

目 录

1. 引言	1
1.1 背景	1
1.2 要解决的问题	1
1.3 小组成员分工	3
2. 系统框架	4
3. 数据结构设计	5
3.1 学生信息初始数据的获取	5
3.2 学生信息数据在内存中的储存	6
3.3 学生信息数据在外部文件的储存	8
3.4 登录用户数据在内存中的储存	9
3.5 登录用户数据在外部文件的储存	10
4. 关键技术	11
4.1 用户登录功能（对应‘密码的检测和管理’）	11
4.1.1 初始化	12
4.1.2 注册	14
4.1.3 密码检测	15
4.1.4 密码修改	17
4.2 学生信息管理系统操作菜单功能（对应‘提供简单的操作菜单’）	18
4.3 根据登录账号的用户组判定 level 的功能	20
4.4 对学生信息的基本操作（对应‘对学生信息实现输入、输出、查找、修改、追加、删除的操作’）	21
4.4.1 查找	22
4.4.2 输入	23
4.4.3 输出	24
4.4.4 追加	25
4.4.5 修改	26
4.4.6 删除	27
4.5 学生信息本地存储（对应‘学生信息的备份和恢复’）	28
4.5.1 存储实时操作记录	29
4.5.2 保存	30
4.5.3 备份	31
4.5.4 恢复	32
4.7 本地存储的加密解密	35
4.7.1 加密	35
4.7.2 解密	35
5. 系统运行结果	36
5.1 运行环境	36
5.2 运行结果	37
5.2.1 登录系统	37
5.2.2 操作系统	42
6. 调试和改进	50
7. 结论	51

参考资料	52
------------	----

1. 引言

学生档案管理系统是一个教育单位不可缺少的部分，它的内容对于学校的决策者和管理者来说都至关重要，所以学生档案管理系统应该能够为用户提供充足的信息和快捷的查询手段。但一直以来人们使用传统人工的方式管理文件档案，这种管理方式存在着许多缺点，如：效率低、保密性差，另外时间一长，将产生大量的文件和数据，这对于查找、更新和维护都带来了不少的困难。

系统所要解决的问题有用户登录、密码管理、查询、添加、删除、更新、保存、备份等功能。

程序主框架编写由洪一芄负责，程序调试由吴诗妍负责，共同完成程序编写、报告撰写。

小节介绍选题的背景、系统所要解决的问题、小组成员分工。

1.1 背景

学生档案管理系统如果开发成功，有以下几个好处：

- (1) 将所有学生信息入档，较于人工管理，检索起来更为迅速、便捷。
- (2) 仅需一个程序，便可省去大面积存放档案的空间，节约成本。
- (3) 当学生档案不断变多时，人工查询、更新、维护都需要大量时间成本，而本系统则可极大程度地避免此类问题。
- (4) 传统人工管理不可避免会因为保存时间增加，带来一定的档案纸张泛黄、老化等，而本系统则可对于所保存的档案始终如一地清晰明了。
- (5) 传统人工管理因存放在一个固定空间，所要存在保密性不佳的问题。而本系统将通过密码管理，仅对密码输入正确的账户开放，并且对存储数据加密，从而达到保密。
- (6) 传统人工管理因档案存放在同一个空间，如需调阅档案则必须进入此空间。因此，仅专门管理人员可进入。而本系统将通过账户等级区分，对不同等级的账户开放不同功能，从而解决调阅档案时的步骤繁琐、耗时耗力。

所以，本系统旨在开发一个查询、增减、更新等功能合为一体的学生档案管理系统，该系统的难度等级：HARD。

1.2 要解决的问题

本系统要提供以下几个功能：

- (1) 登录系统的密码检测和管理功能。第一次进入系统时输入的密码为管理员密码（admin 账户密码）。每次登录前可以选择修改密码或者注

册新账户。同时，新注册、经过修改的账号密码将被加密存储在本地文件中。

- (2) 提供用户简单的操作菜单。登录系统根据账户名判定 level，然后根据 level 提供不同的操作菜单。（测试注册 3 个账户：admin 账户，level 为 3。user 账户，level 为 2。guest 账户，level 为 1。其他新注册的任何账户，level 为 0。）

- (3) 程序的操作功能的实现。包括：

[0]退出该程序

[1]初始化

[2]查找：通过学生学号查找信息，若找到，则显示该学生信息；若未找到，则会询问是否添加该学生

[3]输入：首先输入需要添加几个学生，再依次输入对应学生信息：学号、姓名、性别、生日、电话号

[4]删除：根据输入的学号，删除对应学生的所有信息

[5]修改：输入需要更新信息的学生的对应学号，找到该学生后，依次输入新的信息；若未找到，则会询问是否添加

[6]保存：保存所有信息

[7]输出：打印目前已有所有学生信息

[8]备份：将现有的学生信息备份在一个 txt 文件中，该文件可自行命名，但是需要结尾带.txt

[9]恢复：从指定文件内恢复学生信息

- (4) 学生信息的备份和恢复功能。为了保证数据的安全性，考虑到现实情况可能存在断电停电的风险，进而会使得内存中的数据丢失，所以本系统提供了学生信息的本地存储以及本地备份。当数据出错时，可以有补救措施，例如从备份文件中，恢复学生信息数据。

这部分备份和恢复功能包括：

[0]删除操作的记录

[1]增加的操作记录

[2]修改操作的记录

[3]保存所有的数据

[4]备份所有的数据：将现有的学生信息备份在一个 txt 文件中

[5]恢复已有的数据：从 txt 文件中恢复学生信息至程序

- (5) 对本地存储的信息进行加密解密。本系统中，所有存储到本地的数据采用了异或加密的加密手段，让数据的安全性有了一份保障。

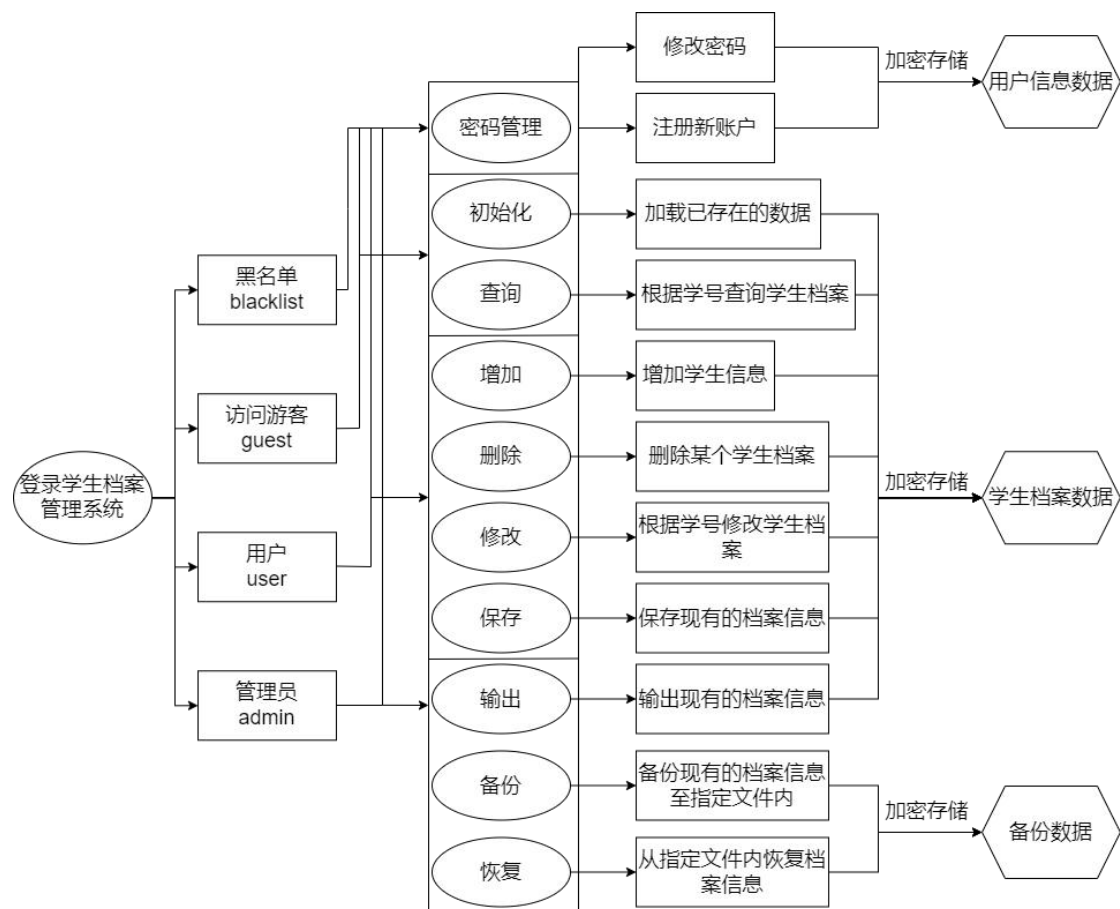
1.3 小组成员分工

描述小组成员组成以及各成员的详细分工情况。

表 1 课程设计任务分工安排

任务分工描述	实际完成人
任务 1：程序主框架编写	Hong
任务 2：程序编写	Hong
任务 3：程序调试	Hong、Wu
任务 4：报告撰写	Hong、Wu

2. 系统框架



2.1 系统框架图

3. 数据结构设计

3.1 学生信息初始数据的获取

本系统的学生信息初始数据由 EXCEL 表格随机生成得到。

201835527779	学雪莲	female	1998.10.19	13433155229
201825267003	生鹏海	male	1998.10.07	13934748249
201884203665	唐闹	male	1999.03.23	18323592446
201822764344	屠兰梦	female	1999.08.18	13264171399
201875520213	浦天和	male	1998.11.27	13957788643
201874473964	典芷	male	1999.06.16	18592547883
201842729250	廖德水	female	1999.08.03	15984859220
201887895587	羿长	female	1999.08.18	18093337655
201816856587	尾昆纶	female	1998.03.17	18322503857
201871450570	买馨蓉	female	1999.07.17	18037445163
201870523052	权向雪	male	1999.03.16	13822053312
201830780640	麻灿	male	1999.12.26	15544657695
201899744418	仇寄柔	male	1998.02.13	13760248823
201867397395	次安宁	female	1999.12.02	17596826479
201812925232	潮曾	female	1998.05.06	15930396681
201887177893	仰幻梅	female	1998.09.07	18866335779
201868122318	祝鹏鲸	male	1998.08.14	14734501343
201842313469	通怀雁	female	1999.05.26	13719577495
201826263551	强孤云	female	1998.09.16	15181872201

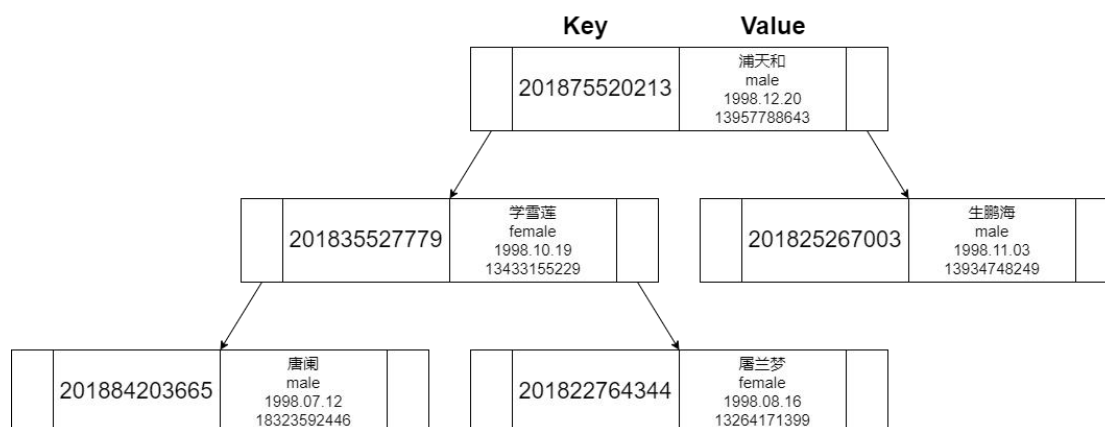
3.1 随机生成的一部分数据

图 3.1 的数据属性，以第一条数据为例，该条学生信息数据的属性如下表所示。

学号	姓名	性别	生日	手机号
201835527779	学雪莲	female	1998.10.19	13433155229

3.2 学生信息数据在内存中的储存

在该系统中，学生信息数据在内存中的储存采用了 stl 库的 map 类。map 类可以保证学生信息的学号（学号作为 key 值）是唯一的，故无法存在学号相同但学生信息不同的情况。



3.2 学生信息数据在内存中的储存图解

map 是一个关联式容器，所谓的关联式容器，也就是类似与关联性数据库。每个元素都有一个键(key)和一个值(value)一一对应。关联式容器一般都是默认进行排序的。排序规则是按照键值的大小。容器的内部结构一般为 RB-tree，或者 hashtable。

映射和多重映射基于某一类型 Key 的键集的存在，提供对 T 类型的数据进行快速和高效的检索。键和值的数据类型一般是不一致的。是一个 pair 类型中的两个分量。

所有类型的 map 容器保存的都是键值对类型的元素。map 容器的元素是 pair<const K, T> 类型的对象，这种对象封装了一个 T 类型的对象和一个与其关联的 K 类型的键。pair 元素中的键是 const，因为修改键会扰乱容器中元素的顺序。

map 算法描述：

```
template<class Key,class T,class Alloc,class Compare = std::less<Key>>
class map{
public:
    typedef Key key_type;    //键值类型
    typedef std::pair<const Key,T>value_type;//元素类型 (键值/实值)

public:
    T& operator[] (const key_type& k){
        return (*((inserter(value_type(k,T()))).first)).second;
    }
};
```


pair 的算法描述:

```
template <class T1,class T2>
struct pair{
    typedef T1 first_type;
    typedef T2 second_type;
    T1 first;    //public
    T2 second;   //public
    pair():first(T1()),second(T2()){};
    pair(const T1& a,const T2& b):first(a),second(b){};
};
```

3.3 学生信息数据在外部文件的储存

学生信息数据在外部的储存是保存在 txt 文件中，且保存下来的数据是经过加密的（具体加密保存的过程会在下一章做阐述）。每一行代表一条数据。每条数据用 # 作为分隔符，代表不同属性的区分。

```
data - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
0039007310:8#询汲必#2#39;8/02/05#36454027067#
003900736138#梳亚呷#3#39;9/05/22#33100969587#
003901257448#耐恨监#3#39;8/07/33#35737965277#
003901741264#加絳#3#39;9/13/31#38586474123#
0039016452;1#撞炼尧#2#39;8/04/05#38761225510#
0039023912:2#物术掌#3#39;9/0:/26#37778439:42#
003902737748#盎脯驼#2#39;8/04/01#36466829033#
00390278:401#昁吮#2#39;8/13/35#38427872461#
003902:37213#北柱#2#39;8/00/23#35;213:7483#
00390300:2:5#矢山驥#2#39;9/0:/07#37074472394#
003903054440#掇盖硅#2#39;9/04/33#3868847077:#
0039030443;8#回沂撼#2#39;9/02/39#34;9117505;#
003903741058#删皆唬闯#2#39;8/11/23#352736;9636#
003903693033#次喝谤皱#3#39;8/12/27#3824656734:#
0039034421;1#凯幽狠#2#39;8/11/24#39:52940;77#
0039042553;0#钵炎炼#2#39;8/01/25#37051656665#
003904560511#阵汲虞#3#39;8/01/01#35773433;5;#
0039053477;8#辑陆阵#3#39;8/06/36#354158;5;5;#
003905454817#萧找擗#2#39;9/07/10#38:56937282#
003905454876#对挺群#2#39;8/04/10#33798937690#
00390669;502#筏韩毗#2#39;9/13/23#39;772:6234#
003906;445:6#滩陇泌#2#39;9/0:/07#38133532:55#
0039070242:4#儒共#2#39;9/07/27#383727:5022#
```

3.3 学生信息数据在外部文件的储存

3.4 登录用户数据在内存中的储存

登录用户数据在内存中的储存是以一维数组的形式。

一维数组算法描述：

```
ElemType arrayname[MAXSIZE];
```

3.5 登录用户数据在外部文件的储存

登录用户数据在外部的储存是保存在 txt 文件中，且保存下来的数据是经过加密的（同样，具体加密保存的过程会在下一章做阐述）。每一行代表一条数据。每条数据用 # 作为分隔符，代表不同属性的区分。



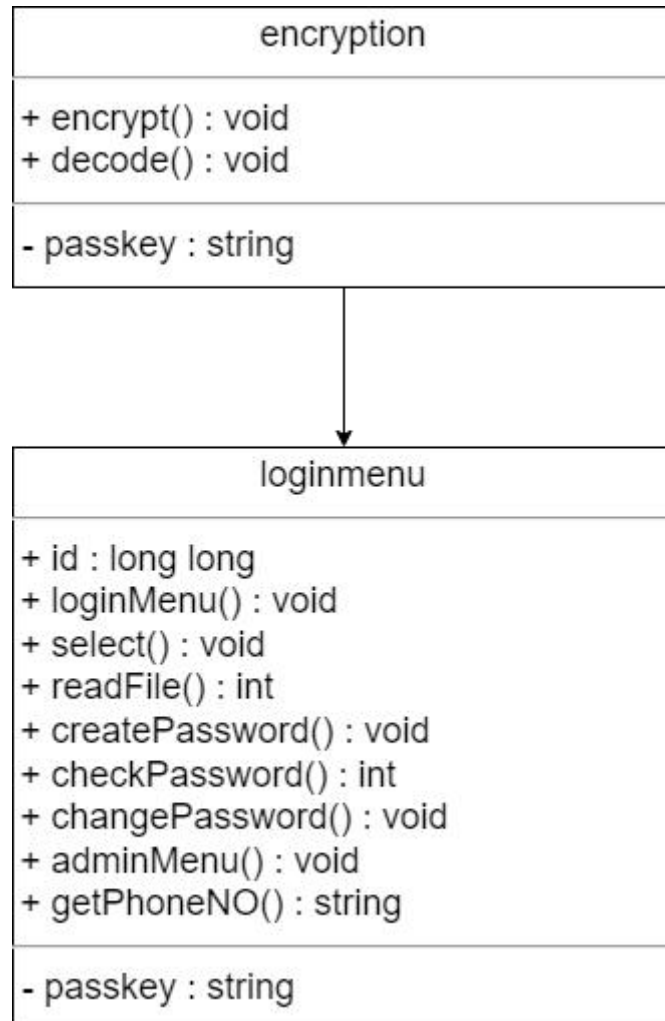
```
*users - 记事本
文件(F) 编辑(E) 格式(O)
dfhl#dfhl
pqds#pqds
bwdrv#bwdrv
```

3.4 登录用户数据在外部文件的储存

4. 关键技术

4.1 用户登录功能（对应‘密码的检测和管理’）

第一次进入系统时输入的密码为管理员密码（admin 账户密码）。每次登录前可以选择修改密码或者注册新账户。同时，新注册、经过修改的账号密码将被加密存储在本地文件 的 users.txt 中。



4.1 登录系统的类图

4.1.1 初始化

进入登录系统时，将进行初始化，目的是为了能够导入已经注册过的账号信息。可以通过如下方式实现：

```
int loginmenu::readFile() { // get password from original database
    string username;
    string password;

    ifss.open("users.txt");
    if (ifss.is_open()) {
        string line;
        int count = 0;
        while (getline(ifss, line)) // while database is not empty
        {
            stringstream ss(line); // ss is an object of class stringstream value is
string line
            bool erase = false; // record that whether this data is operated(if erased,
no need to +=)
            while (getline(ss, line, '#')) // from string ss to get string and store in
"line" , except come up with '#'
            {
                stringstream ll;
                ll << line;
                switch (count % 2) {
                    case 0:
                        //v[1].push_back(line);
                        ll >> username;
                        decode(username, loginKey);
                        break;
                    case 1:
                        //v[2].push_back(line);
                        ll >> password;
                        decode(password, loginKey);
                        break;
                    default:
                        break;
                } // store student's data to different vector (id vector , name
vector ...)

                ll.clear(); // empty stringstream's memory
                ++count;
            }
            users[username] = password;
            ss.clear(); // empty stringstream's memory
        }
    }
}
```

```
        ifss.close();  
        return 1;  
    }  
    else{  
        //write_file();  
        ifss.close();  
        createPassword(); // original user  
        return 0;  
    }  
}
```

4.1.2 注册

注册新用户时，可以通过如下方式实现：

```
void loginmenu::createPassword(){
    string username;
    string password;
    cout << "Please enter username." << endl; cin >> username;

    map<string, string>::iterator it_find;
    it_find = users.find(username);
    if (it_find != users.end()) {
        cout << "[ERROR]The username is already existed." << endl << "Enter any
key back to loginmenu....." << endl;
        cin.get();cin.get();
        loginMenu(); // back to loginmenu
    }
    else{
        cout << "Please enter password." << endl; cin >> password;
        encrypt(username, loginKey); // encryption
        encrypt(password, loginKey); // encryption
        ofss.open("users.txt", ios::app);
        ofss << username << "#" << password;
        ofss << endl;
        ofss.close();
        cout << "Register successful." << endl << "Enter any key back to
loginmenu....." << endl;
        cin.get();cin.get();
        loginMenu(); // back to loginmenu
    }
}
```


4.1.3 密码检测

检查密码是否正确，可以通过如下方式实现：

```
int loginmenu::checkPassword(){
    string username;
    string password;
    cout << "Please enter username." << endl; cin >> username;
    cout << "Please enter password." << endl; cin >> password;

    map<string, string>::iterator it_find;
    it_find = users.find(username);
    if (it_find != users.end() && it_find->second == password) {
        cout << "Login successful." << endl;

        if (username == "guest"){
            guest gue;
            while(1){
                gue.menu(gue.getLevel());
                gue.select(gue.getLevel());
            }
        }
        else if (username == "user"){
            user use;
            while(1){
                use.menu(use.getLevel());
                use.select(use.getLevel());
            }
        }
        else if (username == "admin") {
            admin adm;
            while (true) {
                adm.menu(adm.getLevel());
                adm.select(adm.getLevel());
            }
        }
        else {
            blacklist bla;
            bla.blacklists_menu();
        }
        return 1;
    }
    else {
        cout << "[ERROR]Your password is wrong or the username isn't existed."
        << endl << "Enter any key back to loginmenu....." << endl;
    }
}
```

```
        cin.get();cin.get();  
        loginMenu(); // back to loginmenu  
        return 0;  
    }  
}
```

4.1.4 密码修改

修改密码可以通过如下方式实现：

```
void loginmenu::changePassword(){
    string username;
    string password;
    cout << "Please enter the username." << endl; cin >> username;
    cout << "Please enter the original password." << endl; cin >> password;

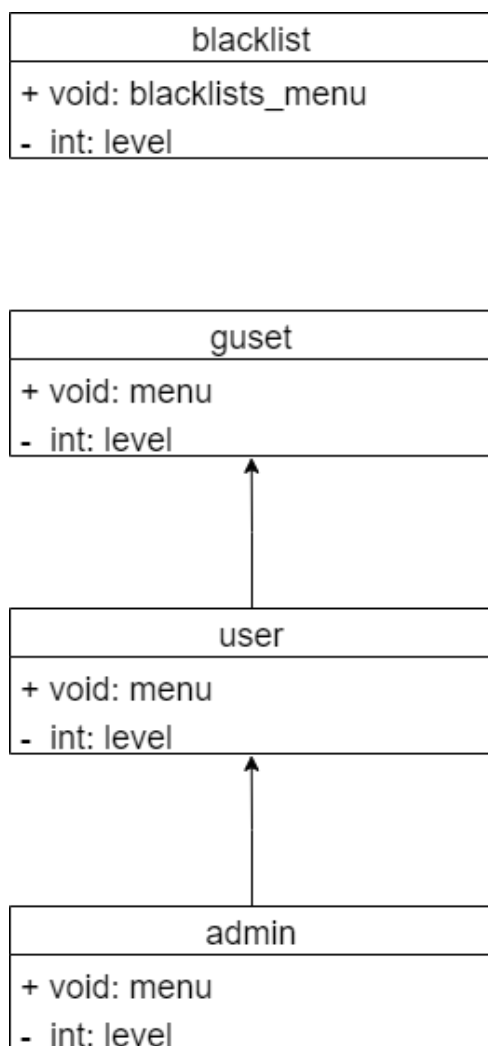
    map<string, string>::iterator it_find;
    it_find = users.find(username);
    if (it_find != users.end() && it_find->second == password) { // firstly check the
username and password
        cout << "Password check successful." << endl;
        cout << "Please enter the new password." << endl; cin >> password;
        users[username] = password;
        encrypt(password, loginKey); // encryption
        encrypt(username, loginKey); // encryption
        ofss.open("users.txt", ios::app);
        ofss << username << "#" << password;
        ofss << endl;
        ofss.close();
        cout << "Password change successful." << endl << "Enter any key back to
loginmenu....." << endl;
        cin.get();cin.get();
        loginMenu(); // back to loginmenu
        return;
    }
    else {
        cout << "[ERROR>Password check failed." << endl << "Enter any key back
to loginmenu....." << endl;
        cin.get();cin.get();
        loginMenu(); // back to loginmenu
        return;
    }
}
```

4.2 学生信息管理系统操作菜单功能（对应‘提供简单的操作菜单’）

登录系统根据账户名判定 level，然后根据 level 提供不同的操作菜单。（目前已注册 3 个账户：admin 账户，level 为 3。user 账户，level 为 2。guest 账户，level 为 1。其他新注册的任何账户，level 为 0。）

某条学生信息的查询是对学校所有教职工（guest）公开的，但对信息的增加、删除、修改以及保存功能，只有校园运维高级用户（user）才能使用。此外，只有校园超级管理员，通常是校长（admin）才可以进行全部数据的输出、备份以及恢复。而识别三种身份要通过身份验证。因此，本系统要提供一个登录界面，通过用户名、密码，验证用户身份。

该系统登陆成功后，通过获取用户的 level，从而给出不同的学生信息管理系统的操作菜单。



4.2 用户组的类图

在该菜单中，level 高的用户组集成 level 低的用户组的菜单，相当于每个用户组可以执行的操作是不同的，该菜单可以通过如下方式实现：

```
void guest::menu(int level) {
    cout<<"\t\t*****"<<endl;
    cout<<"\t\t\t Operation System(Made by Isaac)"<<endl;
    cout<<"\t\t*****"<<endl;
    cout<<"\t\t\t\t [1] Initialize"<<endl;
    cout<<"\t\t\t\t [2] Query"<<endl;
    if (level == 1){

cout<<"\t\t*****"<<endl;
    }
}

void user::menu(int level) {
    guest::menu(level);
    cout<<"\t\t\t\t [3] Insert"<<endl;
    cout<<"\t\t\t\t [4] Delete"<<endl;
    cout<<"\t\t\t\t [5] Update"<<endl;
    cout<<"\t\t\t\t [6] Save"<<endl;
    if (level == 2){

cout<<"\t\t*****"<<endl;
    }
}

void admin::menu(int level) {
    user::menu(level);
    cout<<"\t\t\t\t [7] Display"<<endl;
    cout<<"\t\t\t\t [8] Backup"<<endl;
    cout<<"\t\t\t\t [9] Recover"<<endl;
    if (level == 3){

cout<<"\t\t*****"<<endl;
    }
}
```

4.3 根据登录账号的用户组判定 level 的功能

该系统登陆成功后，通过获取用户的 level，从而给出不同的学生信息管理系统的操作菜单。

本功能可以通过以下方式实现：

```
blacklist::blacklist():level(0){
}
guest::guest():level(1){
}
user::user():level(2){
}
admin::admin():level(3){
}

int guest::getLevel(){
    return level;
}
int user::getLevel(){
    return level;
}
int admin::getLevel(){
    return level;
}
```

4.4 对学生信息的基本操作（对应‘对学生信息实现输入、输出、查找、修改、追加、删除的操作’）

学生的学号取值唯一，当添加学生信息时，若学号已经存在，则提示是否需要修改该学生信息。这个功能是通过自定的 lookup 查找函数实现的。他能从 key 中查询是否存在该学号。

4.4.1 查找

查找学生信息的功能可通过如下方式实现（通过学号查找）：

```
pair<student,bool> school::lookup(long long id){ // lookup a student's data (according
to his id)
    auto it_find = stud.find(id);
    if (it_find != stud.end()){
        cout << "Successful located Student ID: " << std::left << setw(15) << id <<
endl;
        cout << "His/Her data is: " << endl;
        it_find->second.outputData();
        return make_pair(it_find->second,true); // if find this data, student in pair
point to it
    }
    else{
        cout << "[WARNING] Can not find Student ID: " << std::left << setw(15)
<< id << endl;
        cout << "Do you want to add his/her data? (Y/N) (default No)" << endl;
        char choice = 'N'; cin >> choice;
        switch (choice) {
            case 'y':
            case 'Y':
                *this += this->input(id);
                return make_pair(stud.end()->second,false); // if can not find this
data, student in pair point to the last student's data in school
            default:
                cout << "Add operation was cancelled. (id: " << id << ")" <<
endl;
                return make_pair(stud.end()->second,false); // if can not find this
data, student in pair point to the last student's data in school
            }
        }
    }
}
```


4.4.2 输入

对学生信息实现输入功能可通过如下方式实现：

```
student school::input() { // input student's data
    string name, birthDay, phoneNo, sexTemp; int sex; long long id;
    cout << "input student's id:" << endl; cin >> id;
    cout << "input student's name:" << endl; cin >> name;
    cout << "input student's sex: (male or female)" << endl; cin >> sexTemp;
    if (sexTemp == "male") {
        // 1 represent male
        sex = 1;
    }
    else {
        // 0 represent female
        sex = 0;
    }
    cout << "input student's birthday:" << endl; cin >> birthDay;
    cout << "input student's phoneno:" << endl; cin >> phoneNo;
    student student(id,name,sex,birthDay,phoneNo);
    return student;
}
```

4.4.3 输出

对学生信息实现输出功能可通过如下方式实现：

```
int student::outputData() { // get student's data
    //cout << "Student ID: " << id;
    cout << "Name: " << std::left << setw(8) << getName();
    if (getSex() == 1) { // 1 represent male
        cout << std::left << setw(14) << "Sex: male";
    }
    else { // 0 represent female
        cout << std::left << setw(14) << "Sex: female";
    }
    cout << "Birthday: " << std::left << setw(13) << getBirthday();
    cout << "PhoneNO: " << std::left << setw(14) << getPhoneNO() << endl;
    return 1;
}
```

4.4.4 追加

对学生信息实现追加功能可通过如下方式实现（运用重载运算符+）：

```
int school::operator+=(const student& a){ // overload += (insert one student data to
school)
    // return the insertion position and whether the insertion was successful
    pair<map<long long,student>::iterator, bool> ret;
    ret = stud.insert(make_pair(a.id,a));
    if (!ret.second){ // insertion failed
        cout << "[ERROR] This student's data is already existed (id: " << a.id <<
        ")." << endl;
        cout << "Do you want to modify his/her data? (Y/N) (default No)" << endl;
        char choice = 'N'; cin >> choice;
        switch (choice) {
            case 'y':
            case 'Y':
                this->modify(ret.first->first);
                return 0;
            default:
                cout << "Insert student's data failed. Add operation was cancelled.
                (id: " << a.id << ")" << endl;
                return 0;
        }
    }
    else{ // insertion success
        this->saverow(a);
        cout << "Successful insert student's data. (id: " << a.id << ")" << endl;
        return 1;
    }
    //stud[a.id] = a;
}
```

4.4.5 修改

对学生信息实现修改功能可通过如下方式实现：

```
int school::modify(long long id) { // modify a student's data (according to his id)
    // 'this' argument to member function 'getData' has type 'const student', but
    function is not marked const
    if (lookup(id).second){
        cout << "Now modify this data ... ..." << endl;
        stud[id] = input(id);
        this->savemodify(id,stud[id]);
        cout << "Successful modify this data. (id: " << id << ")" << endl;
    }
    return 1;
}
```

4.4.6 删除

删除学生信息的功能可通过如下方式实现（通过学号删除，重载运算符-）：

```
int school::operator-(long long id){ // overload - (delete one student data from school)
    if (stud.erase(id)){ // if success return 1, else return 0
        this->saveerase(id);
        cout << "Successful delete student's data. (id: " << id << ")" << endl;
        return 1;
    }
    else{
        cout << "[ERROR] This student's data is not exist (id: " << id << ")." <<
endl;

        cout << "Do you want to add his/her data? (Y/N) (default No)" << endl;
        char choice = 'N'; cin >> choice;
        switch (choice) {
            case 'y':
            case 'Y':
                *this += this->input(id);
                return 0;
            default:
                cout << "Delete student's data failed. Add operation was cancelled.
(id: " << id << ")" << endl;
                return 0;
            }
        }
    }
}
```

4.5 学生信息本地存储（对应‘学生信息的备份和恢复’）

为了保证数据的安全性，考虑到现实情况可能存在断电停电的风险，进而会使得内存中的数据丢失，所以本系统提供了学生信息的本地存储以及本地备份。当数据出错时，可以有补救措施，例如从备份文件中，恢复学生信息数据。（学生信息是加密存储的，这会在后文进行详细赘述。）

此外，本系统会在数据中实时记录对数据进行的每一次操作，直到超级管理员（admin）保存所有数据时，历史操作记录将被清除。

删除操作的记录以（~）开头，且与下一行的开头数据不相同。

增加的操作记录以数字开头，且只有一行。

修改操作的记录相当于是删除操作+增加操作。所以一条修改操作的记录有二行，以（~）开头，且与下一行的开头数据相同。



```

*data(1) - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
000387751427#患廖魁#3#0022/01/34#37380139781#
00038775:312#炙幻掌#2#0022/07/36#3546800672:#
000388325030#寒兄柏#2#0025/05/31#35:45528:77#
00038800:323#烯兑蛱#2#0025/0:/34#33259275574#
000388056729#诶?蛱#2#0023/06/32#34537859132#
00038806:875#楸料阵#2#0023/11/33#33652516537#
000388694444#硝泽瘤#2#0025/02/23#35763638262#
00038852:307#遮吸躲#3#0025/05/39#33649700306#
00038848:378#岁炎阔#3#0025/0:/09#35787523:64#
000388:95034#妙庚柏闯#3#0023/00/38#38102863430#
000389202625#郇拘牙纓#3#0025/02/38#39:85506660#
0003892805:1#肆姓群#3#0022/11/11#35396662535#
~003902737748
000399294016#样裘垌#3#0023/07/33#33:79837244#
~000382672702
000382672702#塑穹文#3#0022/06/29#386857:0151#
000393:06022#似薇良#2#0023/0:/27#33:241:9621#
000394295824#县叁碎俏#3#0022/0:/21#383928:6:60#
000394496608#划渤梗#2#0025/0:/30#3745813778:#
000395210366#毙嘎#3#0025/02/35#38779617:92#
000395190204#圳找余#2#0022/12/39#35:376:5224#
000395583667#底团敞#2#0022/05/10#37081973:33#
000395412945#廉柑磴#3#0025/0:/29#33488577531#
  
```

删除操作的记录（1行为1条记录）

修改操作的记录（2行为1条记录）

增加操作的记录（1行为1条记录）

4.3 对实时记录对数据的演示

4.5.1 存储实时操作记录

对实时记录对数据进行的每一次操作功能可通过如下方式实现：

1、增加操作的记录

```
int school::saverow(student stud) {
    idTemp = to_string(stud.id); nameTemp = stud.getName(); sexTemp =
to_string(stud.getSex()); birthdayTemp = stud.getBirthday(); phonenoTemp =
stud.getPhoneNO();

    encrypt(idTemp,key); encrypt(nameTemp,key); encrypt(sexTemp,key);
encrypt(birthdayTemp,key); encrypt(phonenoTemp,key); // encryption
    ofst.open("data.txt", ios::app);
    ofst << idTemp << "#" << nameTemp << "#" << sexTemp << "#" <<
birthdayTemp << "#" << phonenoTemp << "#";
    ofst << endl;
    ofst.close();
    return 1;
}
```

2、删除操作的记录

```
int school::saveerase(long long id){
    ofst.open("data.txt", ios::app);
    idTemp = to_string(id);
    encrypt(idTemp,key); // encryption
    ofst << "~" << idTemp << "#";
    ofst << endl;
    ofst.close();
    return 1;
}
```

3、修改操作的记录

```
int school::savemodify(long long id, const student& stud){
    saveerase(id); // when modify, firstly erase original data
    saverow(stud); // then add new data
    return 1;
}
```

4.5.2 保存

保存所有数据的功能可以通过如下方式实现：

```
int school::saveall(school sch) {
    auto stud = sch.getschool();
    if (stud.empty()){
        cout << "[ERROR]There is nothing to save." << endl;
        return 0;
    }
    else{
        ofst.open("data.txt", ios::trunc); // empty original data and replaced with
new data
        ofst.close();
        for (auto & iter : stud) { // save all row
            saverow(iter.second);
        }
        cout << "Save all data successfully." << endl;
        return 1;
    }
}
```


4.5.3 备份

学生信息数据的本地备份功能可以通过如下方式实现：

```
int school::backup(const string& originalDatabase,const string& newDatabase) //
backup original database to new database
{
    ifst.open(originalDatabase,ios::in); // open the source file
    if(!ifst.fail()) // if failed to open the source file
    {
        cout<<"[ERROR][backup]: Fail to open the source file."<<endl;
        ifst.close();
        ofst.close();
        return 0;
    }
    ofst.open(newDatabase,ios::out); // create the new file
    if(!ofst.fail()) // if failed to create the new file
    {
        cout<<"[ERROR][backup]: Fail to create the new file."<<endl;
        ofst.close();
        ifst.close();
        return 0;
    }
    else // start backup
    {
        auto content = ifst.rdbuf(); // get original database's streambuf
        ofst<<content; // backup original streambuf to new database
        ofst.close();
        ifst.close();
        cout << "Backup was done successfully." << endl;
        return 1;
    }
}
```

4.5.4 恢复

学生信息数据的恢复功能可以通过如下方式实现：

```
int school::recover(school& sch, string newDatabase) {
    bool is_original_data = false;
    if (newDatabase != "data.txt") { // when recover from other database
        cout << "[WARNING] This operation will cover the original data. A
backup is recommended. Are you sure to continue recovery? (Y/N) (default No)" <<
endl;

        char choice = 'N'; cin >> choice;
        switch (choice) {
            case 'y':
            case 'Y':
                break;
            default:
                cout << "Recover was cancelled." << endl;
                return 0;
        }
        cout << "Now recover data from " << newDatabase << " database ... ..." <<
endl;

        ofst.open("data.txt", ios::trunc); // if original existed, delete first, then
recover
        ofst.close();
    }
    else { // when recover from original database
        is_original_data = true;
        backup("data.txt", "~data.txt");
        newDatabase = "~data.txt";
    }
    string line;
    long long id; // store student's id
    string name; // store student's name
    int sex; // store student's gender
    string birthDay; // store student's birthday
    string phoneNo; // store student's phone number

    ifst.open(newDatabase);
    if (ifst.is_open()) {
        int count = 0;
        while (getline(ifst, line)) // while database is not empty
        {
            stringstream ss(line); // ss is an object of class stringstream value is
string line
```

```

        bool done = false; // record that whether this data is operated(if erased,
no need to +=)
        while (getline(ss, line, '#')) // from string ss to get string and store in
"line" , except come up with '#'
        {
            stringstream ll;
            if (line[0] == '~'){ // char "~" represents that the data needs to be
deleted
                ll << line.erase(0,1); // erase '~' from line to ensure that id is
long long int
                ll >> idTemp;
                decode(idTemp,key); // decode
                id = strtoll(idTemp.c_str(), nullptr, 10); // convert string to
long long int

                sch - id;
                done = true;
                break;
            }
            ll << line;
            switch (count % 5) {
                case 0:
                    ll >> idTemp;
                    decode(idTemp,key); // decode
                    id = strtoll(idTemp.c_str(), nullptr, 10); // convert string
to long long int

                    break;
                case 1:
                    ll >> name;
                    break;
                case 2:
                    ll >> sexTemp;
                    decode(sexTemp,key); // decode
                    sex = (int)strtoll(sexTemp.c_str(), nullptr, 10); //
convert string to int

                    break;
                case 3:
                    ll >> birthDay;
                    break;
                case 4:
                    ll >> phoneNo;
                    break;
                default:
                    break;
            }
        }
    }
}

```

```
        } // store student's data to different vector (id vector , name
vector ...)

        ll.clear(); // empty stringstream's memory
        ++count;
    }
    if (!done) {
        decode(name, key); decode(birthDay, key); decode(phoneNo, key);
// decode
        sch += sch.input(id, name, sex, birthDay, phoneNo);
    }
    ss.clear(); // empty stringstream's memory
}
ifst.close();

if (is_original_data) { // erase temp(~data.txt)
    backup("~data.txt", "data.txt");
    ofst.open("~data.txt", ios::trunc); // erase temp(~data.txt)
    ofst.close();
    cout << "Successful read data from original database." << endl;
    return 1;
}

cout << "Successful recover data from " << newDatabase << endl;
return 1;
}
else {
    cout << "[ERROR][recover]: Fail to open new database. The database may
not existed in the current folder" << endl;
    ifst.close();
    return 0;
}
}
```

4.7 本地存储的加密解密

无论是将登录系统的账号、密码明文储存在文件中，还是将学生信息管理系统的学生信息明文存储在文件中，都是极不妥当的，在数据文件失窃时，容易轻易被获取到登录系统的账号密码，也可能会导致后果更严重的学生个人信息泄露。所以将内存中的数据加密存储到本地是十分有必要的。

本系统中，所有存储到本地的数据采用了异或加密的加密手段，让数据的安全性有了一份保障。

```
1.  $A \wedge 0 = A$ 
2.  $A \wedge A = 0$ 
3.  $(A \wedge B) \wedge C = A \wedge (B \wedge C)$ 
4.  $(B \wedge A) \wedge A = B \wedge 0 = B$ 
```

4.4 异或加密解密基本思想

4.7.1 加密

账号密码数据加密的 passkey 为 52112202

学生信息数据加密的 passkey 为 202110310252

加密可用如下的方式实现：

```
void encryption::encrypt(string& c, int key[]) {
    int len = c.size();
    for (int i = 0; i < len; i++) {
        c[i] = c[i] ^ key[i % 8];
    }
}
```

4.7.2 解密

解密可用如下的方式实现：

```
void encryption::decode(string& c, int key[]) {
    int len = c.size();
    for (int i = 0; i < len; i++) {
        c[i] = c[i] ^ key[i % 8];
    }
}
```

5. 系统运行结果

5.1 运行环境

硬件：

CPU: Intel(R) Core(TM) i5-8300H CPU @ 2.30GHz 2.30 GHz

RAM: 16.0 GB (8.0 GB 3200 * 2)

软件：

操作系统: Windows 10 x64 家庭中文版 21H2 19044.2006

编码格式: UTF-8

5.2 运行结果

5.2.1 登录系统

5.2.1.1 主界面

打开程序，便会出现登录系统主界面，如下图 1.1 所示。

```
*****
Login System(Made by Isaac)
*****
[1] Login
[2] Change password
[3] Register
[0] Exit
*****
Please enter the operation number.
|
```

5.1 登录主界面

5.2.1.2 登录

在主界面输入 1 后，会跳转至登录页面，输入账号、密码，若登陆成功，则如下图 5.2 所示。

```
*****
Login System(Made by Isaac)
*****
[1] Login
[2] Change password
[3] Register
[0] Exit
*****
Please enter the operation number.
1
Please enter username.
admin
Please enter password.
admin
Login successful.
```

5.2 登录成功

若账号或密码不正确导致登录失败，则如下图 5.3 所示。

```
*****
Login System(Made by Isaac)
*****
[1] Login
[2] Change password
[3] Register
[0] Exit
*****
Please enter the operation number.
1
Please enter username.
admin
Please enter password.
adm
[ERROR]Your password is wrong or the username isn't existed.
Enter any key back to loginmenu.....
|
```

5.3 登陆失败

Level0 登录之后，所显示界面如下图 5.4 所示。

```
*****
Login System(Made by Isaac)
*****
[1] Login
[2] Change password
[3] Register
[0] Exit
*****
Please enter the operation number.
1
Please enter username.
wsy
Please enter password.
202110310251
Login successful.
[WARNING]Your account was blocked. Please contact your supervisor for help.
```

5.4 level0 登录

Level1 登录之后，所显示界面如下图 5.5 所示。

```

*****
Login System(Made by Isaac)
*****
[1] Login
[2] Change password
[3] Register
[0] Exit
*****
Please enter the operation number.
1
Please enter username.
guest
Please enter password.
guest
Login successful.
*****
Operation System(Made by Isaac)
*****
[1] Initialize
[2] Query
*****
Please enter the operation number. Enter '0' to exit this program.

```

5.5 level1 登录

Level2 登录之后，所显示界面如下图 5.6 所示。

```

*****
Login System(Made by Isaac)
*****
[1] Login
[2] Change password
[3] Register
[0] Exit
*****
Please enter the operation number.
1
Please enter username.
user
Please enter password.
user
Login successful.
*****
Operation System(Made by Isaac)
*****
[1] Initialize
[2] Query
[3] Insert
[4] Delete
[5] Update
[6] Save
*****
Please enter the operation number. Enter '0' to exit this program.

```

5.6 level2 登录

Level3 登录之后，所显示界面如下图 5.7 所示。

```

*****
Login System(Made by Isaac)
*****
[1] Login
[2] Change password
[3] Register
[0] Exit
*****
Please enter the operation number.
1
Please enter username.
admin
Please enter password.
admin
Login successful.
*****
Operation System(Made by Isaac)
*****
[1] Initialize
[2] Query
[3] Insert
[4] Delete
[5] Update
[6] Save
[7] Display
[8] Backup
[9] Recover
*****
Please enter the operation number. Enter '0' to exit this program.

```

5.7 level3 登录

5.2.1.3 密码修改

在主界面输入 2 后，会跳转为密码修改，首先输入账号、原始密码，若原始密码正确，则可以输入新的密码，如下图 5.8 所示。

```

Login System(Made by Isaac)
*****
[1] Login
[2] Change password
[3] Register
[0] Exit
*****
Please enter the operation number.
2
Please enter the username.
admin
Please enter the original password.
admin
Password check successful.
Please enter the new password.
123456
Password change successful.
Enter any key back to loginmenu.....

```

5.8 密码修改成功

若输入的原始密码不正确，则如下图 5.9 所示。

```

*****
Login System(Made by Isaac)
*****
[1] Login
[2] Change password
[3] Register
[0] Exit
*****
Please enter the operation number.
2
Please enter the username.
admin
Please enter the original password.
admin
[ERROR>Password check failed.
Enter any key back to loginmenu.....

```

5.9 密码修改失败

5.2.1.4 注册

在主界面输入 3，会跳转至注册，输入新的账号和对应的密码，则可注册成功，如下图 5.10 所示。

```

*****
Login System(Made by Isaac)
*****
[1] Login
[2] Change password
[3] Register
[0] Exit
*****
Please enter the operation number.
3
Please enter username.
wsy
Please enter password.
202110310251
Register successful.
Enter any key back to loginmenu.....

```

5.10 注册

5.2.1.5 退出

在主界面输入 0.则会退出系统，如下图 5.11 所示。

```

*****
Login System(Made by Isaac)
*****
[1] Login
[2] Change password
[3] Register
[0] Exit
*****
Please enter the operation number.
0

```

5.11 退出

5.2.2 操作系统

成功登录后，会跳转至操作界面，如下图 5.12 所示。

```
*****
      Operation System(Made by Isaac)
*****
          [1] Initialize
          [2] Query
          [3] Insert
          [4] Delete
          [5] Update
          [6] Save
          [7] Display
          [8] Backup
          [9] Recover
*****
Please enter the operation number. Enter '0' to exit this program.
```

5.12 操作界面

5.2.2.1 初始化

在操作页面输入 1 后，进行初始化，如下图 5.13 所示。

```
*****
      Operation System(Made by Isaac)
*****
          [1] Initialize
          [2] Query
          [3] Insert
          [4] Delete
          [5] Update
          [6] Save
          [7] Display
          [8] Backup
          [9] Recover
*****
Please enter the operation number. Enter '0' to exit this program.
1

Successful read data from original database.
Initialize successful.
```

5.13 初始化

5.2.2.2 查找

在操作页面输入 1 后，跳转至查找。输入学生学号查找信息，若找到，则显示该学生信息。如下图 5.14 所示。

```

*****
      Operation System(Made by Isaac)
*****
          [1] Initialize
          [2] Query
          [3] Insert
          [4] Delete
          [5] Update
          [6] Save
          [7] Display
          [8] Backup
          [9] Recover
*****
Please enter the operation number. Enter '0' to exit this program.
2
Please enter the student ID that you want to search.
202110310252
Successful located Student ID: 202110310252
His/Her data is:
Name: 洪一芄 Sex: female Birthday: 2003.01.27 PhoneNO: 15821964662
Enter any key back to operation menu.....

```

5.14 查找成功

若未找到，则会询问是否添加该学生。如下图 5.15 所示。

```

*****
      Operation System(Made by Isaac)
*****
          [1] Initialize
          [2] Query
          [3] Insert
          [4] Delete
          [5] Update
          [6] Save
          [7] Display
          [8] Backup
          [9] Recover
*****
Please enter the operation number. Enter '0' to exit this program.
2
Please enter the student ID that you want to search.
2021103
[WARNING] Can not find Student ID: 2021103
Do you want to add his/her data? (Y/N) (default No)
n
Add operation was cancelled. (id: 2021103)
Enter any key back to operation menu.....

```

5.15 查找失败

5.2.2.3 输入

在操作页面输入 3 后，跳转至输入。首先输入需要添加几个学生，再依次输入对应学生信息：学号、姓名、性别、生日、电话号，如下图 5.16 所示。

```

*****
      Operation System(Made by Isaac)
*****
          [1] Initialize
          [2] Query
          [3] Insert
          [4] Delete
          [5] Update
          [6] Save
          [7] Display
          [8] Backup
          [9] Recover
*****
Please enter the operation number. Enter '0' to exit this program.
3
Please enter the number of data to insert.
1
input student's id:
2021103
input student's name:
selina
input student's sex: (male or female)
female
input student's birthday:
10.17
input student's phoneno:
17501622512
Successful insert student's data. (id: 2021103)

```

5.16 输入

5.2.2.4 删除

在操作页面输入 4 后，跳转至删除。根据输入的学号，删除对应学生的所有信息。若找到，则删除对应信息。如下图 5.17 所示。

```

*****
      Operation System(Made by Isaac)
*****
          [1] Initialize
          [2] Query
          [3] Insert
          [4] Delete
          [5] Update
          [6] Save
          [7] Display
          [8] Backup
          [9] Recover
*****
Please enter the operation number. Enter '0' to exit this program.
4
Please enter the student ID that you want to delete.
202110310252
Successful delete student's data. (id: 202110310252)
Enter any key back to operation menu.....

```

5.17 删除成功

若未找到，则会询问是否增加。如下图 5.18 所示。

```

*****
Operation System(Made by Isaac)
*****
[1] Initialize
[2] Query
[3] Insert
[4] Delete
[5] Update
[6] Save
[7] Display
[8] Backup
[9] Recover
*****
Please enter the operation number. Enter '0' to exit this program.
4
Please enter the student ID that you want to delete.
2021101
[ERROR] This student's data is not exist (id: 2021101).
Do you want to add his/her data? (Y/N) (default No)
n
Delete student's data failed. Add operation was cancelled. (id: 2021101)
Enter any key back to operation menu.....

```

5.18 删除失败

5.2.2.5 追加

当找不到输入学号对应的信息时，可追加此学生档案，如下图 5.19 所示。

```

Do you want to add his/her data? (Y/N) (default No)
y
input student's name:
吴诗妍
input student's sex: (male or female)
female
input student's birthday:
11.5
input student's phoneno:
17501622513
Successful insert student's data. (id: 2021101)
Enter any key back to operation menu.....

```

5.19 追加

5.2.2.6 修改

在操作页面输入 5 后，跳转至修改。输入需要修改信息的学生们的对应学号，找到该学生后，依次输入新的信息。如下图 5.20 所示。

```

*****
Operation System(Made by Isaac)
*****
[1] Initialize
[2] Query
[3] Insert
[4] Delete
[5] Update
[6] Save
[7] Display
[8] Backup
[9] Recover
*****
Please enter the operation number. Enter '0' to exit this program.
5
Please enter the student ID that you want to update.
202110310252
Successful located Student ID: 202110310252
His/Her data is:
Name: 洪一芄 Sex: female Birthday: 2003.01.27 PhoneNO: 15821964662
Now modify this data ... ..
input student's name:
洪一芄
input student's sex: (male or female)
male
input student's birthday:
2003.01.27
input student's phoneno:
15821964662
Successful modify this data. (id: 202110310252)
Enter any key back to operation menu.....

```

5.20 修改成功

若未找到，则会询问是否添加。如图 5.21 所示。

```

*****
Operation System(Made by Isaac)
*****
[1] Initialize
[2] Query
[3] Insert
[4] Delete
[5] Update
[6] Save
[7] Display
[8] Backup
[9] Recover
*****
Please enter the operation number. Enter '0' to exit this program.
5
Please enter the student ID that you want to update.
202110310251
[WARNING] Can not find Student ID: 202110310251
Do you want to add his/her data? (Y/N) (default No)
n
Add operation was cancelled. (id: 202110310251)
Enter any key back to operation menu.....

```

5.21 修改失败

5.2.2.7 保存

在操作页面输入 6 后，跳转至保存。保存所有信息。如下图 5.22 所示。

```

*****
Operation System(Made by Isaac)
*****
[1] Initialize
[2] Query
[3] Insert
[4] Delete
[5] Update
[6] Save
[7] Display
[8] Backup
[9] Recover
*****
Please enter the operation number. Enter '0' to exit this program.
6
Save all data successfully.
Enter any key back to operation menu.....

```

5.22 保存

5.2.2.8 输出

在操作页面输入 7 后，跳转至输出。输出所有信息。如下图 5.23 所示。

```

*****
Operation System(Made by Isaac)
*****
[1] Initialize
[2] Query
[3] Insert
[4] Delete
[5] Update
[6] Save
[7] Display
[8] Backup
[9] Recover
*****
Please enter the operation number. Enter '0' to exit this program.
7
Student ID: 2021101      Name: 吴诗妍  Sex: female  Birthday: 11.5      PhoneNO: 17501622513
Student ID: 201810423089  Name: 盈静柏  Sex: female  Birthday: 1999.01.25 PhoneNO: 16645016265
Student ID: 201810424119  Name: 柔悠奕  Sex: male    Birthday: 1998.06.02 PhoneNO: 13311958785
Student ID: 201811145469  Name: 仆高洁  Sex: male    Birthday: 1999.04.13 PhoneNO: 15526954075
Student ID: 201811453245  Name: 居俊    Sex: male    Birthday: 1998.10.11 PhoneNO: 18797445321
Student ID: 201811557290  Name: 詹梅英  Sex: female  Birthday: 1999.07.25 PhoneNO: 18570214712
Student ID: 201812083283  Name: 田若云  Sex: male    Birthday: 1998.09.06 PhoneNO: 17569408840
Student ID: 201812425769  Name: 不涵桃  Sex: female  Birthday: 1999.07.21 PhoneNO: 16677818231
Student ID: 201812498420  Name: 郜函    Sex: female  Birthday: 1999.10.15 PhoneNO: 18636843663
Student ID: 201812925232  Name: 潮曾    Sex: female  Birthday: 1999.03.03 PhoneNO: 15930396681
Student ID: 201813318284  Name: 雀思琪  Sex: female  Birthday: 1998.09.27 PhoneNO: 17265443196
Student ID: 201813346461  Name: 崔浩歌  Sex: female  Birthday: 1998.07.13 PhoneNO: 18499441578
Student ID: 201813356399  Name: 关兴怀  Sex: female  Birthday: 1998.01.19 PhoneNO: 14980144259
Student ID: 201813453079  Name: 司空海凡 Sex: female  Birthday: 1999.12.03 PhoneNO: 15062688434
Student ID: 201813581012  Name: 段干冰珍 Sex: male    Birthday: 1999.11.07 PhoneNO: 18057556148
Student ID: 201813750190  Name: 江雅惠  Sex: female  Birthday: 1999.12.04 PhoneNO: 19843971975
Student ID: 201814147391  Name: 袁又蓝  Sex: female  Birthday: 1999.02.05 PhoneNO: 17240667467
Student ID: 201814672530  Name: 左静逸  Sex: male    Birthday: 1999.02.21 PhoneNO: 15562402959
Student ID: 201815055799  Name: 经兰泽  Sex: male    Birthday: 1999.05.16 PhoneNO: 15604884958

```

5.23 输出

5.2.2.9 备份

在操作页面输入 8 后，跳转至备份。将现有的学生信息备份在一个 txt 文件中，该文件可自行命名，但是需要结尾带.txt。如下图 5.24 所示。

```

*****
      Operation System(Made by Isaac)
*****
      [1] Initialize
      [2] Query
      [3] Insert
      [4] Delete
      [5] Update
      [6] Save
      [7] Display
      [8] Backup
      [9] Recover
*****
Please enter the operation number. Enter '0' to exit this program.
8
Please enter the new file name that you want to backup to (Must end with .txt).
backup.txt
Backup was done successfully.
Enter any key back to operation menu.....

```

5.24 备份

5.2.2.10 恢复

在操作页面输入 9 后，跳转至恢复。将输入的文件信息恢复。如下图 5.25 所示。

```

*****
      Operation System(Made by Isaac)
*****
      [1] Initialize
      [2] Query
      [3] Insert
      [4] Delete
      [5] Update
      [6] Save
      [7] Display
      [8] Backup
      [9] Recover
*****
Please enter the operation number. Enter '0' to exit this program.
9
Please enter the original file name that you want to recover from (Must end with .txt).
backup.txt
[WARNING] This operation will cover the original data. A backup is recommended. Are you sure to continue recovery? (Y/N) (default No)
y
Now recover data from backup.txt database ... ..

```

```

Successful recover data from backup.txt
Enter any key back to operation menu.....

```

5.26 恢复成功

若未成功，则如下图 5.26 所示。

```

Please enter the operation number. Enter '0' to exit this program.
9
Please enter the original file name that you want to recover from (Must end with .txt).
back
[WARNING] This operation will cover the original data. A backup is recommended. Are you sure to continue recovery? (Y/N) (default No)
y
Now recover data from back database ... ..
[ERROR][recover]: Fail to open new database. The database may not existed in the current folder
Enter any key back to operation menu.....

```

5.26 恢复失败

5.2.2.11 退出

在操作页面输入 0 后，退出系统。

```
*****
      Operation System(Made by Isaac)
*****
      [1] Initialize
      [2] Query
      [3] Insert
      [4] Delete
      [5] Update
      [6] Save
      [7] Display
      [8] Backup
      [9] Recover
*****
Please enter the operation number. Enter '0' to exit this program.
0
```

5.26 退出

6. 调试和改进

关于数据的本地存储，一开始没有考虑到数据加密的需求，后来在知乎上看见讨论学习通信息泄露的讨论帖子，我觉得信息安全是个很重要的话题，我也应该在这个程序中实现一个简单的信息加密功能。其次，密码学是一个很重要的学科，它关乎着我们每个网民用户的信息安全。后来，我成功加入了数据加密存储的功能。虽然在这个程序中，用的是简单的异或加密。但是，如果再套一个 MD5 加密模板，完全可以实现更复杂的加密手段。这样类似 MD5 加密的加密方式的对信息安全保证的效果会更好，破译难度也会很大，能更高效、有效的保护用户以及其数据的安全性。

但是，测试的时候发现数据加密解密类设计的不够好，比如加密解密类的头文件里存在学生类，而学生类的头文件里又存在加密类，导致在程序运行的时候卡死，因为编译到加密解密类或者学生类时，它们会相互编译，最后使得程序崩溃。解决方案是在加密解密类的头文件里去掉学生类，原先存在的目的是为了能直接传入学生的信息地址，从而直接加密解密一整条数据。经过改动以后，加密解密函数只是传入字符串，对字符串进行加密解密，最终该问题得到了解决。

7. 结论

针对课程设计要求的各个要素描述本系统的贡献、优缺点、进一步改进的方向；描述整个课程设计过程的经验、体会；描述如何通过本次课程设计达到计算机技术创新的初步能力。

本系统实现了对学生信息的存储和管理。学生信息包括学号、姓名、性别和生日等信息。系统可以对学生信息进行查询、添加、删除和备份。系统进入时还包括一个密码验证的过程，也可以对密码进行管理。

但是本系统与预期还存在一定的缺陷。

比如预期时账户有如下功能：level3 的账户可以使用全部功能；level1、2 账户使用部分功能；level0 账户不能使用任何功能。Level3 的账户可以修改 level0-2 的账户密码、调整 level0-2 的账户的 level 等级、删除 level0-2 的账户；level2 的账户可以修改 level0-1 的账户密码、调整 level0-1 的账户的 level 等级、删除 level0-1 的账户。

但是现在只有如下功能：可以修改自己账户的密码，账号名不能修改；没有设置 level，是根据用户名来提供不同菜单的，就是只有一个 admin 账户可以使用程序全部功能，guest 和 user 账户可以使用部分功能，其余账户为黑名单账户，无法使用任何功能。

通过本次课程设计，我了解到了信息安全的重要性，数据存储的安全性（如果断电，遭到删库等情况时，原始数据的备份）。学会了如何使用 C++操作文件，以及对 map 有了一个较为深刻的理解。

在学生档案管理系统中，计算机技术手段发挥了高效、精准、快速、集成化应用优势，从而能够让这个系统为校园的师生们所用。这个时代是信息化时代，计算机和网络应用技术在社会经济的发展中起着重要的推动作用，它渗透在我们日常生活和社会生活生产技术当中，发挥了综合服务职能。计算机技术，已经与我们的生产、日常生活息息相关。高素质的计算机技术应用能力人才的培养，是不断促进工作效率提高、促进经济发展、提升提供服务水平的重要手段和重要条件。信息化时代更看重工作质量与效率双重提高，其关键在于计算机应用技术的创新，我们仍需砥砺前行。

参考资料

- [1] Pluto_cannon, 加密算法[异或], https://blog.csdn.net/Pluto_cannon/article/details/112431096
- [2] SmileLing~, C++中的 map, https://blog.csdn.net/forever__1234/article/details/89647975
- [3] photon_wa, C++ rdbuf(), <https://blog.csdn.net/photon222/article/details/98644521>
- [4] ZhInen 丶, C++文件操作, https://blog.csdn.net/qq_40279192/article/details/124493566
- [5] Captain_zw, fstream 详解, https://blog.csdn.net/qq_34018840/article/details/106409947
- [6] Exile, 密码学（摘自 crimaster）, <https://zhuanlan.zhihu.com/p/356838671>
- [7] liverycy , C++---STL 标准库之 map 函数全解析 , https://blog.csdn.net/weixin_46369610/article/details/120349280
- [8] 小院 , c 语言从文件删除指定行 , https://blog.csdn.net/weixin_36382265/article/details/117271147