日程管理系统2.0



**开发文档**

组名：天码行空

组长：符泽林

组员：邓文豪

冉弩



**目录：**

1. 需求分析········································2
2. 详细分析········································3
3. 详细设计········································4
4. 技术难点及解决方案······························10
5. 界面展示········································11
6. **需求分析**

1.系统设置两个端口：教师端和学生端，老师可以设置时间区间给学生安排任务，并实时查看，学生自己可以自主安排任务。

2.初次进入系统后用户选择端口并注册账号。

3.注册账号后，用户进入系统时选择端口并进行登录，登录后可以进行日程管理，学生自我管理，教师管理学生，如果未登录系统会给出提示。

4.将日历和备忘录结合，以图表的形式展现给用户，会将初始时间、截止时间、日程内容、完成状态的相应内容展现出来。

5.系统包含学生设置任务、教师安排任务、任务完成状态的显示、教师和学生实时查看的功能。

6.学生完成任务后可以设置状态为完成，同时教师端可以查看相应状态。了解同学学习进度。学生和教师随时查看当前日程状态有助于规划时间。

7.用户可以随时修改和查询信息，包括查询日程内容、修改日程内容、修改账户密码。

8.系统权限分明，教师拥有查看所有学生所有日程的权限，而学生只有查看本人所有日程的权限，并且教师拥有选定任意学生安排任务的权限，而学生本人只能新建和修改自己的日程。

1. **详细分析**

1.本系统初衷为方便学生管理自己任务，同时教师可以监督学生，所以设置两个端口，教师端口权限高于学生端口，教师端可以为学生添加任务。同时教师可以查看所有学生完成情况，学生只能查询自己。

2.为了区分每一个学生的日程，方便多人使用，需要设置账号注册和登陆功能，同时还可以保证每一个学生隐私性。也方便了信息的查询修改等。

3.用户进行的操作有很多操作，如修改信息，添加日程等，都是需要保存以待下一次使用，而不是一次性使用，所以需要添加文件存储功能，保证信息不会遗失，每一次的操作都有效。

4.学生可能需要查看自己当前的任务完成情况，如是否有剩余未完成的任务，教师也需要可以随时查看学生的完成情况，便于督促学生学习，所以需要两个端口都有查看学生日程功能。

5.学生查看自己日程时，往往会因为日程繁杂无法准确记忆，故每完成一个任务学生可以选择将任务标记为已完成。有时候具有特殊情况，也可以给日程添加一个备注。

6.日程管理较为隐私，所以应该设置权限等级，防止学生信息泄露。学生权限只能对自己账号进行操作，教师端口可以对所有管理的学生进行查看和操作。

7.日程随时可能有变动，所以系统需要有添加、删除和修改的功能。

1. **详细设计**

1.下图为系统功能模块图

2. 数据结构

学生结构体数组和教师结构体数组

struct student

{

Wchar\_t szName[STRLENGTH];//定义学生用户名

Wchar\_t password1[NUMLENGTH];//定义学生账户密码

Wchar\_t password2[NUMLENGTH];//定义学生二次输入密码

int nSerialNum;//定义学生序号(从1开始)

int nManagement=0;//定义学生设定的日程事件数

Wchar\_t Management[10][MANAGELENGTH];//定义每个学生设定的具体日程（包括时间） 设定一个学生最多设定10个事件

Wchar\_t FinishStatus[10][MANAGELENGTH];//定义每个学生设定的日程的完成状态

}stu[NUMBER];

/\*定义教师的结构体数组\*/

struct teacher

{

Wchar\_t szName[STRLENGTH];//定义教师用户名

Wchar\_t password1[NUMLENGTH];//定义教师账户密码

Wchar\_t password2[NUMLENGTH];//定义教师二次输入密码

int nSerialNum;//定义老师序号(从1开始)

}teacher[NUMBER];

3. 全局变量

/\*全局变量\*/

int nStudentNum=0;//注册的学生编号（从0开始）（同时也代表着注册人数）

int nTeacherNum=0;//注册的教师编号 （第一个注册的用户编号为0）

int nStuFlag=0;//定义判断学生是否登录的标志变量

int nTeacherFlag=0;//定义判断教师是否登录的标志变量

int studentnum;//定义登录的学生编号

int teachernum;//定义登录的教师编号

4.函数量

18个函数

/\* 一级主菜单选项 （学生端、教师端、退出系统）\*/

void Select\_MainMenu();

/\*二级菜单\*/

//二级菜单学生端目录显示

void OutputStu\_SecondMenu();

//二级菜单：学生创建账户

void Stu\_Register();

// 二级菜单：学生登录

int Stu\_Logon();

//二级菜单：学生修改密码

int Stu\_ResetPassword();

//二级菜单：文件保存

int SaveFile();

//二级菜单：学生日程管理

void Stu\_ScheduleManagement();

//二级菜单教师端目录显示

void OutputTeacher\_SecondMenu();

//二级菜单：教师创建账户

void Teacher\_Register();

//二级菜单：教师登录

int Teacher\_Longon();

//二级菜单：教师修改密码

int Teacher\_ResetPassword();

//二级菜单：教师对学生日程管理

void Teacher\_ScheduleManagement();

/\*二级菜单结束\*/

/\*三级菜单\*/

//三级菜单：新建日程

void CreateSchedule();

//三级菜单：修改日程

void ResetSchedule();

//三级菜单：完成状态

void FinishedCondition();

//三级菜单：设定学生日程

void SetStuSchedule();

//三级菜单：修改学生日程

void ResetStuSchedule();

//三级菜单：查看学生完成状态

void CheckFinishedCondition();

/\*三级菜单结束\*/

5.功能

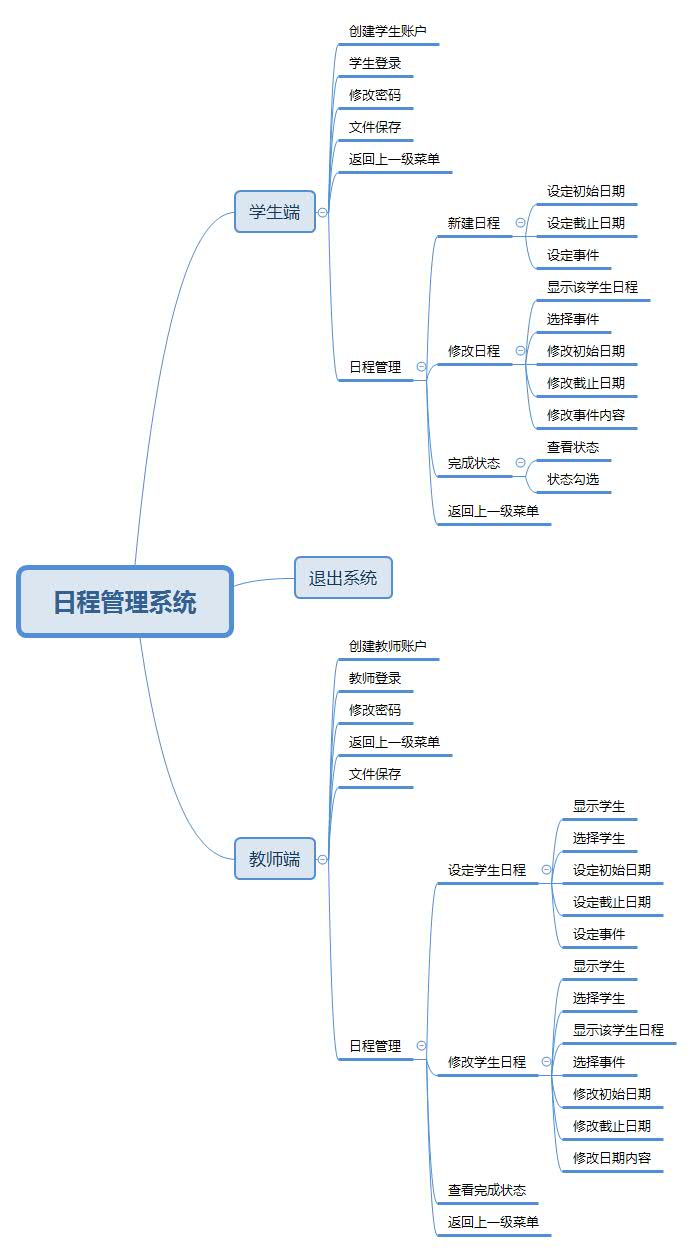
学生账户注册登录

教师账户注册登录

以类似图表的形式输出展示注册的学生的具体日程和完成状态

教师可以进行查看和对学生日程管理

6.系统功能模块图



7.开发说明

开发环境：VS2017

图形库：EasyX\_20190415(beta)

编码字符集：unicode

因此需要新建项目后更改字符集为unicode编码

将图片资源导入与.CPP同目录下的同名文件中

8．主要用到的图形库函数

Outtextxy（）输出文字函数

InputBox（）以弹窗形式让用户输入的函数

MessageBox （）以弹框形式提醒用户进行交互的函数

Loadimage（）和putimage（）导入并显示图片的函数

鼠标交互相关函数

1. **技术难点及解决方案**
   * + 1. **技术难点1**

函数参数多且繁杂

**解决方案**：将很多变量设置为全局变量，对于这种小型系统，此方法比较快捷简单，缺点是不适用于大型系统，因为全局变量可以在任意函数中进行更改，因此可能会带来一些麻烦。

* + - 1. **技术难点2**

学生信息管理的复杂性

**解决方案：**采用结构体数组，因为一个学生对应多个日程，而每个日程又包含多种信息，所以采用多个二维字符数组来保存多个日程的特定信息，如二维数组1保存所有日程的初始时间，二维数组2保存所有日程的截止时间，以此类推，如此一来便解决了多学生、多日程、多信息的问题了。

* + - 1. **技术难点3**

用Eaxyx图形库提供的函数导入设计好的用户界面后，无法用常规方法输出文字

**解决方案：**用InputBox（）函数进行弹窗提示输入

* + - 1. **技术难点4**

鼠标点击特定按钮进行界面切换

**解决方案：**做出按钮变色的界面，通过软件找到按钮所在的区域，利用图形库中的鼠标交互的函数判断鼠标点击动作所在区域，找到对应区域后进行界面切换，达到鼠标点击按钮进行界面切换的效果

* + - 1. **技术难点5**

Eaxyx图形库提供的输出函数只能输出字符数组的内容，无法直接将int型变量直接输出**解决方案：**通过if语句判断数字，再将数字存入字符数组，将字符数组进行输出**五、界面展示**



