项目设计文档

# 一、总体设计

## 1.1通用消息结构体

### 1.1.1结构体定义

typedef struct mymsg{

char usr[20]; //用户名

char pri[10]; //权限 (admin / user)

char name[20]; //姓名

char sex[10]; //性别（"man"/"woman"）

char password[20]; //密码

char dept[20]; //部门

int cmd; //命令

int age; //年龄

int number; //工号 (数据库设置成 主键 自增量 注册时 修改信息时 无法修改)

int salary; //薪水

int inq\_mode; //查询模式

int inq\_cnt; //查询个数

int status; //返回

}msg\_t;

/\*定义一个默认的变量供客户端和服务器测试使用\*/

msg\_tmsg = {

.cmd         = LOG\_USR,

.usr[0]       = 'N',

.pri           = 'U',

.name[0]   = 'N',

.age          = 18,

.sex          = MAN,

.password[0]   = 'N',

.number       = 1,

.salary        = 10000,

.dept[0]       = 'N',

.inq\_mode     = ALL\_INFO,

.inq\_cnt       = 10,

.status        = 1,

};

### 1.1.2结构体分类描述

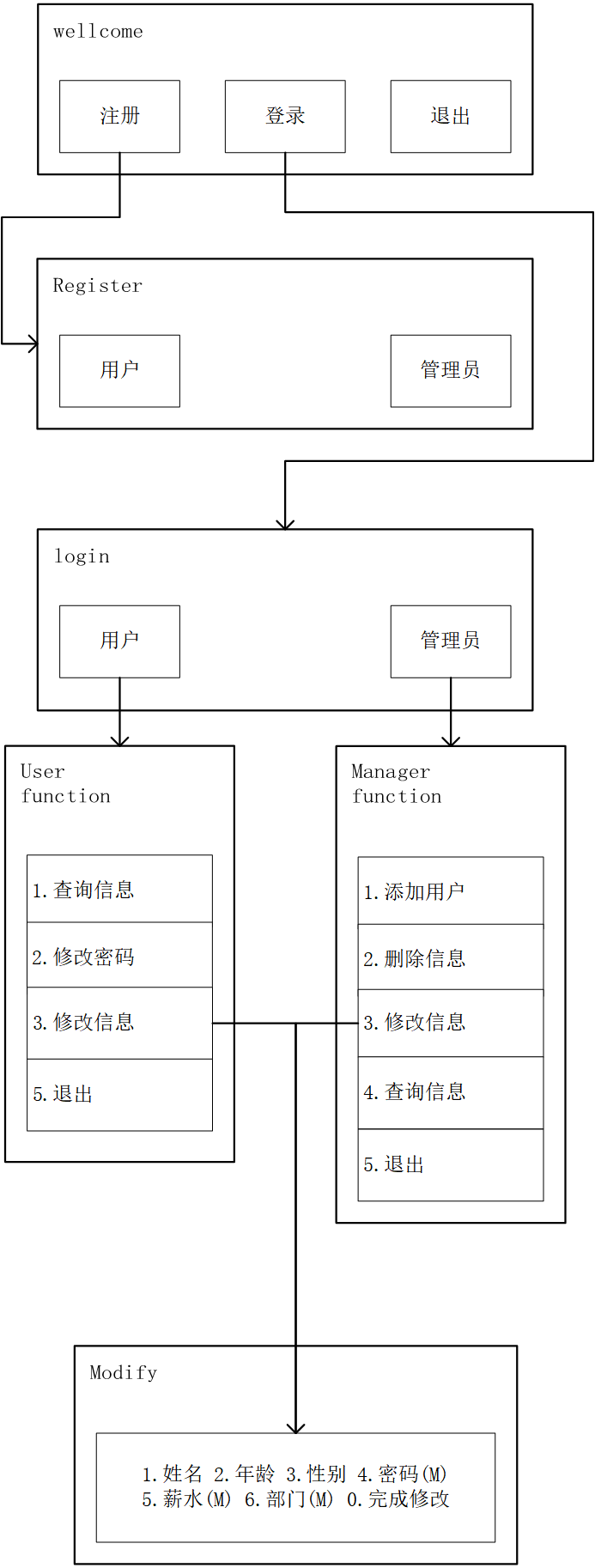
1. 指令类：cmd
2. 数据库类：usr、pri、name、age、
3. 查询模式位：inq\_mode、inq\_cnt(查询计数器)
4. 状态位：status

## 1.3指令描述！！！（写服务器需要注意）

|  |  |  |
| --- | --- | --- |
| REGISTER\_MANAGER | /\*注册管理员\*/ | 客户端发送的结构体msg中要包含注册的用户名和密码。  服务器需要保存用户名，管理员权限，密码这三个值，其它的可以不填或填默认值。注册成功需给客户端返回一个status为OK的消息结构体msg，失败返回NO。 |
| REGISTER\_USR | /\*注册用户\*/ | 客户端发送的结构体msg中要包含注册的用户名和密码。  服务器需要保存用户名，管理员权限，密码这三个值，其它的可以不填或填默认值。注册成功需给客户端返回一个status为OK的消息结构体msg，失败返回NO。 |
| LOG\_MANAGER | /\*管理员登录\*/ | 客户端发送的消息结构体msg中需要包含权限为管理员，用户名，密码三个数据。  服务器对msg中三种消息进行匹配，三者能在数据库中找到则返回给客户端一个status为OK的消息结构体msg，未匹配返回NO。 |
| LOG\_USR | /\*用户登录\*/ | 客户端发送的消息结构体msg中需要包含权限为用户，用户名，密码三个数据。  服务器对msg中三种消息进行匹配，三者能在数据库中找到则返回给客户端一个status为OK的消息结构体msg，未匹配返回NO。 |
| INQUIRE\_INFO | /\*查询信息\*/ | 客户端发送的消息结构体msg中需要包含至少权限，用户名两个信息。当权限为管理员的时候，还需要传送查询方式以及查询数量。  当查询者为管理员的时候 |
| VERIFY\_PASSWORD | /\*修改密码\*/ | 客户端发送的消息结构体msg需要填充用户名和修改的密码。  服务器根据接收到的msg修改用户密码，修改成功需给客户端返回一个status为OK的消息结构体msg，失败返回NO。 |
| VERIFY\_INFO | /\*修改信息\*/ | 客户端发送一个消息结构体，需要填充所有数据库中的信息。  usr: name sex age  服务器根据usr为依据修改用户自己的数据信息。 |
| ADD\_USR | /\*增加用户信息\*/ | 客户端输入用户信息到msg。  服务器根据msg中的信息将用户信息添加到数据库中。 |
| DELECT\_USR | /\*删除用户信息\*/ | 客户端只需填充msg的用户名数据。  服务器根据msg中的信息将该用户从数据库中删除。 |

# 二、客户端设计

## 2.1客户端页面设计



说明：由页面功能，可以分别将对应的函数写在相应的文件中。于是有wellcome.c、reg.c、login.c、usr\_function.c、manager\_function.c这几个文件，最后考虑功能实现函数有共通的功能，所以将具体的实现写在了applib.c中。