

扩展功能说明

1、中断信号捕获与处理

- 当用户从键盘输入程序终止信号（如“CTRL + C”）后，主进程捕获并调用 `signalhandler` 函数，打印出“saving”，并自动退出程序。
- 程序结束会调动默认 `AgendaUI` 析构函数，从而 `private data member AgendaService` 类的对象被销毁，从而 `AgendaService` 的 `private data member Storage` 类指针会被销毁，从而调用 `WriteToFile()` 写入文件进行保存。

```
void signalhandler(int signal){
    cout<<endl<<"Saveing..."<<endl;

    exit(signal); // saveAgenda
}

signal(SIGINT, signalhandler);
```

2、表单字段验证

- 通过匹配 `Regex` 进行邮箱验证，正则表达式格式按照国际域名规则要求：
- 1、域名注册查询只提供英文字母（a-z，不区分大小写）、数字（0-9）、以及“-”（英文中的连词号，即中横线），不能使用空格及特殊字符(如!、\$、&、?等)。

2、“-”不能用作开头和结尾

3、域名最长可达67个字节(包括后缀.com、.net、.org等)

```
#include <regex>

cin>>email;

regex pattern("([0-9A-Za-z\\-\\.]+)@([0-9a-z]+\\. [a-z]{2,3}(\\. [a-z]{2})?)");

while(!regex_match(email, pattern)){
    cout<<"Wrong input format, please input again."<<endl;
    cin>>email;
}
```

- 用户名不做特殊要求，可以包含空格
- 密码不能包含空格
- 手机号必须为11位数字

3、密码加密与密码回显暗文

```
#include <termios.h>
#include <unistd.h>

int mygetch(void){
    struct termios oldt, newt;
    int ch;
    tcgetattr( STDIN_FILENO, &oldt ); // 记录旧设置
    newt = oldt;
    newt.c_lflag &= ~( ICANON | ECHO );//设置不回显
    tcsetattr( STDIN_FILENO, TCSANOW, &newt ); // 设置新的
    ch = getchar();
    tcsetattr( STDIN_FILENO, TCSANOW, &oldt ); // 恢复旧的设置
    return ch;
}
```

1、使用tcgetattr和tcsetattr函数控制Terminal终端

函数原型：

```
int tcgetattr(int fd, struct termios *termios_p);

int tcsetattr(int fd, int optional_actions, const struct termios *termios_p);
```

tcgetattr函数用于获取与终端相关的参数。参数fd为终端的文件描述符，返回的结果保存在termios结构体中，该结构体一般包括如下的成员：

```
tcflag_t c_iflag;

tcflag_t c_oflag;

tcflag_t c_cflag;

tcflag_t c_lflag;

cc_t      c_cc[NCCS];
```

其中 c_lflag部分参数：

键值	说明
ECHO	显示输入字符
ECHOE	如果ICANON同时设置，ERASE将删除输入的字符，WERASE将删除输入的单词
ECHOK	如果ICANON同时设置，KILL将删除当前行
ECHOPR	如果ECHO和ICANON同时设置，将删除打印出的字符（非POSIX）
ECHONL	如果ICANON同时设置，即使ECHO没有设置依然显示换行符
XCASE	在ICANON和XCASE同时设置的情况下，终端只使用大写。如果只设置了XCASE，则输入字符将被转换为小写字符，除非字符使用了转义字符（非POSIX，且Linux不支持该参数）
ICANON	使用标准输入模式

```
string password_hide(){

    char pass[100];
    int len=0;
    char c;
    while(1){
        c=mygetch();
        if(c=='\n'){
            pass[len]='\0';
            cout<<endl;
            break;
        }
        pass[len]=c;
        len++;
        cout<<"*";
    }
    string s=pass;
    return s;
}
```

```

password=password_hide();
cout<<"[Please input your password again] ";
string s=password_hide();
while(s!=password){
    cout<<"Error,set your password again.\n";
    cout<<"[password] ";
    password=password_hide();
    cout<<"[Please input your password again] ";
    s=password_hide();
}

```

- 提供两次密码输入，以便确认无误

2、使用md5加密算法将密码转换为密文后保存到文件，因为不能解密，但每次相同字符串加密结果相同，所以在用户登录或注册时，先转换为密文后进行操作，其余文件操作等未改变。

4、优化UI

```

choice=getOperation();
if(choice=="q"&&m_username==""){
    quitAgenda();
    return;
}
while(!executeOperation(choice)){
    cout<<"Error input,please input again.\n";
    choice=getOperation();
}
}

```

- 对菜单以外的操作必须重新输入
- 此外还有邮箱和手机号码错误输入时的重新输入
- 以及日期格式不对或逻辑错误的重新输入