程序设计实验

学生选课信息管理系统

计算机科学与技术系

191220090

沈天杰

目录
content

需求分析

学生选课信息管理系统

需求分析

系统管理：用户登录与注册(**密码加密输入**)

- 管理员
  - 注销登录
  - 课程录入
  - 查看全部课程
    - 增加课程
    - 删除课程(**支持批量删课**)
    - 修改课程
    - 查看具体课程
      - 选课学生名单
      - **录入课程成绩**
      - 该课助教名单
      - **某一门课程特定助教名单下的学生名单**
  - **课程指选**
  - **批量增加学生**
- 学生
  - 查看全部课程
    - 选课
    - 退课
    - 报名助教
    - 选择助教
  - 注销登录
  - **修改密码**
  - 查看个人课表
    - **打印出担任助教的课程**
  - **查看成绩**

维护信息一致性(写函数读取文件存储在vector中以及更新文件)

提供良好的界面(**在控制台上用库函数布局**)

需求分析

用户分类
- 管理员
- 学生

必须功能
- 管理员登录 以及 学生登录与注册
- 课程的录入，增删，修改;查看全部/具体课程信息（管理员）
- 查看课程信息/个人课表，选退课，报名和选择助教（学生）

拓展功能
- 更贴近真实的界面(控制台界面布局，密码输入)
- 更完善的功能，如增加批量操作,增加成绩信息(管理员，学生更丰富的功能)

数据结构设计

需要处理的信息

课程信息

- 课程编号
- 课程名称
- 授课教师
- 容纳人数
- 目前已选
- 课程类型
- ……

学生信息

- 学生ID
- 学生密码
- 个人课程编号
- 个人助教
- ……

助教信息

- 报名助教的课程编号
- 报名助教的所有学生ID
- ……

信息从文件中来...

课程信息:


```
struct course
{
    std::string id;                 //课程编号
    std::string course_name;        //课程名称
    std::string teacher_name;       //授课老师
    int limit_number;               //容纳人数
    int select_number;              //目前已选
    std::string course_type;        //课程类型
};
```

学生&助教信息:


```
struct Student
{
    std::string student_id;      //学生ID
    std::string student_password;//学生密码
    std::string self_course_id; //个人课程编号
    std::string ass_stu_id;      //个人助教
    std::string score;           //个人成绩
};

struct AssTable
{
    std::string ass_course_id;       //报名助教的课程编号
    std::vector<string> student_ids;//报名助教的所有学生ID
};
```

信息在文件与代码间传输

随用随创建vector<T>，通过函数读取文件信息存入vector中，对vector进行完所需操作。若vector发生变化，再用对应函数把vector的更改更新对应文件。

eg.

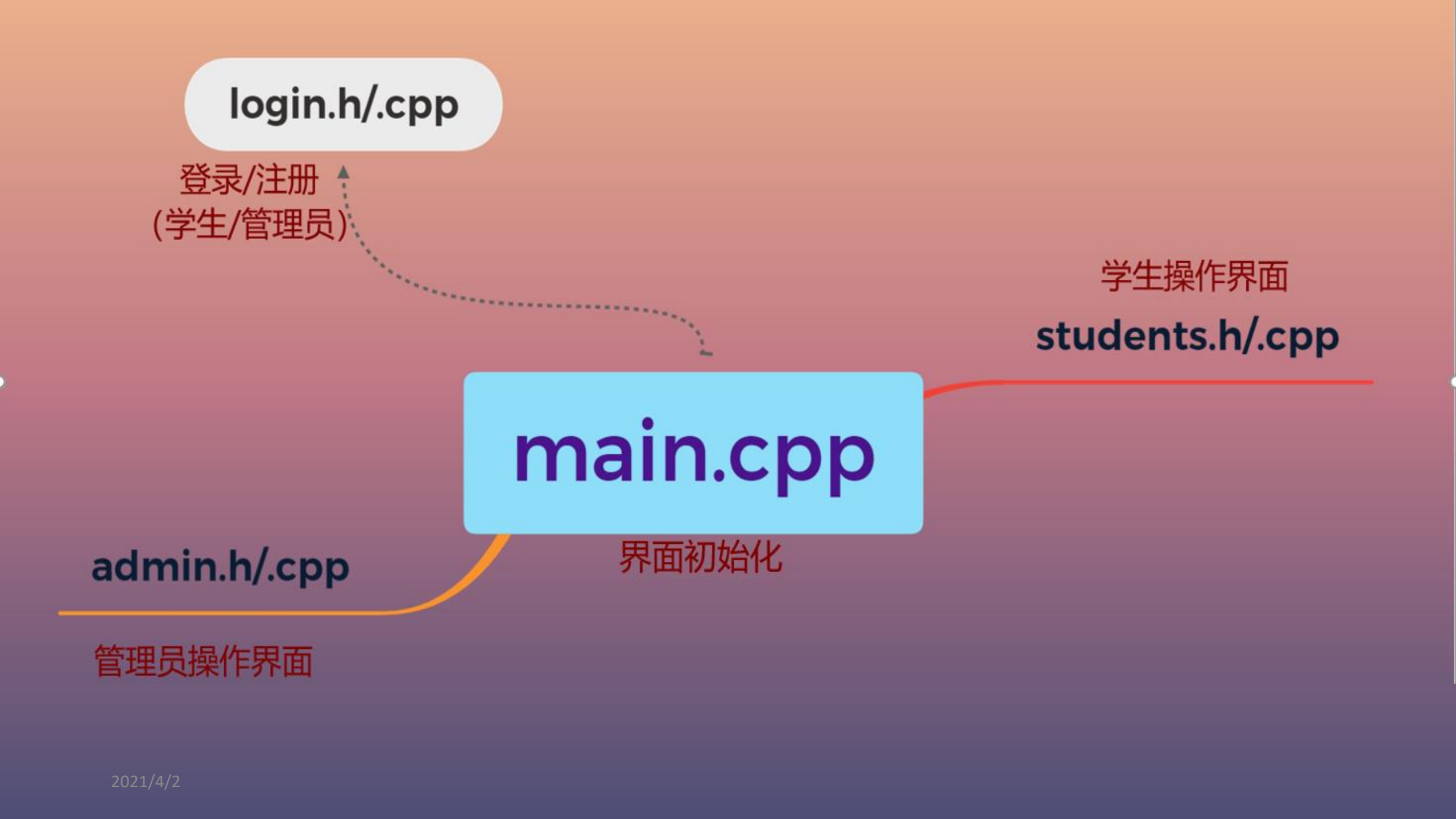
```
void delete_course()
{
    vector <course> course_information;
    storage(course_information, "currentcourse.txt");
    cout << "输入课程编号:";
    string courseId;
    cin >> courseId;
    vector<course>::iterator it;
    for(it=course_information.begin();it!=course_information.end();it++)
        if ((*it).id == courseId)
        {
            if ((*it).select_number > 0)
                cout << "\n删除课程失败，目前已有学生选课。 \n";
            else {
                course_information.erase(it);
                cout << "\n该课程已删除\n";
            }
            current(course_information);
            return;
        }
    cout << "\n删除失败，查无此课。。。\n";
}
```

创建vector，存放文件信息

对vector进行具体操作

vector更新文件信息

模块设计


login.h/.cpp
登录/注册
(学生/管理员)
学生操作界面
students.h/.cpp
main.cpp
界面初始化
admin.h/.cpp
管理员操作界面

login.h/.cpp

- 起始界面
- 输入判断(学生或管理员)
- 移动光标

相应实际代码展示:↓

```
void gotoxy(int x, int y);
void encrypt(string& password);//密码加密
int start_ui();
void log_in(int chosen_path);
void welcome_ui();
void return_menu();//返回主页面
```

main.cpp

- 起始界面
- 选择登录对象

```
#include"login.h"
#include<cstdlib>

int main()
{
    system("title 学生选课信息管理系统");
    system("mode con cols=110 lines=40");//窗口宽度高度
    welcome_ui();
    return 0;
}
```

界面初始化与登录　　main.cpp login.h/.cpp

- 以下用伪代码说明起始界面原理，功能1,2,3在实际代码中存在一定程度的相似:

```
switch(1 or 2 or 3)//选择功能1.学生登录2.学生注册3.管理员登录
string id, password;
cout << "输入学生/管理员ID:     ";
cin >> id;
cout << "输入学生/管理员密码:     ";
cin >> password;
用函数等对id和password验证是否合理(1.stu_log;2.stu_reg;3.Admin)  1.2.相应具体参见students模块
cout<<"登录成功\n";       \\如果合理
进入相应学生/管理员界面操作(stu_func("学生相应txt");/admin_func();)
```

界面初始化与登录　　main.cpp login.h/.cpp

- 以下为实际代码(试图增强容错性)

```
int start_ui()
{
    system("color FD");
    gotoxy(mid+8, 8);cout << "欢迎使用学生选课信息管理系统";
    gotoxy(mid,10); cout << LINE ;
    gotoxy(mid, 11);cout << " 1.学生登录      2.学生注册 3.管理员登录" ;
    gotoxy(mid, 12);cout << LINE ;
    gotoxy(mid+8, 14);cout << "请输入1或2或3进行操作:";
    int i,line=15;
    while( (!(cin >> i))|| (i > 3 || i < 1) ){  //防止输错字符导致错误 超范围错误
        if (cin.fail())
        {
            cin.clear();
            cin.get();
        }
        gotoxy(mid, ++line);cout << "操作失败，请正确输入1或2或3进行操作:";
    }
    return i;
}
```

读取操作选择
以及错误提示

界面初始化与登录　　main.cpp login.h/.cpp

- 以下为实际代码(试图增强容错性)

```
switch(1 or 2 or 3)//选择功能1.学生登录2.学生注册3.管理员登录
string id, password;
cout << "输入学生/管理员ID:     ";
cin >> id;
cout << "输入学生/管理员密码:     ";
cin >> password;
用函数等对id和password验证是否合理(1.stu_log;2.stu_reg;3.Admin)  1.2.相应具体参见students模块
cout<<"登录成功\n";       \\如果合理
进入相应学生/管理员界面操作(stu_func("学生相应txt");/admin_func();)
```

伪代码→实例↙↓↘

```
case 1:     //学生登录
{
    system("cls");
    string stu_id, stu_password;
    gotoxy(centre, 10);cout << "请输入学生ID: ";cin >> stu_id;
    gotoxy(centre, 11);cout << "请输入学生个人密码: ";encrypt(stu_pas
    string stucoursefile = stu_log(stu_id, stu_password);
    gotoxy(centre - 8, 14);cout << Border << "登录成功！ ！(3秒后跳转
    Sleep(jump_time);
    system("cls");
    fstream file;
    file.open("currentcourse.txt", ios::in);
    if (!file) {
        gotoxy(centre, 11);cout << "请管理员先录入课程信息！";
        system("pause");
        return_menu();
    }
    stu_func(stucoursefile);
    system("cls");
    change_password(stu_id, stu_password);
    break;
}
```

2021/4/2

```
system("cls");
string stu_id, stu_password,check, stucoursefile;
gotoxy(centre, 10);cout << "请输入学生ID进行注册: ";     cin >
gotoxy(centre, 11);cout << "请输入学生个人密码进行注册: ";
gotoxy(centre, 12);cout << "请再输入学生个人密码确认: ";
if(check==stu_password)
    stucoursefile = stu_reg(stu_id, stu_password);
else
{
    gotoxy(centre, 14);cout << "注册失败...";gotoxy(centre,
    system("pause");system("cls");welcome_ui();
}
gotoxy(centre - 8, 14);cout << Border << "注册成功！ ！登录成
//建立学生个人文件
ofstream out_file;
out_file.open(stucoursefile);
out_file.close();
Sleep(jump_time);
system("cls");
fstream file;
file.open("currentcourse.txt", ios::in);
if (!file) {
    gotoxy(centre, 11);cout << "请管理员先录入课程信息！";
    system("pause");
    return_menu();
}
stu_func(stucoursefile);
break;
```

```
system("cls");
string admin_id, admin_password;
gotoxy(centre, 10);cout << "请输入管理员ID: ";
cin >> admin_id;
gotoxy(centre, 11);cout << "请输入管理员密码:    ";
encrypt(admin_password);
while (admin_id != "Admin" || admin_password != "admin") {
    cout <<endl<<"\t\t"<< Border << "登录失败. ." << Border <<
    system("pause");
    system("cls");
    gotoxy(centre, 10);cout << "请输入管理员ID:   ";
    cin >> admin_id;
    gotoxy(centre, 11);cout << "请输入管理员密码:     ";
    cin >> admin_password;
}
gotoxy(centre - 8, 14);cout << Border << "登录成功！ !(3秒后跳转
Sleep(jump_time);
system("cls");
admin_func();
break;
```

管理员/学生界面操作 admin.h/.cpp students.h/.cpp

- 以下用伪代码说明操作界面原理，成功登录或注册后执行相应的func()函数进行交互:

```
1  func();      //分为admin_func()和stu_func("学生个人文件")
2  打印欢迎界面;
3  输入操作对应的数字1-7;
4  1为退出登录;
5  while(不输入1){          //执行2-7操作
6  switch (choice)      //2-7功能由相应函数各自实现
7      case2 3 4 5 6 7...}
```

admin.h/.cpp

- 1.总界面函数
- 管理员操作界面(含功能1注销登录)　　void admin_func();
- 2.功能函数
- 录入信息(功能2)　　调用storage和current(只录入一次，创建currentcourse.txt)
- 查看课表(功能3)　　void print_schedule(vector <course> course_information);
- 课程增删改(功能4,5,6)　　void add_course();
  void delete_course();
  void change_course();
- 查看具体课程(功能7)　　void detail_course();
- 3.辅助函数
- 打印功能管理员选择　　void print_title();
- 将文件中数据存入vector中　　void storage(vector <course>& course_information);
- 创建current_course.txt　　void current(vector <course>& course_information);
- 某门课学生/助教清单（写在students.cpp中的辅助函数）实现功能7
- void stu_list(string courseID);　　void ass_list(string courseID);

admin.h/.cpp

```
void admin_func()
{
begin:
    cout << border << endl << "1.注销登录 2.录入课程信息 3.查看全部课程信息 4.增加课程 5.删除课程 6.修改课程 7.查看具体课程\n" << border
    int choice;
    while (!(cin >> choice)) { ... }
    if (choice == 1) { ... }
    if (choice == 2) { ... }
paths:
    while (choice > 1 && choice < 8)
    {
        switch (choice)
        {
        case 2:cout<<"\n    信息已录入~ "<<endl; break;
        case 3: {vector<course> schedule;storage(schedule, "currentcourse.txt");print_schedule(schedule);break;}
        case 4:add_course();break;
        case 5:delete_course();break;
        case 6:change_course();break;
        case 7:detail_course();break;
        default:break;
        }
        cout <<endl<< border << endl << "1.注销登录 2.录入课程信息 3.查看全部课程信息 4.增加课程 5.删除课程 6.修改课程 7.查看具体课程\n"
        cin >> choice;
        if (choice == 1) { ... }
    }
    while (choice > 7 || choice < 1) { ... }
}
```

功能1，2

管理员操作界面

功能3-7

Students.h/.cpp

- 1.总界面函数
- 学生操作界面(含功能1注销登录)`void stu_func(string stucoursefile);`
- 登录/注册学生账号 `string stu_log(string stu_id, string stu_password);`
  (在login.cpp中调用) `string stu_reg(string stu_id, string stu_password);`
- 2.功能函数
- 查看课表(功能2) 复用admin.cpp中 print_schedule 函数
- 课程选退(功能3,5) `void select_self_course(const string stucoursefile);`
  `void delete_self_course(const string stucoursefile);`
- 查看个人课表(功能4) `void self_schedule(const string stucoursefile);`
- 报名/选择助教(功能6,7) `void apply_ass(const string stucoursefile);`
  `void select_ass(const string stucoursefile);`
- 助教退课告知(功能5，7) `void ass_existOrNot(const string stucoursefile);`
- 选课不足提示(功能4，5) `void course_num(const vector<Student> stu_self);`

Students.h/.cpp

- 3. 辅助函数
- a)处理文件函数：用vector存储文件信息，方便数据操作
- 读取学生总文件　　void read_stu_txt(vector<Student>& stu_information);
- 存储学生个人文件　　void store_self(vector<Student>&stu_self,string stucoursefile);
- 读取助教文件　　void read_ass_txt(vector<AssTable>& ass_information);
- 更新助教文件　　void current_ass(vector<AssTable> ass_information);
- b)完备基本功能函数
- 打印功能学生选择　　void print_stu_title();
- 排序个人文件　　void sort_course(const string stucoursefile);
- 字符串分割函数　　vector<string> Mysplit(string str,string pattern);
- 自定义student大小比较　　bool comp(const Student s1,const Student s2);
- 匹配课程编号和课程名称　　void courseID_map(couMap& course_map);

Students.h/.cpp

```
void stu_func(const string stucoursefile)
{
    ass_existOrNot(stucoursefile);//提示助教是否退课
    cout << border << endl
        << "1.注销登录 2.查看课程信息 3.选课 4.查看个人课表 5.退课 6.举手报名助教 7.选择个人助教\n"
        << border <<endl<< endl << "请选择所需要的功能: ";
    int choice;
    while (!(cin >> choice)) { ... }
    if (choice == 1) { ... }
    int apply_ass_times = 1;
paths:
    while (choice > 1 && choice < 8)
    {
        switch (choice)
        {
        case 2: {vector<course> schedule;storage(schedule, "currentcourse.txt");print_schedule(schedule);break;}
        case 3:select_self_course(stucoursefile);sort_course(stucoursefile);break;
        case 4:self_schedule(stucoursefile);break;
        case 5:delete_self_course(stucoursefile);sort_course(stucoursefile);break;
        case 6: if (apply_ass_times < 3) { apply_ass(stucoursefile);apply_ass_times++; }
                else cout << "已报名两门助教，无法再次报名!\n";break;
        case 7:select_ass(stucoursefile);break;
        default:break;
        }
        cout <<endl<< border << endl
            << "1.注销登录 2.查看课程信息 3.选课 4.查看个人课表 5.退课 6.举手报名助教 7.选择个人助教\n"
            << border << endl << endl << "请选择所需要的功能: ";
        cin >> choice;
        if (choice == 1) { ... }
    }
    while (choice > 7 || choice < 1) { ... }
}
```

学生操作界面

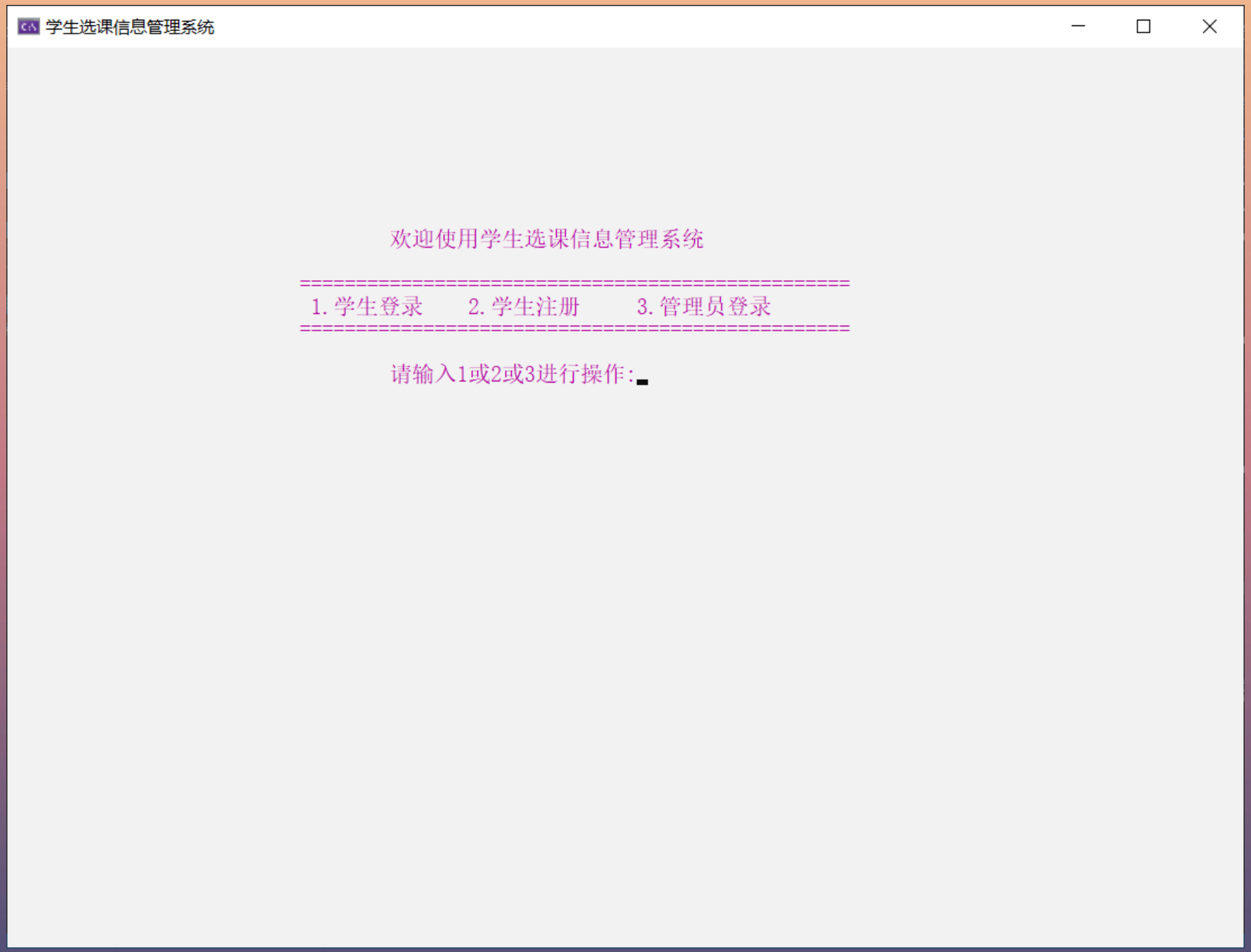
界面设计

最初仿照PPT制作的界面

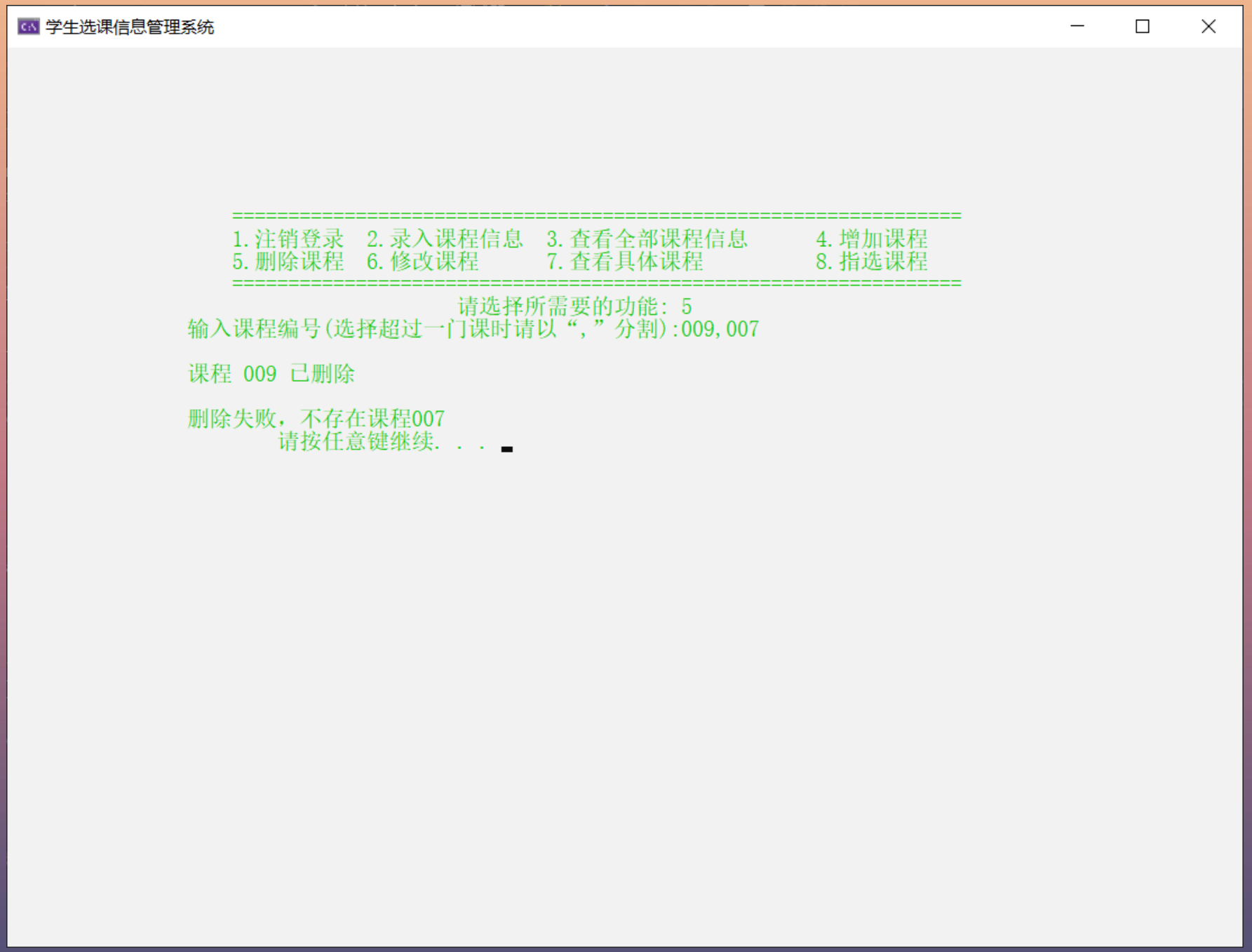
```
================================================
1.学生登录        2.学生注册        3.管理员登录
================================================

请输入1或2或3进行操作:   a
操作失败，请正确输入1或2或3进行操作:4
操作失败，请正确输入1或2或3进行操作:3
输入管理员ID:    Admin
输入管理员密码: admin
**********    登录成功！！     **********

=============================================================================================
1.注销登录 2.录入课程信息 3.查看全部课程信息 4.增加课程 5.删除课程 6.修改课程 7.查看具体课程
=============================================================================================

请选择所需要的功能:
```

```
================================================
1.学生登录        2.学生注册        3.管理员登录
================================================

请输入1或2或3进行操作:   2
输入学生ID进行注册: 191220090
输入学生个人密码进行注册: 1234
**********    注册成功！！登录成功！  ！**********

=============================================================================================
1.注销登录 2.查看课程信息 3.选课 4.查看个人课表 5.退课 6.举手报名助教 7.选择个人助教
=============================================================================================

请选择所需要的功能: 1
Log Out !!

================================================
1.学生登录        2.学生注册        3.管理员登录
================================================
```

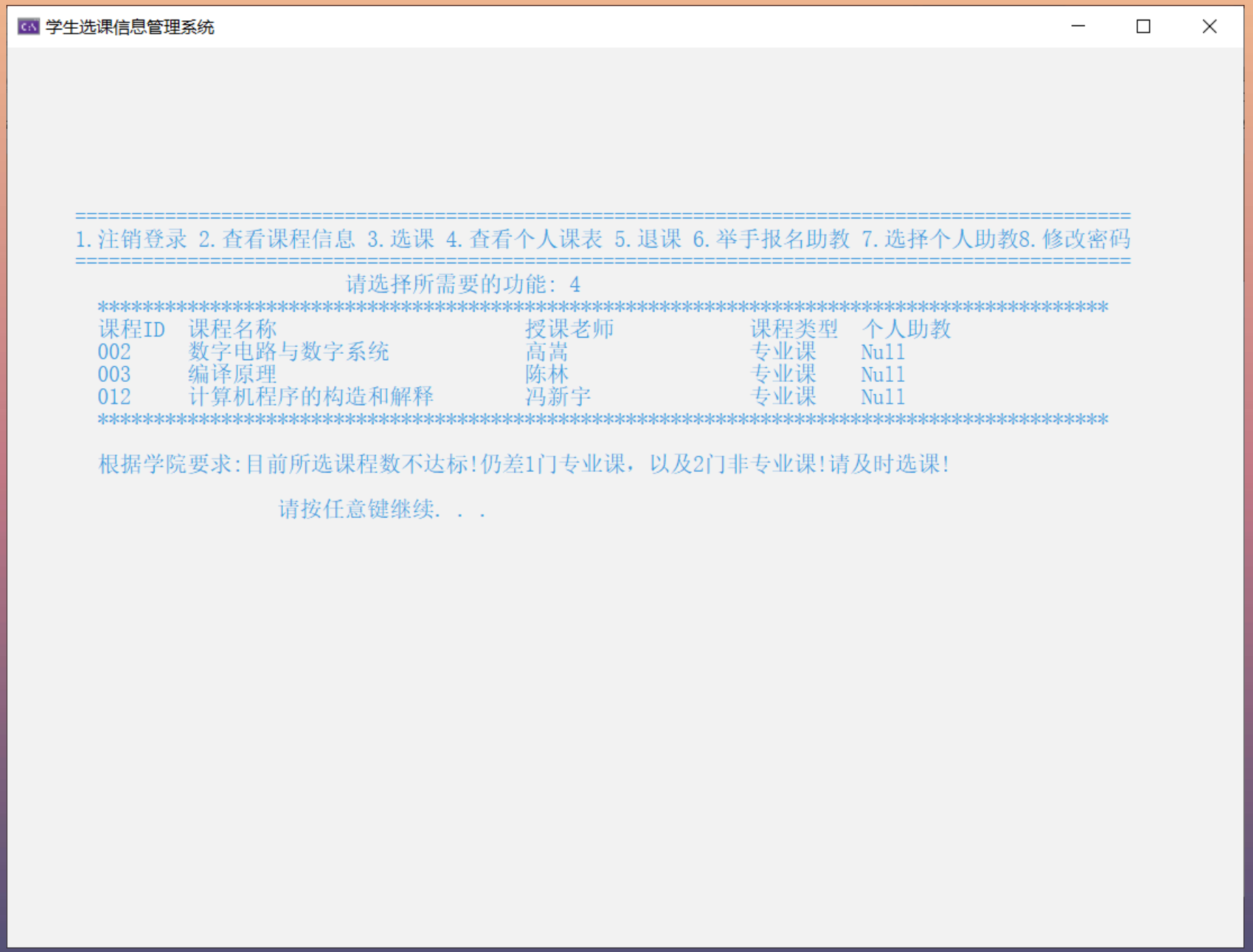
登录界面


学生选课信息管理系统
欢迎使用学生选课信息管理系统
==========
1.学生登录 2.学生注册 3.管理员登录
==========
请输入1或2或3进行操作:

管理员界面


学生选课信息管理系统
1. 注销登录 2. 录入课程信息 3. 查看全部课程信息 4. 增加课程
5. 删除课程 6. 修改课程 7. 查看具体课程 8. 指选课程
请选择所需要的功能: 5
输入课程编号(选择超过一门课时请以“,”分割):009,007
课程 009 已删除
删除失败，不存在课程007
请按任意键继续. . .

学生界面


学生选课信息管理系统
1.注销登录 2.查看课程信息 3.选课 4.查看个人课表 5.退课 6.举手报名助教 7.选择个人助教8.修改密码
请选择所需要的功能: 4
课程ID 课程名称 授课老师 课程类型 个人助教
002 数字电路与数字系统 高嵩 专业课 Null
003 编译原理 陈林 专业课 Null
012 计算机程序的构造和解释 冯新宇 专业课 Null
根据学院要求:目前所选课程数不达标!仍差1门专业课，以及2门非专业课!请及时选课!
请按任意键继续. . .

拓展功能

1.布局

- 使用cpp中移动光标函数+system()函数+Sleep()函数etc排版控制台的窗口。

```
void gotoxy(int x, int y)//光标移动
{
    HANDLE handle = GetStdHandle(STD_OUTPUT_HANDLE);
    COORD coord;
    coord.X = x;
    coord.Y = y;
    SetConsoleCursorPosition(handle, coord);
}
```

调用库函数功能
system(“pause”);　　停顿
system(“cls”);　　清屏
system(“color 03”);　颜色
sleep(jump_time);停顿时间

1.光标移动分别居中输出

2.一些新增操作

- 1.登录密码**不显示明文**(解决 回删 的bug)　　任意键盘输入均会以*打印，可回删
- 2.学生注册时密码**确认两次**

a)**管理员功能新增**

- 3.指选课程(某门课选课为0时且可以容纳人数时 管理员**批量导入**全体学生指定选该门课)
- 4.查看某门课程详细信息后可**录入成绩**，同时相应学生**文件更新成绩**
- 5.查看助教详细信息后可以进一步查看某一门课程特定**助教名单下的学生名单**
- **6.支持批量删课**(若删除课程数目大于1，课程间以英文半角“,”间隔)
- 7.批量**建立num个学生账号**，默认从学生文件最后一个人学号开始，若无则191220001开始,默认密码123

b)**学生功能新增**

- 7.学生增加**修改密码**功能
- 8.查看个人课表时若已担任助教打印出**担任助教的课程**
- 9.学生可以**查看自己课程成绩**

3.一些优化(可能不算拓展)

- 1.鲁棒性 如:

对输入数字合法性判断两例

```
while( (!(cin >> num))|| (num > 2 || num < 1)) {    //防止输错字符导致错误 超范围错误
    if (cin.fail())
    {
        cin.clear();
        cin.get();
    }
    cout << "操作失败，请正确输入1或2进行操作:";
}
```

```
try {//容错
    stoi(cou_infor[2]);
    stoi(cou_infor[3]);
}
catch (std::invalid_argument&) {
    cout << "\t\t\t课程上限人数和目前已选未正确输入数字\n";
}
int limited_num = stoi(cou_infor[2]);
```

- 2.函数尽可能只执行功能，不打印提示语句，使得一个功能函数在不同场合合理调用（实现了一部分）

用户手册

2021/4/2

用户手册

- 1.基本操作与PPT所给规范相同，输入模式按照题目PPT所示，在此不赘述。
- 2.学生个人文件微小变化，增加成绩一栏，与助教之间tab分割。
- 3.进入某个功能或者退回到主页面时候 页面切换 有3秒时间。
- 4.学生注册时需要输入两次密码确认。
- 5.学生登录成功后可以选择修改密码，修改成功后请重新登录。
- 6.指选功能：管理员端可统一导入某些必选课（如 形势与政策）。
  （在无人选此门课以及课程容纳人数允许的情况下）
- 7.查看具体课程（管理员功能7）后， 可继续选择查看这名助教的学生。
- 8.查看具体学生（管理员功能7）后， 可录入学生成绩，学生端可查看已录入的成绩。
- 9.管理员端可批量建立一批学生文件。

- 注1：所有标点符号均为**英文**输入法下的标点,并且全是半角符号。文件中除了换行符和逗号以外没有多余的空格，文件的每一行表示一条信息。
- 注2：请根据界面提示信息操作，尤其错误操作之后。