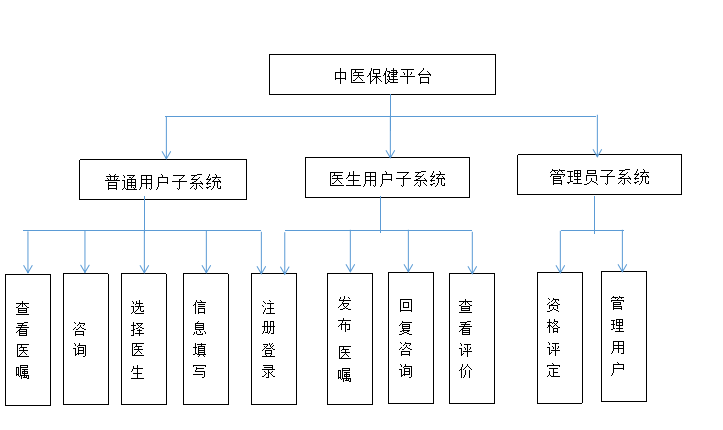
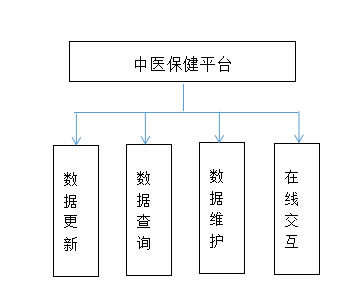
**中医保健平台的案例分析（系统设计部分）**

1. 系统设计分析
2. 总体设计
3. 系统的功能结构图设计

该平台包含三个子系统，即普通用户子系统、医生用户子系统、管理员子系统；而每个子系统都具有其相应的功能，其功能结构如下图所示。



而由功能结构图可知，该平台可分解成多个功能单一的模块，主要为包含三个模块，如下图所示。



每个模块的功能说明如下。

数据更新：主要实现管理员对用户的管理，用户注册，普通用户基本信息的录入、修改；医生用户发布的医嘱和评价的记录。

数据查询：主要实现普通用户信息的显示，医嘱的公布；医生用户的评价和资料的显示和查看。

数据维护：主要实现对数据库的数据进行维护，包括是否符合数据的输入、输出要求，避免数据的重复、丢失等错误发生。

在线交互：主要实现普通用户和医生用户以在线的方式，进行实时的交流。

1. 系统流程图设计

本平台的用户分为三个类型，即普通用户、医生用户以及管理员用户。普通用户通过注册登录，填写基本信息，浏览医生的基本信息及综合评价，选择医生进行在线咨询，在咨询结束后，查看医生发布的医嘱并发表评价；医生用户注册时，录入相关信息供管理员审核后，登录平台，回复发来的咨询信息，并发布咨询者医嘱并查看自我的评价；管理员主要根据医生的相关信息对医生用户的注册资格，进行审核，并维护平台环境，管理平台用户，即取消不良用户的注册资格。

该平台的数据流程图如下：

1. 数据库设计

数据库设计主要包括概念结构、逻辑结构设计和物理结构设计，而概念结构设计已在系统分析阶段通过E-R模型表示。以下将进行逻辑结构和物理结构的设计。

1. 逻辑结构设计

基于系统分析中的概念模型，在数据库的逻辑结构设计中，主要采用满足3NF的数据模型表示。其中，E-R图中的每个实体，以及实体与实体间的联系皆对应逻辑结构中的关系。为了便于区分各个关系表，将每个关系表命名为“表名拼音缩写”。以下展示该平台的数据库中部分主要关系表的定义及属性设置。

表4-1 普通用户登录表：g-user

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 是否主键 | 是否为空 | 说明 | 其他 |
| Uno | varchar(30) | 是 | not null | 用户名 | 唯一 |
| Paswrd | varchar(30) | 否 | not null | 密码 | 无 |
| Name | varchar(30) | 否 | null | 昵称 | 无 |
| Sex | bit | 否 | null | 性别 | 无 |
| Age | Varchar(15) | 否 | null | 年龄 | 无 |
| History | Varchar(100) | 否 | null | 病史 | 无 |

表4-2 问诊单内容表：ask

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 是否主键 | 是否为空 | 说明 | 其他 |
| No | int | 是 | Not null | 序号 | 自增、主键约束 |
| Uno | varchar(30) | 是 | Not null | 用户名 | 无 |
| Bpre | varchar(100) | 否 | null | 血压 | 无 |
| Slp | varchar(100) | 否 | null | 睡眠情况 | 无 |
| Temp | varchar(100) | 否 | null | 冷热情况 | 无 |
| Sweat | varchar(100) | 否 | null | 出汗情况 | 无 |
| Pain | varchar(100) | 否 | null | 疼痛情况 | 无 |
| Discomfort | varchar(100) | 否 | null | 不适情况 | 无 |
| Eye | varchar(100) | 否 | null | 眼部不适 | 无 |
| Cough | varchar(100) | 否 | null | 咳嗽情况 | 无 |
| Drink | Varchar(100) | 否 | null | 饮水情况 | 无 |
| Stime | varchar(100) | 否 | null | 大便频率 | 无 |
| Sshap | varchar(100) | 否 | null | 大便性状 | 无 |
| Scolor | varchar(100) | 否 | null | 大便颜色 | 无 |
| Pee | varchar(100) | 否 | null | 小便情况 | 无 |
| Pcolor | varchar(100) | 否 | null | 小便色质 | 无 |

表4-3 保健医嘱：healplan

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 是否主键 | 是否为空 | 说明 | 其他 |
| No | int | 是 | not null | 序号 | 自增 |
| Uno | varchar(30) | 否 | not null | 普通用户名 | 无 |
| Dno | varchar(30) | 否 | not null | 医生账号 | 无 |
| Sport | varchar(100) | 否 | null | 运动建议 | 无 |
| Food | varchar(100) | 否 | null | 饮食建议 | 无 |
| Cmedicine | varchar(100) | 否 | null | 中医建议 | 无 |
| Upadate | date | 否 | null | 更新时间 | 无 |

表4-4 服务评价：assess

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 是否主键 | 是否为空 | 说明 | 其他 |
| No | int | 是 | not null | 序号 | 自增 |
| Uno | varchar(15) | 否 | not null | 普通用户名 | 无 |
| Dno | varchar(15) | 否 | not null | 医生账号 | 无 |
| Content | varchar(100) | 否 | not null | 评价内容 | 无 |
| Time | date | 否 | null | 发表时间 | 无 |

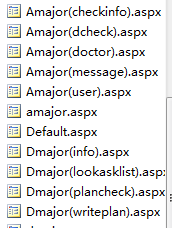
1. 物理结构设计
2. 代码设计方案

为了提高代码的可读性和可维护性，减少代码的冗余和复杂性，对代码的设计主要采用以下原则。

1. 代码的命名规则

该平台的前端界面涉及许多的界面设计工具，如Label、TextBox、Button等，界面设计工具的命名主要采用区间码的方式，采用“所处界面简写+工具类型简写+该工具输入内容的简写”，如登录界面中的密码输入框，则命名为logtpawd。

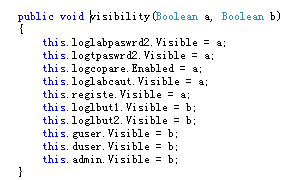
每个页面的命名主要采用“用户类型+（主要页面内容）”，如下图所示，Amajor（）表示管理员的相关界面，Amajor（checkinfo）表示管理员的审核界面。

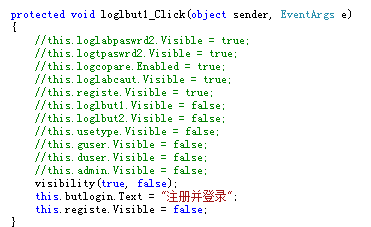


代码中相关函数的命名规则主要采用“实现功能+内容”，如：查询信息的函数名定义为selectinfo。

1. 代码模块化
2. 定义公共函数

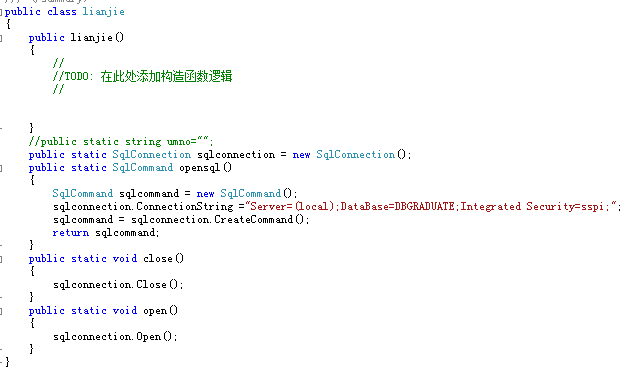
在平台操作过程中，同个页面中需要反复使用的功能主要通过定义相关的函数，并调用该函数实现相应功能，能有效的避免代码冗余，提高代码的可维护性。如下图的，“public void visibilty(Boolean a,Boolean b)”函数，定义了某些界面工具的可见性问题，而在同一页面的其他位置被反复的调用两次。



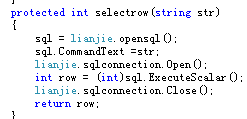


1. 定义类

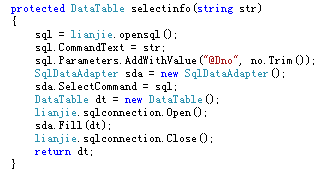
该平台需要反复的对数据库中的数据进行操作，如查询、插入、更新等，而每次的操作都需要与数据库建立连接，主要通过定义一个全局类并调用该类来实现对数据库的操作。该平台与数据库的连接类定义为：



Amajor页面，通过调用该类，实现查询功能。



Dmajor(plancheck)页面，通过调用该类查询功能。



1. 输入输出设计方案
2. 输出设计

已注册的普通用户，在登录平台后，以分主题的方式，分别显示用户的个人基本信息、问诊单信息、医嘱信息以及用户的评价记录。而普通用户通过相应的操作可对该信息进行修改、更新。

已注册的医生用户，以表格的方式，显示了个人信息、在诊病人、结诊病人以及评价；表头包括，用户名称、日期、内容三部分。

管理员主要以表格的方式，显示用户的名称、日期等基本情况。

1. 输入设计

新注册的普通用户的问诊单，主要以表格的格式，以多选题的方式进行填写。主要包括题目、选项。已注册的医生用户主要通过发布医嘱功能为用户发布医嘱，其中，包括普通用户名称以及医嘱内容，而医嘱内容主要包含三个方面，即饮食、运动、中医保健。

1. 程序模块说明
2. 数据查询模块

通过与数据库建立连接，定义相应的查询命令语句，根据查询条件，从数据库查询相应的数据信息，并返回查询结果。

protected DataTable selectinfo(int a,String one, String two)

{

sql = lianjie.opensql();

if (a == 1)

{

sql.CommandText = one;

sql.Parameters.AddWithValue("@Dno", this.lbdname.Text.Trim());

sql.Parameters.AddWithValue("@Uno", two.Trim());

}

else

{

sql.CommandText = one;

sql.Parameters.AddWithValue("@Dno", this.lbdname.Text.Trim());

sql.Parameters.AddWithValue("@YorN", two.Trim());

}

SqlDataAdapter sda = new SqlDataAdapter();

sda.SelectCommand = sql;

DataTable dt = new DataTable();

lianjie.sqlconnection.Open();

sda.Fill(dt);

lianjie.sqlconnection.Close();

return dt;

}

1. 数据更新模块

通过调用公共类，实现与数据库的连接，并输入相应的数据信息，更新数据中的数据，若数据更新失败，发出提示信息。

protected void butumcun\_Click(object sender, EventArgs e)

{

string[] str = new string[14];//采用数组存放选中选项字符创

str[0] = this.que1.Text.Trim();

str[1]=state(pan3);//调用state()函数遍历每个选项Panel中的CheckBox是否选中，

//连接选中项的字符串，并返回

str[2]=state(pan2);

str[3]=state(pan4);

str[4]=state(pan5);

str[5] = state(pan6);

str[6] = state(pan7);

str[7] = state(pan8);

str[8] = state(pan9);

str[9] = state(pan10);

str[10] = state(pan12);

str[11] = state(pan11);

str[12] = state(pan13);

str[13] = state(pan14);

DateTime time = System.DateTime.Now;

sql=opensql();//实例化Sqlcommand、Sqlconnection对象

sql.CommandText =

"insert into asklist(Uno,Bpre,Slp,Temp,Sweat,Pain,Discomfort,Eye,Cough,Drink,Stime,Sshap,Scolor,Pee,Pcolor,Utime) values(@Uno,@Bpre,@Slp,@Temp,@Sweat,@Pain,@Discomfort,@Eye,@Cough,@Drink,@Stime,@Sshap,@Scolor,@Pee,@Pcolor,@Utime);";//实例化Sqlcommand的插入语句

sql.Parameters.AddWithValue("@Uno", this.umlabname.Text.Trim());//为Sqlcommand参数赋值

sql.Parameters.AddWithValue("@Bpre", str[0]);

sql.Parameters.AddWithValue("@Slp", str[1]);

sql.Parameters.AddWithValue("@Temp", str[2]);

sql.Parameters.AddWithValue("@Sweat",str[3]);

sql.Parameters.AddWithValue("@Pain",str[4]);

sql.Parameters.AddWithValue("@Discomfort",str[5]);

sql.Parameters.AddWithValue("@Eye",str[6]);

sql.Parameters.AddWithValue("@Cough", str[7]);

sql.Parameters.AddWithValue("@Drink", str[8]);

sql.Parameters.AddWithValue("@Stime", str[9]);

sql.Parameters.AddWithValue("@Sshap", str[10]);

sql.Parameters.AddWithValue("@Scolor", str[11]);

sql.Parameters.AddWithValue("@Pee", str[12]);

sql.Parameters.AddWithValue("@Pcolor", str[13]);

sql.Parameters.AddWithValue("@Utime", time.ToString());

sqlconnection.Open();//打开Sqlconnection连接SQL数据库

int row = (int)sql.ExecuteNonQuery();

sqlconnection.Close();//关闭Sqlconnection连接SQL数据库

if (row == 1)

{

allpan(true, 3);//设置选项的可见性

allpan(false,1);//设置选项的可用性

this.butumcun.Visible = false;

return;

}

else

{

tishi.Renew("问诊单更新失败", this);

return;

}

}

1. 登录模块

主要是通过对登录用户的登录名、密码以及登录的类型进行验证，判断用户的登录信息是否正确，若通过验证，则登录；否则，显示错误。

protected void revbutgo\_Click(object sender, EventArgs e)

{

SqlCommand sql1 = opensql();//查询该用户名是否通过审核，定义变量来res1\res2,分别表示未通过审核和已通过审核的用户

SqlCommand sql2 = opensql();

sql1.CommandText =

"select count(\*) from d\_user where Dno=@Dno and checkorno=@checkorno;";

sql1.Parameters.AddWithValue("@Dno", this.revtname.Text.Trim());

sql1.Parameters.AddWithValue("@checkorno", "0");

sql2.CommandText =

"select count(\*) from d\_user where Dno=@Dno and checkorno=@checkorno;";

sql2.Parameters.AddWithValue("@Dno", this.revtname.Text.Trim());

sql2.Parameters.AddWithValue("@checkorno", "1");

sqlconnection.Open();

int res1 = (int)sql1.ExecuteScalar();

int res2 = (int)sql2.ExecuteScalar();

sqlconnection.Close();

if (res1 == 1)

{

this.revlabres.Text = "该用户等待管理员审核！";

}

else if (res2 == 1)

{

tishi.Refresh("该用户已经通过管理员审核，您可直接去登录", this);

return;

}

else

{

tishi.Refresh("该医生用户不存在", this);

return;

}

//this.regprocess.Visible = true;

this.reviewcheck.Visible = true;

}