```
#include "network.h"

//***************************服务器数据库**************************************

//创建数据库用于保存账号密码
sqlite3 * Create_Sqlite(void)
{
    sqlite3 * datebase;
    int ret = sqlite3_open("chat.db", &datebase);
    if(ret != SQLITE_OK)
    {
        printf("数据库打开失败\n");
        return NULL;
    }
    printf("数据库打开成功\n");

    return datebase;
}

//注册时保存用户账号和密码
int Save_User(Msg *msg, sqlite3 *datebase)
{
    //打开表
    char *errmsg = NULL;
    char buf[100] = "create table if not exists user(NAME TEXT,Password TEXT,Sign_name
TEXT,primary key(NAME))";
    int ret = sqlite3_exec(datebase, buf, NULL, NULL, &errmsg);
    if(ret != SQLITE_OK)
    {
        printf("打开表失败\n");
        printf("sqlite3_exec: %s\n",errmsg);
        return -1;
    }
    printf("打开表成功\n");

    //判断表中是否已存在相同账号名
    char **resultp = NULL;
    int nrow,ncolumn;
    char *sq1 = "select NAME from user";
    ret = sqlite3_get_table(datebase, sq1, &resultp, &nrow, &ncolumn, &errmsg);
    if(ret != SQLITE_OK)
    {
        printf("数据库操作失败\n");
        printf("sqlite3_get_table: %s\n",errmsg);
        return -1;
    }

    //插入数据
    sprintf(buf,"insert into user values('%s', '%s', '你的程序员小可爱已上线')",msg->fromname, msg->password);
    ret = sqlite3_exec(datebase, buf, NULL, NULL,&errmsg);
    if(ret != SQLITE_OK)
    {
        printf("插入数据失败\n");
        printf("sqlite3_exec: %s\n",errmsg);
        return -1;
    }

    return 0;
}

//登录时检查用户账号和密码
int Entry_User(Msg *msg, sqlite3 *datebase)
{
    //打开表
    char *errmsg = NULL;
    char buf[100] = "create table if not exists user(NAME TEXT,Password TEXT,Sign_name TEXT,primary key(NAME))";
    int ret = sqlite3_exec(datebase, buf, NULL, NULL, &errmsg);
    if(ret != SQLITE_OK)
    {
        printf("打开表失败\n");
        printf("sqlite3_exec: %s\n",errmsg);
        return -1;
    }
    printf("打开表成功\n");

    //判断账号密码
    char **resultp = NULL;
    int nrow,ncolumn;
    char *sq1 = "select * from user";
    ret = sqlite3_get_table(datebase, sq1, &resultp, &nrow, &ncolumn, &errmsg);
    if(ret != SQLITE_OK)
    {
        printf("数据库操作失败\n");
        printf("sqlite3_get_table: %s\n",errmsg);
        return -1;
    }
    int i;
    int j = 0 ;                    //表示密码不对
    int flag = 0;                    //表示账号不存在
    for(i=3; i<(nrow+1)*ncolumn ;i+=3)
    {
        if(strcmp(resultp[i], msg->fromname) == 0)
        {
            if (strcmp(resultp[i+1], msg->password) != 0)
            {
                j = 1;
            }
            flag = 1;
            break;
        }
    }


    ret = 0;
    if (flag != 1)
    {
        ret = -2;         //表示名字不存在
    }
    else if (j == 1)
    {
        ret = -3;         //表示密码不对
    }
    else
        strcpy(msg->signname,resultp[i+2]);       //保存个性签名

    sqlite3_free_table(resultp); //释放空间
    return ret;                  //表示登录成功
}

//修改数据库（个性签名)
int revise_sign_sqlite(Msg * msg)
{
    sqlite3 * datebase = Create_Sqlite();

    char *errmsg = NULL;
    char buf[200];
    sprintf(buf,"update user set Sign_name = '%s' where Name = '%s'",msg->signname,msg->fromname);
    int ret = sqlite3_exec(datebase, buf, NULL, NULL,&errmsg);
    if(ret != SQLITE_OK)
    {
        printf("修改数据失败\n");
        printf("sqlite3_exec: %s\n",errmsg);
        sqlite3_close(datebase);    //关闭表
        return -1;
    }

    printf("修改数据成功\n");
    sqlite3_close(datebase);

    return 0;
}

//修改密码(数据库)
int revise_password_sqlite(Msg * msg)
{
    sqlite3 * datebase = Create_Sqlite();

    char *errmsg = NULL;
    char buf[200];
    sprintf(buf,"update user set Password = '%s' where Name =
'%s'",msg->password,msg->fromname);
    int ret = sqlite3_exec(datebase, buf, NULL, NULL,&errmsg);
    if(ret != SQLITE_OK)
    {
        printf("修改数据失败\n");
        printf("sqlite3_exec: %s\n",errmsg);
        sqlite3_close(datebase);    //关闭表
        return -1;
    }

    printf("修改数据成功\n");
    sqlite3_close(datebase);

    return 0;
}

//***************************客户端数据库***************************************

//创建数据库用于保存聊天记录
sqlite3 * Create_user_sqlite(Msg * msg)
{
    sqlite3 * datebase;
    char name[23];
    sprintf(name ,"%s.db",msg->fromname);
    int ret = sqlite3_open(name, &datebase);

    return datebase;
```

```
}

//保存聊天记录到数据库中
void save_Chat(Msg *msg)
{
	sqlite3 * datebase = Create_user_sqlite(msg);
	//打开表
	char *errmsg = NULL;
	char buf[200] = "create table if not exists chat(时间 TEXT,发送者 TEXT,接收者 TEXT,内容 TEXT)";
	int ret = sqlite3_exec(datebase, buf, NULL, NULL, &errmsg);
	if(ret != SQLITE_OK)
	{
		printf("打开表失败\n");
		return ;
	}

	time_t t;
	t = time(&t);
	char time[100];
	strcpy(time,ctime(&t));
	int len = strlen(time);
	time[len-1] = '\0';		//去掉回车
	sprintf(buf,"insert into chat values('%s','%s','%s','%s')",time,msg->fromname,msg->localname,msg->msg);
	ret = sqlite3_exec(datebase, buf, NULL, NULL,&errmsg);
	if(ret != SQLITE_OK)
	{
		printf("插入数据失败\n");
		return ;
	}
	sqlite3_close(datebase);
}

//查看聊天记录(调用数据库)
void see_chat(Msg * msg)
{
	sqlite3 * datebase = Create_user_sqlite(msg);
	//打开表
	char *errmsg = NULL;
	char buf[200] = "create table if not exists chat(时间 TEXT,发送者 TEXT,接收者 TEXT,内容 TEXT)";
	int ret = sqlite3_exec(datebase, buf, NULL, NULL, &errmsg);
	if(ret != SQLITE_OK)
	{
        printf("打开表失败\n");
        return ;
    }
    //查看聊天记录
    char **resultp = NULL;
    int nrow,ncolumn;
    char *sq1 = "select * from chat";
    ret = sqlite3_get_table(datebase, sq1, &resultp, &nrow, &ncolumn, &errmsg);
    if(ret != SQLITE_OK)
    {
        printf("数据库操作失败\n");
        printf("sqlite3_get_table: %s\n",errmsg);
        return ;
    }
    int i;
    for(i=0; i<(nrow+1)*ncolumn ;i++)
    {
        if(i%4 == 0)
        {
            printf("\n");
            printf("%-25s",resultp[i]);
        }
        else
            printf("%15s",resultp[i]);
    }
    printf("\n");

    sqlite3_free_table(resultp); //释放空间

    sqlite3_close(datebase);

    sleep(5);
}
```