暨南大學
JINAN UNIVERSITY

本科课程设计

课 程 名 称：C++程序设计

课 程 编 号：08060226

学 生 姓 名：葉小選（組長）、徐佳乐

学　　　　号：2021152956(组长) 、2021150791

学　　　　院：信息科学与技术学院

系：计算机

专　　　　业：计算机科学与技术

指 导 教 师：张晓刚

教 师 单 位：

开 课 时 间：2022～2023　学年度第 2　学期

暨南大学教务处

2023　年 4 月 6 日

目录

一、题目及需求分析

题目：高校水电费管理系统

1．选做此课题的目的与意义

学习如何用 C++程序来编写一个管理系统，加深对于 C++语言的了解和实际运用。

2．系统的功能需求分析

问题描述：

住宿学生信息包括：学号、姓名、性别、年龄、班级、用电量、用水量等信息。教工信息包括职工号、姓名、性别、年龄、工作部门、用电量、用水量等信息。能计算出学生和教工每月所要交的电费和水费。定义一个人员类，实现学生和教工共同的信息和行为。

功能要求：

（1）添加功能：程序能够添加不同学生和教工的记录，提供选择界面供用户选择所要添加的类别，要求编号号要唯一，如果添加了重复编号的记录时，则提示数据添加重复并取消添加。

（2）查询功能：可根据姓名、用水量、用电量信息对已添加的学生或教工记录进行查询，如果未找到，给出相应的提示信息，如果找到，则显示相应的记录信息。

（3）显示功能：可显示当前系统中所有学生和教工的记录，每条记录占据一行。

（4）编辑功能：可根据查询结果对相应的记录进行修改，修改时注意编号的唯一性。

（5）删除功能：主要实现对已添加的学生或教工记录进行删除。如果当前系统中没有相应的记录，则提示“记录为空！”并返回操作。

（6）统计功能：能根据多种参数进行统计。能统计学生和教工的用水用电量、所要交纳的电费和水费、未交纳水电费的人员信息等。

（7）保存功能：可将当前系统中各类记录存入文件中，存入方式任意。

（8）读取功能：可将保存在文件中的信息读入到当前系统中，供用户进行使用。

（9）计算电费和水费。学生每月都有一定额度的水电是免费使用的，超过的部分需要交费。

二、系统设计

组员分工：构思、代码实现由组长（叶小选）完成，绘图、PPT 制作由组员（徐佳乐）完成。

1. 根据所选题目，通过用例图描述系统总体功能。

设计思路：

程序在打开后的第一个画面应该是主页面，主页面需要有两个选项，一个是普通用

户登录（学生、教工），另一个是管理员登录，在普通用户登录和学生用户登录都设立一个登陆界面。

在登陆界面，需要有两个输入，分别是账号输入和密码输入，并且包含初次登陆的选项，即注册账户。

在注册账户页面，有两个选项，一个是教师用户注册，一个是学生用户注册，在选择对应身份后，会先进行账号注册，要求输入一次账号和连续输出两次密码，两次密码会进行校验，如果不一样则返回错误重新输入。账号注册完成再进入注册信息补充，分别需要输入他们的用户信息，包括姓名、年龄、工（学）号、单位（班级）、性别这些基础信息，他们的用水量和用电量会初始化为 0。在填写工（学）号的时候会进行校验，工（学）号将会作为关键字，不允许重复，注册时输入的工（学）号，会与后台存储的工（学）号进行对比，当发现一样的返回错误。

管理员账户不提供注册，默认为创作者本人拥有，仅能透过给与管理员密钥提升普通用户身份组进管理员才能新增管理员数量。

当登陆完成——以管理员登录的情况。会出现一个界面，包括查询未缴费名单、学生查询、教工查询和全部查询、修改用户信息、新增管理员用户、删除管理员用户。学生查询会按班查询（输入班号），然后输出整个班级的信息，教工查询会按部查询（输入部门），然后输出整个部门教工信息。信息包括用户的关键字、部门(班级），用水量、用电量、总共需要缴付的水电费。新增管理员用户则是输入关键字，把该名用户提升到管理员。删除管理员用户则是输入关键字，把该名用户从管理员身份里删除。管理员身份为一个身份标识符，每名管理员都拥有一个身份标识符，当该标识符不匹配，则不属于管理员。修改用户信息则是有两个选项，一个是显示学生的，一个是显示教工的，选择对应的类别后，会显示当前全部的学生或教工的信息，之后再输入对应的关键字，以对指定用户进行信息修改，包括对用水量，用电量和删除该用户的选项。

当登陆完成——以普通用户登录的情况。学生和教工的页面都是差不多的，但仍会分出两个页面。普通用户页面显示当前用户的个人信息，包括他的名字，学号，所属班级，年龄，用电量，用水量以及他当前应该缴付的水电费。拥有修改选项，用户修改用户的密码，需要用户输入当前密码，输入的当前密码会与改名用户的密码（未修改前）进行对比，当密码相匹配时，会进行密码修改操作，新增临时数组$_1$存放用户第一次输入的密码，当输入第二次密码时候再新增临时数组$_2$，输入完成后会对两个临时数组进行比较，当两个临时数组完全一样时，临时数组一的内容覆盖原本的旧密码，并且释放两个临时数组。拥有缴费选项，用于给用户进行缴费，当缴费完成后，会对该名用户的用水量、用电量以及当前应该缴付的水电费进行清 0 操作。

当前需要缴费的金额：(用水量-免费用水额度)*0.5+(用电量-免费用电额度)*0.7，当>0 时，需要缴费，当<=0 时，无需缴费。

新增余额系统，余额系统会存储用户的金额进行自动扣费服务，设立 cnt_ddl 计数为缴费的日期，设立 pay_ddl，当 cnt_ddl 到达一个阈值会进行“缴费操作”，缴费操作流程先计算当前需要缴费的金额$p，然后$p 与余额进行比较，当$p<余额，则直接进行自动缴费，并且留下记录，输出缴费成功信息。当$p>余额，则提示用户需要对余额进行储值，当到达 pay_ddl 阈值，则会进行$p 和余额的比较，若仍处于$p>余额，则输出请缴费，若$p<余额，则输出缴费成功。

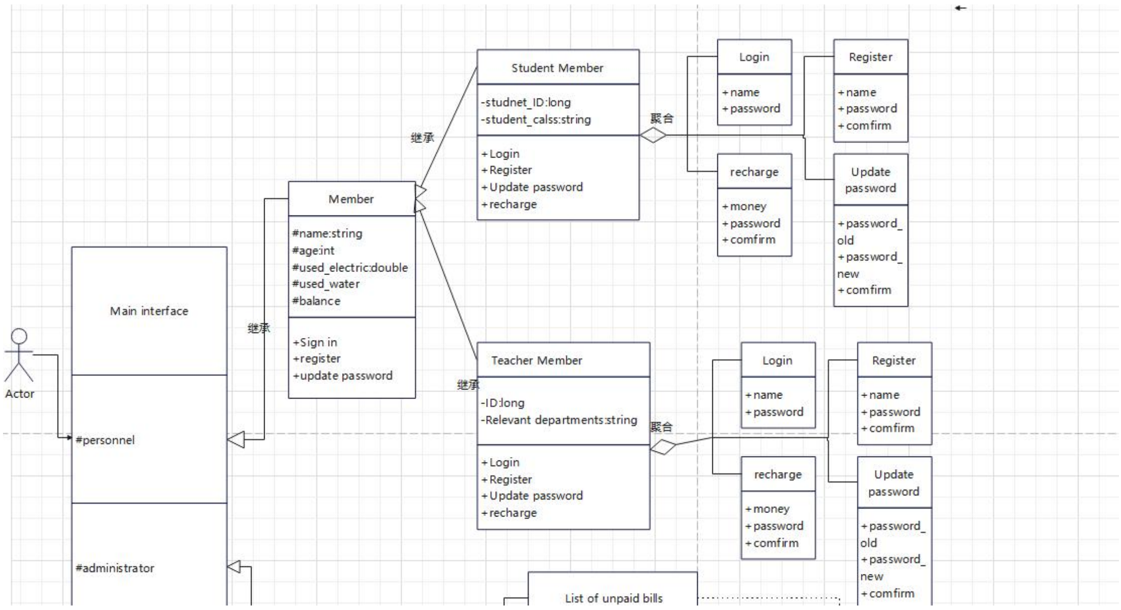
新增情况，在注册里面完成。编辑功能在用户个人页面里完成，统计功能，对全部

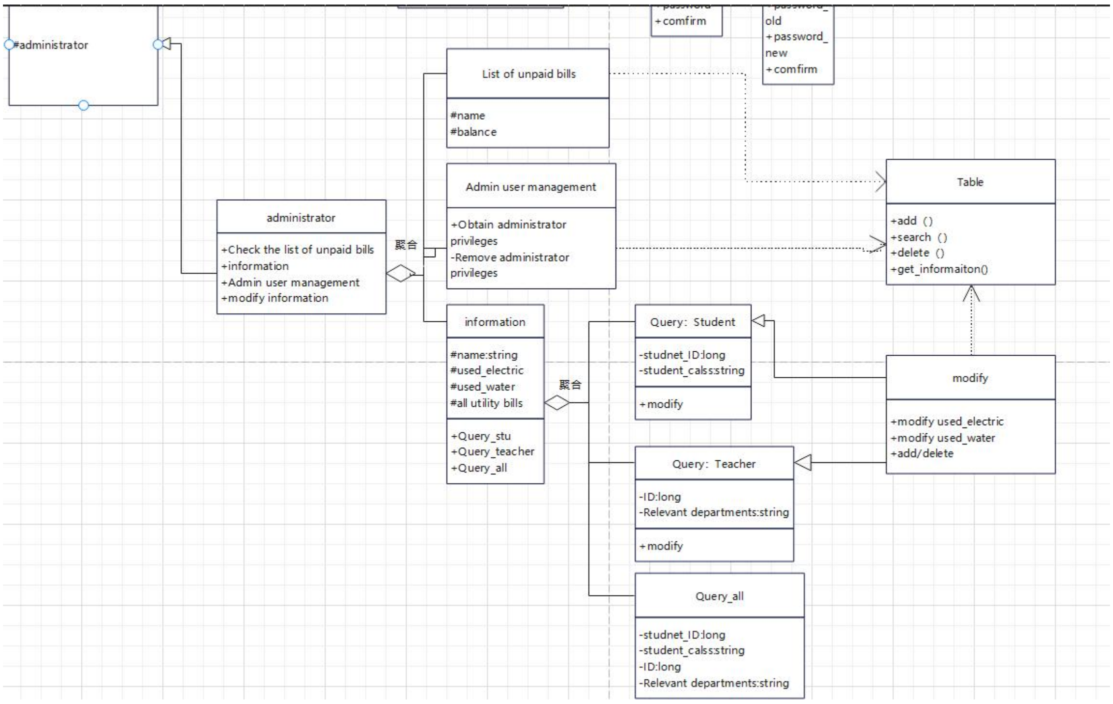
用户的统计仅管理员内部完成，而对用户个人的统计则又用户部分独立完成。

保存方面-把当前的用户信息以 txt 文件保存，实现对文件的输入。

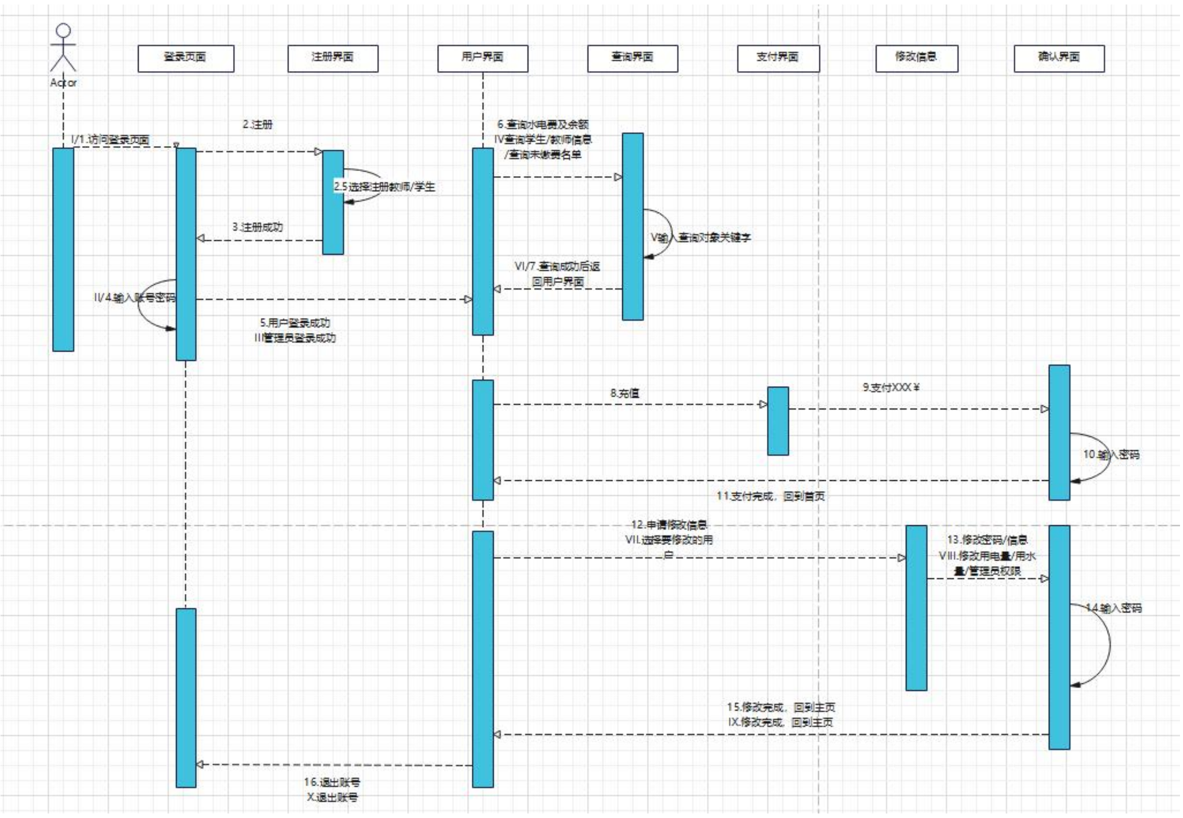
读取方面，对 txt 文件进行读取，并且显示在页面上。

2. 使用类图描述系统总体设计。





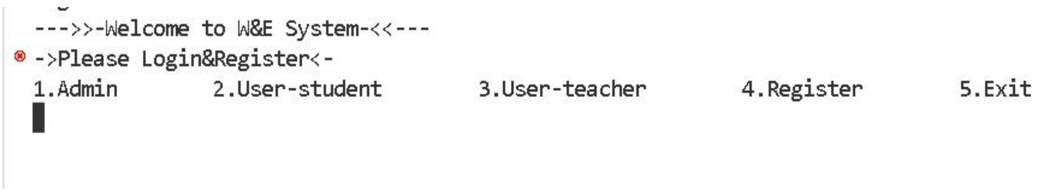
3. 使用时序图描述系统的主要业务操作。



三、系统运行及调试

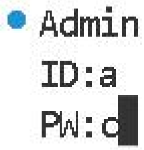
1. 系统运行及调试

主菜单界面：

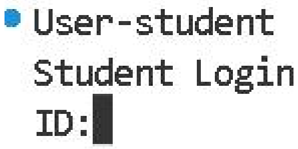


```
---->>-Welcome to W&E System-<<---
->Please Login&Register<-
1.Admin        2.User-student        3.User-teacher        4.Register        5.Exit
```

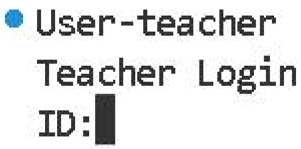
登陆界面：



```
Admin
ID:a
PW:c
```

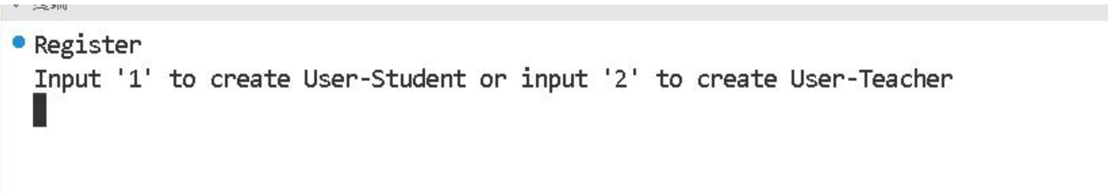


```
User-student
Student Login
ID:
```

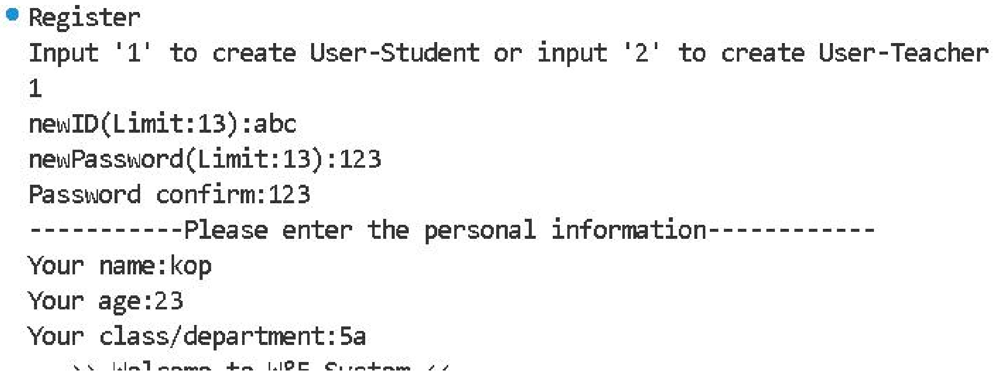


```
User-teacher
Teacher Login
ID:
```

注册界面：

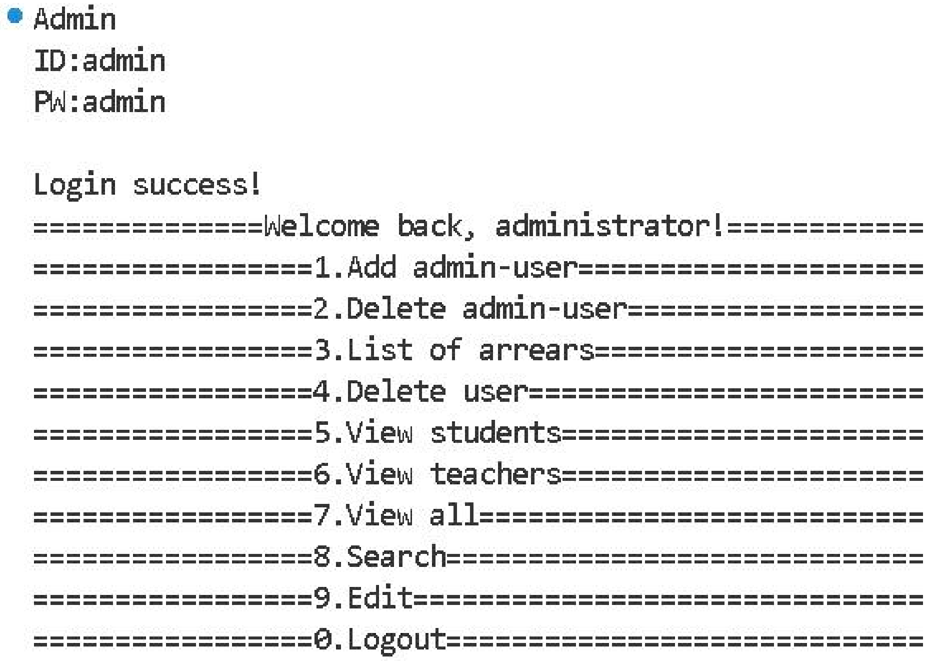


```
Register
Input '1' to create User-Student or input '2' to create User-Teacher
```



```
Register
Input '1' to create User-Student or input '2' to create User-Teacher
1
newID(Limit:13):abc
newPassword(Limit:13):123
Password confirm:123
-----------Please enter the personal information------------
Your name:kop
Your age:23
Your class/department:5a
   >> Welcome to W&F System <<
```

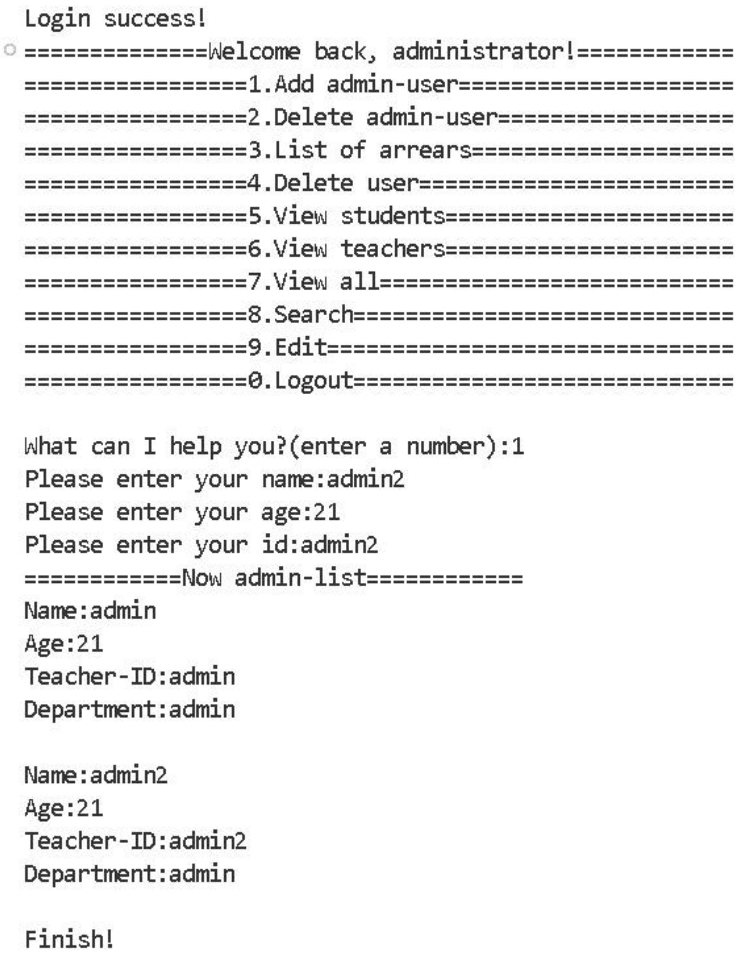
管理员界面：



```
Admin
ID:admin
PW:admin

Login success!
==============Welcome back, administrator!===========
=================1.Add admin-user====================
=================2.Delete admin-user=================
=================3.List of arrears===================
=================4.Delete user=======================
=================5.View students=====================
=================6.View teachers=====================
=================7.View all==========================
=================8.Search============================
=================9.Edit==============================
=================0.Logout============================
```

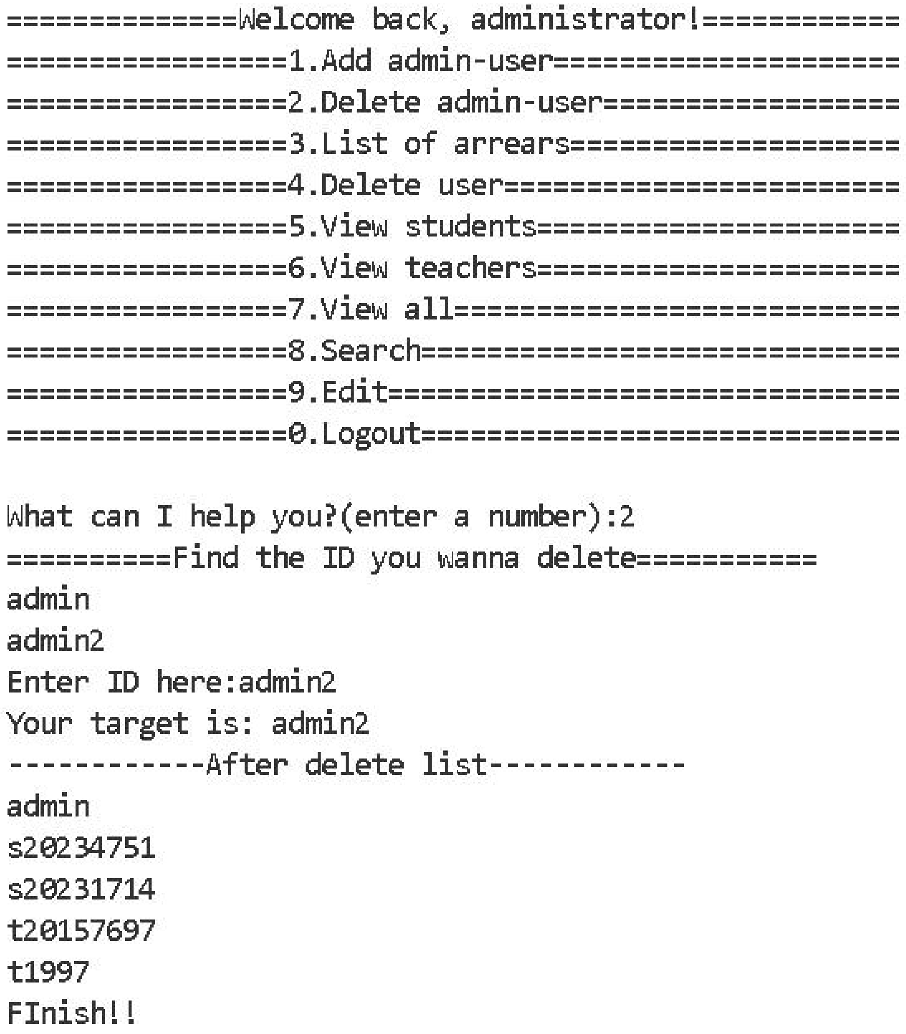
管理员功能实现：

1.



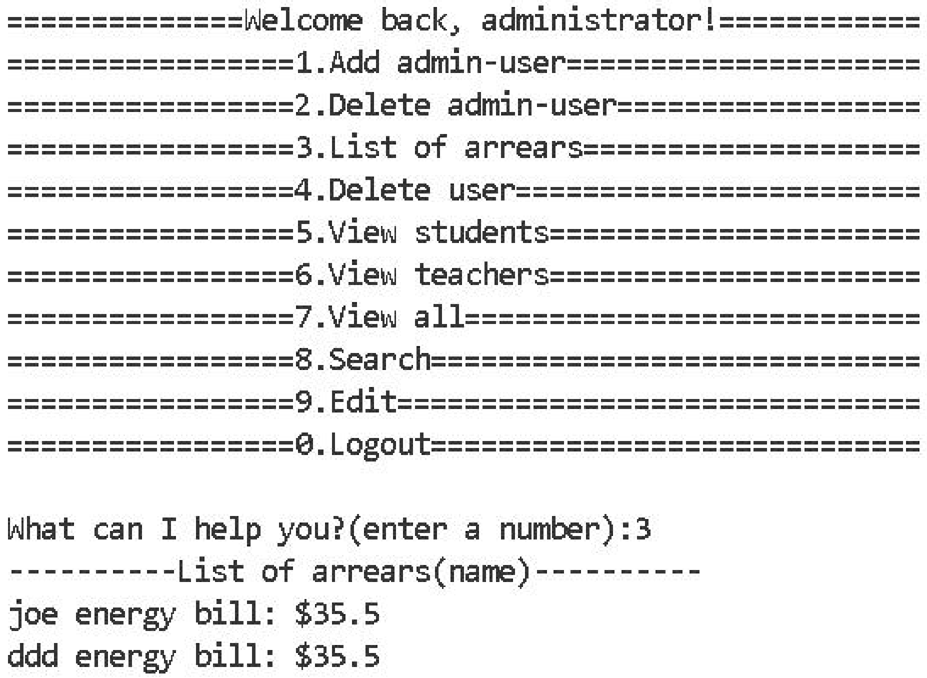
```
Login success!
==============Welcome back, administrator!===========
=================1.Add admin-user====================
=================2.Delete admin-user=================
=================3.List of arrears===================
=================4.Delete user=======================
=================5.View students=====================
=================6.View teachers=====================
=================7.View all==========================
=================8.Search============================
=================9.Edit==============================
=================0.Logout============================

What can I help you?(enter a number):1
Please enter your name:admin2
Please enter your age:21
Please enter your id:admin2
============Now admin-list===========
Name:admin
Age:21
Teacher-ID:admin
Department:admin

Name:admin2
Age:21
Teacher-ID:admin2
Department:admin

Finish!
```

2.



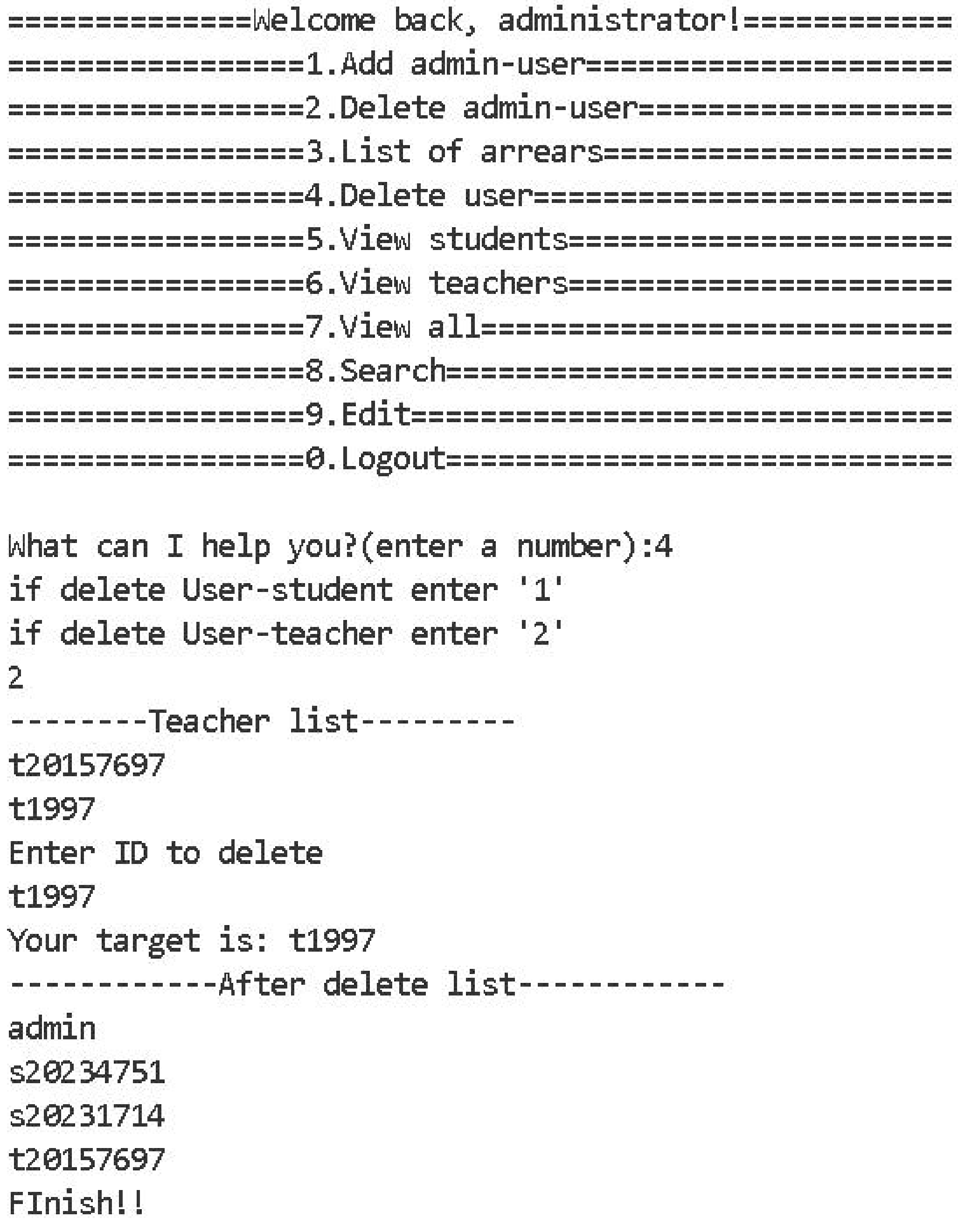
```
==============Welcome back, administrator!============
=================1.Add admin-user=====================
=================2.Delete admin-user==================
=================3.List of arrears====================
=================4.Delete user========================
=================5.View students======================
=================6.View teachers======================
=================7.View all===========================
=================8.Search=============================
=================9.Edit===============================
=================0.Logout=============================

What can I help you?(enter a number):2
==========Find the ID you wanna delete===========
admin
admin2
Enter ID here:admin2
Your target is: admin2
------------After delete list------------
admin
s20234751
s20231714
t20157697
t1997
FInish!!
```

3.



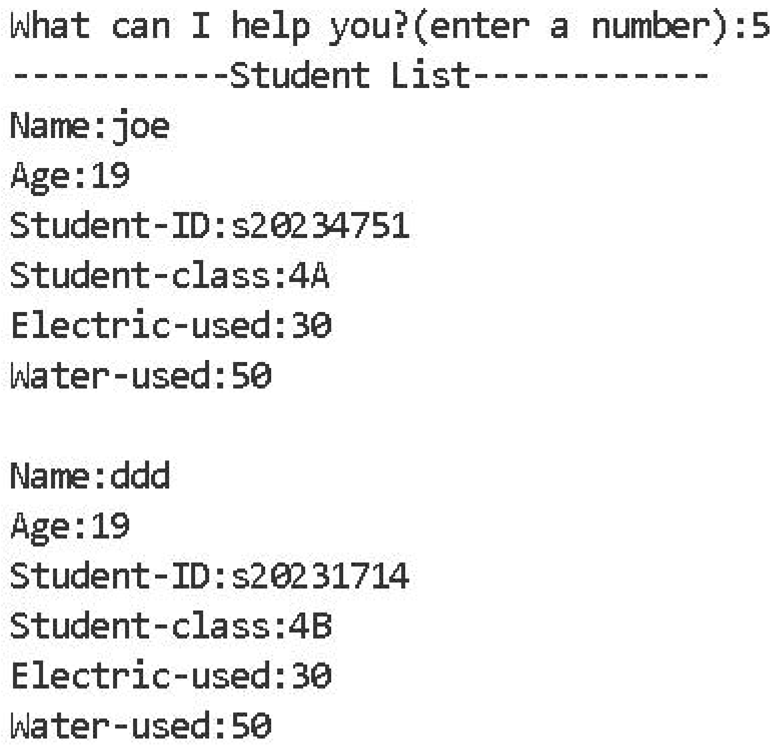
```
==============Welcome back, administrator!============
=================1.Add admin-user=====================
=================2.Delete admin-user==================
=================3.List of arrears====================
=================4.Delete user========================
=================5.View students======================
=================6.View teachers======================
=================7.View all===========================
=================8.Search=============================
=================9.Edit===============================
=================0.Logout=============================

What can I help you?(enter a number):3
----------List of arrears(name)----------
joe energy bill: $35.5
ddd energy bill: $35.5
```

4.



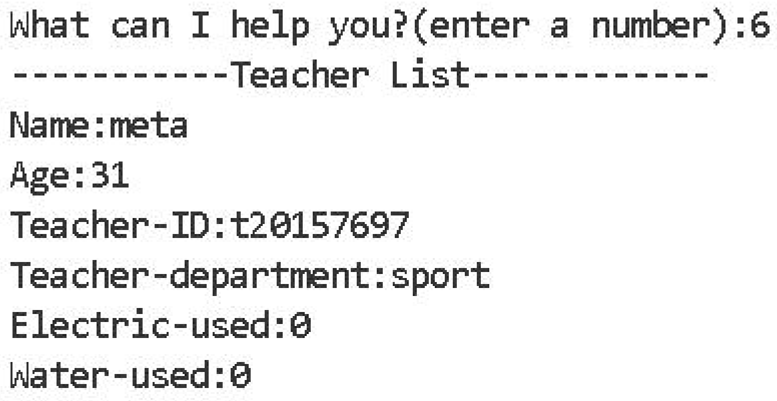
```
==============Welcome back, administrator!============
=================1.Add admin-user=====================
=================2.Delete admin-user==================
=================3.List of arrears====================
=================4.Delete user========================
=================5.View students======================
=================6.View teachers======================
=================7.View all===========================
=================8.Search=============================
=================9.Edit===============================
=================0.Logout=============================

What can I help you?(enter a number):4
if delete User-student enter '1'
if delete User-teacher enter '2'
2
--------Teacher list---------
t20157697
t1997
Enter ID to delete
t1997
Your target is: t1997
------------After delete list------------
admin
s20234751
s20231714
t20157697
FInish!!
```

5.



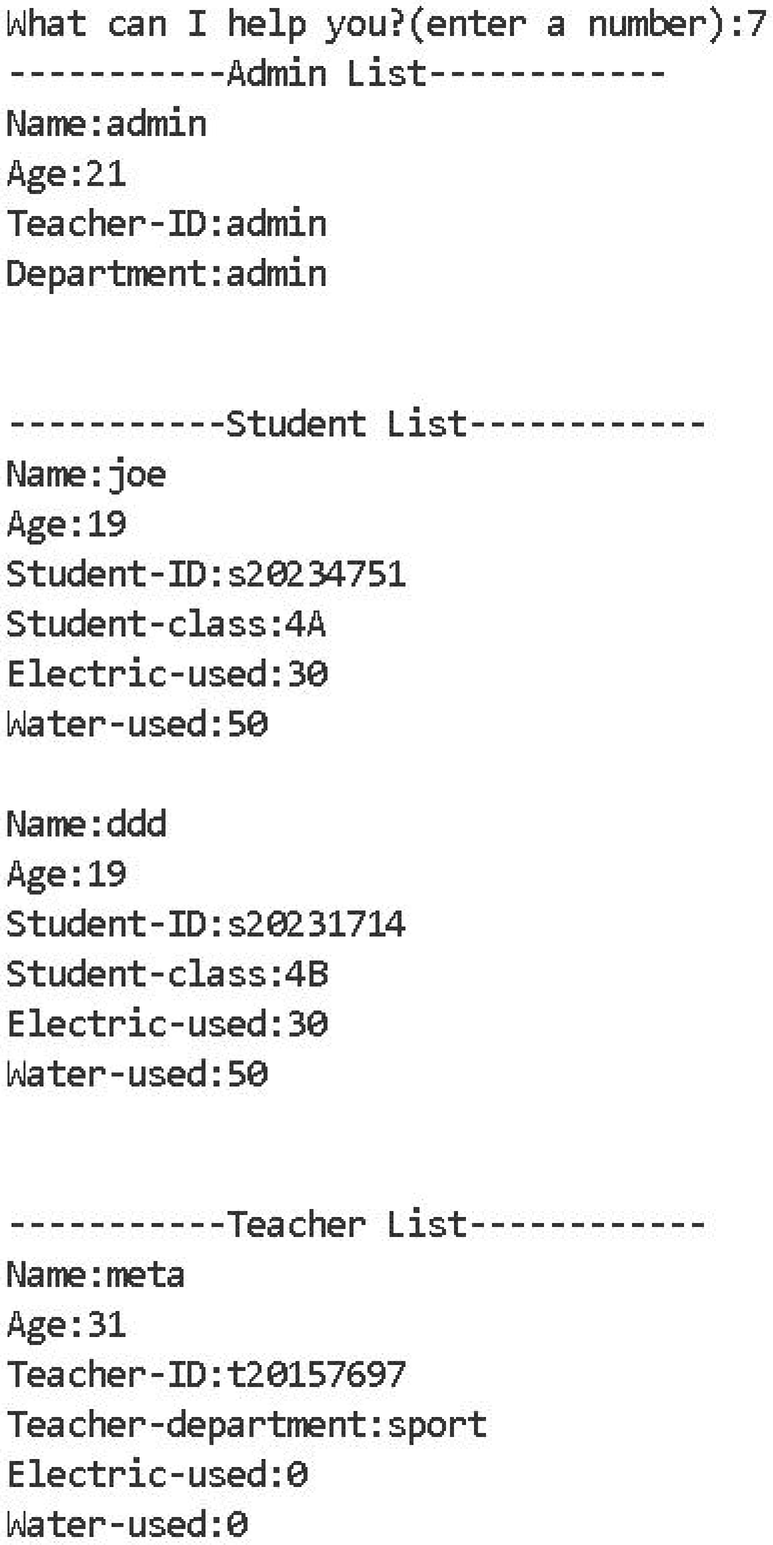
```
What can I help you?(enter a number):5
-----------Student List------------
Name:joe
Age:19
Student-ID:s20234751
Student-class:4A
Electric-used:30
Water-used:50

Name:ddd
Age:19
Student-ID:s20231714
Student-class:4B
Electric-used:30
Water-used:50
```

6.



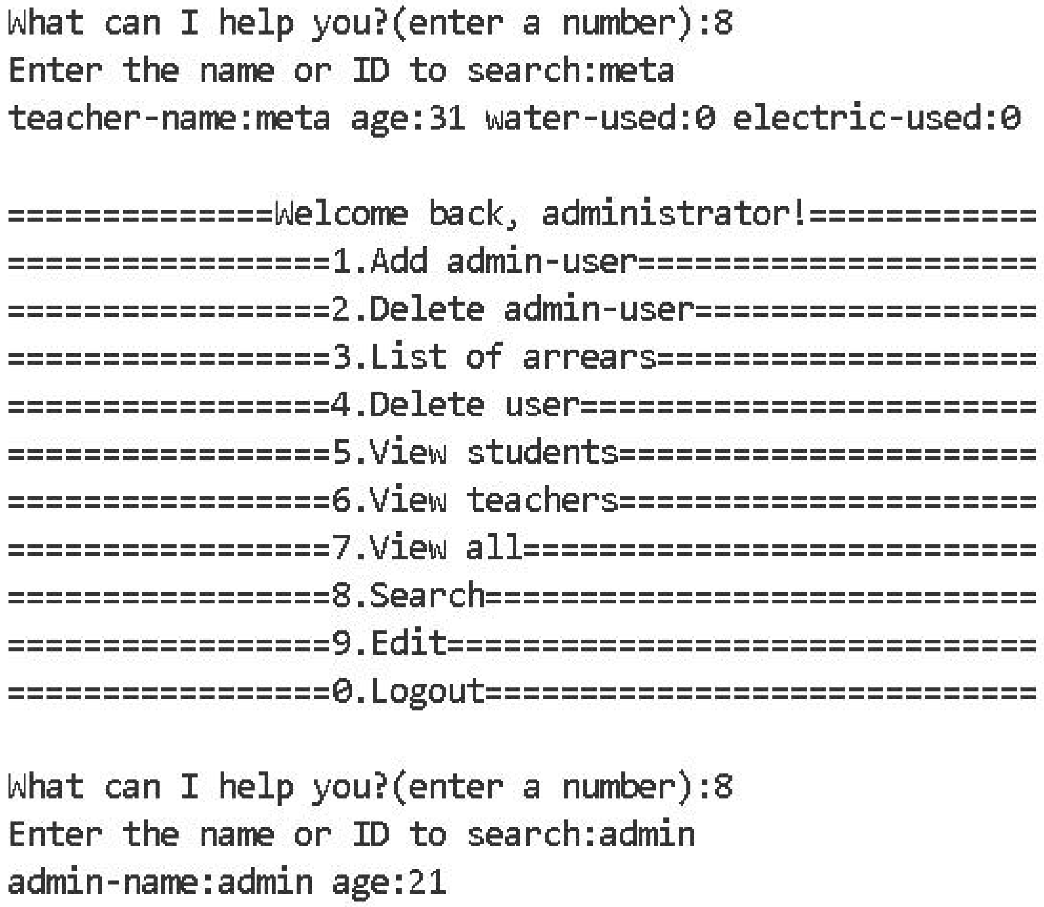
```
What can I help you?(enter a number):6
-----------Teacher List------------
Name:meta
Age:31
Teacher-ID:t20157697
Teacher-department:sport
Electric-used:0
Water-used:0
```

7.



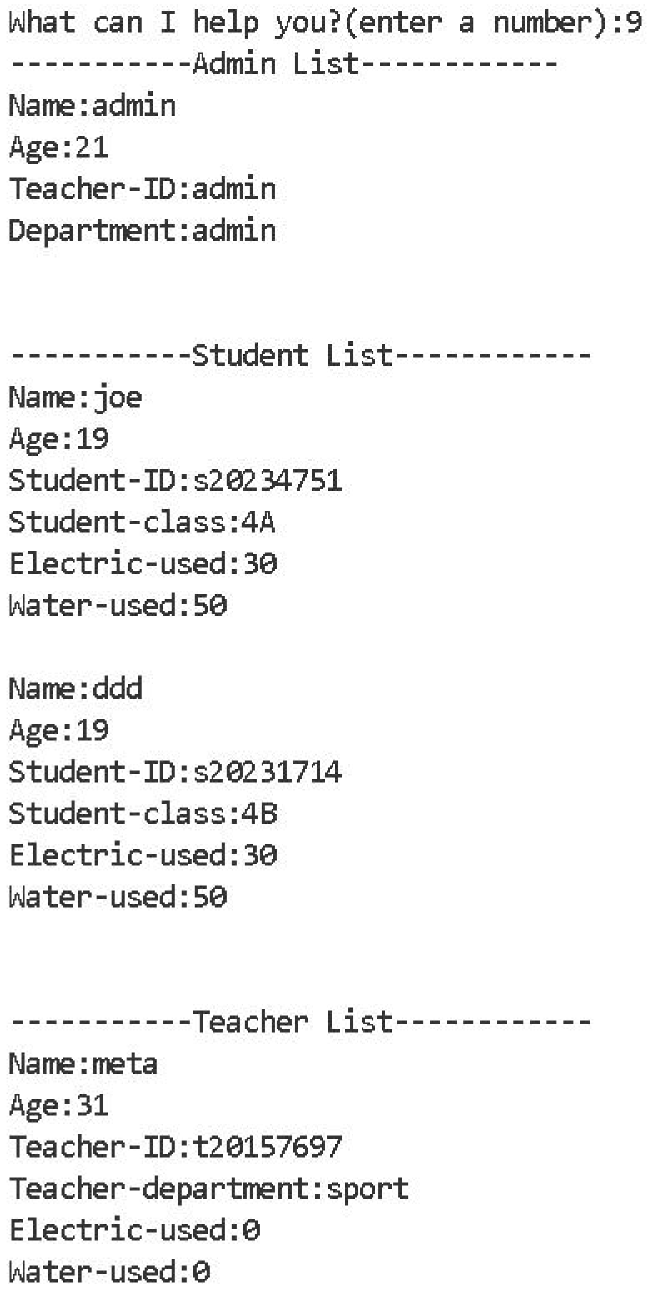
```
What can I help you?(enter a number):7
-----------Admin List------------
Name:admin
Age:21
Teacher-ID:admin
Department:admin


-----------Student List------------
Name:joe
Age:19
Student-ID:s20234751
Student-class:4A
Electric-used:30
Water-used:50

Name:ddd
Age:19
Student-ID:s20231714
Student-class:4B
Electric-used:30
Water-used:50


-----------Teacher List------------
Name:meta
Age:31
Teacher-ID:t20157697
Teacher-department:sport
Electric-used:0
Water-used:0
```

8.

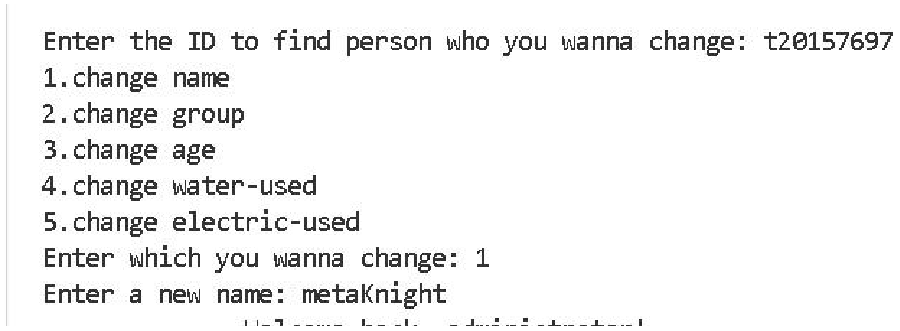


```
What can I help you?(enter a number):8
Enter the name or ID to search:meta
teacher-name:meta age:31 water-used:0 electric-used:0

==============Welcome back, administrator!============
=================1.Add admin-user=====================
=================2.Delete admin-user==================
=================3.List of arrears====================
=================4.Delete user========================
=================5.View students======================
=================6.View teachers======================
=================7.View all===========================
=================8.Search=============================
=================9.Edit===============================
=================0.Logout=============================

What can I help you?(enter a number):8
Enter the name or ID to search:admin
admin-name:admin age:21
```

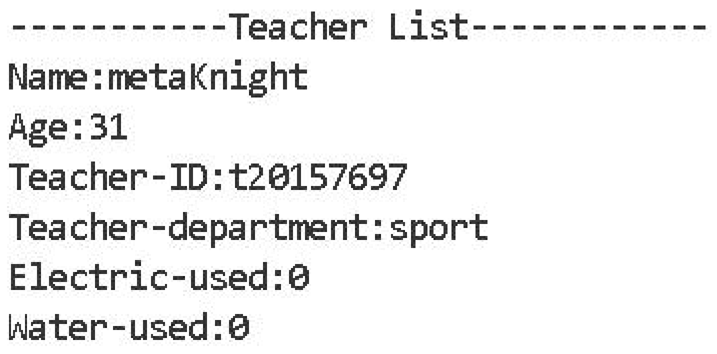
9.



```
What can I help you?(enter a number):9
-----------Admin List------------
Name:admin
Age:21
Teacher-ID:admin
Department:admin


-----------Student List------------
Name:joe
Age:19
Student-ID:s20234751
Student-class:4A
Electric-used:30
Water-used:50

Name:ddd
Age:19
Student-ID:s20231714
Student-class:4B
Electric-used:30
Water-used:50


-----------Teacher List------------
Name:meta
Age:31
Teacher-ID:t20157697
Teacher-department:sport
Electric-used:0
Water-used:0
```

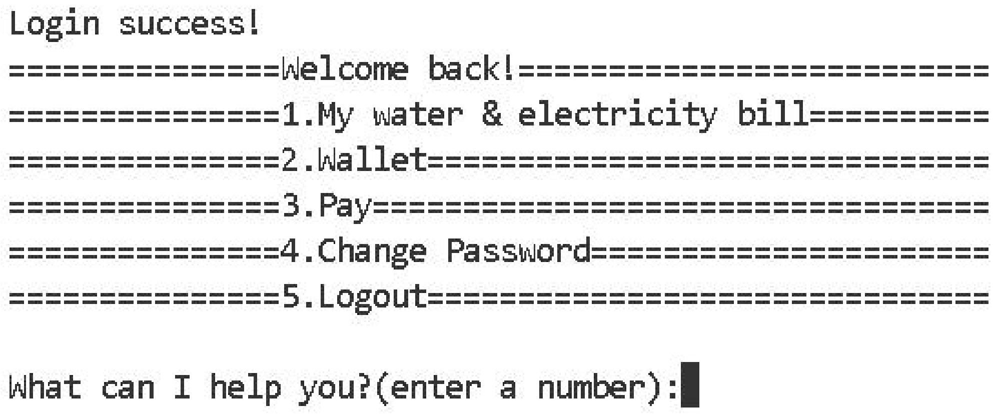


```
Enter the ID to find person who you wanna change: t20157697
1.change name
2.change group
3.change age
4.change water-used
5.change electric-used
Enter which you wanna change: 1
Enter a new name: metaKnight
              Welcome back, administrator!
```



```
-----------Teacher List------------
Name:metaKnight
Age:31
Teacher-ID:t20157697
Teacher-department:sport
Electric-used:0
Water-used:0
```

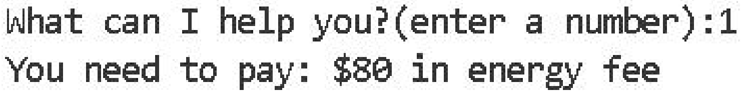
用户界面：



```
Login success!
===============Welcome back!==========================
===============1.My water & electricity bill==========
===============2.Wallet===============================
===============3.Pay==================================
===============4.Change Password======================
===============5.Logout===============================

What can I help you?(enter a number):
```

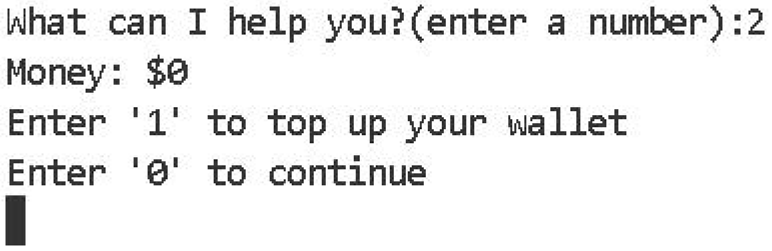
用户界面功能：

1.

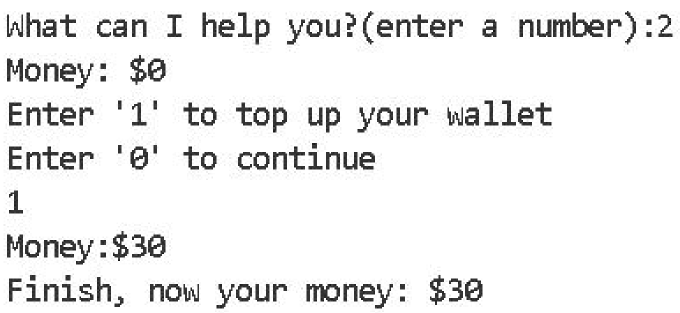


```
What can I help you?(enter a number):1
You need to pay: $80 in energy fee
```

2.

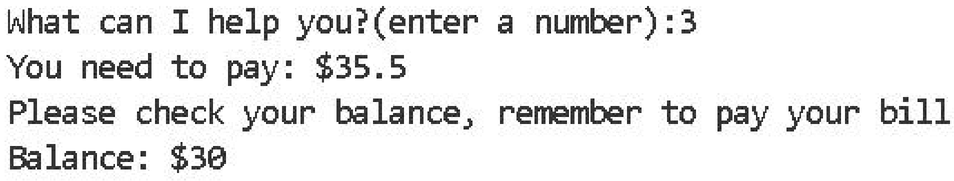


```
What can I help you?(enter a number):2
Money: $0
Enter '1' to top up your wallet
Enter '0' to continue
```

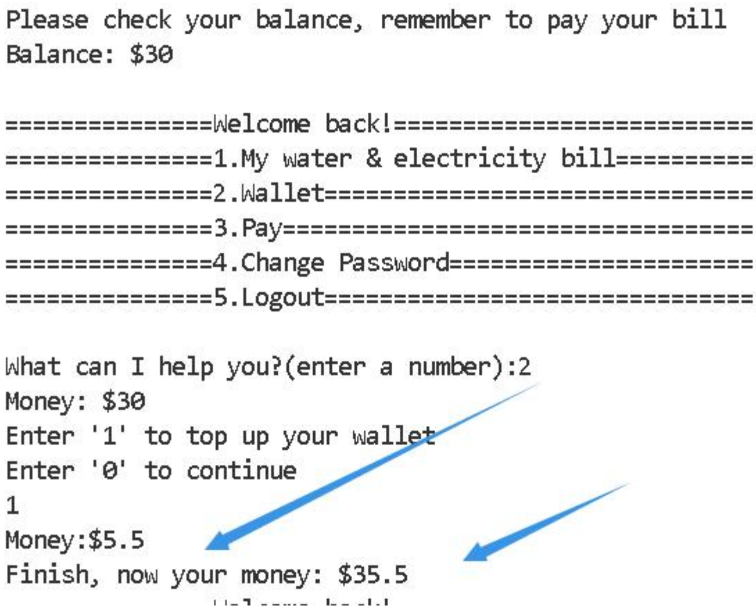


```
What can I help you?(enter a number):2
Money: $0
Enter '1' to top up your wallet
Enter '0' to continue
1
Money:$30
Finish, now your money: $30
```

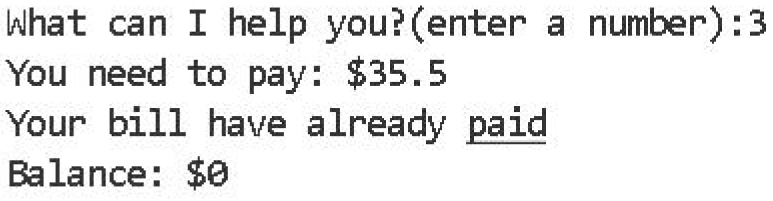
3.



```
What can I help you?(enter a number):3
You need to pay: $35.5
Please check your balance, remember to pay your bill
Balance: $30
```

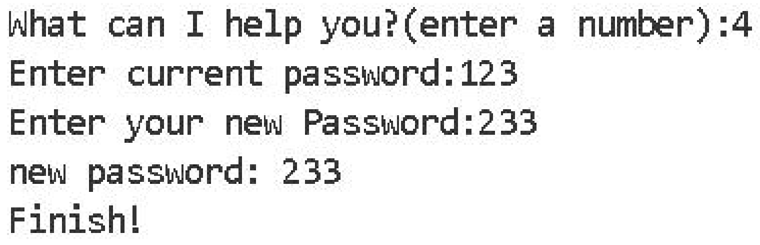


```
Please check your balance, remember to pay your bill
Balance: $30

===============Welcome back!==========================
===============1.My water & electricity bill==========
===============2.Wallet===============================
===============3.Pay==================================
===============4.Change Password======================
===============5.Logout===============================

What can I help you?(enter a number):2
Money: $30
Enter '1' to top up your wallet
Enter '0' to continue
1
Money:$5.5
Finish, now your money: $35.5
```

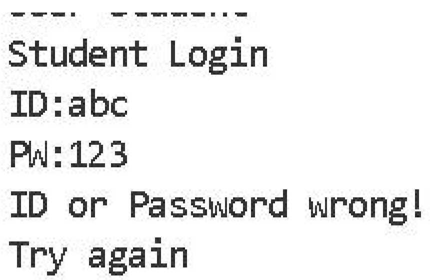


```
What can I help you?(enter a number):3
You need to pay: $35.5
Your bill have already paid
Balance: $0
```

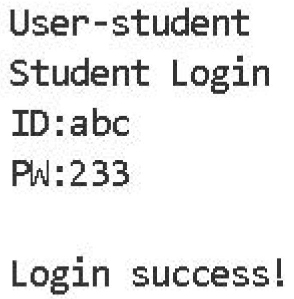
4.



```
What can I help you?(enter a number):4
Enter current password:123
Enter your new Password:233
new password: 233
Finish!
```



```
Student Login
ID:abc
PW:123
ID or Password wrong!
Try again
```



```
User-student
Student Login
ID:abc
PW:233

Login success!
```

2. 系统问题及其解决方案。

最常出现的问题就是输入输出时候的输入流未清理干净的问题，比如要求用户输入账户密码是采用 getline 来获取，但是需要提前知道调入这个 getline 之前有没有留在流里的字符没被舍弃，常见的例如 cin 捕获到回车符会停止输入捕捉，但是并不会丢弃回车符，会撤回会仍然留在输出流里面，导致下一个输入被跳过，最开始用的是 cin.ignore()来无视一个字符，但是这样做会导致当流是空的时候，就会忽略掉用户输入的第一个字符，也就是最开始 getline 的时候用户输入 abc，’a’字符会被忽视不被读取，造成严重错误。之后查找办法说可以利用 cin.peek()来检测是否未空，当 cin.peek()!=EOF 时执行

cin.ignore()来忽视掉流里的回车符，但是这样做依然没有显著效果。最后决定针对每次的 cin 单字符捕获后用 getchar（）吃掉回车符。

除此以外就是 switch 里的变量被跳过处理问题，c++不允许在 case 里单独实现功能，需要改用函数来实现，因此每个 case 的单独功能实现都被改成调用函数实现。

对于数据的存储问题，用了 C++的 vector 容器来存取数据，用一个类专门存放用户的账号密码信息，作为暂存的信息类，之后再把这些存放好的信息类导入进对应的成员类当中。

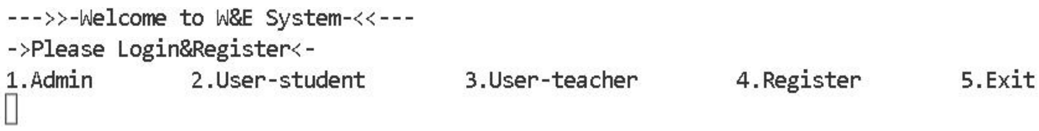
3．可扩展功能及设计实现构想。

针对自动缴费方面，当用户的钱包余额满足需要扣除的费用时，会自动扣除功能

针对管理员可以单独创建新超级用户。

四、设计总结

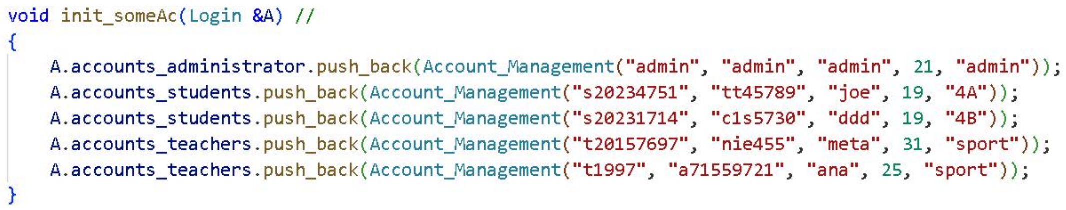
在最开始的主页面设计，因为没完成函数的逐步退出功能，因此做成了 while 循环反复执行 index 的进入，模拟退出到主菜单的功能。



```
--->>-Welcome to W&E System-<<---
->Please Login&Register<-
1.Admin        2.User-student        3.User-teacher        4.Register        5.Exit
```

进入主菜单的页面，有五个功能，User 的登陆选项为了减少分支带来的不断函数调用深入问题，以尽量减少深度调用改为平面化处理，对功能进行展开，因此有五个选项，分别对应管理员、教师、学生和第一次登录用的注册以及退出系统功能。

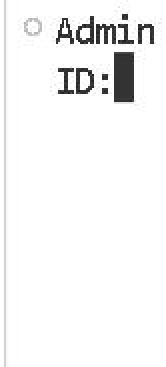
为了方便测试，在系统运行时先完成了对成员的预载入，作为 NPC 存在。



```
void init_someAc(Login &A) //
{
    A.accounts_administrator.push_back(Account_Management("admin", "admin", "admin", 21, "admin"));
    A.accounts_students.push_back(Account_Management("s20234751", "tt45789", "joe", 19, "4A"));
    A.accounts_students.push_back(Account_Management("s20231714", "c1s5730", "ddd", 19, "4B"));
    A.accounts_teachers.push_back(Account_Management("t20157697", "nie455", "meta", 31, "sport"));
    A.accounts_teachers.push_back(Account_Management("t1997", "a71559721", "ana", 25, "sport"));
}
```

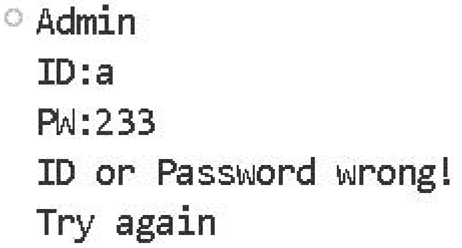
这些 NPC 的存在也可以作为检测之后登陆测试代码的选择，他们对应了不同的成员，有管理员、学生和老师。

一号功能管理员登录



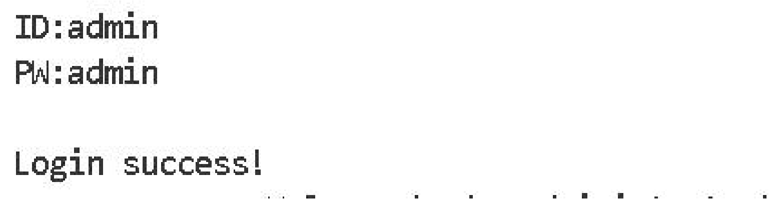
```
Admin
ID:
```

进入界面会看到提示输入 ID 和密码，同时这部分也完成了对数据的检测，也就是登录的密码是否正确。



```
Admin
ID:a
PW:233
ID or Password wrong!
Try again
```

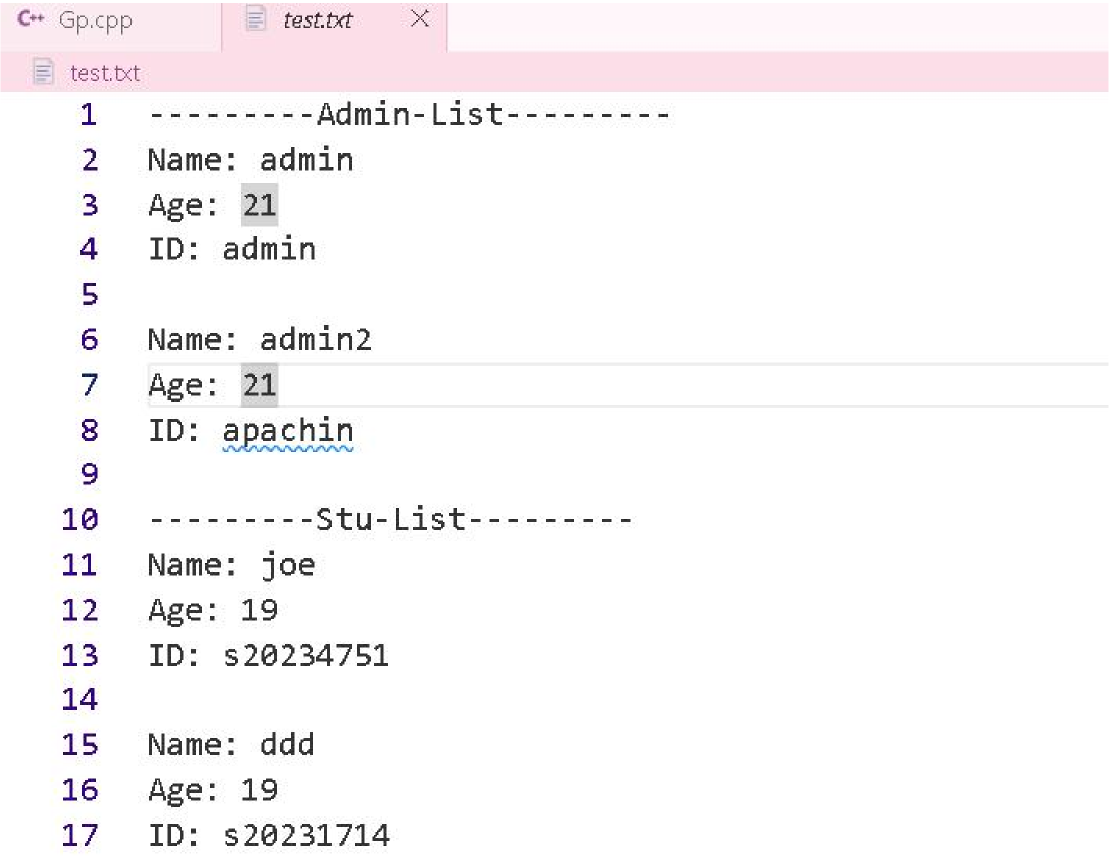
回顾 init 函数会发现最开始的管理员仅有 admin 一位，当输入不正确的信息，则会输出错误信息，提示登陆失败，完成了对登录的检测。



```
ID:admin
PW:admin

Login success!
```

当输入正确的账号密码后，则会正确登录。

另外对每次的成员变动都会把记录更新在 txt 文档当中。



```
---------Admin-List---------
Name: admin
Age: 21
ID: admin

Name: admin2
Age: 21
ID: apachin

---------Stu-List---------
Name: joe
Age: 19
ID: s20234751

Name: ddd
Age: 19
ID: s20231714
```

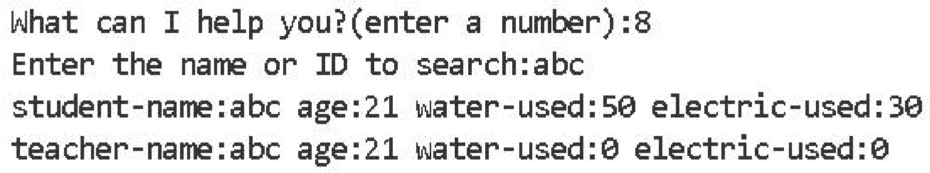
透过文档可以查看到最新的用户信息。

这里的 ID 是作为独立的成员识别码存在，ID 是不允许重复的，每当新用户注册时，第一个填写的 ID 需要具有唯一性，当出现重复时就会出现错误，而名字是允许重名的，密码也是。

在管理员界面实现了九个功能，由于管理员是不需要交付水电费这些功能的，因此不在管理员里增设缴付功能，但是管理员可以透过

编辑功能调整用户的水电费用量和成员信息。

对于八号功能查找上，因为是关键查找，所以不提供用户信息，仅留下一个输入栏让用户输入，用户有两个选择，如果输入 id 关键查找只会出来一个人，但是输入名字可以找出对应名字重复的人，提供了指定查找和范围查找。



```
What can I help you?(enter a number):8
Enter the name or ID to search:abc
student-name:abc age:21 water-used:50 electric-used:30
teacher-name:abc age:21 water-used:0 electric-used:0
```

可以看到他们来自学生和老师，但是因为同名都会被查找出来。

由于程序很多都是与查找相关，因此对于 for 迭代器的使用非常频繁，C++的 auto 变量可以很轻松的解决类型问题，不用再去翻看用了什么类型声明的变量。

在界面选择上，尽量用系统调用 cls 来完成对屏幕的清理，但是有些罗列出来的数据需要停留，因此有限地使用屏幕清理功能。

比较麻烦的是对成员变量的生成和导入，最开始想的是用成员类生成一个 new 对象建立起来后再把他 push 进去准备好的成员 vector 容器，但是实际操作的时候发现了成员并没有 push 进去，也没有报错，访问的时候 vector 为空，后来改成在 push 里直接生成构造函数反而可以成功 push 进去 vector 向量中。

对密码的检测方面，因为每一次的用户进入都需要经过 login，也就是在 login 里的成员变量可以捕获到用户的登录信息，因此可以对用户进行身份验证，这个身份信息会一直保存直到下一次新的登录信

息覆盖掉他，于是乎就可以透过访问变量来获取成员的账号密码，省去了跳去对成员的访问。

五、参考文献

https://cplusplus.com/reference/vector/vector/

https://cplusplus.com/reference/vector/vector/erase/

https://blog.csdn.net/u013654125/article/details/77321081

http://c.biancheng.net/view/338.html

https://www.runoob.com/cplusplus/cpp-files-streams.html

https://zhuanlan.zhihu.com/p/352961501

https://blog.csdn.net/chongzi_daima/article/details/103508585

https://blog.csdn.net/wangkai_123456/article/details/76598917

六、附录

给出系统的核心代码

```
#include <iostream>

#include <string.h>

#include <algorithm>

#include <vector>

#include <string>

#include <fstream>

#include <stdlib.h>
```

```cpp
using namespace std;

class Member
{
public:
    string name;
    int age;
    double used_electric;
    double used_water;
    void displayInfo()
    {
        cout << "name:" << name << endl;
        cout << "age:" << age << endl;
        cout << "electric:" << used_electric << endl;
        cout << "water:" << used_water << endl;
    }
};

class Student : public Member
{
public:
    Student(){}; // default
    Student(string name, int age, string cls, string id,
double electric = 30, double water = 50, double money = 0)
    {
        this->name = name;
        this->age = age;
        this->student_class = cls;
        this->student_ID = id;
        this->used_electric = electric;
        this->used_water = water;
        this->money = money;
    }

    void displayInfo()
    {
        cout << "Name:" << name << endl;
        cout << "Age:" << age << endl;
        cout << "Student-ID:" << student_ID << endl;
        cout << "Student-class:" << student_class << endl;
        cout << "Electric-used:" << used_electric << endl;
        cout << "Water-used:" << used_water << endl;
    }

    string getStuID() const
    {
        return student_ID;
    }
    string getStuCls() const
    {
        return student_class;
    }
    double getMoney() const
    {
        return money;
    }
    void addMoney(double cash)
    {
        this->money += cash;
    }
    void changeCds(string newCds)
    {
        this->student_class=newCds;
    }
private:
    string student_class;
    string student_ID;
    double money;
};

class Teacher : public Member
{
public:
    Teacher(){}; // default
    Teacher(string name, int age, string dept, string id, double electric = 0, double water = 0, double money = 30)
    {
        this->name = name;
        this->age = age;
        this->teacher_department = dept;
        this->teacher_ID = id;
        this->used_electric = electric;
        this->used_water = water;
        this->money = money;
    }

    void displayInfo()
    {
        cout << "Name:" << name << endl;
        cout << "Age:" << age << endl;
        cout << "Teacher-ID:" << teacher_ID << endl;
        cout << "Teacher-department:" << teacher_department << endl;
        cout << "Electric-used:" << used_electric << endl;
        cout << "Water-used:" << used_water << endl;
    }
    string getTecID() const
    {
        return teacher_ID;
    }
    string getTecCls() const
    {
        return teacher_department;
    }
    double getMoney() const
    {
        return money;
    }
    void addMoney(double cash)
    {
        this->money += cash;
    }
    void changeCds(string newCds)
    {
        this->teacher_department=newCds;
    }
private:
    string teacher_department;
    string teacher_ID;
    double money;
};

class Administrator : public Member
{
public:
    Administrator(){}; // default
    Administrator(string name, int age, string dept, string id, double electric = 30, double water = 10, double money = 9999)
    {
        this->name = name;
        this->age = age;
        this->admin_department = dept;
        this->admin_ID = id;
        this->used_electric = electric;
        this->used_water = water;
        this->money = money;
    }

    void displayInfo()
    {
        cout << "Name:" << name << endl;
        cout << "Age:" << age << endl;
        cout << "Teacher-ID:" << admin_ID << endl;
        cout << "Department:" << admin_department << endl;
    }
    string getAdminID() const
    {
        return admin_ID;
    }
    string getAdminCls() const
    {
        return admin_department;
    }

private:
    string admin_department;
    string admin_ID;
    double money;
};
class Account_Management
{
public:
    Account_Management(const string &username, const
string &password, const string &nickname, const int &age,
const string &cds) // cds=class or department
    {
        this->username = username;
        this->password = password;
        this->nickname = nickname;
        this->age = age;
        this->cds = cds;
    }
    bool check(const string &username, const string
&password) const
    {
        return this->username == username && this->password
== password;
    }
    string getUsername() const
```

```
{
    return username;
}
string getPassword() const
{
    return password;
}

string getNickname() const
{
    return nickname;
}

int getAge() const
{
    return age;
}

string getCds() const
{
    return cds;
}

void changePwd(string newPWD)
```

```
    {
        this->password = newPWD;
    }

private:
    string username;
    string password;
    int age;
    string nickname;
    string cds;
};

class Login
{
public:
    int login_check;
    char user_id[13];
    char user_pw[13];
    int len_userID;
    int len_userPW;
    vector<Account_Management> accounts_students; // DATA
SAVE
    vector<Account_Management> accounts_teachers;
    vector<Account_Management> accounts_administrator;

    void index()
    {
        // Index part
        cout << "--->>-Welcome to W&E System-<<---" << endl;
        while (1)
        {
            cout << "->Please Login&Register<-" << endl;
            cout <<
"1.Admin        2.User-student        3.User-teacher
    4.Register        5.Exit" << endl;
            cin >> login_check;
            if (login_check == 1 || login_check == 2 ||
login_check == 3 || login_check == 4)
            {
                system("cls");
                break;
            }
            if (login_check == 5)
            {
                cout << "See you next time" << endl;
                exit(1);
            }
        }
        login();
    }
    void displayAccounts() const
    {
        cout << "--------------------------" << endl;
        cout << "Administrator List" << endl;
        cout << "--------------------------" << endl;
        for (const auto &account : accounts_administrator)
        {
            cout << "Username: " << account.getUsername() << endl;
            cout << "Password: " << account.getPassword() << endl;
            cout << "Nickname:" << account.getNickname() << endl;
            cout << "Age:" << account.getAge() << endl;
            cout << "Class/Department:" << account.getCds() << endl;
            cout << endl;
        }
        cout << endl;
        cout << "-------------------------" << endl;
        cout << "Student List" << endl;
        cout << "-------------------------" << endl;
        for (const auto &account : accounts_students)
        {
            cout << "Username: " << account.getUsername()
<< endl;
            cout << "Password: " << account.getPassword()
<< endl;
            cout << "Nickname:" << account.getNickname() <<
endl;
            cout << "Age:" << account.getAge() << endl;
            cout << "Class/Department:" << account.getCds()
<< endl;
            cout << endl;
        }
        cout << endl;
        cout << "-------------------------" << endl;
        cout << "Teacher List" << endl;
        cout << "-------------------------" << endl;
        for (const auto &account : accounts_teachers)
        {
            cout << "Username: " << account.getUsername() << endl;
            cout << "Password: " << account.getPassword() << endl;
            cout << "Nickname:" << account.getNickname() << endl;
            cout << "Age:" << account.getAge() << endl;
            cout << "Class/Department:" << account.getCds() << endl;
            cout << endl;
        }
    }

    void Build() // data->member
    {
        for (auto it = accounts_administrator.begin(); it < accounts_administrator.end(); ++it) // ok
        {
            Mana_Adm.push_back(Administrator((*it).getNickname(), (*it).getAge(), (*it).getCds(), (*it).getUsername()));
        }
        // if(!Mana_Adm.empty())
        // cout<<"success"<<endl;
        for (auto it = accounts_students.begin(); it <
accounts_students.end(); ++it) // ok
        {
            Mana_Stu.push_back(Student((*it).getNickname
(), (*it).getAge(), (*it).getCds(),
(*it).getUsername()));
        }
        for (auto it = accounts_teachers.begin(); it <
accounts_teachers.end(); ++it) // ok
        {
            Mana_Tec.push_back(Teacher((*it).getNickname
(), (*it).getAge(), (*it).getCds(),
(*it).getUsername()));
        }
    }

    void index_login()
    {
        int n;
        if (card_login == 0) // admin part
        {
            cout << "==============Welcome back,
administrator!============" << endl;
            cout << "=================1.Add
admin-user=====================" << endl;
            cout << "=================2.Delete
admin-user==================" << endl;
            cout << "=================3.List of
arrears====================" << endl;
            cout << "=================4.Delete
user========================" << endl;
            cout << "=================5.View
students======================" << endl;
            cout << "=================6.View
teachers======================" << endl;
            cout << "=================7.View
all===========================" << endl;
            cout <<
"=================8.Search=============================
" << endl;
```

```
            cout <<
"=================9.Edit===============================
" << endl;
            cout <<
"=================0.Logout=============================
" << endl;
            cout << endl;
            cout << "What can I help you?(enter a number):";
            cin >> n;
            getchar();
            switch (n)
            {
            case 1:
                ADM_f1();
                break;
            case 2:
                ADM_f2();
                break;
            case 3:
                ADM_f3();
                break;
            case 4:
```

```
    ADM_f4();
    break;
case 5:
    ADM_f5();
    break;
case 6:
    ADM_f6();
    break;
case 7:
    ADM_f7();
    break;
case 8:
    ADM_f8();
    break;
case 9:
    ADM_f9();
    break;
default:
    ADM_f10();
    break;
}
if (n != 0)
```

```
            {
                index_login();
            }
        }
        if (card_login == 1 || card_login == 2)
        {
            cout << "===============Welcome
back!==========================" << endl;
            cout << "===============1.My water &
electricity bill==========" << endl;
            cout <<
"===============2.Wallet===============================
" << endl;
            cout <<
"===============3.Pay==================================
" << endl;
            cout << "===============4.Change
Password======================" << endl;
            cout <<
"===============5.Logout===============================
" << endl;
            cout << endl;
            cout << "What can I help you?(enter a number):";
            cin >> n;
            getchar();
            switch (n)
            {
            case 1:
                USR_f1();
                break;
            case 2:
                USR_f2();
                break;
            case 3:
                USR_f3();
                break;
            case 4:
                USR_f4();
                break;
            case 5: // out exec
                break;
            default:
                cout << "Error input" << endl;
                index_login();
```

```
        }
        if (n != 5)
        {
            index_login();
        }
        USR_f5();
    }
    index();
}

void transTxt() // save to txt
{
    ofstream f("test.txt");
    f << "---------Admin-List---------" << endl;
    for (auto &it : Mana_Adm)
    {
        f << "Name: " << it.name << endl;
        f << "Age: " << it.age << endl;
        f << "ID: " << it.getAdminID() << endl;
        f << endl;
    }
    f << "---------Stu-List---------" << endl;
    for (auto &it : Mana_Stu)
```

```cpp
        {
            f << "Name: " << it.name << endl;
            f << "Age: " << it.age << endl;
            f << "ID: " << it.getStuID() << endl;
            f << endl;
        }
        f << "---------Tec-List---------" << endl;
        for (auto &it : Mana_Tec)
        {
            f << "Name: " << it.name << endl;
            f << "Age: " << it.age << endl;
            f << "ID: " << it.getTecID() << endl;
            f << endl;
        }
    }

private:
    vector<Administrator> Mana_Adm;
    vector<Student> Mana_Stu;
    vector<Teacher> Mana_Tec;
    int card_login = 0; // identity 0admin 1stu 2teach
    void login()
    {
    }
    switch (login_check)
    {
    case 1:
        cout << "Admin" << endl;
        Login_Branch(login_check);
        break;
    case 2:
        cout << "User-student" << endl;
        Login_Branch(login_check);
        break;
    case 3:
        cout << "User-teacher" << endl;
        Login_Branch(login_check);
        break;
    case 4:
        cout << "Register" << endl;
        Add_Account(login_check);
    default:
        index();
        break;
    }
}
```

```
void Login_Branch(int login_check)
{
    char *id_login = new char[13]();
    int len_id;
    char *pwd_login = new char[13]();
    int len_pwd;

    if (login_check == 2)
        cout << "Student Login" << endl;
    if (login_check == 3)
        cout << "Teacher Login" << endl;

    // id
    cout << "ID:";
    cin.ignore();
    Login_Info_get(id_login, len_id);
    strcpy(user_id, id_login); // send to public
    len_userID = len_id;

    // pw
    cout << "PW:";
    Login_Info_get(pwd_login, len_pwd);
    strcpy(user_pw, pwd_login);
    len_userPW = len_pwd;
```

```
        // check is_valid
        if (login_check == 1) // admin part
        {
            bool login_is_valid = false;
            for (const auto &account : accounts_administrator)
            {
                if (account.check(user_id, user_pw))
                {
                    login_is_valid = true;
                    break;
                }
            }
            if (login_is_valid)
            {
                cout << endl;
                cout << "Login success!" << endl;
                card_login = 0;
                index_login();
                cout << endl;
            }
            else
```

```
    {
        cout << "ID or Password wrong!" << endl;
        cout << "Try again" << endl;
        index();
    }
}
if (login_check == 2) // student part
{
    bool login_is_valid = false;
    for (const auto &account : accounts_students)
    {
        if (account.check(user_id, user_pw))
        {
            login_is_valid = true;
            break;
        }
    }

    if (login_is_valid)
    {
        cout << endl;
        cout << "Login success!" << endl;
        card_login = 1;
        index_login();
        cout << endl;
    }
    else
    {
        cout << "ID or Password wrong!" << endl;
        cout << "Try again" << endl;
        index();
    }
}
if (login_check == 3) // teacher part
{
    bool login_is_valid = false;
    for (const auto &account : accounts_teachers)
    {
        if (account.check(user_id, user_pw))
        {
            login_is_valid = true;
            break;
        }
    }

    if (login_is_valid)
```

```
        {
            cout << endl;
            cout << "Login success!" << endl;
            card_login = 2;
            index_login();
            cout << endl;
        }
        else
        {
            cout << "ID or Password wrong!" << endl;
            cout << "Try again" << endl;
            index();
        }
    }
    delete[] id_login;
    delete[] pwd_login;
    id_login = pwd_login = nullptr;
}
void Add_Account(int login_check)
{
    char *new_id_login = new char[13]();
    char *tmp_pwd = new char[13]();
```

```
        char *new_pwd_login = new char[13]();
        int flag;

        cout << "Input '1' to create User-Student or input '2' to create User-Teacher" << endl; // branch
        while (1)
        {
            cin >> flag;
            getchar();
            if (flag > 2 || flag < 1)
            {
                cout << "invalid input" << endl;
                continue;
            }
            if (flag == 2 || flag == 1)
                break;
        }

        cout << "newID(Limit:13):";
        cin.getline(new_id_login, 13, '\n');
        for(auto&it:accounts_administrator)
        {
            if(it.getUsername()==new_id_login)
```

```
    {
        cout << endl;
        cout << "Account already exist" << endl;
        cout << endl;
        exit(1);
    }
}
for(auto&it:accounts_students)
{
    if(it.getUsername()==new_id_login)
    {
        cout << endl;
        cout << "Account already exist" << endl;
        cout << endl;
        exit(1);
    }
}
 for(auto&it:accounts_teachers)
{
    if(it.getUsername()==new_id_login)
    {
        cout << endl;
```

```
                cout << "Account already exist" << endl;
                cout << endl;
                exit(1);
            }
        }
        cout << "newPassword(Limit:13):";
        cin.getline(tmp_pwd, 13, '\n');
        cout << "Password confirm:";
        cin.getline(new_pwd_login, 13, '\n');
        // check_is_valid?

        // send to AM
        // success register, addition materials ↓
        string temp_nickname;
        int temp_age;
        string temp_cds; // class or department
        // id -> login username
        if (flag == 1)
        {
            if (strcmp(tmp_pwd, new_pwd_login) == 0)
            {
                cout << "-----------Please enter the
personal information-----------" << endl;
```

```
                cout << "Your name:";
                cin >> temp_nickname;
                cout << "Your age:";
                cin >> temp_age;
                cout << "Your class/department:";
                cin >> temp_cds;

                accounts_students.push_back(Account_Mana
gement(new_id_login, new_pwd_login, temp_nickname,
temp_age, temp_cds));
                Mana_Stu.push_back(Student(temp_nickname,
temp_age, temp_cds, new_id_login));
            }
            else
            {
                cout << "Not the same password,please try
again" << endl;
                index();
            }
        }
        if (flag == 2)
        {
            // send to AM
            if (strcmp(tmp_pwd, new_pwd_login) == 0)
            {
                cout << "-----------Please enter the personal information------------" << endl;
                cout << "Your name:";
                cin >> temp_nickname;
                cout << "Your age:";
                cin >> temp_age;
                cout << "Your class/department:";
                cin >> temp_cds;

                accounts_teachers.push_back(Account_Management(new_id_login, new_pwd_login, temp_nickname, temp_age, temp_cds));
                Mana_Tec.push_back(Teacher(temp_nickname, temp_age, temp_cds, new_id_login));
            }
            else
            {
                cout << "Not the same password,please try again" << endl;
                index();
```

```
        }
    }
    transTxt();
    delete[] tmp_pwd;
    delete[] new_id_login;
    delete[] new_pwd_login;
    tmp_pwd = new_id_login = new_pwd_login = nullptr;
}
void Login_Info_get(char *arr, int &len)
{
    cin.getline(arr, 13, '\n');
    len = strlen(arr);
}

// switch part
void ADM_f1() // Add admin-user ok
{
    string add_temp_nickname;
    int add_temp_age;
    string add_temp_id;
    string add_temp_pwd;
    cout << "Please enter your name:";
    getline(cin, add_temp_nickname);
```

```
        cout << "Please enter your age:";
        cin >> add_temp_age;
        getchar();
        cout << "Please enter your id:";
        getline(cin, add_temp_id);
        for(auto&it:Mana_Adm)
        {
            if(it.getAdminID()==add_temp_id)
                {
                    cout<<"Error: ID already exist"<<endl;
                    cout<<endl;
                    index_login();
                }
        }
        cout<<"Please enter your pwd: ";
        getline(cin,add_temp_pwd);

        Mana_Adm.push_back(Administrator(add_temp_nickname, add_temp_age, "admin", add_temp_id));
        accounts_administrator.push_back(Account_Management(add_temp_id,add_temp_pwd,add_temp_nickname,add_temp_age,"admin"));
        cout << "============Now admin-list============"
<< endl;
        for (auto &it : Mana_Adm)
        {
            it.displayInfo();
            cout << endl;
        }
        transTxt();
        cout << "Finish!" << endl;
    }
    void ADM_f2() // delete admin ok
    {
        string d_id;
        cout << "==========Find the ID you wanna
delete===========" << endl;
        for (auto &admin : Mana_Adm)
            cout << admin.getAdminID() << endl;
        cout << "Enter ID here:";
        getline(cin, d_id);
        cout << "Your target is: " << d_id << endl;
        for (auto it = Mana_Adm.begin(); it <
Mana_Adm.end();)
```

```
        {
            if ((*it).getAdminID() == d_id)
            {
                it = Mana_Adm.erase(it);
            }
            else
                ++it;
        }

        cout << "------------After delete list------------" << endl;
        for (auto &admin : Mana_Adm)
            cout << admin.getAdminID() << endl;
        for (auto &stu : Mana_Stu)
            cout << stu.getStuID() << endl;
        for (auto &tec : Mana_Tec)
            cout << tec.getTecID() << endl;


        transTxt();
        cout << "FInish!!" << endl;
    }
    void ADM_f3() // ListArrears ok
    {
        cout << "----------List of
arrears(name)----------" << endl;
        for (auto it = Mana_Stu.begin(); it <
Mana_Stu.end();)
        {
            if ((((*it).used_electric - 15) * 0.5 +
((*it).used_water - 10) * 0.7) > 0)
            {
                cout << (*it).name << " energy bill: $" <<
((*it).used_electric - 15) * 0.5 + ((*it).used_water - 10)
* 0.7 << endl;
                ++it;
            }
            else
                ++it;
        }
        for (auto it = Mana_Tec.begin(); it <
Mana_Tec.end();)
        {
            if ((((*it).used_electric - 15) * 0.5 +
((*it).used_water - 10) * 0.7) > 0)
            {
                cout << (*it).name << " energy bill: $" <<
((*it).used_electric - 15) * 0.5 + ((*it).used_water - 10)
* 0.7 << endl;
                ++it;
            }
            else
                ++it;
        }
        cout << endl;
    }
    void ADM_f4() // delete User ok
    {
        int n;
        string d_id;
        cout << "if delete User-student enter '1' " << endl;
        cout << "if delete User-teacher enter '2' " << endl;
        cin >> n;
        getchar();
        if (n < 1 || n > 2)
        {
            cout << "Error input" << endl;
            return;
        }
        if (n == 1)
        {
            cout << "--------Student list---------" <<
endl;
            for (auto &stu : Mana_Stu)
                cout << stu.getStuID() << endl;
            cout << "Enter ID to delete" << endl;
            getline(cin, d_id);
            cout << "Your target is: " << d_id << endl;
            for (auto it = Mana_Stu.begin(); it <
Mana_Stu.end();)
            {
                if ((*it).getStuID() == d_id)
                {
                    it = Mana_Stu.erase(it);
                }
                else
                    ++it;
            }
            cout << "------------After delete
list------------" << endl;
```

```
      for (auto &admin : Mana_Adm)
        cout << admin.getAdminID() << endl;
      for (auto &stu : Mana_Stu)
        cout << stu.getStuID() << endl;
      for (auto &tec : Mana_Tec)
        cout << tec.getTecID() << endl;

      transTxt();
      cout << "FInish!!" << endl;
    }
    if (n == 2)
    {
      cout << "--------Teacher list---------" <<
endl;
      for (auto &tec : Mana_Tec)
        cout << tec.getTecID() << endl;
      cout << "Enter ID to delete" << endl;
      getline(cin, d_id);
      cout << "Your target is: " << d_id << endl;
      for (auto it = Mana_Tec.begin(); it <
Mana_Tec.end();)
      {
```

```
                if ((*it).getTecID() == d_id)
                {
                    it = Mana_Tec.erase(it);
                }
                else
                    ++it;
            }
            cout << "------------After delete
list------------" << endl;
            for (auto &admin : Mana_Adm)
                cout << admin.getAdminID() << endl;
            for (auto &stu : Mana_Stu)
                cout << stu.getStuID() << endl;
            for (auto &tec : Mana_Tec)
                cout << tec.getTecID() << endl;

            transTxt();
            cout << "FInish!!" << endl;
        }
    }
    void ADM_f5() // view stu ok
    {
        cout << "-----------Student List------------" <<
endl;
        for (auto &stu : Mana_Stu)
        {
            stu.displayInfo();
            cout << endl;
        }
        cout << endl;
    }
    void ADM_f6() // view Tec ok
    {
        cout << "-----------Teacher List------------" <<
endl;
        for (auto &tec : Mana_Tec)
        {
            tec.displayInfo();
            cout << endl;
        }
        cout << endl;
    }
    void ADM_f7() // viewAll ok
    {
        cout << "-----------Admin List------------" <<
endl;
        for (auto &it : Mana_Adm)
        {
            it.displayInfo();
            cout << endl;
        }
        cout << endl;
        ADM_f5();
        ADM_f6();
    }
    void ADM_f8() // searchKey ok
    {
        string key;
        cout << "Enter the name or ID to search:";
        getline(cin, key);
        for (const auto &admin : Mana_Adm)
        {
            if (admin.name == key)
                cout << "admin-name:" << admin.name << "
age:" << admin.age << endl;
            else if (admin.getAdminID() == key)
```

```
                cout << "admin-name:" << admin.name << " age:" << admin.age << endl;
        }

        // stu list
        for (const auto &stu : Mana_Stu)
        {
            if (stu.name == key)
                cout << "student-name:" << stu.name << " age:" << stu.age << " water-used:" << stu.used_water << " electric-used:" << stu.used_electric << endl;
            else if (stu.getStuID() == key)
                cout << "student-name:" << stu.name << " age:" << stu.age << " water-used:" << stu.used_water << " electric-used:" << stu.used_electric << endl;
        }

        // tec list
        for (const auto &tec : Mana_Tec)
        {
            if (tec.name == key)
```

```
                cout << "teacher-name:" << tec.name << "
age:" << tec.age << " water-used:" << tec.used_water << "
electric-used:" << tec.used_electric << endl;
            else if (tec.getTecID() == key)
                cout << "teacher-name:" << tec.name << "
age:" << tec.age << " water-used:" << tec.used_water << "
electric-used:" << tec.used_electric << endl;
        }
        cout<<endl;
    }
    void ADM_f9() //Edit
    {
        string key;

        string change_name;
        string change_cds;
        int change_age;
        double change_water;
        double change_elec;

        int n;
        ADM_f7();
        cout<<"Enter the ID to find person who you wanna change: ";
        getline(cin,key);

        cout<<"1.change name"<<endl;
        cout<<"2.change group"<<endl;
        cout<<"3.change age"<<endl;
        cout<<"4.change water-used"<<endl;
        cout<<"5.change electric-used"<<endl;

        cout<<"Enter which you wanna change: ";
        cin>>n;
        getchar();
        for(auto& it:Mana_Stu)
        {
            if(key==it.getStuID())
            {
                switch(n)
                {
                    case 1: //change name
                        cout<<"Enter a new name: ";
                        getline(cin,change_name);
                        it.name=change_name;
                        break;
                    case 2: //change group
                        cout<<"Enter a new group: ";
                        getline(cin,change_cds);
                        it.changeCds(change_cds);
                        break;
                    case 3://change age
                        cout<<"Enter a new age: ";
                        cin>>change_age;
                        getchar();
                        it.age=change_age;
                        break;
                    case 4://change water
                        cout<<"Change water-used: ";
                        cin>>change_water;
                        getchar();
                        it.used_water=change_water;
                        break;
                    case 5://change elec
                        cout<<"Change elec-used: ";
                        cin>>change_elec;
                        getchar();
```

```
                it.used_electric=change_elec;
                break;
            default:
                break;
        }
        transTxt();
        cout<<"Finish!"<<endl;
    }
}
for(auto& it:Mana_Tec)
{
    if(key==it.getTecID())
    {
         switch(n)
        {
            case 1: //change name
                cout<<"Enter a new name: ";
                getline(cin,change_name);
                it.name=change_name;
                break;
            case 2: //change group
                cout<<"Enter a new group: ";
                getline(cin,change_cds);

                it.changeCds(change_cds);

                break;

            case 3://change age

                cout<<"Enter a new age: ";

                cin>>change_age;

                getchar();

                it.age=change_age;

                break;

            case 4://change water

                cout<<"Change water-used: ";

                cin>>change_water;

                getchar();

                it.used_water=change_water;

                break;

            case 5://change elec

                cout<<"Change elec-used: ";

                cin>>change_elec;

                getchar();

                it.used_electric=change_elec;

                break;

            default:
```

```
                        break;
                }
            }
        }
    }
    void ADM_f10() // exit ok
    {
        cout << "Logout" << endl;
        return;
    }

    void USR_f1() // energy bill ok
    {
        double sum;
        for (auto &it : Mana_Stu)
        {
            if (it.getStuID() == user_id)
            {
                sum = it.used_water + it.used_electric;
                cout << "You need to pay: $" << sum << " in
energy fee" << endl;
            }
        }
        for (auto &it : Mana_Tec)
        {
            if (it.getTecID() == user_id)
            {
                sum = it.used_water + it.used_electric;
                cout << "You need to pay: $" << sum << " in
energy fee" << endl;
            }
        }
    }
    void USR_f2() // money ok
    {
        int t;
        double cash = 0;
        for (auto &it : Mana_Stu)
        {
            if (it.getStuID() == user_id)
            {
                cout << "Money: $" << it.getMoney() << endl;
                cout << "Enter '1' to top up your wallet"
<< endl;
                cout << "Enter '0' to continue" << endl;
                cin >> t;
                getchar();
                if (t == 1)
                {
                    cout << "Money:$";
                    cin >> cash;
                    getchar();
                    it.addMoney(cash);
                    cout << "Finish, now your money: $" <<
it.getMoney() << endl;
                }
                else
                    return;
            }
        }
        for (auto &it : Mana_Tec)
        {
            if (it.getTecID() == user_id)
            {
                cout << "Money: $" << it.getMoney() << endl;
                cout << "Enter '1' to top up your wallet"
<< endl;
```

```
                cout << "Enter '0' to continue" << endl;
                cin >> t;
                getchar();
                if (t == 1)
                {
                    cout << "Money:$";
                    cin >> cash;
                    getchar();
                    it.addMoney(cash);
                    cout << "Finish, now your money: $" <<
it.getMoney() << endl;
                }
                else
                    return;
            }
        }
    }
    void USR_f3() // pay
    {
        double need_pay;
        for (auto &it : Mana_Stu)
        {
            if (it.getStuID() == user_id)
            {
                need_pay = (it.used_electric - 15) * 0.5 + (it.used_water - 10) * 0.7;
                if (need_pay > 0)
                {
                    cout << "You need to pay: $" << need_pay << endl;
                    if (it.getMoney() >= need_pay)
                    {
                        cout << "Your bill have already paid" << endl;
                        it.addMoney((-1) * need_pay);
                        cout << "Balance: $" << it.getMoney() << endl;
                        // do clear energy bill
                        it.used_electric = 0;
                        it.used_water = 0;
                    }
                    if (it.getMoney() < need_pay)
                    {
                        cout << "Please check your balance,
remember to pay your bill" << endl;
                        cout << "Balance: $" <<
it.getMoney() << endl;
                        cout<<endl;
                    }
                }
            }
        }
        for (auto &it : Mana_Tec)
        {
            if (it.getTecID() == user_id)
            {
                need_pay = (it.used_electric - 15) * 0.5 +
(it.used_water - 10) * 0.7;
                if (need_pay > 0)
                {
                    cout << "You need to pay: $" << need_pay
<< endl;
                    if (it.getMoney() >= need_pay)
                    {
                        cout << "Your bill have already
paid" << endl;
                        it.addMoney((-1) * need_pay);
                        cout << "Balance: $" <<
it.getMoney() << endl;
                        it.used_electric = 0;
                        it.used_water = 0;
                    }
                    if (it.getMoney() < need_pay)
                    {
                        cout << "Please check your balance,
remember to pay your bill" << endl;
                        cout << "Balance: $" <<
it.getMoney() << endl;
                    }
                }
            }
        }
    }
    void USR_f4() // change shadow
    {
        string temp_pwd;
        for (auto &it : Mana_Stu) // stu
        {
            if (user_id == it.getStuID())
            {
                cout << "Enter current password:";
                getline(cin, temp_pwd);
                if (user_pw == temp_pwd)
                {
                    cout << "Enter your new Password:";
                    getline(cin, temp_pwd);
                    for (auto &it : accounts_students)
                    {
                        if (it.getUsername() == user_id) //
primary key
                        {
                            it.changePwd(temp_pwd);
                            cout << "new password: " <<
it.getPassword() << endl;
                            cout << "Finish!" << endl;
                        }
                    }
                }
            }
        }
        for (auto &it : Mana_Tec) // stu
        {
            if (user_id == it.getTecID())
            {
                cout << "Enter current password:";
                getline(cin, temp_pwd);
                for (auto &it : accounts_teachers)
                {
                    if (it.getUsername() == user_id)
                    {
                        it.changePwd(temp_pwd);
                        cout << "new password: " <<
it.getPassword() << endl;
                        cout << "Finish!" << endl;
                    }
                }
            }
        }
        transTxt();
    }
    void USR_f5() // exit ok
    {
        cout << "Logout" << endl;
        return;
    }
};

void init_someAc(Login &A) //
{
    A.accounts_administrator.push_back(Account_Management("admin", "admin", "admin", 21, "admin"));
    A.accounts_students.push_back(Account_Management("s20234751", "tt45789", "joe", 19, "4A"));
    A.accounts_students.push_back(Account_Management("s20231714", "c1s5730", "ddd", 19, "4B"));
    A.accounts_teachers.push_back(Account_Management("t20157697", "nie455", "meta", 31, "sport"));
    A.accounts_teachers.push_back(Account_Management("t1997", "a71559721", "ana", 25, "sport"));
}

int main()
{
    int identity;

    Login login_used;

    init_someAc(login_used);

    login_used.Build(); // build member info

    login_used.transTxt();

    login_used.index();

    return 0;

}
```