大概应该包括 项目背景 技术路线 需求分析 关键代码 技术创新 项目展示 未来展望

第十组 项目三 项目总结

组长：程锦国 组员：张瀚霖，骆致远，张谦 技术顾问：王子捷

1. 项目背景

设计“通讯录管理系统”使其具有数据插入、修改、删除、显示和查询等功能。

1. 技术路线

第一天：张谦开发项目基本架构，包括头文件，用到的类，结构体，及其所需函数的规划。

第二天：程锦国负责对项目的前四个功能实现，并交由骆致远同学完成代码的规范美观和页面完善。

第三天：程锦国负责对项目的密码系统进行实现，张瀚霖负责对项目的黑名单系统及来电显示，邮件的实现。

第四天：张谦负责对项目进行测试查找bug，程锦国于张瀚霖负责解决bug。

第五天：张谦负责项目的ppt，以及答辩。

1. 需求分析

基础部分：

1. 基本数据：人名、工作单位、电话号码，生日和E-mail地址

拓展：最近三次通话记录以及通话时长

2， 可对记录中的信息进行修改，具体：

A.人名、工作单位、电话号码、EM可以修改，通话记录与时长无法修改

B.以上所有数据可以删除

C.可以储存新的联系人

3. 可按人名进行查询

4. 将所有数据存放到.txt或者其他类似的数据储存文件中，确保不会再程序运行完成后丢失

进一步拓展：

1. 黑名单 添加一个来电测试模块。如果是黑名单电话来电或短信，直接拦截下来，用户也可以查看被拦截的邮件信息。
2. 应用锁：手机通讯录里存放的都是比较重要的个人隐私信息，要查看其中内容可以设置一个锁设备当刚运行程序时需输入密码才能进行后续操作（输入密码时只显示\*\*\*\*\*，不显示具体密码，并且有人性化的密码容错机制
3. 来电显示：模拟来电显示
4. 模拟邮箱邮件到达
5. 关键代码
6. 通讯录基础部分
7. 新增联系人

void Guidance::AddInfo(AddressBook\* MyAddressBook, int\* AddressTop)

{

cout << endl;

cout << "Please input infomation as below format:" << endl;

cout << "At first input the name" << endl;

//cout << "JohnSmith SCU 12312341234 2020/07/14 123123123@163.com 2020/01/01/08:01:01 2020/01/01/08:01:02 2020/01/01/08:01:02" << endl;

cout << "If what to quit this function, please input 0." << endl;

while (true) {

cin >> MyAddressBook[\*AddressTop].sName;

if (MyAddressBook[\*AddressTop].sName[0] == '0') {

break;

}

cout << "please enter the workunit:" << endl;

cin >> MyAddressBook[\*AddressTop].sWorkUnit;

cout << "please enter the number:" << endl;

cin >> MyAddressBook[\*AddressTop].sPhoneNumber;

cout << "please enter the birthay:" << endl;

cin >> MyAddressBook[\*AddressTop].sBirthday;

cout << "please enter the mail:" << endl;

cin >> MyAddressBook[\*AddressTop].sE\_mailAddress;

cout << "please enter the num1 call:" << endl;

cin >> MyAddressBook[\*AddressTop].sCallLog1;

cout << "please enter the num2 call:" << endl;

cin >> MyAddressBook[\*AddressTop].sCallLog2;

cout << "please enter the num3 call:" << endl;

cin >> MyAddressBook[\*AddressTop].sCallLog3;

MyAddressBook[\*AddressTop].flag = 1;

(\*AddressTop)++;

cout << "ok,i get it!" << endl;

Sleep(1000);

system("cls");

break;

}

}

1. 修改联系人

void Guidance::ModifyInfo(AddressBook\* MyAddressBook, int AddressTop)

{

while (true) {

cout << endl;

cout << "Please input the name of the information you want to change." << endl;

cout << "If what to quit this function, please input 0." << endl;

char name[20];

cin >> name;

if (name[0] == '0') {

break;

}

int flag = 0;

for (int i = 0; i < AddressTop; i++) {

if (strcmp(MyAddressBook[i].sName, name) == 0 && MyAddressBook[i].flag == 1) {

cout << "We have found the name, please input new information." << endl;

cout << "please enter the new workunit:" << endl;

cin >> MyAddressBook[i].sWorkUnit;

cout << "please enter the new number:" << endl;

cin >> MyAddressBook[i].sPhoneNumber;

cout << "please enter the new birthday:" << endl;

cin >> MyAddressBook[i].sBirthday;

cout << "please enter the new mail:" << endl;

cin >> MyAddressBook[i].sE\_mailAddress;

flag = 1;

cout << "ok,i finish!" << endl;

Sleep(1000);

system("cls");

}

}break;

if (flag == 0) {

cout << "Do not find the name." << endl;

Sleep(1000);

system("cls");

break;

}

}

}

1. 寻找联系人

void Guidance::CheckAddressBookList(AddressBook\* MyAddressBook, int AddressTop)

{

while (true) {

cout << endl;

cout << "Please input the name of the information you want to check." << endl;

cout << "If what to quit this function, please input 0." << endl;

char name[20];

cin >> name;

if (name[0] == '0') {

break;

}

int flag = 0;

for (int i = 0; i < AddressTop; i++) {

if (strcmp(MyAddressBook[i].sName, name) == 0 && MyAddressBook[i].flag == 1) {

cout << "We have found the name, below is its information." << endl;

cout << "Name: " << MyAddressBook[i].sName << endl;

cout << "WorkUnit: " << MyAddressBook[i].sWorkUnit << endl;

cout << "PhoneNumber: " << MyAddressBook[i].sPhoneNumber << endl;

cout << "Birthday: " << MyAddressBook[i].sBirthday << endl;

cout << "E\_mailAddress: " << MyAddressBook[i].sE\_mailAddress << endl;

cout << "sCallLog1: " << MyAddressBook[i].sCallLog1 << endl;

cout << "sCallLog2: " << MyAddressBook[i].sCallLog2 << endl;

cout << "sCallLog3: " << MyAddressBook[i].sCallLog3 << endl;

flag = 1;

}

}

if (flag == 0) {

cout << "Do not find the name." << endl;

Sleep(1000);

system("cls");

break;

}

cout << "do you want to back to the menu?you can take any key to back!" << endl;

getchar();

getchar();

system("cls");

break;

}

}

1. 删除联系人

void Guidance::DeleteInfo(AddressBook\* MyAddressBook, int AddressTop)

{

while (true) {

cout << endl;

cout << "Please input the name of the information you want to delete." << endl;

cout << "If what to quit this function, please input 0." << endl;

char name[20];

cin >> name;

if (name[0] == '0') {

Sleep(1000);

system("cls");

break;

}

int flag = 0;

for (int i = 0; i < AddressTop; i++) {

if (strcmp(MyAddressBook[i].sName, name) == 0 && MyAddressBook[i].flag == 1) {

cout << "We have found the name, and deleted its information." << endl;

MyAddressBook[i].flag = 0;

flag = 1;

Sleep(1000);

system("cls");

}

}break;

if (flag == 0) {

cout << "Do not find the name." << endl;

Sleep(1000);

system("cls");

break;

}

}

}

1. 程序与文件交互

void get\_data(int\* AddressTop) {

std::ifstream in("source.txt");

if (in.is\_open())

{

while (!in.eof()) {

in >> MyAddressBook[\*AddressTop].sName;

in >> MyAddressBook[\*AddressTop].sWorkUnit;

in >> MyAddressBook[\*AddressTop].sPhoneNumber;

in >> MyAddressBook[\*AddressTop].sBirthday;

in >> MyAddressBook[\*AddressTop].sE\_mailAddress;

in >> MyAddressBook[\*AddressTop].sCallLog1;

in >> MyAddressBook[\*AddressTop].sCallLog2;

in >> MyAddressBook[\*AddressTop].sCallLog3;

if (MyAddressBook[\*AddressTop].sCallLog3[0] != '\0') {

(\*AddressTop)++;

}

MyAddressBook[\*AddressTop - 1].flag = 1;

}

in.close();

}

}

void save\_data(int AddressTop) {

std::ofstream out("source.txt");

if (out.is\_open())

{

for (int i = 0; i < AddressTop; i++) {

if (MyAddressBook[i].flag == 1) {

out << MyAddressBook[i].sName << " " << MyAddressBook[i].sWorkUnit << " ";

out << MyAddressBook[i].sPhoneNumber << " " << MyAddressBook[i].sBirthday << " ";

out << MyAddressBook[i].sE\_mailAddress << " " << MyAddressBook[i].sCallLog1 << " ";

out << MyAddressBook[i].sCallLog2 << " " << MyAddressBook[i].sCallLog3 << endl;

}

}

out.close();

}

}

1. 密码系统
2. 密码隐秘输入功能

void Guidance::Password()

{

ifstream \_file("data.txt");

cout << "welcome to address book system!" << endl;

if (!\_file)

{

cout << "Password not created." << endl;

Change\_Password();

cout << "succeed in creating,Please log in with password." << endl << endl;

}

log\_in();

}

void Guidance::log\_in()

{

ifstream \_file("data.txt");

cout << "please input the PIN" << endl;

char sYourInput[100];

char sPassword[100];

int timeLimit = 3;

\_file >> sPassword;

\_file.close();

while (1)

{

char ch = \_getch();

int i;

for (i = 0; ch != '\r'; i++)

{

sYourInput[i] = ch;

cout << "\*";

ch = \_getch();

}

sYourInput[i] = '\0';

cout << endl;

if (mycmp(sYourInput, sPassword) == 1)

{

cout << "Welcome to the system!" << endl;

break;

}

else if (timeLimit != 0)

{

cout << "There are " << timeLimit << " chances left." << endl;

timeLimit--;

}

else

{

cout << "Chances ran out,quiting system." << endl;

exit(1);

}

}

}

1. 密码修改功能

void Guidance::Change\_Password()

{

cout << "Now,start creating a new password." << endl;

ofstream out("data.txt", ios\_base::out);

char Cpass[100];

cin >> Cpass;

out << Cpass;

out.close();

}

1. 模拟电话邮件输入及黑名单系统
2. 模拟来电功能及模拟黑户功能

void BlackList::ResistNum()

{

std::ofstream add;

add.open("BlackList.txt", std::ios::app);

add.close();

std::ifstream in("BlackList.txt");

if (!in)

{

std::cout << "open file failed." << std::endl;

}

else

{

std::string CallNum;

std::cout << "The phone call is from:" << std::endl;

std::cin >> CallNum;

std::string read;

bool i = true;

while (getline(in, read))

{

if (CallNum == read)

{

std::cout << "The call is form blacklist number,refused." << std::endl;

i = false;

break;

}

}

if (i == true)

{

std::cout << "your phone is ringing!" << std::endl;

}

i = true;

}

Sleep(2000);

system("cls");

}

1. 模拟邮件功能及验证黑户功能

void BlackList::ResistEmail()

{

std::ofstream add;

add.open("BlackList.txt", std::ios::app);

add.close();

std::ifstream in("BlackList.txt");

if (!in)

{

std::cout << "open file failed." << std::endl;

}

else

{

std::string Email;

std::cout << "The email is from:" << std::endl;

std::cin >> Email;

std::string content;

std::cout << "The content of the email is:" << std::endl;

std::cin >> content;

std::string read;

bool i = true;

while (getline(in, read))

{

if (Email == read)

{

std::cout << "The email is from blacklist address,refused." << std::endl;

std::string check;

std::cout << "Check the resisted Email?Please enter yes or no." << std::endl;

std::cin >> check;

if (check == "yes")

{

std::cout << "The content is:" << content << std::endl;

}

i = false;

break;

}

}

if (i == true)

{

std::cout << "New email available." << std::endl;

}

i = true;

}

Sleep(2000);

system("cls");

}

1. 黑名单新增功能

void BlackList::AddBlackNum()

{

bool repeat = false;

std::string BlackListNum;

std::cout << "enter new blacklist phone number." << std::endl;

std::cin >> BlackListNum;

std::ofstream add;

add.open("BlackList.txt", std::ios::app);

std::string get;

std::ifstream in("BlackList.txt");

if (!in)

{

std::cout << "open file failed." << std::endl;

}

else

{

while (getline(in, get))

{

if (BlackListNum == get)

{

repeat = true;

break;

}

}

}

if (repeat == false)

{

add << BlackListNum << "\n";

std::cout << "Black phone number added successfully." << std::endl;

std::cout << "The blacklist before adding has:" << std::endl;

BlackListDisp();

add.close();

cout << "Take any key to back!" << endl;

getchar();

getchar();

Sleep(2000);

system("cls");

}

else

{

std::cout << "Blacklist number repeat." << std::endl;

std::cout << "The blacklist before adding has:" << std::endl;

BlackListDisp();

add.close();

cout << "Take any key to back!" << endl;

getchar();

getchar();

Sleep(2000);

system("cls");

}

}

void BlackList::AddBlackEmail()

{

bool repeat = false;

std::string BlackListEmail;

std::cout << "enter new blacklist email adress." << std::endl;

std::cin >> BlackListEmail;

std::ofstream add;

add.open("BlackList.txt", std::ios::app);

std::string get;

std::ifstream in("BlackList.txt");

if (!in)

{

std::cout << "open file failed." << std::endl;

}

else

{

while (getline(in, get))

{

if (BlackListEmail == get)

{

repeat = true;

break;

}

}

}

if (repeat == false)

{

add << BlackListEmail << "\n";

std::cout << "Black email address added successfully." << std::endl;

std::cout << "The blacklist before adding has:" << std::endl;

BlackListDisp();

add.close();

cout << "Take any key to back!" << endl;

getchar();

getchar();

Sleep(1000);

system("cls");

}

else

{

std::cout << "Blacklist email repeat." << std::endl;

std::cout << "The blacklist before adding has:" << std::endl;

BlackListDisp();

add.close();

cout << "Take any key to back!" << endl;

getchar();

getchar();

Sleep(2000);

system("cls");

}

}

1. 删除黑名单功能

void BlackList::DeleteBlackInfo()

{

char cLine[200] = { '\0' };

std::string sTempAns;

std::string sContent;

int iNumOfContact;

int LineNum = 0;

std::cout << "You have chosen to delete the BlackList information." << std::endl;

BlackListDisp();

std::cout << "Please follow the guidelines to enter the changed BlackList information." << std::endl;

std::cout << "The number of line you want to delete (from top to bottom):";

std::cin >> iNumOfContact;

std::ifstream in("BlackList.txt", std::ios::in);

std::string lines;

while (getline(in, lines))

{

LineNum++;

}

if (iNumOfContact <= LineNum)

{

if (in.is\_open())

{

int iTemp = 0;

while (in.getline(cLine, sizeof(cLine)))

{

iTemp++;

if (iTemp != iNumOfContact)

{

sTempAns += charToStr(cLine);

}

}

in.close();

std::ofstream out("BlackList.txt", std::ios::out);

if (out.is\_open())

{

out.flush();

out << sTempAns;

out.close();

}

else

{

std::cout << "Error opening file!" << std::endl;

exit(1);

}

std::cout << "Wonderful!BlackList information has been deleted successfully!" << std::endl;

Sleep(2000);

system("cls");

}

else

{

std::cout << "Error opening file!" << std::endl;

exit(1);

}

}

else

{

std::cout << "Wrong number of line!" << std::endl;

cout << "Take any key to back!" << endl;

getchar();

getchar();

Sleep(2000);

system("cls");

}

}

1. 技术创新
2. 通讯录的基本操作用了引用方法，在不声明全局变量的情况下，使变量在不同代码文件之间产生影响。
3. 对于密码系统
4. 输入的密码会有\*的屏蔽效果
5. 可以修改密码
6. 利用system（“cls”）功能，清除屏幕多余字符。美化屏幕。
7. 按照黑名单的行对黑名单内容进行删除。

1. 项目展示（答辩实操）
2. 未来展望

可以完善的地方：

1. 做成一个图形化界面，而不是黑框字符界面。
2. 如果可以接入网络，就可以实现电话与email的接入。而不必自己手动模拟。
3. 对于密码系统，如果输入三次密码没有通过（已实现），直接进入锁定状态，并将软件冻结一段时间。

尽管已经下了不少功夫，我们的交互界面做的还是不太友好，而且代码的技术含量也并不算高。希望今后有机会能做得更好。