            }
        }
        /*判断身份证号码格式*/
        if (key_3 == 1)
        {
            key_4 = judge_IDnumber(id); //0 表示长度不合乎要求，1 表示内容不合乎要求，2 表示合乎要求
            if (key_4 == 0)
            {
                prt_hz24(670, 553 + 64, "长度有误", 0, "HZK\\Hzk24h");
                state = 0;
            }
            else if (key_4 == 1)
            {
                prt_hz24(670, 553 + 64, "内容有误", 0, "HZK\\Hzk24h");
                state = 0;
            }
        }

        if (state == 1)
        {
            wr_user_inf(username, key1, id);
            prt_hz24(670, 553, "注册成功", 0, "HZK\\Hzk24h");
            delay(1000);
            return 1;
        }
    }

    if (MousePress(566, 614, 614, 638)) //点击取消键
    {
        return 1;
    }
    if (MousePress(900, 0, 1024, 50)) //点击退出
```

```
        {
            return -1;
        }
        if (MousePress(0, 0, 124, 50)) //点击返回
        {
            return 1;
        }
    }
}
```

REFOUND.C

```
/*该文件用于忘记密码操作*/
#include "common.h"
#include "lgstate.h"
#include "register.h"

int refound_user(void) //重设密码界面
{
    int flag = 0; //反馈标志变量
    char state = 1;
    char username[12 + 1] = {'\0'}; //username
    char key1[12 + 1] = {'\0'};    //password
    char key2[12 + 1] = {'\0'};    //ensure password
    char id[18 + 1] = {'\0'};      //IDnumber
    Bar1(0, 0, 1024, 768, 0x9eef);  //重设界面

    //标题
    prt_hz16_size(285, 150, 5, 5, "新型冠状病毒", 0xffe0, "HZK\\Hzk16s");
    prt_hz16_size(285, 250, 5, 5, "疫情通报系统", 0xffe0, "HZK\\Hzk16s");

    /*交互界面*/
    //退出键
    Bar1(900, 5, 1024, 55, 0x7bef); //退出框阴影
    Bar1(895, 0, 1019, 50, 59004);  //退出框实体
    prt_hz24(935, 12, "退出", 000000, "HZK\\Hzk24h");

    //返回键
    Bar1(5, 5, 127, 53, 0x7bef); //0x7bef 灰色
    Bar1(0, 0, 124, 50, 59004);
    prt_hz24(38, 13, "返回", 0x0000, "HZK\\Hzk24h");

    Bar1(374, 423, 661, 451, 0);
    Bar1(376, 425, 659, 449, 0xffff); //账号框
```

```
    Bar1(374, 487, 661, 515, 0);
    Bar1(376, 489, 659, 513, 0xffff); //密码框

    Bar1(374, 551, 661, 579, 0);
    Bar1(376, 553, 659, 577, 0xffff); //确认密码框

    Bar1(374, 551 + 64, 661, 579 + 64, 0);
    Bar1(376, 553 + 64, 659, 577 + 64, 0xffff); //身份证框

    Bar1(361, 614 + 100, 429, 648 + 100, 0x7bef);
    Bar1(356, 609 + 100, 424, 643 + 100, 0xffff); //确认框

    Bar1(561, 614 + 100, 629, 648 + 100, 0x7bef);
    Bar1(556, 609 + 100, 624, 643 + 100, 0xffff); //取消框

    prt_hz24(307, 425, "账号：", 0, "HZK\\Hzk24h");
    prt_hz24(259, 489, "重设密码：", 0, "HZK\\Hzk24h");
    prt_hz24(259, 553, "确认密码：", 0, "HZK\\Hzk24h");
    prt_hz24(259, 553 + 64, "身份证：", 0, "HZK\\Hzk24h");
    prt_hz24(366, 614 + 100, "确认", 0, "HZK\\Hzk24h");
    prt_hz24(566, 614 + 100, "取消", 0, "HZK\\Hzk24h");

    Mouse_Init(); //鼠标初始化
    while (1)
    {

        char key_0 = 0;                                 //按下重设键后的状态，成功 1，账号未
输入 2，密码未输入 3，密码未确认 4，身份证未输入 5
        char key_1 = 0;                                //按下重设键后的状态，账号 存在 0，不
存在 1
        char key_2 = 0;                                 //按下重设键后的状态，密码格式，0 表
示长度不合乎要求，1 表示合乎要求
        char key_3 = 0;                                 //按下重设键后的状态，两次密码，0 表
示不相同，1 表示相同
        char key_4 = 0;                                 //按下重设键后的状态，身份证号码是否
正确 0 错误 1 正确
        MouseShow(&mouse);                             //鼠标循环
        if (flag == 0 && MouseIn(895, 0, 1019, 50)) //退出框反馈
        {
            MouseOff(&mouse);
            Bar1(895, 0, 1019, 50, 56603);
            prt_hz24d(935, 12, "退出", 000000, "HZK\\Hzk24h");
            MouseOn(mouse);
```

```
            flag = 1;
        }
        if (flag == 0 && MouseIn(0, 0, 124, 50)) //返回框反馈
        {
            MouseOff(&mouse);
            Bar1(0, 0, 124, 50, 56603);
            prt_hz24d(38, 13, "返回", 000000, "HZK\\Hzk24h");
            MouseOn(mouse);
            flag = 1;
        }
        if (flag == 0 && MouseIn(366, 614 + 100, 414, 638 + 100)) //确认框反馈
        {
            MouseOff(&mouse);
            Bar1(356, 609 + 100, 424, 643 + 100, 0xeeee);
            prt_hz24d(366, 614 + 100, "确认", 0, "HZK\\Hzk24h");
            MouseOn(mouse);
            flag = 1;
        }
        if (flag == 0 && MouseIn(566, 614 + 100, 614, 638 + 100)) //取消框反馈
        {
            MouseOff(&mouse);
            Bar1(556, 609 + 100, 624, 643 + 100, 0xeeee);
            prt_hz24d(566, 614 + 100, "取消", 0, "HZK\\Hzk24h");
            MouseOn(mouse);
            flag = 1;
        }

        else if (flag == 1 && !(MouseIn(900, 0, 1024, 50)) && !(MouseIn(0, 0,
124, 50)) && !(MouseIn(366, 614 + 100, 414, 638 + 100)) && !(MouseIn(566, 614
+ 100, 614, 638 + 100)))
        {
            MouseOff(&mouse);
            Bar1(895, 0, 1019, 50, 59004);
            Bar1(356, 609 + 100, 424, 643 + 100, 0xffff);
            Bar1(556, 609 + 100, 624, 643 + 100, 0xffff);
            Bar1(0, 0, 124, 50, 59004);
            prt_hz24d(38, 13, "返回", 000000, "HZK\\Hzk24h");
            prt_hz24d(935, 12, "退出", 000000, "HZK\\Hzk24h");
            prt_hz24d(366, 614 + 100, "确认", 0, "HZK\\Hzk24h");
            prt_hz24d(566, 614 + 100, "取消", 0, "HZK\\Hzk24h");
            MouseOn(mouse);
            flag = 0;
        }
        else if (MousePress(378, 425, 659, 449)) //输入账号
```

```
        {
            prt_hz24d(700, 454, "（回车键结束输入）", 0, "HZK\\Hzk24h");
            MouseOff(&mouse);
            Bar1(378, 425, 659, 449, 0xffff);
            //        strcpy(username,"\0");
            Getinfo(380, 427, username, 12); //gets.h
            MouseOn(mouse);
            Bar1(670, 425, 670 + 200, 425 + 28, 0x9eef);
        }
        else if (MousePress(378, 489, 659, 513)) //输入密码
        {
            MouseOff(&mouse);
            Bar1(378, 489, 659, 513, 0xffff);
            //        strcpy(key1,"\0");
            Getcode(380, 491, key1, 12);
            MouseOn(mouse);
            Bar1(670, 425 + 64, 670 + 200, 425 + 28 + 64, 0x9eef);
        }
        else if (MousePress(378, 553, 659, 577)) //重新输入密码
        {
            MouseOff(&mouse);
            Bar1(378, 553, 659, 577, 0xffff);
            //        strcpy(key2,"\0");
            Getcode(380, 555, key2, 12);
            MouseOn(mouse);
            Bar1(670, 425 + 64 * 2, 670 + 200, 425 + 28 + 64 * 2, 0x9eef);
        }
        else if (MousePress(378, 553 + 64, 659, 577 + 64)) //输入身份证
        {
            MouseOff(&mouse);
            Bar1(378, 553 + 64, 659, 577 + 64, 0xffff);
            //        strcpy(id,"\0");
            Getinfo(380, 555 + 64, id, 18);
            MouseOn(mouse);
            Bar1(670, 425 + 64 * 3, 670 + 200, 425 + 28 + 64 * 3, 0x9eef);
        }
        if (MousePress(366, 614 + 100, 414, 638 + 100)) //点击确认键
        {
            /*判断是否输入完全*/
            state = 1;
            key_0 = complete_register(username[0], key1[0], key2[0], id[0]); //
成功 1，账号未输入 2，密码未输入 3，密码未确认 4，身份证未输入 5
            switch (key_0)
            {
```

```
case 2:
    prt_hz24(670, 425, "账号未输入", 0, "HZK\\Hzk24h");
    state = 0;
    break;
case 3:
    prt_hz24(670, 489, "密码未输入", 0, "HZK\\Hzk24h");
    state = 0;
    break;
case 4:
    prt_hz24(670, 553, "请确认密码", 0, "HZK\\Hzk24h");
    state = 0;
    break;
case 5:
    prt_hz24(670, 553 + 64, "身份证号码未输入", 0, "HZK\\Hzk24h");
    state = 0;
    break;
}
/*判断是否注册*/
if (key_0 == 1)
{
    key_1 = judge_same_name(username); //0 表示重名，1 表示不重名
    if (key_1 == 1)
    {
        prt_hz24(670, 553, "账户未注册", 0, "HZK\\Hzk24h");
        state = 0;
    }
}
/*判断密码格式*/
if (key_1 == 0)
{
    key_2 = judge_password(key1); //0 表示长度不合乎要求，1 表示合乎要求
    if (key_2 == 0)
    {
        prt_hz24(670, 553, "密码过短", 0, "HZK\\Hzk24h");
        state = 0;
    }
}
/*判断两次密码是否相同*/
if (key_2 == 1)
{
    key_3 = compare_password(key1, key2); //0 表示不相同，1 表示相同
    if (key_3 == 0)
    {
        prt_hz24(670, 553, "密码不一致", 0, "HZK\\Hzk24h");
```

```
                state = 0;
            }
        }

        /*判断身份证号码格式*/
        if (key_3 == 1)
        {
            key_4 = right_id(username, id); //0 表示不正确，1 表示正确

            if (key_4 == 0)
            {
                prt_hz24(670, 553 + 64, "身份证错误", 0, "HZK\\Hzk24h");
                state = 0;
            }
        }
        if (state == 1)
        {
            user_change_password(username, key1);
            prt_hz24(670, 553, "密码重设成功", 0, "HZK\\Hzk24h");
            delay(1000);
            return 1;
        }
    }

    if (MousePress(566, 614, 614, 638)) //点击取消键
    {
        return 1;
    }
    if (MousePress(900, 0, 1024, 50)) //点击退出
    {
        return -1;
    }
    if (MousePress(0, 0, 124, 50)) //点击返回
    {
        return 1;
    }
  }
}
```

LGSTATE.C

```
#include "common.h"
#include "lgstate.h"
/*  FileName: report.c
    Author:   刘子恒
    Date:     2020/9/7
    function：char judge_password(char mima[]) ;密码格式
           char judge_IDnumber(char id[]); 身份证号格式
           char compare_password(char key1[],char key2[]); 确认密码是否相同
           char right_id(char account[],char id[]);确认身份证号码是否相同
           char judge_same_name(char account[]); 是否注册
           char right_password(char account[],char key[]); 密码正确
           char judge_userfrozen(char account[]); 判断用户是否被冻结
           char user_freeze(char account[]) ; 冻结账户
           char user_unfreeze(char account[]);函数功能：将帐户解冻
           char complete_register(char account,char key1,char key2,char id); 完
成注册输入
           char complete_login(char account,char key); 完成登录输入
           char complete_admin(char key);完成管理员登录输入
           void wr_user_inf(char account[],char key[],char id[]);将注册信息写入文
件
           void wr_user(char account[]);将本次登录的账号写入文件
           char admin_password(char key[]);判断管理员登录时密码是否相符*/

/*函数功能：判断密码是否合乎要求
  返回值：0 表示长度不合乎要求，1 表示合乎要求*/
char judge_password(char mima[])
{
    int i;
    int flag = 1;
    for (i = 0; i < 6; i++)
    {
        if (mima[i] == '\0') //若密码长度小于 6，则长度错误
        {
            flag = 0;
            break;
        }
    }
    return flag;
}

/*函数功能：判断身份证号是否合乎要求
  返回值：0 表示长度不合乎要求，1 表示内容不合乎要求，2 表示合乎要求*/
```

```
char judge_IDnumber(char id[])
{
    int i;
    int flag = 2;
    for (i = 0; i < 19; i++)
    {
        if (id[i] == '\0') //若提前结束，则长度错误
        {
            break;
        }
        if ((i < 17) && (id[i] < '0' || id[i] > '9')) //若在除最后一位之外出现非数字字符，则内容错误
        {
            flag = 1;
            break;
        }
        if ((i == 17) && (id[i] != 'x' && id[i] != 'X') && (id[i] < '0' || id[i] > '9')) //若在最后一位出现不是 x 或 X 的非数字字符，则内容错误
        {
            flag = 1;
            break;
        }
    }
    if (i < 18 && flag != 1)
    {
        flag = 0;
    }
    return flag;
}

/*函数功能：判断两次输入的密码是否相同
   返回值：0 表示不相同，1 表示相同*/
char compare_password(char key1[], char key2[])
{
    int i;
    int flag = 0;
    for (i = 0; i < 13; i++)
    {
        if (key1[i] != key2[i])
        {
            flag = 0;
            break;
        }
        if (key1[i] == '\0')
```

```
        {
            flag = 1;
            break;
        }
    }
    return flag;
}

/*函数功能：判断注册的账号是否与已有账号重名
  返回值：0 表示重名，1 表示不重名
  函数功能：判断登录账号是否已经注册
  返回值：0 表示已注册，1 表示尚未注册*/
char judge_same_name(char account[])
{
    FILE *fp;
    USER *u = NULL;
    int i, j;
    int flag = 1;
    int length;
    if ((fp = fopen("UserInfo.dat", "rb+")) == NULL)
    {
        printf("\nUserInfo.dat error file");
        delay(3000);
        exit(1); //打开失败，退出程序
    }
    fseek(fp, 0, SEEK_END);
    length = ftell(fp) / sizeof(USER);
    for (i = 0; i < length; i++)
    {
        if ((u = (USER *)malloc(sizeof(USER))) == NULL)
        {
            printf("memory error JudgeSameName");
            delay(3000);
            exit(1); //分配空间不足，退出程序
        }
        fseek(fp, i * sizeof(USER), SEEK_SET);
        fread(u, sizeof(USER), 1, fp);
        for (j = 0; j < 13; j++)
        {
            if (u->UserName[j] != account[j])
            {
                break;
            }
            if (account[j] == '\0')
```

```
            {
                j = 13;
                break;
            }
        }
        if (j == 13)
        {
            flag = 0;
            break;
        }
        free(u);
        u = NULL;
    }
    if (u != NULL)
    {
        free(u);
        u = NULL;
    }
    //setbkcolor(RED);
    if (fclose(fp) != 0) //关闭文件
    {
        printf("\nError in closing file UserInfo.");
        delay(3000);
        exit(1);
    }
    return flag;
}

/*函数功能：判断账号密码是否匹配
  返回值：0 表示不正确，1 表示正确*/
char right_password(char account[], char key[])
{
    FILE *fp;
    USER *u = NULL;
    int i, j, k = 0;
    int flag = 0;
    int length;
    if ((fp = fopen("UserInfo.dat", "rb+")) == NULL)
    {
        printf("\nUserInfo.dat error file");
        delay(3000);
        exit(1); //打开失败，退出程序
    }
    fseek(fp, 0, SEEK_END);
```

```
length = ftell(fp) / sizeof(USER);
for (i = 0; i < length; i++)
{
    if ((u = (USER *)malloc(sizeof(USER))) == NULL)
    {
        printf("memory error RightPassword");
        delay(3000);
        exit(1); //分配空间不足，退出程序
    }
    fseek(fp, i * sizeof(USER), SEEK_SET);
    fread(u, sizeof(USER), 1, fp);
    for (j = 0; j < 13; j++)
    {
        if (u->UserName[j] != account[j])
        {
            break;
        }
        if (account[j] == '\0')
        {
            k = 1; //存在此账户
            break;
        }
    }
    if (k == 1)
    {
        for (j = 0; j < 13; j++)
        {
            if (u->password[j] != key[j])
            {
                flag = 0;
                break;
            }
            if (key[j] == '\0')
            {
                flag = 1;
                break;
            }
        }
        break;
    }
    free(u);
    u = NULL;
}
if (u != NULL)
```

```
    {
        free(u);
    }

    if (fclose(fp) != 0) //关闭文件
    {
        printf("\nError in closing file UserInfo.");
        delay(3000);
        exit(1);
    }
    return flag;
}

/*函数功能：判断用户是否被冻结
  返回值：1 表示冻结，0 表示未冻结*/
char judge_userfrozen(char account[])
{
    FILE *fp;
    USER *u = NULL;
    int i, j, k = 0;
    char flag = 0;
    int length;
    if ((fp = fopen("UserInfo.dat", "rb+")) == NULL)
    {
        printf("\nUserInfo.dat error file");
        delay(3000);
        exit(1); //打开失败，退出程序
    }
    fseek(fp, 0, SEEK_END);
    length = ftell(fp) / sizeof(USER);
    for (i = 0; i < length; i++)
    {
        if ((u = (USER *)malloc(sizeof(USER))) == NULL)
        {
            printf("memory error RightPassword");
            delay(3000);
            exit(1); //分配空间不足，退出程序
        }
        fseek(fp, i * sizeof(USER), SEEK_SET);
        fread(u, sizeof(USER), 1, fp);
        for (j = 0; j < 13; j++)
        {
            if (u->UserName[j] != account[j])
            {
```

```
                break;
            }
            if (account[j] == '\0')
            {
                k = 1; //存在此账户
                break;
            }
        }
        if (k == 1)
        {
            if (u->UserState == '1')
            {
                flag = 0;
            }
            else
            {
                flag = 1;
            }
            break;
        }
        free(u);
        u = NULL;
    }
    if (u != NULL)
    {
        free(u);
    }

    if (fclose(fp) != 0) //关闭文件
    {
        printf("\nError in closing file UserInfo.");
        delay(3000);
        exit(1);
    }
    return flag;
}

/*函数功能：将帐户冻结
返回值：0 帐户已被冻结 1：帐户冻结成功 2：帐户不存在*/
int user_freeze(char account[])
{
    FILE *fp;
    USER *u = NULL;
    int i, j, k = 0;
```

```
int length;
int flag;
if ((fp = fopen("UserInfo.dat", "rb+")) == NULL)
{
    printf("\nUserInfo.dat error file");
    delay(3000);
    exit(1); //打开失败，退出程序
}
fseek(fp, 0, SEEK_END);
length = ftell(fp) / sizeof(USER);
for (i = 0; i < length; i++)
{
    k = 0;
    if ((u = (USER *)malloc(sizeof(USER))) == NULL)
    {
        printf("memory error RightPassword");
        delay(3000);
        exit(1); //分配空间不足，退出程序
    }
    fseek(fp, i * sizeof(USER), SEEK_SET);
    fread(u, sizeof(USER), 1, fp);
    for (j = 0; j < 12; j++)
    {
        if (u->UserName[j] != account[j])
        {
            break;
        }
        if (account[j] == '\0')
        {
            k = 1; //存在此账户
            break;
        }
    }
    if (k == 1)
    {
        if (u->UserState == '2')
        {
            flag = 0;
        }
        if (u->UserState == '1')
        {
            flag = 1;
            u->UserState = '2';
        }
```

```
        }
        else if (flag != 0 && flag != 1)
        {
            flag = 2;
        }
        fseek(fp, i * sizeof(USER), SEEK_SET);
        fwrite(u, sizeof(USER), 1, fp);
        free(u);
        u = NULL;
    }
    if (u != NULL)
    {
        free(u);
    }
    if (fclose(fp) != 0) //关闭文件
    {
        printf("\nError in closing file UserInfo.");
        delay(3000);
        exit(1);
    }
    return flag;
}

/*函数功能：将帐户解冻
返回值：0 帐户已被解冻 1：帐户解冻成功 2：帐户不存在*/
char user_unfreeze(char account[])
{
    FILE *fp;
    USER *u = NULL;
    int i, j, k = 0;
    int length;
    char flag;
    if ((fp = fopen("UserInfo.dat", "rb+")) == NULL)
    {
        printf("\nUserInfo.dat error file");
        delay(3000);
        exit(1); //打开失败，退出程序
    }
    fseek(fp, 0, SEEK_END);
    length = ftell(fp) / sizeof(USER);
    for (i = 0; i < length; i++)
    {
        k = 0;
        if ((u = (USER *)malloc(sizeof(USER))) == NULL)
```

```
    {
        printf("memory error RightPassword");
        delay(3000);
        exit(1); //分配空间不足，退出程序
    }
    fseek(fp, i * sizeof(USER), SEEK_SET);
    fread(u, sizeof(USER), 1, fp);
    for (j = 0; j < 12; j++)
    {
        if (u->UserName[j] != account[j])
        {
            break;
        }
        if (account[j] == '\0')
        {
            k = 1; //存在此账户
            break;
        }
    }
    if (k == 1)
    {
        if (u->UserState == '1')
        {
            flag = 0;
        }
        if (u->UserState == '2')
        {
            flag = 1;
            u->UserState = '1';
        }
    }
    else if (flag != 0 && flag != 1)
    {
        flag = 2;
    }
    fseek(fp, i * sizeof(USER), SEEK_SET);
    fwrite(u, sizeof(USER), 1, fp);
    free(u);
    u = NULL;
}
if (u != NULL)
{
    free(u);
}
```

```
    if (fclose(fp) != 0) //关闭文件
    {
        printf("\nError in closing file UserInfo.");
        delay(3000);
        exit(1);
    }
    return flag;
}

/*函数功能：判断账号身份证号是否匹配
  返回值：0 表示不正确，1 表示正确*/
char right_id(char account[], char id[])
{
    FILE *fp;
    USER *u = NULL;
    int i, j, k = 0;
    int flag = 0;
    int length;
    if ((fp = fopen("UserInfo.dat", "rb+")) == NULL)
    {
        printf("\nUserInfo.dat error file");
        delay(3000);
        exit(1); //打开失败，退出程序
    }
    fseek(fp, 0, SEEK_END);
    length = ftell(fp) / sizeof(USER);
    for (i = 0; i < length; i++)
    {

        if ((u = (USER *)malloc(sizeof(USER))) == NULL)
        {
            printf("memory error RightPassword");
            delay(3000);
            exit(1); //分配空间不足，退出程序
        }
        fseek(fp, i * sizeof(USER), SEEK_SET);
        fread(u, sizeof(USER), 1, fp);
        k = 0;
        for (j = 0; j < 13; j++)
        {
            if (u->UserName[j] != account[j])
            {
                break;
            }
```

```
            if (account[j] == '\0')
            {
                k = 1; //存在此账户
                break;
            }
        }
        if (k == 1)
        {
            for (j = 0; j < 19; j++)
            {
                if (u->IDnumber[j] != id[j])
                {
                    flag = 0;
                    break;
                }
                if (id[j] == '\0')
                {
                    flag = 1;
                    break;
                }
            }
            break;
        }
        free(u);
        u = NULL;
    }
    if (u != NULL)
    {
        free(u);
    }
    fclose(fp);
    return flag;
}

/*函数功能：判断是否输入完整 （用于账户注册）
  返回值：1 表示全部输入，2 表示账号未输入，3 表示密码未输入，4 表示未确认密码，5 表示身份证号未输入*/
char complete_register(char account, char key1, char key2, char id)
{
    if (account == '\0')
    {
        return 2;
    }
    if (key1 == '\0')
```

```
    {
        return 3;
    }
    if (key2 == '\0')
    {
        return 4;
    }
    if (id == '\0')
    {
        return 5;
    }
    return 1;
}

/*函数功能：判断是否输入完整 （用于账户登录）
  返回值：1 表示全部输入，2 表示账号未输入，3 表示密码未输入*/
char complete_login(char account, char key)
{
    if (account == '\0')
    {
        return 2;
    }
    if (key == '\0')
    {
        return 3;
    }
    return 1;
}

/*函数功能：判断是否输入完整 （用于管理员登录）
  返回值：1 表示全部输入，2 表示密码未输入*/
char complete_admin(char key)
{
    if (key == '\0')
    {
        return 2;
    }
    return 1;
}

/*函数功能：将注册信息写入文件*/
void wr_user_inf(char account[], char key[], char id[])
{
    FILE *fp;
```

```
    USER *u;
    int i;
    if ((fp = fopen("UserInfo.dat", "rb+")) == NULL)
    {
        printf("\nUserInfo.dat error file");
        delay(3000);
        exit(1); //打开失败，退出程序
    }
    if ((u = (USER *)malloc(sizeof(USER))) == NULL)
    {
        printf("memory error WrUserInf");
        delay(3000);
        exit(1); //分配空间不足，退出程序
    }
    for (i = 0; i < 13; i++)
    {
        u->UserName[i] = account[i];
    }
    for (i = 0; i < 13; i++)
    {
        u->password[i] = key[i];
    }
    for (i = 0; i < 19; i++)
    {
        u->IDnumber[i] = id[i];
    }
    // for(i=0;i<4;i++)
    // {
    //     u->reportnum[i]='0';
    // }
    // u->reportnum[4]='\0';
    u->UserState = '1';
    u->n[0] = '\r';
    u->n[1] = '\n';
    fseek(fp, 0, SEEK_END);
    fwrite(u, sizeof(USER), 1, fp);
    free(u);
    if (fclose(fp) != 0) //关闭文件
    {
        printf("\nError in closing file UserInfo.");
        delay(3000);
        exit(1);
    }
}
```

```
/*函数功能：将新密码写入文件*/
void user_change_password(char account[], char key[])
{
    FILE *fp;
    USER *u = NULL;
    int i, j, k = 0;
    int length;
    int mimalength;
    if ((fp = fopen("UserInfo.dat", "rb+")) == NULL)
    {
        printf("\nUserInfo.dat error file");
        delay(3000);
        exit(1); //打开失败，退出程序
    }
    fseek(fp, 0, SEEK_END);
    length = ftell(fp) / sizeof(USER);
    for (i = 0; i < length; i++)
    {
        k = 0;
        if ((u = (USER *)malloc(sizeof(USER))) == NULL)
        {
            printf("memory error RightPassword");
            delay(3000);
            exit(1); //分配空间不足，退出程序
        }
        fseek(fp, i * sizeof(USER), SEEK_SET);
        fread(u, sizeof(USER), 1, fp);
        for (j = 0; j < 12; j++)
        {
            if (u->UserName[j] != account[j])
            {
                break;
            }
            if (account[j] == '\0')
            {
                k = 1; //存在此账户
                break;
            }
        }
        if (k == 1)
        {
            for (j = 0; j < 13; j++)
            {
```

```
            if (key[j] == '\0')
            {
                mimalength = j + 1;
                break;
            }
        }
        for (j = 0; j < mimalength; j++)
        {
            u->password[j] = key[j];
        }
        for (j = mimalength; j < 13; j++)
        {
            u->password[j] = '\0';
        }
    }
    fseek(fp, i * sizeof(USER), SEEK_SET);
    fwrite(u, sizeof(USER), 1, fp);
    free(u);
    u = NULL;
  }
  if (u != NULL)
  {
    free(u);
  }
  if (fclose(fp) != 0) //关闭文件
  {
    printf("\nError in closing file UserInfo.");
    delay(3000);
    exit(1);
  }
}

/*函数功能：将本次登录的账号写入文件*/
void wr_user(char account[])
{
  FILE *fp;
  if ((fp = fopen("usernow.dat", "rb+")) == NULL)
  {
    printf("usernow.dat file error\n");
    delay(3000);
    exit(1);
  }
  fseek(fp, 0, SEEK_SET);
  fwrite(account, sizeof(char), 13, fp);
```

```
    if (fclose(fp) != 0) //关闭文件
    {
        printf("\nError in closing file usernow.");
        delay(3000);
        exit(1);
    }
}

/*函数功能：将本次登录的账号读出*/
void re_user(char account[])
{
    FILE *fp;
    if ((fp = fopen("usernow.dat", "rb+")) == NULL)
    {
        printf("usernow.dat file error\n");
        delay(3000);
        exit(1);
    }
    fseek(fp, 0, SEEK_SET);
    fread(account, sizeof(char), 13, fp);
    if (fclose(fp) != 0) //关闭文件
    {
        printf("\nError in closing file usernow.");
        delay(3000);
        exit(1);
    }
}

/*函数功能：判断管理员登录时密码是否相符
  返回值：0 表示不正确，1 表示正确*/
char admin_password(char key[])
{
    char adpassword[5 + 1] = "admin";
    int i;
    int flag = 1;
    for (i = 0; i < 5; i++)
    {
        if (key[i] != adpassword[i])
        {
            flag = 0;
            break;
        }
    }
    if (key[5] != '\0')
```

```
    {
        flag = 0;
    }
    return flag;
}

/*函数功能：判断 id 是否存在
  返回值：0 表示存在，1 表示不存在*/
char judge_same_id(char idnumber[], char account[])
{
    FILE *fp;
    USER *u = NULL;
    int i, j;
    int flag = 1;
    int length;
    if ((fp = fopen("UserInfo.dat", "rb+")) == NULL)
    {
        printf("\nUserInfo.dat error file");
        delay(3000);
        exit(1); //打开失败，退出程序
    }
    fseek(fp, 0, SEEK_END);
    length = ftell(fp) / sizeof(USER);
    for (i = 0; i < length; i++)
    {
        if ((u = (USER *)malloc(sizeof(USER))) == NULL)
        {
            printf("memory error JudgeSameName");
            delay(3000);
            exit(1); //分配空间不足，退出程序
        }
        fseek(fp, i * sizeof(USER), SEEK_SET);
        fread(u, sizeof(USER), 1, fp);
        for (j = 0; j < 19; j++)
        {
            if (u->IDnumber[j] != idnumber[j])
            {
                break;
            }
            if (idnumber[j] == '\0')
            {
                j = 19;
                break;
            }
```

```
        }
        if (j == 19)
        {
            flag = 0;
            strcpy(account, u->UserName);
            break;
        }
        free(u);
        u = NULL;
    }
    if (u != NULL)
    {
        free(u);
        u = NULL;
    }
    if (fclose(fp) != 0) //关闭文件
    {
        printf("\nError in closing file UserInfo.");
        delay(3000);
        exit(1);
    }
    return flag;
}
```

INQUIRY.C

```
/*****************************************
FileName: inquiry.c
Author:   刘子恒
Date:    2020/10/24
Description: 该文件主要用于用户通报数据功能
Others: 无
Function List:
        inquiry  用户查询主函数
        drawarea 输出区域信息
*****************************************/
#include "choice.h"
#include "common.h"
#include "dayinfo.h"
#include "drawc.h"
#include "gettime.h"
#include "inquiry.h"
// 用户查询主函数
int inquiry(void)
```

```
{
    //初始化
    int flag = 0, areakey, datekey, judge;
    char area[6];
    char alinf[6] = {'\0'};
    char aldea[6] = {'\0'};
    char alhea[6] = {'\0'};
    char alnow[6] = {'\0'};
    //char filename[5+1]={'\0'};
    char cyear[5] = {'\0'};
    char cmonth[3] = {'\0'};
    char cday[3] = {'\0'};
    CHARTDATA data[135];
    SAVEDAY *dd = NULL;
    int alsuminf, alsumdea, alsumheal, alnowinf;
    int suminf, sumdea, sumheal, nowinf;
    int year, month, day, count = 0, today;
    re_area(&areakey);
    countareacases(14, &alsuminf, &alsumdea, &alsumheal, &alnowinf);
    itoa(alsuminf, alinf, 10);
    itoa(alsumdea, aldea, 10);
    itoa(alsumheal, alhea, 10);
    itoa(alnowinf, alnow, 10);
    Bar1(0, 0, 1024, 768, 0x9efc);

    if ((dd = (SAVEDAY *)malloc(sizeof(SAVEDAY))) == NULL)
    {
        printf("memory error CountCase");
        delay(3000);
        exit(1); //分配空间不足，退出程序
    }

    gett(&year, &month, &day);        //得到今天时间
    today = findday(year, month, day); //找到对应日期再数据库中的位置

    Bar1(248, 168, 752, 482, 0);
    Bar1(250, 170, 750, 480, 0x8edf); //弹窗底色

    prt_hz24d(380, 200, "请选择您要查询的日期", 0, "HZK\\Hzk24h");

    Bar1(348, 413, 452, 457, 0x4b32);
    Bar1(350, 415, 450, 455, 0x8ff2);
    prt_hz24d(376, 423, "确认", 0, "HZK\\Hzk24h"); //确认键
    Bar1(548, 413, 652, 457, 0x4b32);
```

```
    Bar1(550, 415, 650, 455, 0x8ff2);
    prt_hz24d(576, 423, "取消", 0, "HZK\\Hzk24h"); //取消键

    judge = choose_date(345, 255, &datekey);
    if (judge == 0)
    {
        return 5;
    }
    if (datekey > today) //判断日期是否合法
    {
        Bar1(0, 0, 1024, 768, 0x9efc);
        prt_hz24d(380, 200, "数据有误", 0, "HZK\\Hzk24h");
        delay(2000);
        return 5;
    }
    getoneday(dd, areakey, datekey); //读取相应日期信息

    Bar1(0, 0, 1024, 768, 0x9efc);

    Bar1(600, 730, 650, 740, 0xcccc);

    Bar1(500, 730, 550, 740, 0xcccc);
    //绘制退出框
    Bar1(900, 5, 1024, 55, 0x7bef); //退出框阴影
    Bar1(895, 0, 1019, 50, 0xccff); //退出框实体
    prt_hz24(935, 12, "退出", 0xfc00, "HZK\\Hzk24h");

    //返回键
    Bar1(5, 5, 127, 53, 0x7bef); //0x7bef 灰色
    Bar1(0, 0, 124, 50, 0xccff);
    prt_hz24(38, 13, "返回", 0x0000, "HZK\\Hzk24h");

    //标题
    prt_hz16_size(300, 20, 5, 5, "疫情信息查询", 0xffe0, "HZK\\Hzk16s");
//0xffe0 亮黄色

    put_asc16_size(185, 200 + 50, 2, 2, alinf, 0x0000);
    prt_hz24(100, 200 + 50, "总感染", 0x0000, "HZK\\Hzk24k");
    put_asc16_size(185 + 150, 200 + 50, 2, 2, aldea, 0x0000);
    prt_hz24(100 + 150, 200 + 50, "总死亡", 0x0000, "HZK\\Hzk24k");
    put_asc16_size(185 + 150 + 150, 200 + 50, 2, 2, alhea, 0x0000);
    prt_hz24(100 + 150 + 150, 200 + 50, "总治愈", 0x0000, "HZK\\Hzk24k");
    put_asc16_size(185 + 150 + 150 + 150 + 20, 200 + 50, 2, 2, alnow, 0x0000);
```

```
    prt_hz24(100 + 150 + 150 + 150, 200 + 50, "现存感染", 0x0000,
"HZK\\Hzk24k");

    drawarea(areakey); //输出相应地区信息

    countareacases(areakey, &suminf, &sumdea, &sumheal, &nowinf); //计算对应区域
信息

    gett(&year, &month, &day);

    put_asc16_size(285 + 80, 600, 2, 2, cyear, 0x0000);
    put_asc16_size(485 + 80, 600, 2, 2, cmonth, 0x0000);
    put_asc16_size(685 + 80, 600, 2, 2, cday, 0x0000);

    itoa(suminf, alinf, 10);
    itoa(sumdea, aldea, 10);
    itoa(sumheal, alhea, 10);
    itoa(nowinf, alnow, 10);
    put_asc16_size(185 + 80, 200, 2, 2, alinf, 0x0000);
    prt_hz24(100, 200, "选区总感染", 0x0000, "HZK\\Hzk24k");
    put_asc16_size(185 + 150 + 80 + 80, 200, 2, 2, aldea, 0x0000);
    prt_hz24(100 + 150 + 80, 200, "选区总死亡", 0x0000, "HZK\\Hzk24k");
    put_asc16_size(185 + 150 + 150 + 80 + 80 + 80, 200, 2, 2, alhea, 0x0000);
    prt_hz24(100 + 150 + 150 + 80 + 80, 200, "选区总治愈", 0x0000,
"HZK\\Hzk24k");
    put_asc16_size(185 + 150 + 150 + 150 + 80 + 80 + 80 + 80, 200, 2, 2,
alnow, 0x0000);
    prt_hz24(100 + 150 + 150 + 150 + 80 + 80 + 80, 200, "选区现存感染", 0x0000,
"HZK\\Hzk24k");

    put_asc16_size(120, 160, 2, 2, dd->year, 0x0000);
    put_asc16_size(165 + 60, 160, 2, 2, dd->month, 0x0000);
    put_asc16_size(165 + 60 * 2, 160, 2, 2, dd->day, 0x0000);
    put_asc16_size(165 + 60 * 3, 160, 2, 2, dd->infestor, 0x0000);
    put_asc16_size(165 + 60 * 4, 160, 2, 2, dd->heal, 0x0000);
    put_asc16_size(165 + 60 * 5, 160, 2, 2, dd->death, 0x0000);
    put_asc16_size(165 + 60 * 6, 160, 2, 2, dd->allinf, 0x0000);
    put_asc16_size(165 + 60 * 7, 160, 2, 2, dd->allhea, 0x0000);
    put_asc16_size(165 + 60 * 8, 160, 2, 2, dd->alldea, 0x0000);

    drawchart(data, areakey, count); //画折线图

    Mouse_Init();
    while (1)
```

```
    {
        MouseShow(&mouse);

        if (flag == 0 && MouseIn(895, 0, 1019, 50)) //退出框反馈
        {
            MouseOff(&mouse);
            Bar1(895, 0, 1019, 50, 0x9ea0);
            prt_hz24d(935, 12, "退出", 0x0000, "HZK\\Hzk24h");
            MouseOn(mouse);
            flag = 1;
        }
        if (flag == 0 && MouseIn(0, 0, 124, 50)) //返回框反馈
        {
            MouseOff(&mouse);
            Bar1(0, 0, 124, 50, 0x9ea0);
            prt_hz24d(38, 13, "返回", 0x0000, "HZK\\Hzk24h");
            MouseOn(mouse);
            flag = 1;
        }

        else if (flag == 1 && !(MouseIn(895, 0, 1019, 50)) && !(MouseIn(0, 0,
124, 50)))
        {
            MouseOff(&mouse);
            Bar1(895, 0, 1019, 50, 0xccff);
            Bar1(0, 0, 124, 50, 0xccff);

            prt_hz24d(935, 12, "退出", 0x0000, "HZK\\Hzk24h");
            prt_hz24d(38, 13, "返回", 0x0000, "HZK\\Hzk24h");

            MouseOn(mouse);
            flag = 0;
        }
        if (MousePress(600, 730, 650, 740) && count > 0)
        {
            count--;
            Bar1(228, 280, 1024, 710, 0x9efc);
            drawchart(data, areakey, count); //画折线图
        }
        if (MousePress(500, 730, 550, 740) && count < 2)
        {
            count++;
            Bar1(228, 280, 1024, 710, 0x9efc);
            drawchart(data, areakey, count); //画折线图
```

```
        }

        if (MousePress(895, 0, 1019, 50)) //点击退出
        {
            return -1;
        }
        else if (MousePress(0, 0, 124, 50)) //点击返回
        {
            free(dd);
            return 5;
        }
    }
}
// 输出区域信息
void drawarea(int areakey)
{
    if (areakey == 0)
    {
        prt_hz24(790, 90, "已选中益民小区", 0x0000, "HZK\\Hzk24k");
    }
    if (areakey == 1)
    {
        prt_hz24(790, 90, "已选中北门小区", 0x0000, "HZK\\Hzk24k");
    }
    if (areakey == 2)
    {
        prt_hz24(790, 90, "已选中光明小区", 0x0000, "HZK\\Hzk24k");
    }
    if (areakey == 3)
    {
        prt_hz24(790, 90, "已选中桂园小区", 0x0000, "HZK\\Hzk24k");
    }
    if (areakey == 4)
    {
        prt_hz24(790, 90, "已选中南池小区", 0x0000, "HZK\\Hzk24k");
    }
    if (areakey == 5)
    {
        prt_hz24(790, 90, "已选中丽景小区", 0x0000, "HZK\\Hzk24k");
    }
    if (areakey == 6)
    {
        prt_hz24(790, 90, "已选中风韵荷都", 0x0000, "HZK\\Hzk24k");
    }
```

```
    if (areakey == 7)
    {
        prt_hz24(790, 90, "已选中恒大名都", 0x0000, "HZK\\Hzk24k");
    }
    if (areakey == 8)
    {
        prt_hz24(790, 90, "已选中古槐社区", 0x0000, "HZK\\Hzk24k");
    }
    if (areakey == 9)
    {
        prt_hz24(790, 90, "已选中仙营社区", 0x0000, "HZK\\Hzk24k");
    }
    if (areakey == 10)
    {
        prt_hz24(790, 90, "已选中南池社区", 0x0000, "HZK\\Hzk24k");
    }
    if (areakey == 11)
    {
        prt_hz24(790, 90, "已选中北湖社区", 0x0000, "HZK\\Hzk24k");
    }
    if (areakey == 12)
    {
        prt_hz24(790, 90, "已选中任城区", 0x0000, "HZK\\Hzk24k");
    }
    if (areakey == 13)
    {
        prt_hz24(790, 90, "已选中高新区", 0x0000, "HZK\\Hzk24k");
    }
    if (areakey == 14)
    {
        prt_hz24(790, 90, "已选中市", 0x0000, "HZK\\Hzk24k");
    }
}
```

DRAWC.C

```
/*******************************************
FileName: drawc.c
Author:   刘子恒
Date:    2020/10/16
Description: 该文件主要用于绘制折线图
Others: 无
Function List:
       drawchart(); 绘制折线图函数
```

```
        readata(); 读取数据函数
        areafunc();    地区编号转换函数
        giveweight();改变权重函数
        findday(); 寻找对应日期数据函数
*****************************************/
#include "common.h"
#include "drawc.h"
#include "gettime.h"

/*函数功能：画折线图*/

void drawchart(CHARTDATA *data, int areakey, int count)
{
    int i, j;
    float x = 1.0;
    int inf, hea, dea, alinf, alhea, aldea;
    giveweight(&areakey, &x);      //由不同区域给与不同权重防止折线图过大
    readata(data, areakey, count); //读取数据
    Line3(221, 706, 221, 300, 3, 0x0000);
    Line3(221, 706, 920, 706, 3, 0x0000);
    for (i = 0; i < 10; i++) //绘制箭头
    {
        Line3(221 - i, 300 + 2 * i, 221 + i, 300 + 2 * i, 1, 0x0000);
    }
    for (i = 0; i < 10; i++) //绘制箭头
    {
        Line3(920 - 2 * i, 706 + i, 920 - 2 * i, 706 - i, 1, 0x0000);
    }
    for (i = 0; i < 135 - 1; i++) //绘制折线图
    {
        Line2(224 + (5 * i), 700 - x * data[i].allinf, 224 + 5 * (i + 1), 700 -
x * data[i + 1].allinf, 0x1111);
        Line2(224 + (5 * i), 700 - x * data[i].alldea, 224 + 5 * (i + 1), 700 -
x * data[i + 1].alldea, 0x2222);
        Line2(224 + (5 * i), 700 - x * data[i].allhea, 224 + 5 * (i + 1), 700 -
x * data[i + 1].allhea, 0x3333);
        Line2(224 + (5 * i), 700 - x * data[i].heal, 224 + 5 * (i + 1), 700 - x
* data[i + 1].heal, 0x4444);
        Line2(224 + (5 * i), 700 - x * data[i].death, 224 + 5 * (i + 1), 700 -
x * data[i + 1].death, 0x5555);
        Line2(224 + (5 * i), 700 - x * data[i].infestor, 224 + 5 * (i + 1), 700
- x * data[i + 1].infestor, 0x6666);
    }
}
```

```
/*函数功能：读取数据*/
void readata(CHARTDATA *data, int areakey, int count)
{
    char area[8 + 1] = "";
    const char t[4] = {".DAT"};
    int i, j, dnum, year, month, day;
    FILE *fp;
    char filename[10] = "start";
    SAVEDAY *d = NULL;
    memset(data, '\0', sizeof(CHARTDATA) * 135);
    areafunc(areakey, area); //得到地区编号对应区域

    if ((d = (SAVEDAY *)malloc(sizeof(SAVEDAY))) == NULL)
    {

        delay(3000);
        exit(1); //分配空间不足，退出程序
    }
    gett(&year, &month, &day);                          //找到当日日期
    dnum = findday(year, month, day) - 135 * (count + 1); //从当日开始画图
    if (dnum < 0)
        dnum = 0; //防止越界

    while (area[i] != '\0')
    {
        filename[0] = area[i];
        filename[1] = '\0';
        strcat(filename, t);
        filename[5] = '\0';
        i++;

        if ((fp = fopen(filename, "rb+")) == NULL)
        {
            put_asc16_size(185 + 150 + 80 + 80, 600, 2, 2, filename, 0x0000);
            delay(3000);
            exit(1); //打开失败，退出程序
        }
        for (j = 0; j < 135; j++)
        {
            fseek(fp, (dnum + j) * sizeof(SAVEDAY), SEEK_SET);
            fread(d, sizeof(SAVEDAY), 1, fp);
            data[j].infestor = data[j].infestor + atoi(d->infestor); //该日新增感
染
            data[j].death = data[j].death + atoi(d->death);         //该日新增死亡
```

```
            data[j].heal = data[j].heal + atoi(d->heal);            //该日治愈
            data[j].allinf = data[j].allinf + atoi(d->allinf);      //到该日为止
总感染数
            data[j].alldea = data[j].alldea + atoi(d->alldea);      //到该日为止
总治愈数
            data[j].allhea = data[j].allhea + atoi(d->allhea);      //到该日为止
总死亡数
        }
        if (fclose(fp) != 0) //关闭文件
        {
            printf("\nError in closing file .");
            delay(3000);
            exit(1);
        }
    }
    free(d);
}
/*函数功能：地区编号转换*/
void areafunc(int areakey, char area[])
{
    switch (areakey)
    {
    case 0:
        strcpy(area, "0");
        break;
    case 1:
        strcpy(area, "1");
        break;
    case 2:
        strcpy(area, "2");
        break;
    case 3:
        strcpy(area, "3");
        break;
    case 4:
        strcpy(area, "4");
        break;
    case 5:
        strcpy(area, "5");
        break;
    case 6:
        strcpy(area, "6");
        break;
    case 7:
```

```
        strcpy(area, "7");
        break;
    case 8:
        strcpy(area, "01");
        break;
    case 9:
        strcpy(area, "23");
        break;
    case 10:
        strcpy(area, "45");
        break;
    case 11:
        strcpy(area, "67");
        break;
    case 12:
        strcpy(area, "0123");
        break;
    case 13:
        strcpy(area, "4567");
        break;
    case 14:
        strcpy(area, "01234567");
        break;
    }
}
/*函数功能：赋予权重*/
void giveweight(int *areakey, float *x)
{

    if (*areakey < 8)
    {
        *x = 3;
    }
    if (*areakey < 12 && *areakey > 7)
    {
        *x = 2;
    }
    if (*areakey > 11 && *areakey < 14)
    {
        *x = 1;
    }
    if (*areakey == 14)
    {
        *x = 0.5;
```

```
    }
}
/*函数功能：寻找某天对应编号*/
int findday(int year, int month, int day)
{
    FILE *fp;
    SAVEDAY *d = NULL;
    int length, i, count;
    if ((fp = fopen("0.DAT", "rb+")) == NULL)
    {
        put_asc16_size(185 + 150 + 80 + 80, 600, 2, 2, "file error", 0x0000);
        delay(3000);
        exit(1); //打开失败，退出程序
    }
    if ((d = (SAVEDAY *)malloc(sizeof(SAVEDAY))) == NULL)
    {

        delay(3000);
        exit(1); //分配空间不足，退出程序
    }
    fseek(fp, 0, SEEK_END);
    length = ftell(fp) / sizeof(SAVEDAY);
    for (i = 0; i < length; i++)
    {
        fseek(fp, i * sizeof(SAVEDAY), SEEK_SET);
        fread(d, sizeof(SAVEDAY), 1, fp);
        if (year == atoi(d->year) && month == atoi(d->month) && day ==
atoi(d->day))
        {
            //找到对应日期后停止
            break;
        }
    }
    free(d);
    if (fclose(fp) != 0) //关闭文件
    {
        printf("\nError in closing file .");
        delay(3000);
        exit(1);
    }
    return i;
}
```

REPORT.C

```
/*****************************************
FileName: report.c
Author:   刘子恒
Date:    2020/10/27
Description: 该文件主要用于用户通报数据功能
Others: 无
Function List:
       report()  用户通报函数
*****************************************/
#include "common.h"
#include "dayinfo.h"
#include "gettime.h"
#include "report.h"

/*****************************
Function: int report(void)
Description: 用户通报页面
Calls:    re_area()
          gett()
          wr_day_inf()
Called by: main()
Table Accessed: area.DAT
Table Updated:  area.DAT(area:0-7)
Input:  无
Output: 无
Return: -1   -> 退出
       5   -> 数据可视化界面
*****************************/
int report(void)
{
    //初始化
    int flag = 0, areakey;
    char area[6];
    char Ye[4 + 1] = {'\0'};
    char Mo[3] = {'\0'};
    char Da[3] = {'\0'};
    char In[6] = {'\0'};
    char De[6] = {'\0'};
    char He[6] = {'\0'};
    char filename[6];
    int key1 = 0, key2 = 0;
    int year, month, day;
```

```
char User[13];

re_area(&areakey);
gett(&year, &month, &day);
itoa(year, Ye, 10);
itoa(month, Mo, 10);
itoa(day, Da, 10);
Bar1(0, 0, 1024, 768, 0x9efc);

//绘制退出框
Bar1(900, 5, 1024, 55, 0x7bef); //退出框阴影
Bar1(895, 0, 1019, 50, 59004);  //退出框实体
prt_hz24(935, 12, "退出", 0x0000, "HZK\\Hzk24h");

//返回键
Bar1(5, 5, 127, 53, 0x7bef); //0x7bef 灰色
Bar1(0, 0, 124, 50, 59004);
prt_hz24(38, 13, "返回", 0x0000, "HZK\\Hzk24h");

//标题
prt_hz16_size(285, 80, 5, 5, "新型冠状病毒", 0xffe0, "HZK\\Hzk16s");
prt_hz16_size(285, 180, 5, 5, "疫情信息通报", 0xffe0, "HZK\\Hzk16s");

Bar1(374, 403 - 64, 661, 431 - 64, 0);
Bar1(376, 405 - 64, 659, 429 - 64, 0xffff); //年

Bar1(374, 403, 661, 431, 0);
Bar1(376, 405, 659, 429, 0xffff); //月

Bar1(374, 467, 661, 495, 0);
Bar1(376, 469, 659, 493, 0xffff); //日

Bar1(374, 531, 661, 559, 0);
Bar1(376, 533, 659, 557, 0xffff); //新增感染

Bar1(374, 531 + 64, 661, 559 + 64, 0);
Bar1(376, 533 + 64, 659, 557 + 64, 0xffff); //新增死亡

Bar1(374, 531 + 64 + 64, 661, 559 + 64 + 64, 0);
Bar1(376, 533 + 64 + 64, 659, 557 + 64 + 64, 0xffff); //新增治愈

Bar1(361, 614 + 100, 429, 648 + 100, 0x7bef);
Bar1(356, 609 + 100, 424, 643 + 100, 0xffff); //确认
```

```
Bar1(561, 614 + 100, 629, 648 + 100, 0x7bef);
Bar1(556, 609 + 100, 624, 643 + 100, 0xffff); //取消框

prt_hz24(307, 405 - 64, "年：", 0, "HZK\\Hzk24h");
prt_hz24(307, 405, "月：", 0, "HZK\\Hzk24h");
prt_hz24(307, 469, "日：", 0, "HZK\\Hzk24h");
prt_hz24(259, 533, "新增感染", 0, "HZK\\Hzk24h");
prt_hz24(259, 533 + 64, "新增死亡", 0, "HZK\\Hzk24h");
prt_hz24(259, 533 + 64 + 64, "新增治愈", 0, "HZK\\Hzk24h");
prt_hz24(366, 614 + 100, "确认", 0, "HZK\\Hzk24h");
prt_hz24(566, 614 + 100, "取消", 0, "HZK\\Hzk24h");

if (areakey == 0)
{
   prt_hz24(790, 90, "已选中益民小区", 0x0000, "HZK\\Hzk24k");
   strcpy(filename, "0.DAT");
}
if (areakey == 1)
{
   prt_hz24(790, 90, "已选中北门小区", 0x0000, "HZK\\Hzk24k");
   strcpy(filename, "1.DAT");
}
if (areakey == 2)
{
   prt_hz24(790, 90, "已选中光明小区", 0x0000, "HZK\\Hzk24k");
   strcpy(filename, "2.DAT");
}
if (areakey == 3)
{
   prt_hz24(790, 90, "已选中桂园小区", 0x0000, "HZK\\Hzk24k");
   strcpy(filename, "3.DAT");
}
if (areakey == 4)
{
   prt_hz24(790, 90, "已选中南池小区", 0x0000, "HZK\\Hzk24k");
   strcpy(filename, "4.DAT");
}
if (areakey == 5)
{
   prt_hz24(790, 90, "已选中丽景小区", 0x0000, "HZK\\Hzk24k");
   strcpy(filename, "5.DAT");
}
if (areakey == 6)
{
```

```
        prt_hz24(790, 90, "已选中风韵荷都", 0x0000, "HZK\\Hzk24k");
        strcpy(filename, "6.DAT");
    }
    if (areakey == 7)
    {
        prt_hz24(790, 90, "已选中恒大名都", 0x0000, "HZK\\Hzk24k");
        strcpy(filename, "7.DAT");
    }

    re_user(User);

    Mouse_Init();
    while (1)
    {
        MouseShow(&mouse);

        if (flag == 0 && MouseIn(895, 0, 1019, 50)) //退出框反馈
        {
            MouseOff(&mouse);
            Bar1(895, 0, 1019, 50, 56603);
            prt_hz24d(935, 12, "退出", 0x0000, "HZK\\Hzk24h");
            MouseOn(mouse);
            flag = 1;
        }
        if (flag == 0 && MouseIn(0, 0, 124, 50)) //返回框反馈
        {
            MouseOff(&mouse);
            Bar1(0, 0, 124, 50, 56603);
            prt_hz24d(38, 13, "返回", 0x0000, "HZK\\Hzk24h");
            MouseOn(mouse);
            flag = 1;
        }

        else if (flag == 1 && !(MouseIn(895, 0, 1019, 50)) && !(MouseIn(0, 0,
124, 50)))
        {
            MouseOff(&mouse);
            Bar1(895, 0, 1019, 50, 59004);
            Bar1(0, 0, 124, 50, 59004);
            prt_hz24d(935, 12, "退出", 0x0000, "HZK\\Hzk24h");
            prt_hz24d(38, 13, "返回", 0x0000, "HZK\\Hzk24h");

            MouseOn(mouse);
            flag = 0;
```

```
        }
        else if (MousePress(376, 405 - 64, 659, 429 - 64)) //输入*年*
        {
            MouseOff(&mouse);
            Bar1(376, 405 - 64, 659, 429 - 64, 0xffff);
            Getinfo(380, 406 - 64, Ye, 4); //gets.h
            MouseOn(mouse);
        }
        else if (MousePress(378, 405, 659, 429)) //输入*月*
        {
            MouseOff(&mouse);
            Bar1(378, 405, 659, 429, 0xffff);
            Getinfo(380, 406, Mo, 2); //gets.h
            MouseOn(mouse);
        }
        else if (MousePress(378, 469, 659, 493)) //输入*日*
        {
            MouseOff(&mouse);
            Bar1(378, 469, 659, 493, 0xffff);
            Getinfo(380, 470, Da, 2);
            MouseOn(mouse);
        }
        else if (MousePress(376, 533, 659, 557)) //输入*新增感染*
        {
            MouseOff(&mouse);
            Bar1(376, 533, 659, 557, 0xffff);
            Getinfo(380, 534, In, 5);
            MouseOn(mouse);
        }
        else if (MousePress(378, 533 + 64, 659, 557 + 64)) //输入*新增死亡*
        {
            MouseOff(&mouse);
            Bar1(378, 533 + 64, 659, 557 + 64, 0xffff);
            Getinfo(380, 534 + 64, De, 5);
            MouseOn(mouse);
        }
        else if (MousePress(378, 533 + 64 + 64, 659, 557 + 64 + 64)) //输入*新增
治愈*
        {
            MouseOff(&mouse);
            Bar1(378, 533 + 64 + 64, 659, 557 + 64 + 64, 0xffff);
            Getinfo(380, 534 + 64 + 64, He, 5);
            MouseOn(mouse);
        }
```

```
        if (MousePress(356, 609 + 100, 424, 643 + 100))
        {
            key1 = judge_date(Ye, Mo, Da);
            key2 = judge_cases(In, De, He);

            re_area(area);
            if (key1 == 0 && key2 == 0)
            {
                wr_day_inf(filename, Ye, Mo, Da, In, De, He, User);
                prt_hz24d(700, 554, "通报成功", 0, "HZK\\Hzk24h");
                delay(1000);
                return 5;
            }
            else
            {
                prt_hz24d(700, 554, "数据有误", 0, "HZK\\Hzk24h");
                delay(2000);
                Bar1(698, 550, 800, 600, 0x9efc);
            }
        }

        if (MousePress(895, 0, 1019, 50)) //点击退出
        {
            return -1;
        }
        else if (MousePress(0, 0, 124, 50)) //点击返回
        {
            return 5;
        }
    }
}
```

GETTIME.C

```
/*获取当日时间*/
#include "common.h"

void gett(int *year, int *month, int *day)
{
    time_t timep;
    struct tm *p;
    time(&timep);
    p = gmtime(&timep);
    if ((p->tm_hour) < 8) /*将世界标准时间转换为中国时间*/
```

```
        *day = p->tm_mday - 1;
    else
        *day = p->tm_mday;     /*获取当前月份日数,范围是 1-31*/
    *month = 1 + p->tm_mon;    /*获取当前月份,范围是 0-11,所以要加 1*/
    *year = 1900 + p->tm_year; /*获取当前年份,从 1900 开始，所以要加 1900*/
}
```

DAYINFO.C

```
/*********************************************
FileName:dayinfo.c
Author:   刘子恒
Date:    2020/10/16
Description: 该文件主要用于用户通报数据功能
Others: 无
Function List:
        wr_day_inf();   将每日疫情信息写入文件
       countcases();    计算单个文件中各项总数
       countallcases(); 计算所有文件中各项总数
       judge_date();    判断输入的日期是否合法
       judge_cases();       判断输入的疫情信息是否合法
       countareacases();   读取某区域各项数据
       wr_area();       将将要通报的文件名写入文件
       re_area();       将将要通报的文件名读出
        wr_database();      将数据写入数据库
       modify_database();  修改数据库中的数据
       getoneday();     获取某天在数据库中的位置

*********************************************/
#include "common.h"
#include "dayinfo.h"
#include "drawc.h"

/*函数功能：将每日疫情信息写入临时存储文件*/
void wr_day_inf(char area[], char Ye[], char Mo[], char Da[], char In[], char
De[], char He[], char User[])
{
    FILE *fp = NULL;
    DAY *d = NULL;
    int i;
    if ((fp = fopen("userepo.dat", "rb+")) == NULL)
    {
        printf(" error file");
        delay(3000);
```

```
    exit(1); //打开失败，退出程序
}
if ((d = (DAY *)malloc(sizeof(DAY))) == NULL)
{
    printf("memory error Wrfile");
    delay(3000);
    exit(1); //分配空间不足，退出程序
}
memset(d, '\0', sizeof(DAY)); //将结构体每位都写入
for (i = 0; i < 5; i++)
{
    d->area[i] = area[i];
}
for (i = 0; i < 4; i++)
{
    d->year[i] = Ye[i];
}
for (i = 0; i < 2; i++)
{
    d->month[i] = Mo[i];
}
for (i = 0; i < 2; i++)
{
    d->day[i] = Da[i];
}
for (i = 0; i < 6; i++)
{
    d->infestor[i] = In[i];
}
for (i = 0; i < 6; i++)
{
    d->death[i] = De[i];
}
for (i = 0; i < 6; i++)
{
    d->heal[i] = He[i];
}
for (i = 0; i < 13; i++)
{
    d->username[i] = User[i];
}
d->infostate = '0';
d->n[0] = '\r'; //在记事本中打开可以换行
d->n[1] = '\n';
```

```
    fseek(fp, 0, SEEK_END);
    fwrite(d, sizeof(DAY), 1, fp);
    free(d);
    if (fclose(fp) != 0) //关闭文件
    {
        printf("\nError in closing file .");
        delay(3000);
        exit(1);
    }
    delay(500);
}

/*函数功能：计算单个文件中各项总数 */
void countcases(char *filename, int *suminf, int *sumdea, int *sumheal, int
*nowinf)
{
    int i, j;
    int length;
    FILE *fp;
    SAVEDAY *d = NULL;
    *suminf = 0;
    *sumdea = 0; //初始化总数
    *sumheal = 0;
    *nowinf = 0;

    if ((fp = fopen(filename, "rb+")) == NULL)
    {
        printf("\n%s error file", filename);
        delay(3000);
        exit(1); //打开失败，退出程序
    }
    fseek(fp, 0, SEEK_END);
    length = ftell(fp) / sizeof(SAVEDAY);

    if ((d = (SAVEDAY *)malloc(sizeof(SAVEDAY))) == NULL)
    {
        printf("memory error CountCase");
        delay(3000);
        exit(1); //分配空间不足，退出程序
    }
    fseek(fp, (length - 1) * sizeof(SAVEDAY), SEEK_SET); // 读取数据库最后一天的数
据
    fread(d, sizeof(SAVEDAY), 1, fp);
```

```
    *suminf = atoi(d->allinf);
    *sumheal = atoi(d->allhea);
    *sumdea = atoi(d->alldea);
    *nowinf = (*suminf) - (*sumdea) - (*sumheal);

    if (d != NULL)
    {
        free(d);
        d = NULL;
    }
    if (fclose(fp) != 0) //关闭文件
    {
        printf("\nError in closing file %s", filename);
        delay(3000);
        exit(1);
    }
}

/*函数功能：计算所有文件中各项总数 */
void  countallcases(int  filenumbers,  int  *alsuminf,  int  *alsumdea,  int
*alsumheal, int *alnowinf)
{
    int i;
    char tempfilename[10] = {"0.DAT"};
    const char t[4] = {".DAT"};
    int countifn, countdea, countheal, countnoinf;
    *alsuminf = 0;
    *alsumdea = 0;
    *alsumheal = 0;
    *alnowinf = 0;

    for (i = 0; i < filenumbers + 1; i++) // 使用循环逐个读取
    {
        itoa(i, tempfilename, 10);
        strcat(tempfilename, t);
        countcases(tempfilename, &countifn, &countdea, &countheal, &countnoinf);
        *alsuminf = *alsuminf + countifn;
        *alsumdea = *alsumdea + countdea;
        *alsumheal = *alsumheal + countheal;
        *alnowinf = *alsuminf - *alsumdea - *alsumheal;
    }
}

/*函数功能：判断输入的日期是否合法 */
```

```
int judge_date(char year[], char month[], char day[])
{
    int i, m, d;
    int flaga = 1;
    int flagb = 1;
    int flagc = 1;
    int flagd = 1;
    int mon[13] = {1, 31, 29, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31}; // 每月相对日数

    int lea = strlen(month);
    int leb = strlen(day);
    int lec = strlen(year);
    for (i = 0; i < lec; i++)
    {
        if (year[i] > '9' || year[i] < '0' || year[0] != '2' || year[1] != '0' || year[2] != '2' || year[3] != '0')
        {
            flagc = 1;
            break;
        }
        else
            flagc = 0;
    }
    for (i = 0; i < lea; i++)
    {
        if (month[i] > '9' || month[i] < '0')
        {
            flaga = 1;
            break;
        }
        else
            flaga = 0;
    }
    for (i = 0; i < leb; i++)
    {
        if (day[i] > '9' || day[i] < '0')
        {
            flagb = 1;
            break;
        }

        else
            flagb = 0;
```

```
    }

    if (flaga == 0 && flagb == 0)
    {
        m = atoi(month);
        d = atoi(day);
        if (12 < m || m < 1 || mon[0] > d || mon[m] < d)
            flagd = 1;
        else
            flagd = 0;
    }
    return (flaga + flagb + flagc + flagd); // 只有均为零才为零
}

/*函数功能：判断输入的病例是否合法 */
int judge_cases(char inf[], char dea[], char hea[])
{
    int i, m, d;
    int flaga = 1;
    int flagb = 1;
    int flagc = 1;

    int lea = strlen(inf);
    int leb = strlen(dea);
    int lec = strlen(hea);
    for (i = 0; i < lea; i++)
    {
        if (inf[i] > '9' || inf[i] < '0')
        {
            flaga = 1;
            break;
        }
        else
            flaga = 0;
    }
    for (i = 0; i < leb; i++)
    {
        if (dea[i] > '9' || dea[i] < '0')
        {
            flagb = 1;
            break;
        }
        else
            flagb = 0;
```

```
    }
    for (i = 0; i < lec; i++)
    {
        if (hea[i] > '9' || hea[i] < '0')
        {
            flagc = 1;
            break;
        }
        else
            flagc = 0;
    }
    return (flaga + flagb + flagc); // 均为 0 才为 0
}

/*函数功能：将某区域数据读出*/
void countareacases(int areakey, int *suminf, int *sumdea, int *sumheal, int
*nowinf)
{
    char area[8 + 1] = "";
    const char t[4] = {".DAT"};
    int i = 0, j, length;
    FILE *fp;
    char filename[10] = "start";
    SAVEDAY *d = NULL;
    areafunc(areakey, area); // 得到相应地区号包含地区的序列
    *suminf = 0;
    *sumdea = 0;
    *sumheal = 0;
    *nowinf = 0;

    if ((d = (SAVEDAY *)malloc(sizeof(SAVEDAY))) == NULL)
    {

        delay(3000);
        exit(1); //分配空间不足，退出程序
    }

    while (area[i] != '\0') // 使用循环逐个文件打开
    {
        filename[0] = area[i];
        filename[1] = '\0';
        strcat(filename, t);
        filename[5] = '\0';
        i++;
```

```
        if ((fp = fopen(filename, "rb+")) == NULL)
        {
            put_asc16_size(185 + 150 + 80 + 80, 600, 2, 2, filename, 0x0000);
            delay(3000);
            exit(1); //打开失败，退出程序
        }
        fseek(fp, 0, SEEK_END);
        length = ftell(fp) / sizeof(SAVEDAY);
        fseek(fp, (length - 1) * sizeof(SAVEDAY), SEEK_SET);
        fread(d, sizeof(SAVEDAY), 1, fp);
        *suminf = atoi(d->allinf) + *suminf;
        *sumdea = atoi(d->alldea) + *sumdea;
        *sumheal = atoi(d->allhea) + *sumheal;

        if (fclose(fp) != 0) //关闭文件
        {
            printf("\nError in closing file .");
            delay(3000);
            exit(1);
        }
    }
    free(d);
    *nowinf = *suminf - *sumdea - *sumheal;
}

/*函数功能：将将要通报的文件名写入文件*/
void wr_area(int areakey)
{
    FILE *fp;
    if ((fp = fopen("areanow.dat", "rb+")) == NULL)
    {
        printf("areanow.dat file error\n");
        delay(3000);
        exit(1);
    }
    fseek(fp, 0, SEEK_SET);
    fwrite(&areakey, sizeof(int), 1, fp);
    if (fclose(fp) != 0) //关闭文件
    {
        printf("\nError in closing file areanow.");
        delay(3000);
        exit(1);
    }
```

```
}

/*函数功能：将将要通报的文件名读出*/
void re_area(int *areakey)
{
    FILE *fp;
    if ((fp = fopen("areanow.dat", "rb+")) == NULL)
    {
        printf("areanow.dat file error\n");
        delay(3000);
        exit(1);
    }
    fseek(fp, 0, SEEK_SET);
    fread(areakey, sizeof(int), 1, fp);
    if (fclose(fp) != 0) //关闭文件
    {
        printf("\nError in closing file areanow.");
        delay(3000);
        exit(1);
    }
}

/*函数功能：将数据写入数据库*/
void wr_database(DAY *a)
{
    FILE *fp;
    SAVEDAY *d;
    const char t[4] = {".DAT"};
    char filename[10] = "start";
    int i, j, flag = 1, temp, length;
    memset(filename, '\0', sizeof(char) * 10);
    filename[0] = a->area[0];
    filename[1] = '\0';
    strcat(filename, t);
    filename[5] = '\0';
    if ((fp = fopen(filename, "rb+")) == NULL)
    {
        put_asc16_size(185 + 150 + 80 + 80, 600, 2, 2, filename, 0x0000);
        delay(3000);
        exit(1); //打开失败，退出程序
    }
    if ((d = (SAVEDAY *)malloc(sizeof(SAVEDAY))) == NULL)
    {
        printf("memory error Wrfile");
```

```
        delay(3000);
        exit(1); //分配空间不足，退出程序
    }

    memset(d, '\0', sizeof(SAVEDAY));
    fseek(fp, 0, SEEK_END);
    length = ftell(fp) / sizeof(SAVEDAY);
    for (i = 0; i < length; i++)
    {
        fseek(fp, i * sizeof(SAVEDAY), SEEK_SET);
        fread(d, sizeof(SAVEDAY), 1, fp);
        if (strcmp(a->year, d->year) == 0 && strcmp(a->month, d->month) == 0 &&
strcmp(a->day, d->day) == 0)
        {
            flag = 0; // 找到相应日期
            break;
        }
    }
    if (flag == 0)
    {
        fseek(fp, i * sizeof(SAVEDAY), SEEK_SET);
        fread(d, sizeof(SAVEDAY), 1, fp);
        temp = atoi(d->infestor) + atoi(a->infestor);
        itoa(temp, d->infestor, 10);
        temp = atoi(d->death) + atoi(a->death);
        itoa(temp, d->death, 10);
        temp = atoi(d->heal) + atoi(a->heal);
        itoa(temp, d->heal, 10);
        temp = atoi(d->allinf) + atoi(a->infestor);
        itoa(temp, d->allinf, 10);
        temp = atoi(d->alldea) + atoi(a->death);
        itoa(temp, d->alldea, 10);
        temp = atoi(d->allhea) + atoi(a->heal);
        itoa(temp, d->allhea, 10);
        /*修改当日日期数据*/

        fseek(fp, i * sizeof(SAVEDAY), SEEK_SET);
        fwrite(d, sizeof(SAVEDAY), 1, fp);
        i++;

        for (; i < length; i++)
        {
            fseek(fp, i * sizeof(SAVEDAY), SEEK_SET);
            fread(d, sizeof(SAVEDAY), 1, fp);
```

```
            temp = atoi(d->allinf) + atoi(a->infestor);
            itoa(temp, d->allinf, 10);
            temp = atoi(d->alldea) + atoi(a->death);
            itoa(temp, d->alldea, 10);
            temp = atoi(d->allhea) + atoi(a->heal);
            itoa(temp, d->allhea, 10);
            fseek(fp, i * sizeof(SAVEDAY), SEEK_SET);
            fwrite(d, sizeof(SAVEDAY), 1, fp);
            /*更新该日日期信息*/

            //prt_hz24d(470,706,"成功",0,"HZK\\Hzk24h");
        }
    }
    free(d);
    if (fclose(fp) != 0) //关闭文件
    {
        printf("\nError in closing file .");
        delay(3000);
        exit(1);
    }
    delay(500);
}
/*函数功能：修改数据库中信息*/
void modify_database(SAVEDAY *a, char *area)
{
    FILE *fp = NULL;
    SAVEDAY *d = NULL;
    const char t[4] = {".DAT"};
    char filename[10] = "start";
    int i, j, flag = 1, temp, length;
    memset(filename, '\0', sizeof(char) * 10);
    filename[0] = *area;
    filename[1] = '\0';
    strcat(filename, t);
    filename[5] = '\0';
    if ((fp = fopen(filename, "rb+")) == NULL)
    {
        put_asc16_size(185 + 150 + 80 + 80, 600, 2, 2, filename, 0x0000);
        delay(3000);
        exit(1); //打开失败，退出程序
    }
    if ((d = (SAVEDAY *)malloc(sizeof(SAVEDAY))) == NULL)
    {
        printf("memory error Wrfile");
```

```
        delay(3000);
        exit(1); //分配空间不足，退出程序
    }

    memset(d, '\0', sizeof(SAVEDAY));
    fseek(fp, 0, SEEK_END);
    length = ftell(fp) / sizeof(SAVEDAY);
    for (i = 0; i < length; i++)
    {
        fseek(fp, i * sizeof(SAVEDAY), SEEK_SET);
        fread(d, sizeof(SAVEDAY), 1, fp);
        if (strcmp(a->year, d->year) == 0 && strcmp(a->month, d->month) == 0 &&
strcmp(a->day, d->day) == 0)
        {
            flag = 0; //找到相应日期
            break;
        }
    }
    if (flag == 0)
    {
        strcpy(d->infestor, a->infestor);
        strcpy(d->death, a->death);
        strcpy(d->heal, a->heal);
        fseek(fp, i * sizeof(SAVEDAY), SEEK_SET);
        fwrite(d, sizeof(SAVEDAY), 1, fp);
        i++;

        for (; i < length; i++)
        {
            fseek(fp, i * sizeof(SAVEDAY), SEEK_SET);
            fread(d, sizeof(SAVEDAY), 1, fp);
            temp = atoi(d->allinf) + atoi(a->infestor);
            itoa(temp, d->allinf, 10);
            temp = atoi(d->alldea) + atoi(a->death);
            itoa(temp, d->alldea, 10);
            temp = atoi(d->allhea) + atoi(a->heal);
            itoa(temp, d->allhea, 10);

            fseek(fp, i * sizeof(SAVEDAY), SEEK_SET);
            fwrite(d, sizeof(SAVEDAY), 1, fp);
            //将日期更新并写入
            //prt_hz24d(470,706,"成功",0,"HZK\\Hzk24h");
        }
    }
```

```
    free(d);
    if (fclose(fp) != 0) //关闭文件
    {
        printf("\nError in closing file .");
        delay(3000);
        exit(1);
    }
    delay(500);
}
/*函数功能：获取某天的数据*/
void getoneday(SAVEDAY *a, int areakey, int daykey)
{
    char area[8 + 1] = "";
    const char t[4] = {".DAT"};
    int i = 0, j, temp;
    FILE *fp;
    SAVEDAY *d = NULL;
    char filename[10] = "start";
    areafunc(areakey, area);
    memset(a, '\0', sizeof(SAVEDAY));

    if ((d = (SAVEDAY *)malloc(sizeof(SAVEDAY))) == NULL)
    {
        printf("memory error Wrfile");
        delay(3000);
        exit(1); //分配空间不足，退出程序
    }

    while (area[i] != '\0')
    {
        filename[0] = area[i];
        filename[1] = '\0';
        strcat(filename, t);
        filename[5] = '\0';
        i++;

        if ((fp = fopen(filename, "rb+")) == NULL)
        {
            put_asc16_size(185 + 150 + 80 + 80, 600, 2, 2, filename, 0x0000);
            delay(3000);
            exit(1); //打开失败，退出程序
        }
        fseek(fp, daykey * sizeof(SAVEDAY), SEEK_SET);
```

```
        fread(d, sizeof(SAVEDAY), 1, fp);
        strcpy(a->year, d->year);
        strcpy(a->month, d->month);
        strcpy(a->day, d->day);
        temp = atoi(d->infestor) + atoi(a->infestor);
        itoa(temp, a->infestor, 10);
        temp = atoi(d->death) + atoi(a->death);
        itoa(temp, a->death, 10);
        temp = atoi(d->heal) + atoi(a->heal);
        itoa(temp, a->heal, 10);
        temp = atoi(d->allinf) + atoi(a->allinf);
        itoa(temp, a->allinf, 10);
        temp = atoi(d->alldea) + atoi(a->alldea);
        itoa(temp, a->alldea, 10);
        temp = atoi(d->allhea) + atoi(a->allhea);
        itoa(temp, a->allhea, 10);

        if (fclose(fp) != 0) //关闭文件
        {
            printf("\nError in closing file .");
            delay(3000);
            exit(1);
        }
    }
    free(d);
}
```

ADUSER.C

```
/*****************************************
FileName: aduser.c
Author:   刘子恒
Date:    2020/10/26
Description: 该文件主要用于用户管理功能
Others: 无
Function List:
        aduser();       用户管理主函数
        aduser_menu();  用户管理主界面
        aduserdraw();   界面翻页函数
        aduserbutton(); 绘制按钮
        user_ice();       冻结用户
        no_iced();      解冻用户
        button1();    按钮反馈
        button3();    按钮反馈
```

```
       put_User();       打印用户数据
       put_run();      打印用户是否可通报
*******************************************/
#include "aduser.h"
#include "common.h"
#include "lgstate.h"
/*****************************
Function: int admin_menu(void)
Description: 选择操作菜单
Calls:    aduser_menu()
         user_ice()
         no_iced()
Called by: main()
Table Accessed: 无
Table Updated:  无
Input:  无
Output: 无
Return: 8  -> 返回管理员页面
*****************************/
int aduser(void)
{
    int choose = 0;
    int choose1 = 0;
    while (choose != -1)
    {
       choose = aduser_menu(&choose1);
       switch (choose)
       {
       case 1:
          user_ice(); //冻结用户函数
          break;
       case 3:
          no_iced(); //解冻用户函数
          break;
       case -1:
          return 8;
          //返回
       case -2:
          CloseSVGA(); //退出系统
          delay(1000);
          exit(0);
       case -3:
          break; //控制上一页和下一页
       }
```

```
    }
}
/*****************************
Function: int aduser_menu(int *choose1)
Description: 用户管理主界面
Calls:    aduserdraw()
         aduserbutton()
Called by: admin_menu()
Table Accessed: UserInfo.dat
Table Updated:  无
Input:  *choose1
Output: 无
Return: 1  -> 冻结用户
       3  -> 解冻用户
       -1  -> 返回管理员页面
       -2  -> 退出程序
       -3  -> 控制翻页
*****************************/
int aduser_menu(int *choose1)
{
    int choose = 0;
    int flag = 0;
    int long q = 0;
    int m = 0;
    int n = 0;
    int u = 0;
    FILE *fp = NULL;
    if ((fp = fopen("UserInfo.dat", "rb+")) == NULL) //读取文件
    {
        CloseSVGA();
        printf("Unable to open UserInfo.dat!\n");
        getch();
        exit(0);
    }
    fseek(fp, 0, 2);
    q = ftell(fp);
    m = q / sizeof(USER); //m 为总用户数量
    n = m / 5;           //n 为页数
    u = m % 5;           //u 为最后一页的用户数量
    if (u != 0)
    {
        n = n + 1;
    }
    fclose(fp);
```

```
    aduserdraw(choose1); //绘制页面
    Mouse_Init();
    while (choose != -1)
    {
        MouseShow(&mouse);
        aduserbutton(&flag, choose1, n);
        if (MousePress(156, 668, 356, 768)) //冻结用户
        {
            return 1;
        }
        if (MousePress(668, 668, 868, 768)) //解冻用户
        {
            return 3;
        }
        if (MousePress(0, 668, 98, 768)) //返回
        {
            return -1;
        }
        if (MousePress(926, 668, 1024, 768)) //退出
        {
            return -2;
        }
        if (*choose1 == 0 && MousePress(425, 533, 599, 633)) //初始界面点击下一页
        {
            *choose1 = *choose1 + 1;
            return -3;
        }
        if (*choose1 == n - 1 && MousePress(425, 533, 599, 633)) //最后一页界面点
击上一页
        {
            *choose1 = *choose1 - 1;
            return -3;
        }
        if (*choose1 > 0 && MousePress(562, 533, 736, 633) && (*choose1 != n -
1)) //非第一页后点击下一页
        {
            *choose1 = *choose1 + 1;
            return -3;
        }
        if (*choose1 > 0 && MousePress(288, 533, 462, 633) && (*choose1 != n -
1)) //非第一页后点击上一页
        {
            *choose1 = *choose1 - 1;
            return -3;
```

```
        }
    }
    return 0;
}
/*****************************
Function: void aduserdraw(int *choose1)
Description: 绘制界面
Calls:    put_User()
Called by: aduser_menu()
Table Accessed: UserInfo.dat
Table Updated:  无
Input:  *choose1
Output: 无
Return: 无
*****************************/
void aduserdraw(int *choose1)
{
    FILE *fp = NULL;
    int long q = 0;
    int m = 0;
    int n = 0;
    int u = 0;
    if ((fp = fopen("UserInfo.dat", "rb+")) == NULL)
    {
        CloseSVGA();
        printf("Unable to open UserInfo.dat!\n");
        getch();
        exit(0);
    }
    fseek(fp, 0, 2);
    q = ftell(fp);
    m = q / sizeof(USER);
    n = m / 5;
    u = m % 5;
    if (u != 0)
    {
        n = n + 1;
    }
    fclose(fp);
    Bar1(0, 0, 1024, 768, 0x875a);
    Line3(0, 98, 1024, 98, 2, 0);
    Line3(0, 198, 1024, 198, 2, 0);
    Line3(0, 298, 1024, 298, 2, 0);
    Line3(0, 398, 1024, 398, 2, 0);
```

```
    Line3(0, 498, 1024, 498, 2, 0);
    Bar1(0, 666, 100, 768, 0x5d75);
    Bar1(0, 668, 98, 768, 0xb7b1);
    prt_hz24d(25, 706, "返回", 0, "HZK\\Hzk24h");
    Bar1(924, 666, 1024, 768, 0x5d75);
    Bar1(926, 668, 1024, 768, 0xb7b1);
    prt_hz24d(951, 706, "退出", 0, "HZK\\Hzk24h");
    Bar1(154, 666, 358, 768, 0x4b32);
    Bar1(156, 668, 356, 768, 0xa794);
    prt_hz24d(208, 706, "冻结用户", 0, "HZK\\Hzk24h");
    Bar1(666, 666, 870, 768, 0x4b32);
    Bar1(668, 668, 868, 768, 0xa794);
    prt_hz24d(720, 706, "解冻用户", 0, "HZK\\Hzk24h");
    put_User(choose1); //绘制用户
    if (*choose1 == 0)
    {
        Bar1(250, 510, 750, 650, 0x875a);
        Bar1(423, 531, 601, 635, 0);
        Bar1(425, 533, 599, 633, 0x37dc);
        prt_hz24d(476, 571, "下一页", 0, "HZK\\Hzk24h");
    }
    if (*choose1 > 0 && (*choose1 != n - 1))
    {
        Bar1(250, 510, 750, 650, 0x875a);
        Bar1(286, 531, 464, 635, 0);
        Bar1(288, 533, 462, 633, 0x37dc);
        prt_hz24d(339, 571, "上一页", 0, "HZK\\Hzk24h");
        Bar1(560, 531, 738, 635, 0);
        Bar1(562, 533, 736, 633, 0x37dc);
        prt_hz24d(613, 571, "下一页", 0, "HZK\\Hzk24h");
    }
    if (*choose1 == n - 1)
    {
        Bar1(250, 510, 750, 650, 0x875a);
        Bar1(423, 531, 601, 635, 0);
        Bar1(425, 533, 599, 633, 0x37dc);
        prt_hz24d(476, 571, "上一页", 0, "HZK\\Hzk24h");
    }
}
/*****************************
Function: void aduserbutton(int *flag,int *choose1,int n)
Description: 绘制按钮
Calls:    无
Called by: aduser_menu()
```

```
Table Accessed: 无
Table Updated:  无
Input:  *flag,*choose1,n
Output: 无
Return: 无
*****************************/
void aduserbutton(int *flag, int *choose1, int n)
{
    if (*flag == 0 && MouseIn(0, 668, 98, 768))
    {
        MouseOff(&mouse);
        Bar1(0, 668, 98, 768, 0x1639);
        prt_hz24d(25, 706, "返回", 0, "HZK\\Hzk24h");
        MouseOn(mouse);
        *flag = -1;
    }
    if (*flag == 0 && MouseIn(926, 668, 1024, 768))
    {
        MouseOff(&mouse);
        Bar1(926, 668, 1024, 768, 0x1639);
        prt_hz24d(951, 706, "退出", 0, "HZK\\Hzk24h");
        MouseOn(mouse);
        *flag = -2;
    }
    if (*flag == 0 && MouseIn(156, 668, 356, 768))
    {
        MouseOff(&mouse);
        Bar1(156, 668, 356, 768, 0xb9d7);
        prt_hz24d(208, 706, "冻结用户", 0, "HZK\\Hzk24h");
        MouseOn(mouse);
        *flag = -3;
    }
    if (*flag == 0 && MouseIn(668, 668, 868, 768))
    {
        MouseOff(&mouse);
        Bar1(668, 668, 868, 768, 0xb9d7);
        prt_hz24d(720, 706, "解冻用户", 0, "HZK\\Hzk24h");
        MouseOn(mouse);
        *flag = -5;
    }
    if (*choose1 == 0 && (*flag == 0) && MouseIn(425, 533, 599, 633))
    {
        MouseOff(&mouse);
        Bar1(425, 533, 599, 633, 0xdf70);
```

```
        prt_hz24d(476, 571, "下一页", 0, "HZK\\Hzk24h");
        MouseOn(mouse);
        *flag = 1;
    }
    if (*choose1 == n - 1 && (*flag == 0) && MouseIn(425, 533, 599, 633))
    {
        MouseOff(&mouse);
        Bar1(425, 533, 599, 633, 0xdf70);
        prt_hz24d(476, 571, "上一页", 0, "HZK\\Hzk24h");
        MouseOn(mouse);
        *flag = 4;
    }
    if (*choose1 > 0 && (*flag == 0) && MouseIn(288, 533, 462, 633) &&
(*choose1 != n - 1))
    {
        MouseOff(&mouse);
        Bar1(288, 533, 462, 633, 0xdf70);
        prt_hz24d(339, 571, "上一页", 0, "HZK\\Hzk24h");
        MouseOn(mouse);
        *flag = 2;
    }
    if (*choose1 > 0 && (*flag == 0) && MouseIn(562, 533, 736, 633) &&
(*choose1 != n - 1))
    {
        MouseOff(&mouse);
        Bar1(562, 533, 736, 633, 0xdf70);
        prt_hz24d(613, 571, "下一页", 0, "HZK\\Hzk24h");
        MouseOn(mouse);
        *flag = 3;
    }
    else if (*flag == -1 && !(MouseIn(0, 668, 98, 768)))
    {
        MouseOff(&mouse);
        Bar1(0, 668, 98, 768, 0xb7b1);
        prt_hz24d(25, 706, "返回", 0, "HZK\\Hzk24h");
        MouseOn(mouse);
        *flag = 0;
    }
    else if (*flag == -2 && !(MouseIn(926, 668, 1024, 768)))
    {
        MouseOff(&mouse);
        Bar1(926, 668, 1024, 768, 0xb7b1);
        prt_hz24d(951, 706, "退出", 0, "HZK\\Hzk24h");
        MouseOn(mouse);
```

```
        *flag = 0;
    }
    else if (*flag == -3 && !(MouseIn(156, 668, 356, 768)))
    {
        MouseOff(&mouse);
        Bar1(156, 668, 356, 768, 0xa794);
        prt_hz24d(208, 706, "冻结用户", 0, "HZK\\Hzk24h");
        MouseOn(mouse);
        *flag = 0;
    }
    else if (*flag == -5 && !(MouseIn(668, 668, 868, 768)))
    {
        MouseOff(&mouse);
        Bar1(668, 668, 868, 768, 0xa794);
        prt_hz24d(720, 706, "解冻用户", 0, "HZK\\Hzk24h");
        MouseOn(mouse);
        *flag = 0;
    }
    else if (*choose1 == 0 && (*flag == 1) && !(MouseIn(425, 533, 599, 633)))
    {
        MouseOff(&mouse);
        Bar1(425, 533, 599, 633, 0x37dc);
        prt_hz24d(476, 571, "下一页", 0, "HZK\\Hzk24h");
        MouseOn(mouse);
        *flag = 0;
    }
    else if (*choose1 == n - 1 && (*flag == 4) && !(MouseIn(425, 533, 599,
633)))
    {
        MouseOff(&mouse);
        Bar1(425, 533, 599, 633, 0x37dc);
        prt_hz24d(476, 571, "上一页", 0, "HZK\\Hzk24h");
        MouseOn(mouse);
        *flag = 0;
    }
    else if (*choose1 > 0 && (*flag == 2) && !(MouseIn(288, 533, 462, 633)) &&
(*choose1 != n - 1))
    {
        MouseOff(&mouse);
        Bar1(288, 533, 462, 633, 0x37dc);
        prt_hz24d(339, 571, "上一页", 0, "HZK\\Hzk24h");
        MouseOn(mouse);
        *flag = 0;
    }
```

```
    else if (*choose1 > 0 && (*flag == 3) && !(MouseIn(562, 533, 736, 633)) &&
(*choose1 != n - 1))
    {
        MouseOff(&mouse);
        Bar1(562, 533, 736, 633, 0x37dc);
        prt_hz24d(613, 571, "下一页", 0, "HZK\\Hzk24h");
        MouseOn(mouse);
        *flag = 0;
    }
}
/*****************************
Function: int user_ice(void)
Description: 冻结用户
Calls:    user_freeze()
Called by: aduser_menu()
Table Accessed: 无
Table Updated:  无
Input:  无
Output: 无
Return: 1->返回用户管理主函数
*****************************/
int user_ice(void)
{
    char name[12 + 1] = "";
    int choose = 0;
    int flag = 0;
    int key;
    MouseOff(&mouse);
    Bar1(298, 198, 702, 402, 0);
    Bar1(300, 200, 700, 400, 0x8edf);
    prt_hz24d(356, 228, "请输入您要冻结的用户账号", 0, "HZK\\Hzk24h");
    Bar1(402, 258, 598, 312, 0);
    Bar1(404, 260, 596, 310, 0xffff); //输入框
    Bar1(348, 333, 452, 377, 0x4b32);
    Bar1(350, 335, 450, 375, 0x8ff2);
    prt_hz24d(376, 343, "确认", 0, "HZK\\Hzk24h"); //确认键
    Bar1(548, 333, 652, 377, 0x4b32);
    Bar1(550, 335, 650, 375, 0x8ff2);
    prt_hz24d(576, 343, "取消", 0, "HZK\\Hzk24h"); //取消
    Mouse_Init();
    while (choose != -1)
    {
        MouseShow(&mouse);
        button1(&flag);
```

```
    if (MousePress(404, 260, 596, 310))
    {
        MouseOff(&mouse);
        Getinfo(404, 260, name, 12);
        MouseOn(mouse);
    }
    if (MousePress(350, 335, 450, 375))
    {
        choose = -1;
    }
    if (MousePress(550, 335, 650, 375))
    {

        return 1;
    }
}
key = user_freeze(name);
if (key == 1)
{
    Bar1(400, 250, 600, 350, 0x8edf);
    Bar1(403, 253, 597, 347, 0x327d);
    Bar1(405, 255, 595, 345, 0x8edf);
    prt_hz24d(452, 283, "冻结成功", 0, "HZK\\Hzk24h");
    delay(1500);
    return 1;
}
if (key == 0)
{
    Bar1(300, 200, 700, 400, 0x8edf);
    Bar1(303, 203, 697, 397, 0x327d);
    Bar1(305, 205, 695, 395, 0x8edf);
    prt_hz24d(416, 252, "该用户已被冻结", 0, "HZK\\Hzk24h");
    prt_hz24d(440, 323, "请重新输入", 0, "HZK\\Hzk24h");
    delay(1500);
    return 1;
}
if (key == 2)
{
    Bar1(300, 200, 700, 400, 0x8edf);
    Bar1(303, 203, 697, 397, 0x327d);
    Bar1(305, 205, 695, 395, 0x8edf);
    prt_hz24d(428, 252, "未找到该用户", 0, "HZK\\Hzk24h");
    prt_hz24d(440, 323, "请重新输入", 0, "HZK\\Hzk24h");
    delay(1500);
```

```
        return 1;
    }
    return 0;
}

/*****************************
Function: void button1(int *flag)
Description: 按钮反馈
Calls:    无
Called by: aduser_menu()
Table Accessed: 无
Table Updated:  无
Input:  *flag
Output: 无
Return: 无
*****************************/
void button1(int *flag)
{
    if (*flag == 0 && MouseIn(350, 335, 450, 375))
    {
        MouseOff(&mouse);
        Bar1(348, 333, 452, 377, 0xf0a0);
        Bar1(350, 335, 450, 375, 0x1dbe);
        prt_hz24d(376, 343, "确认", 0, "HZK\\Hzk24h");
        MouseOn(mouse);
        *flag = 1;
    }
    if (*flag == 0 && MouseIn(550, 335, 650, 375))
    {
        MouseOff(&mouse);
        Bar1(548, 333, 652, 377, 0xf0a0);
        Bar1(550, 335, 650, 375, 0x1dbe);
        prt_hz24d(576, 343, "取消", 0, "HZK\\Hzk24h");
        MouseOn(mouse);
        *flag = 2;
    }
    else if (*flag == 1 && !(MouseIn(350, 335, 450, 375)))
    {
        MouseOff(&mouse);
        Bar1(348, 333, 452, 377, 0x4b32);
        Bar1(350, 335, 450, 375, 0x8ff2);
        prt_hz24d(376, 343, "确认", 0, "HZK\\Hzk24h");
        MouseOn(mouse);
        *flag = 0;
```

```
    }
    else if (*flag == 2 && !(MouseIn(550, 335, 650, 375)))
    {
        MouseOff(&mouse);
        Bar1(548, 333, 652, 377, 0x4b32);
        Bar1(550, 335, 650, 375, 0x8ff2); //重新选择
        prt_hz24d(576, 343, "取消", 0, "HZK\\Hzk24h");
        MouseOn(mouse);
        *flag = 0;
    }
}
/*****************************
Function: int no_iced(void)
Description:解冻用户
Calls:    int no_iced(void)()
Called by: aduser_menu()
Table Accessed: 无
Table Updated:  无
Input:  无
Output: 无
Return: 1->返回用户管理主函数
*****************************/
int no_iced(void)
{
    char name[12 + 1] = "";
    int choose = 0;
    int flag = 0;
    char key;

    MouseOff(&mouse);
    Bar1(298, 198, 702, 402, 0);
    Bar1(300, 200, 700, 400, 0x8edf);
    prt_hz24d(356, 235, "请输入您要解冻的用户账号", 0, "HZK\\Hzk24h");
    Bar1(402, 278, 598, 317, 0);
    Bar1(404, 280, 596, 315, 0xffff); //输入框
    Bar1(348, 348, 452, 387, 0x4b32);
    Bar1(350, 350, 450, 385, 0x8ff2); //确认
    prt_hz24d(376, 355, "确认", 0, "HZK\\Hzk24h");
    Bar1(548, 348, 652, 387, 0x4b32);
    Bar1(550, 350, 650, 385, 0x8ff2); //取消
    prt_hz24d(576, 355, "取消", 0, "HZK\\Hzk24h");
    Mouse_Init();
    while (choose != -1)
    {
```

```
    MouseShow(&mouse);
    button3(&flag);
    if (MousePress(404, 280, 596, 315))
    {
        MouseOff(&mouse);
        Getinfo(404, 280, name, 12);
        MouseOn(mouse);
    }
    if (MousePress(350, 350, 450, 385))
    {
        choose = -1;
    }
    if (MousePress(550, 350, 650, 385))
    {
        return 1;
    }
}
key = user_unfreeze(name);
if (key == 1)
{
    Bar1(400, 250, 600, 350, 0x8edf);
    Bar1(403, 253, 597, 347, 0x327d);
    Bar1(405, 255, 595, 345, 0x8edf);
    prt_hz24d(452, 283, "解冻成功", 0, "HZK\\Hzk24h"); //有个 bug
    delay(1500);
    return 1;
}
if (key == 0)
{
    Bar1(300, 200, 700, 400, 0x8edf);
    Bar1(303, 203, 697, 397, 0x327d);
    Bar1(305, 205, 695, 395, 0x8edf);
    prt_hz24d(416, 252, "该用户未被冻结", 0, "HZK\\Hzk24h");
    prt_hz24d(440, 323, "请重新输入", 0, "HZK\\Hzk24h");
    delay(1500);
    return 1;
}
if (key == 2)
{
    Bar1(300, 200, 700, 400, 0x8edf);
    Bar1(303, 203, 697, 397, 0x327d);
    Bar1(305, 205, 695, 395, 0x8edf);
    prt_hz24d(428, 252, "未找到该用户", 0, "HZK\\Hzk24h");
    prt_hz24d(440, 323, "请重新输入", 0, "HZK\\Hzk24h");
```

```
        delay(1500);
        return 1;
    }
    return 0;
}
/*****************************
Function: void button3(int *flag)
Description: 按钮反馈
Calls:    无
Called by: aduser_menu()
Table Accessed: 无
Table Updated:  无
Input:  *flag
Output: 无
Return: 无
*****************************/
void button3(int *flag)
{
    if (*flag == 0 && MouseIn(350, 350, 450, 385))
    {
        MouseOff(&mouse);
        Bar1(348, 348, 452, 387, 0xf0a0);
        Bar1(350, 350, 450, 385, 0x1dbe);
        prt_hz24d(376, 355, "确认", 0, "HZK\\Hzk24h");
        MouseOn(mouse);
        *flag = 1;
    }
    if (*flag == 0 && MouseIn(550, 350, 650, 385))
    {
        MouseOff(&mouse);
        Bar1(548, 348, 652, 387, 0xf0a0);
        Bar1(550, 350, 650, 385, 0x1dbe);
        prt_hz24d(576, 355, "取消", 0, "HZK\\Hzk24h");
        MouseOn(mouse);
        *flag = 2;
    }
    else if (*flag == 1 && !(MouseIn(350, 350, 450, 385)))
    {
        MouseOff(&mouse);
        Bar1(348, 348, 452, 387, 0x4b32);
        Bar1(350, 350, 450, 385, 0x8ff2);
        prt_hz24d(376, 355, "确认", 0, "HZK\\Hzk24h");
        MouseOn(mouse);
        *flag = 0;
```

```
    }
    else if (*flag == 2 && !(MouseIn(550, 350, 650, 385)))
    {
        MouseOff(&mouse);
        Bar1(548, 348, 652, 387, 0x4b32);
        Bar1(550, 350, 650, 385, 0x8ff2);
        prt_hz24d(576, 355, "取消", 0, "HZK\\Hzk24h");
        MouseOn(mouse);
        *flag = 0;
    }
}
/*****************************
Function: int put_User(int *choose1)
Description:打印用户信息
Calls:    put_run()
Called by: aduserdraw()
Table Accessed: UserInfo.dat
Table Updated:  无
Input:  *flag
Output: 无
Return: -1
*****************************/
int put_User(int *choose1)
{
    FILE *fp = NULL;

    USER *e = NULL;
    // char s[13]="";
    int i = 0, page, u, count, length;
    // char ordernum[5];
    if ((fp = fopen("UserInfo.dat", "rb+")) == NULL)
    {
        CloseSVGA();
        printf("Unable to open UserInfo.dat!\n");
        getch();
        exit(0);
    }
    if ((e = (USER *)malloc(sizeof(USER))) == NULL)
    {

        delay(3000);
        exit(1); //分配空间不足，退出程序
    }
```

```
    fseek(fp, 0, SEEK_END);

    length = ftell(fp) / sizeof(USER);
    page = length / 5;
    u = length % 5;

    if (page == 0 || (page - *choose1) == 0)
        count = u;
    else
        count = 5;

    fseek(fp, 0, SEEK_SET);
    for (i = 0; i < count; i++)
    {
        fseek(fp, (i + 5 * (*choose1)) * sizeof(USER), SEEK_SET);
        fread(e, sizeof(USER), 1, fp);

        prt_hz16_asc16_size(10,  20  +  100  *  i,  2,  2,  e->UserName,  0,
"HZK\\Hzk16k");

        prt_hz16_asc16_size(160 + 40, 20 + 100 * (i), 2, 2, e->password, 0,
"HZK\\Hzk16k");
        prt_hz16_asc16_size(300 + 40, 20 + 100 * (i), 2, 2, e->IDnumber, 0,
"HZK\\Hzk16k");
        put_run(e->UserState, i);
    }
    fclose(fp);
    return -1;
}
/*****************************
Function: void put_run(char s,int i)
Description:判断用户是否冻结
Calls:    无
Called by: put_User()
Table Accessed: 无
Table Updated:  无
Input:  *flag
Output: s , i;
Return: 无
*****************************/
void put_run(char s, int i)
{
    if (s == '1')
    {
```

```
        prt_hz24d(910, 20 + 100 * (i), "可通报", 0, "HZK\\Hzk24h");
    }
    if (s == '2')
    {
        prt_hz24d(910, 20 + 100 * (i), "冻结", 0, "HZK\\Hzk24h");
    }
}
```

ADOP.C

```
/*******************************************
FileName: adop.c
Author:   刘子恒
Date:    2020/10/27
Description: 该文件主要用于管理员选择操作界面
Others: 无
Function List:
       admin_menu() 选择操作菜单函数
       admindraw()  画图函数
       admindraw2() 画图函数
       adminbutton()按钮反馈函数
*******************************************/
#include "adop.h"
#include "common.h"

/*****************************
Function: int admin_menu(void)
Description: 选择操作菜单
Calls:    admindraw()
         admindraw2()
         adminbutton()
Called by: main()
Table Accessed: 无
Table Updated:  无
Input:  无
Output: 无
Return: -1  -> 退出
       0   -> 欢迎界面函数
       9   -> 用户信息管理函数
       10  -> 审核用户通报函数
       11  -> 管理疫情数据函数
*****************************/
int admin_menu(void)
{
```

```
    int flag = 0;
    admindraw(); //画图函数
    Mouse_Init();
    while (1)
    {
        MouseShow(&mouse);
        admin_button(&flag);              //反馈函数
        if (MousePress(400, 450, 650, 500)) //审核用户通报
            return 10;
        else if (MousePress(400, 550, 650, 600)) //管理疫情数据
            return 11;
        else if (MousePress(400, 650, 650, 700)) //用户信息管理
            return 9;
        else if (MousePress(0, 0, 150, 55)) //返回
            return 0;
        else if (MousePress(874, 0, 1024, 55)) //退出
            return -1;
    }
}
/*****************************
Function: admindraw(void)
Description: 画图函数
Calls:    无
Called by: admin_menu()
Table Accessed: 无
Table Updated:  无
Input:  无
Output: 无
Return: 无
*****************************/
void admindraw(void)
{
    Bar1(0, 0, 1024, 768, 1658);

    Bar1(5, 5, 155, 60, 0x5b4f);
    Bar1(0, 0, 150, 55, 59004);
    prt_hz24(45, 12, "返回", 0, "HZK\\Hzk24h");
    Bar1(869, 5, 1029, 60, 0x5b4f);
    Bar1(874, 0, 1024, 55, 59004);
    prt_hz24(919, 12, "退出", 0, "HZK\\Hzk24h");
    admindraw2();

    Bar1(406, 456, 656, 506, 0x7bef);
    Bar1(400, 450, 650, 500, 44925);
```

```
    Circlefill(420, 475, 10, 0xffff); //地点管理

    Bar1(406, 556, 656, 606, 0x7bef);
    Bar1(400, 550, 650, 600, 44925);
    Circlefill(420, 575, 10, 0xffff); //司机信息管理

    Bar1(406, 656, 656, 706, 0x7bef);
    Bar1(400, 650, 650, 700, 44925);
    Circlefill(420, 675, 10, 0xffff); //用户信息查询

    prt_hz24(453, 462, "审核用户通报", 0, "HZK//HzK24h");
    prt_hz24(453, 562, "管理疫情数据", 0, "HZK//HzK24h");
    prt_hz24(453, 662, "用户信息管理", 0, "HZK//HzK24h");
    Line3(426, 250, 426, 400, 3, 0);
    Line3(426, 250, 626, 250, 3, 0);
    Line3(626, 250, 626, 400, 3, 0);
    Circlefill(526, 190, 60, 0);
    Circlefill(526, 190, 55, 1658); //管理员图标
}
/*****************************
Function: admindraw2(void)
Description: 画图函数
Calls:    无
Called by: admin_menu()
Table Accessed: 无
Table Updated:  无
Input:  无
Output: 无
Return: 无
*****************************/
void admindraw2(void)
{
    prt_hz16_size(100, 100, 2, 2, "尊", 0, "HZK\\Hzk16s");
    prt_hz16_size(100, 150, 2, 2, "敬", 0, "HZK\\Hzk16s");
    prt_hz16_size(100, 200, 2, 2, "的", 0, "HZK\\Hzk16s");
    prt_hz16_size(100, 250, 2, 2, "管", 0, "HZK\\Hzk16s");
    prt_hz16_size(100, 300, 2, 2, "理", 0, "HZK\\Hzk16s");
    prt_hz16_size(100, 350, 2, 2, "员", 0, "HZK\\Hzk16s");
    prt_hz16_size(140, 100, 2, 2, "欢", 0, "HZK\\Hzk16s");
    prt_hz16_size(140, 150, 2, 2, "迎", 0, "HZK\\Hzk16s");
    prt_hz16_size(140, 200, 2, 2, "您", 0, "HZK\\Hzk16s");
    prt_hz16_size(230, 100, 3, 3, "请", 0, "HZK\\Hzk16s");
    prt_hz16_size(230, 160, 3, 3, "选", 0, "HZK\\Hzk16s");
    prt_hz16_size(230, 220, 3, 3, "择", 0, "HZK\\Hzk16s");
```

```
    prt_hz16_size(230, 280, 3, 3, "您", 0, "HZK\\Hzk16s");
    prt_hz16_size(230, 340, 3, 3, "要", 0, "HZK\\Hzk16s");
    prt_hz16_size(230, 400, 3, 3, "执", 0, "HZK\\Hzk16s");
    prt_hz16_size(230, 460, 3, 3, "行", 0, "HZK\\Hzk16s");
    prt_hz16_size(230, 520, 3, 3, "的", 0, "HZK\\Hzk16s");
    prt_hz16_size(230, 580, 3, 3, "操", 0, "HZK\\Hzk16s");
    prt_hz16_size(230, 640, 3, 3, "作", 0, "HZK\\Hzk16s");
}
/*****************************
Function: admin_button(int *flag)
Description: 按钮反馈函数
Calls:    无
Called by: admin_menu()
Table Accessed: 无
Table Updated:  无
Input:  &flag
Output: 无
Return: 无
*****************************/
void admin_button(int *flag)
{
    if (*flag == 0 && MouseIn(0, 0, 150, 55)) //返回框反馈
    {
        MouseOff(&mouse);
        Bar1(0, 0, 150, 55, 56603);
        prt_hz24d(45, 12, "返回", 0, "HZK\\Hzk24h");
        MouseOn(mouse);
        *flag = 1;
    }
    if (*flag == 0 && MouseIn(874, 0, 1024, 55))
    {
        MouseOff(&mouse);
        Bar1(874, 0, 1024, 55, 56603);
        prt_hz24d(919, 12, "退出", 0, "HZK\\Hzk24h");
        MouseOn(mouse);
        *flag = 2;
    }
    if (*flag == 0 && MouseIn(400, 450, 650, 500))
    {
        MouseOff(&mouse);
        Bar1(400, 450, 650, 500, 46876);
        Circlefill(420, 475, 10, 0x429a);
        prt_hz24d(453, 462, "审核用户通报", 0, "HZK//HzK24h");
        MouseOn(mouse);
```

```
    *flag = 3;
}
if (*flag == 0 && MouseIn(400, 550, 650, 600))
{
    MouseOff(&mouse);
    Bar1(400, 550, 650, 600, 46876);
    Circlefill(420, 575, 10, 0x429a);
    prt_hz24d(453, 562, "管理疫情数据", 0, "HZK//HzK24h");
    MouseOn(mouse);
    *flag = 4;
}
if (*flag == 0 && MouseIn(400, 650, 650, 700))
{
    MouseOff(&mouse);
    Bar1(400, 650, 650, 700, 46876);
    Circlefill(420, 675, 10, 0x429a);
    prt_hz24d(453, 662, "用户信息管理", 0, "HZK//HzK24h");
    MouseOn(mouse);
    *flag = 5;
}
else if (*flag == 1 && !(MouseIn(0, 0, 150, 55)))
{
    MouseOff(&mouse);
    Bar1(0, 0, 150, 55, 59004);
    prt_hz24d(45, 12, "返回", 0, "HZK\\Hzk24h");
    MouseOn(mouse);
    *flag = 0;
}
else if (*flag == 2 && !(MouseIn(874, 0, 1024, 55)))
{
    MouseOff(&mouse);
    Bar1(874, 0, 1024, 55, 59004);
    prt_hz24d(919, 12, "退出", 0, "HZK\\Hzk24h");
    MouseOn(mouse);
    *flag = 0;
}
else if (*flag == 3 && !(MouseIn(400, 450, 650, 500)))
{
    MouseOff(&mouse);
    Bar1(400, 450, 650, 500, 44925);
    Circlefill(420, 475, 10, 0xffff);
    prt_hz24d(453, 462, "审核用户通报", 0, "HZK//HzK24h");
    MouseOn(mouse);
    *flag = 0;
```

```
    }
    else if (*flag == 4 && !(MouseIn(400, 550, 650, 600)))
    {
        MouseOff(&mouse);
        Bar1(400, 550, 650, 600, 44925);
        Circlefill(420, 575, 10, 0xffff);
        prt_hz24d(453, 562, "管理疫情数据", 0, "HZK//HzK24h");
        MouseOn(mouse);
        *flag = 0;
    }
    else if (*flag == 5 && !(MouseIn(400, 650, 650, 700)))
    {
        MouseOff(&mouse);
        Bar1(400, 650, 650, 700, 44925);
        Circlefill(420, 675, 10, 0xffff);
        prt_hz24d(453, 662, "用户信息管理", 0, "HZK//HzK24h");
        MouseOn(mouse);
        *flag = 0;
    }
}
```

MAIN.C

```
/************************
程序主函数
**********************/
#include "common.h"
#include "main.h"
void main(void)
{

    int page = 0; //页面切换变量

    SetSVGA64k();
    Mouse_Init();

    while (1)
    {
        switch (page)
        {
        case 0: //欢迎页面
            page = welcome();
            break;
        case 1: //登录页面
```

```
    page = login_user();
    break;

case 2: //管理员登录页面
    page = login_admin();
    break;
case 3: //注册页面
    page = register_user();
    break;

case 4: //忘记密码页面
    page = refound_user();
    break;

case 5: //可视化界面
    page = visual_page();
    break;

case 6: //通报界面
    page = report();
    break;

case 7: //数据查询界面
    page = inquiry();
    break;

case 8: //管理员界面
    page = admin_menu();
    break;

case 9: //冻结用户界面
    page = aduser();
    break;

case 10: //数据审核页面
    page = admin_review();
    break;

case 11: //数据修改页面
    page = admin_modify();
    break;

case 12: //数据展示页面（柱状图）
    page = showinfo_main();
```

```
            break;

        case -1:        //退出程序
            CloseSVGA(); //关闭图形界面
            exit(0);
        }
    }
}
```

11.代码分工

组员	文件名	代码量	占比
刘子恒	adop.c	162	3.1%
	aduser.c	566	10.9%
	dayinfo.c	509	9.8%
	inquiry.c	229	4.4%
	drawc.c	188	3.6%
	gettime.c	14	0.3%
	lgstate.c	729	14.0%
	main.c	56	1.1%
	refound.c	207	4.0%
	register,c	217	4.2%
	report.c	198	3.8%
总行数		**3075**	**59.2%**
韩海若	welcome.c	75	1.4%
	admod.c	327	6.3%
	areview.c	328	6.3%
	choice.c	418	8.0%
	draw.c	114	2.2%
	login.c	194	3.7%
	showinfo.c	189	3.6%
	visual.c	367	7.1%
	admin.c	108	2.1%
总行数		**2120**	**40.8%**
总行数		**5195**	