**“抱朴子”化学反应数据库**

**C语言项目报告**

**项目成员：\_\_\_\_\_张朔\_\_张海波\_\_吴浩翔\_\_\_\_\_\_**

**填写日期：\_\_\_\_\_\_\_\_\_2021.7.8\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

1. **摘要（Abstract）**

“抱朴子”化学反应数据库是一个由我们自己开发的数据库，我们希望它能够满足一个化学工作者的许多重要需求，如对反应信息的查询、对其他工作者科研方向和经历的查询等。虽然与现今已有的化学反应数据库还存在很大的差距，但对于初入C语言项目开发的小白而言，本项目的开发是一次有意义的尝试和探索。

1. **问题描述（Problem Statement）**

在化学家的不断探索中，对前人工作的学习和借鉴是不可或缺的，特别是在有机合成工作中。对于每一步反应的进行，合成工作者都希望从前人走过的路中获得启示。因此Reaxys和Scifinder应运而生。

诚然，在两者的帮助下化学工作者可以快速找寻到所需反应的相关文献，但却难免遇到文章内容老旧，Supporting Information对于反应的描述过于简略，十分不利于反应的重复，甚至反应完全重复不出来的情况，即很多时候我们难以得到反应所需要的准确信息。因此，我们需要一个将反应阐释的更为明确的由自己人所构建的数据库，解决上述诸如“无法重复同一实验”的难题，助力科研工作。

1. **组内分工（Group Division）**

小组成员信息、具体分工和所占工作量如下表：

**注**:重点工作标红

|  |  |  |  |
| --- | --- | --- | --- |
| 小组成员 | 成员信息 | 具体分工 | 工作量所占比重 |
| 张朔（组长） | 2020级唐敖庆班化学方向学生 | 1. Python爬虫的实现 2. Python与SQL Server的连接与交互 3. 全部UI的设计与背景绘制 4. 项目功能的构想与页面链接间逻辑 5. 十四个页面的构建，全部页面风格统一重构与优化 6. 部分页面跳转的实现及小部分代码的修改 7. My SQL数据库的搭建与SQL Server数据库的搭建 8. 基础数据的查找及一半数据的录入 9. 数据库使用教学，结题PPT的制作，撰写项目计划书、接口设计、软件需求说明书、环境配置文件三个报告相关文档。 | 35% |
| 张海波（组员） | 2020级唐敖庆班化学方向学生 | 1. C#窗体主体代码的实现 2. 控件使用和参数的调试 3. 三个窗口的构建 4. 登录注册功能的实现 5. 窗口间跳转和数据交互功能的实现 6. 窗口与数据库的连接 7. 数据录入和读取功能的实现 8. 窗口与用户的交互功能的实现 9. 完成开题PPT的制作，及C#部分代码教学。 | 35% |
| 吴浩翔（组员） | 2020级唐敖庆班化学方向学生 | 1. 负责九个窗口的构建 2. 一半数据的录入 3. 进行Mysql使用的简单教学 4. 完成中期答辩PPT的制作，撰写项目报告、详细设计文档。 | 30% |

1. **分析（Analysis）**

**4.1 项目整体设计**

本项目总共设计了26个窗口和同一个数据库下的10个数据表，其中窗口按功能和目的可以分为四个部分：用户的登录与注册、对于某种化合物的检索、对于某位科研工作者的检索以及信息的录入功能；而数据表可以根据对相应数据和内容的读取需求在不同窗口进行特定调用。

用户接触的控件中，picturebox、textbox和button这三种控件起到了主要作用。其中picturebox和textbox控件又可分为用户可操作和不可操作两类，前者用于输入或是录入信息，后者用于数据和内容的呈现。而button控件是用户对应用进行控制的最主要工具，可进行“确认”、“取消”、“跳转”等操作以对涉及的各项功能进行实现。

**4.2 项目整体框架**

26个窗口分别命名为page1-page18（为除录入外的窗口）和inf1-inf8（为录入的窗口）。以窗口page1为程序主入口，通过“登录”和“注册”按钮跳转至相应登陆了窗口page2和注册窗口page3，完成注册和登录后进入窗口page4，以此为本项目主要功能实现的主页面，继而分化成三条路线以分别进行化合物检索、科研工作者检索和数据录入。

在对同一物质或是作者进行检索时，为保证不同窗口检索内容的统一，设置不可见textbox控件，用于进行不同窗口间数据传递的同时做到不影响用户的视觉体验；而使用录入窗口inf1-inf8时，虽然录入内容可能相互关联，但为保证一个窗口对应一张数据表和快速保存已录入部分，在此部分窗口均设置了“完成”按钮用于上传以及写入的部分至数据库。

**4.3 项目其他功能**

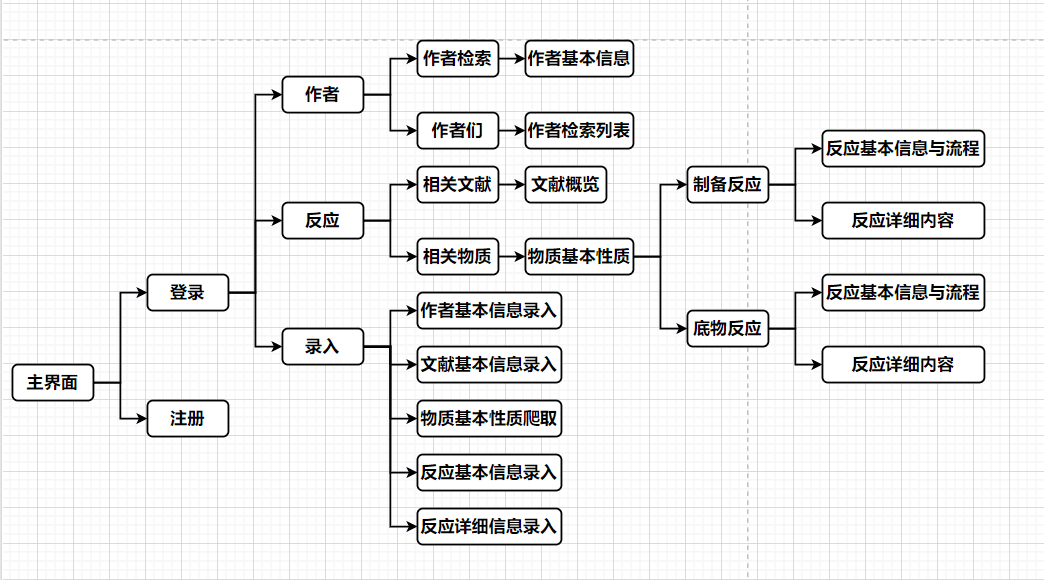
注册、登录、录入、退出等功能实现时，为提供给用户更好的交互性和适用性，在进行相应操作或是操作出现问题时会出现提示窗口，用于告知程序运行情况和操作不当的信息。

1. **设计（Design）**

本项目的整体框架设计逻辑主要分为界面设计、功能设计、数据库设计和交互设计四个方面。下面将逐一阐述。

**5.0 项目主体逻辑**

用户进入主界面后可按流程图中过程进行操作以实现对所需数据的查询和相关信息的录入。



**5.1 界面设计**

采用Photoshop进行窗口背景设计，以针对主页、注册登录界面和功能 界面建立不同的背景图片。并使用开源控件库Sunny UI进行窗口控件设计。

**5.2 功能设计**

完成一个本地的化学数据库，来满足对反应的简单查询以及用户工作的检索，以助力研究人员的工作。

**5.2.1用户登录**

用户输入无效的用户名或是错误的密码，点击登录按钮，提示登录失败。

用户输入有效的用户名以及正确的密码，点击登录按钮，提示登录成功，跳转page4，作者检索界面。

**5.2.2用户注册**

用户提交获取、在数据库中添加用户相关信息。

**5.2.3检索功能**

5.2.3.1实现由统一命名法的化学式精准查询。

5.2.3.2实现通过用户查询进入作者个人主页，查看主页中的工作内容。

5.2.3.3实现通过作者目录进行检索。

**5.2.4个人主页编辑**

个人主页编辑主要可分为以下几个部分：

5.2.4.1基本信息/ [basic](javascript:;) [information](javascript:;)；

5.2.4.2个人简历/Personal resume；

5.2.4.3工作经历/ [research](javascript:;) [experience](javascript:;)；

5.2.4.4研究方向/ Enrollment information；

5.2.4.5代表性成果/ Representative achievement。

**5.2.5录入工作内容**

录入工作分为以下模块：

5.2.5.1 Topic & Doi；

5.2.5.2 Abstract & Supporting Information；

5.2.5.3 Supporting中每一步的真实详细的实验过程；

5.2.5.4选题的出发点与实验思路。

**5.2.6 窗口切换设计**

5.2.6.1通过“主页”按钮返回page4检索界面；

5.2.6.2通过功能按钮跳转至各相关功能界面；

5.2.6.3通过“返回”按钮跳转上一界面。

**5.3数据库设计**

**5.3.1 介绍**

数据库当中存放若干表格，包括但不限于author, reaction和notice，它提供了后端的搜索功能。

**5.3.2 数据库环境描述**

该数据库由sqlserver数据库管理系统建立并维护。

**5.3.3 逻辑分析与设计——模块功能**

5.3.3.1用户管理：有关用户的信息有：用户名，密码，邮箱。用户可以使用用户名进行登录，也可以在自己的个人页面上进行编辑，录入相关反应及数据等内容。

5.3.3.2化学反应管理：关于某一个化学反应，包括的信息有：反应物，生成物，反应条件，所需试剂，反应时长，反应检测手段，产物的检验和纯化等。

5.3.3.3录入反应管理：当用户录入一个反应是时候，应至少输入以下内容：反应物，生成物，反应条件，反应所需试剂四方面。

**5.3.4数据设计**

5.3.4.1 用户基本信息数据录入

5.3.4.2 用户工作信息录入

5.3.4.3 用户论文基本信息录入

5.3.4.4 物质详细信息录入

**5.4物质基本性质数据的爬取**

5.4.1 Python数据的爬取

5.4.2 Python与SQL server的连接

5.4.3 爬取数据录入特定数据库

5.4.4特定数据库数据与主数据库之间的交互

**5.5交互设计**

**5.5.1数据库的读写**

5.5.1.1 数据库连接

5.5.1.2 数据库读取

5.5.1.3 数据库添写

**5.5.2 按钮功能**

5.5.2.1文本框功能（TextBox，RichTextBox，LinkLabel）

5.5.2.2 分页功能（TabControlMenu，Pagination）

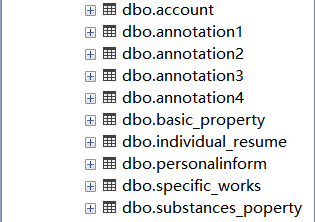
5.5.2.3 窗口功能（Form）

5.5.3.1 窗口跳转（Button）

1. **实施（Implementation）**

**6.1 核心模块Ⅰ**：数据库存储部分。

**6.1.1总览表**

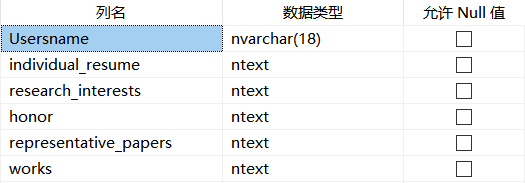


**6.1.2具体表**

i)用户基本信息存储



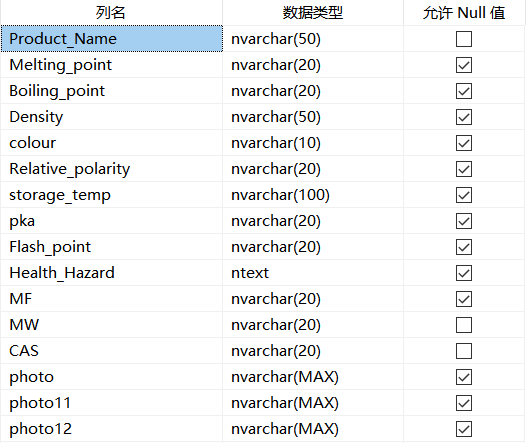
**ii)用户工作信息存储**



**iii)用户论文基本信息存储**



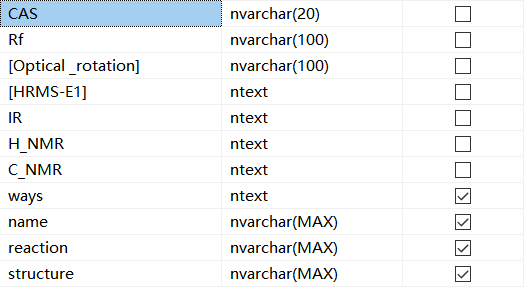
**iv)物质详细信息存储**



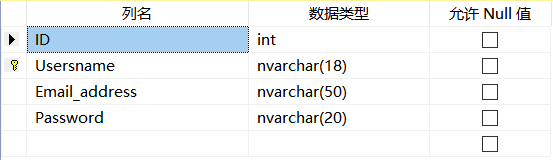
**v)文献基本信息存储**



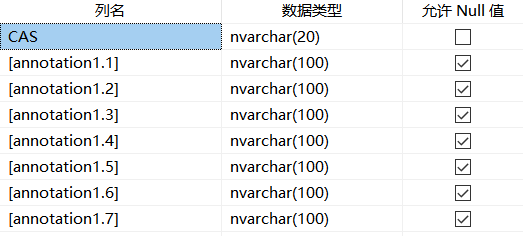
**vi)反应基本信息存储**

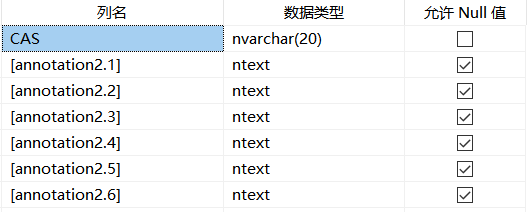


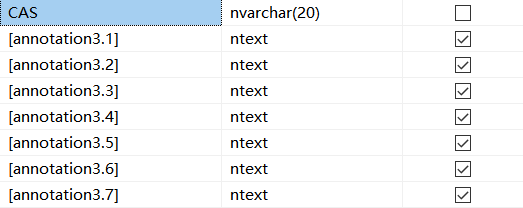
**vii)用户名与密码存储**

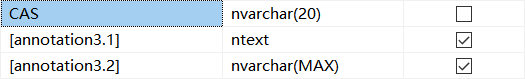


**viii)反应详细信息存储**









**6.2 核心模块Ⅱ**：窗体设计部分

**6.2.1登录与注册**

登录与注册功能的实现其实就是连接数据库后再进行对数据库中数据和内容的读取和录入。在登录时，程序会对输入内容进行检验，以验证是否为已有账号，进而决定是否可以登录成功；在注册时，程序会对输入内容进行检测，若为已有帐号、输入格式错误或是其他原因则会导致登陆失败。

登录代码：

private void btnlogin\_Click(object sender, EventArgs e)

{

string M\_str\_name = page2b1.Text;

string M\_str\_Pwd = page2b2.Text;

string constr = "Server=.;DataBase=Baopuzi;Integrated Security=True";

//建立SqlConnection对象

SqlConnection con = new SqlConnection(constr);

//打开链接

con.Open();

SqlCommand com = new SqlCommand("select Usersname,Password from account where Usersname = '" + M\_str\_name + "' and password = '" + M\_str\_Pwd + "'", con);

SqlDataAdapter da = new SqlDataAdapter(com);

DataSet ds = new DataSet();

int n = da.Fill(ds, "account");

if (n != 0)

{

con.Close();

MessageBox.Show("登录成功了( •̀ ω •́ )y", "抱朴子", MessageBoxButtons.OK, MessageBoxIcon.Information);

this.Hide();

Form page4 = new page4();

page4.ShowDialog();

}

else

{

MessageBox.Show("用户名或密码错误，请重新输入！", "抱朴子");

page2b1.Text = "";

page2b2.Text = "";

page2b1.Focus();

}

con.Close();

}

注册代码：

private void QueRen\_Click(object sender, EventArgs e)

{

string Usersname = page3c1.Text;

string email = page3c2.Text;

string password = page3c3.Text;

string JiaoYan = page3c4.Text;

try

{

if (!doperate.ValidateEmail(email.Trim()))

{

MessageBox.Show("邮箱格式不正确", "信息", MessageBoxButtons.OK, MessageBoxIcon.Information);

page3c2.ClearUndo();

page3c2.Focus();

}

else if (!doperate.ValidateNum(password.Trim()))

{

MessageBox.Show("密码格式不正确", "信息", MessageBoxButtons.OK, MessageBoxIcon.Information);

page3c3.ClearUndo();

page3c3.Focus();

}

else if (!doperate.ValidateNum(JiaoYan.Trim()))

{

MessageBox.Show("再次输入密码格式不正确", "信息", MessageBoxButtons.OK, MessageBoxIcon.Information);

page3c4.ClearUndo();

page3c4.Focus();

}

else if (password != JiaoYan)

{

MessageBox.Show("前后输入密码不一致", "信息", MessageBoxButtons.OK, MessageBoxIcon.Information);

page3c3.ClearUndo();

page3c4.ClearUndo();

page3c3.Focus();

}

else

{

if (Usersname == "")

{

MessageBox.Show("用户名不能为空！", "抱朴子", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

else

{

// 数据库配置

string connString = @"Server=.;DataBase=Baopuzi;Integrated Security=True;MultipleActiveResultSets=True";

SqlConnection conn = new SqlConnection(connString);

// 连接

conn.Open();

// 查找用户名是否存在

string sql = String.Format("select Usersname from account where Usersname = '{0}'", Usersname);

SqlCommand command = new SqlCommand(sql, conn);

SqlDataReader datareader = command.ExecuteReader();

int i = 0;

for (i = 0; datareader.Read(); i++) { }

if (i > 0)

{

// 已注册，弹出提示信息

MessageBox.Show("用户名已存在了哦", "抱朴子"); datareader.Close();

}

else

{

// 未注册，向数据表中插入相应的用户名和密码

string sql1 = String.Format("insert into account(Usersname, Email\_address, Password) values('{0}','{1}','{2}')", Usersname, email, password);

SqlCommand command1 = new SqlCommand(sql1, conn);

int j = command1.ExecuteNonQuery();

if (j > 0)

{

MessageBox.Show("注册成功啦", "抱朴子");

this.DialogResult = DialogResult.Retry;

this.Close();

}

}

// 断开数据库连接

conn.Close();

}

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message, "抱朴子", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

this.Hide();

Form page4 = new FrmLogin();

page4.ShowDialog();

}

**6.2.2检索和录入**

此部分本质上和之前的登录与注册相同，只是调用的数据表不完全相同，故在一部分细节上需要修改。

检索代码：（以窗口page10中的代码为例）

private void page10\_Load(object sender, EventArgs e)

{

string wuzhi = this.page10j4.Text;

SqlConnection conn = new SqlConnection("Server =.; DataBase = Baopuzi; Integrated Security = True");

conn.Open();

SqlCommand cmd = new SqlCommand("select\*from substances\_poperty where Product\_Name ='" + wuzhi + "'", conn);

SqlDataReader sdr = cmd.ExecuteReader();

while (sdr.Read())

{

if (sdr["Product\_Name"].ToString() == page10j4.Text)//如果与条件相符就将记录显示到相应文本框里

page10j5.Text = "化学式:" + sdr["MF"].ToString();

page10j6.Text = "分子量:" + sdr["MW"].ToString();

page10j7.Text = sdr["CAS"].ToString();

page10j8.Text = "熔点:" + sdr["Melting\_point"].ToString();

page10j9.Text = "沸点:" + sdr["Boiling\_point"].ToString();

page10j10.Text = "密度:" + sdr["Density"].ToString();

page10j11.Text = "颜色:" + sdr["colour"].ToString();

page10j12.Text = "极性:" + sdr["Relative\_polarity"].ToString();

page10j13.Text = "储存:" + sdr["storage\_temp"].ToString();

page10j14.Text = "酸度:" + sdr["pka"].ToString();

page10j15.Text = "闪点:"+ sdr["Flash\_point"].ToString();

page10j16.Text = "毒理学性质:" + sdr["Health\_Hazard"].ToString();

page10j19.Text = sdr["photo"].ToString();

this.page10j3.Image = Image.FromFile(page10j19.Text);

}

sdr.Close();

sdr.Dispose();

cmd.Dispose();

}

录入代码：（以窗口inf2中的代码为例）

private void inf2b1\_Click(object sender, EventArgs e)

{

string Doi = inf2b5.Text;

string Topic = inf2b4.Text;

string Abstract = inf2b6.Text;

string sunthetic\_route = inf2b10.Text;

if (inf2b4.Text == ""|inf2b5.Text == ""|inf2b6.Text == ""|inf2b7.Text == ""|inf2b8.Text == "")

{

MessageBox.Show("内容没有填完哦", "抱朴子", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

else

{

// 数据库配置

string connString = @"Server=.;DataBase=Baopuzi;Integrated Security=True;MultipleActiveResultSets=True";

SqlConnection conn = new SqlConnection(connString);

// 连接

conn.Open();

// 未录入，向数据表中插入相应内容

string sql1 = String.Format("insert into specific\_works(Topic, Author, Doi, Abstract, sunthetic\_route) values('{0}','{1}','{2}','{3}','{4}')", Topic, "0", Doi, Abstract, sunthetic\_route);

SqlCommand command1 = new SqlCommand(sql1, conn);

int j = command1.ExecuteNonQuery();

if (j > 0)

{

MessageBox.Show("录入成功啦", "抱朴子");

this.DialogResult = DialogResult.Retry;

this.Close();

}

// 断开数据库连接

conn.Close();

this.Hide();

Form inf1 = new inf1();

inf1.StartPosition = System.Windows.Forms.FormStartPosition.CenterScreen;

inf1.ShowDialog();

}

}

**6.2.3其它交互功能**

控件的使用代码或是其它小环节的代码在上面已有所展示，接下来展示其它环节。

对输入内容的验证：

#region 验证文本框输入为邮箱

/// <summary>

/// 验证文本框输入为邮箱

/// </summary>

/// <param name="M\_str\_email">输入字符串</param>

/// <returns>返回一个bool类型的值</returns>

public bool ValidateEmail(string M\_str\_Email)

{

return Regex.IsMatch(M\_str\_Email, @"(^[A-Za-z0-9]+([\_\.][A-Za-z0-9]+)\*@([A-Za-z0-9\-]+\.)+[A-Za-z]{2,6}$)");

}

#endregion

上传图片：

private void inf2b11\_Click(object sender, EventArgs e)

{

OpenFileDialog ofd = new OpenFileDialog();

ofd.Title = "请选择上传的图片";

ofd.Filter = "图片格式|\*.jpg|\*.png";

ofd.Multiselect = false;

if (ofd.ShowDialog() == System.Windows.Forms.DialogResult.OK)

{

//获得文件的完整路径

string filePath = ofd.FileName;

inf2b10.Text = filePath;

}

}

窗口跳转：

private void page5e1\_Click(object sender, EventArgs e)

{

this.Hide();

Form page4 = new page4();

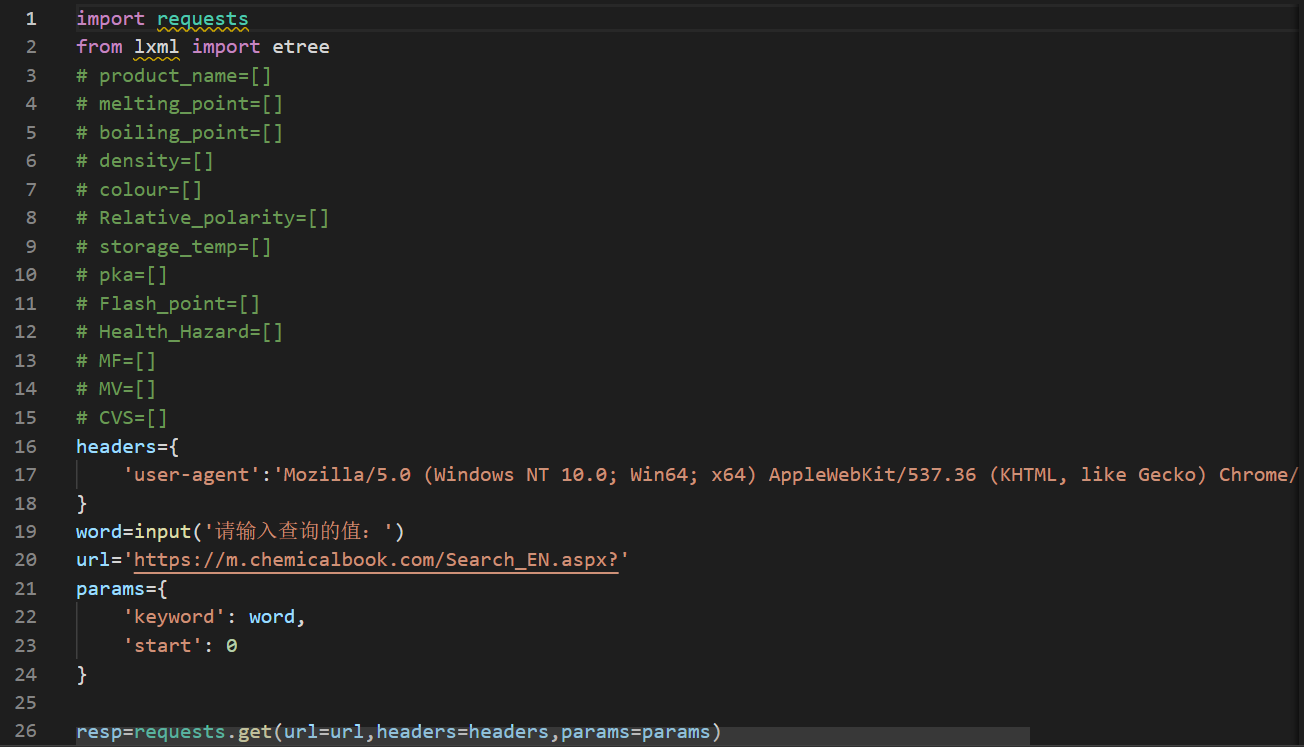
page4.StartPosition = System.Windows.Forms.FormStartPosition.CenterScreen;

page4.Show();

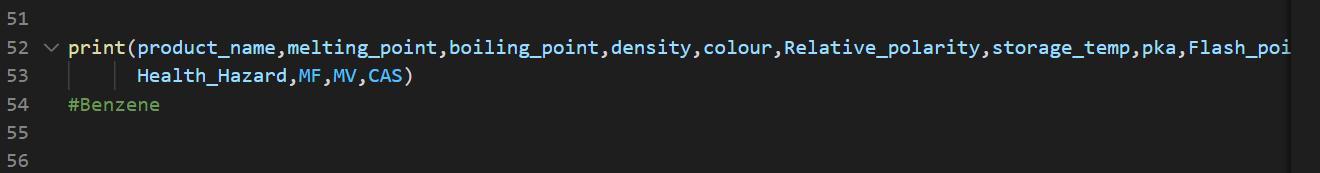
}

**6.3 核心模块Ⅲ**：爬虫设计部分（注：polacode插件不好用了）

**6.2.1仅爬虫部分**

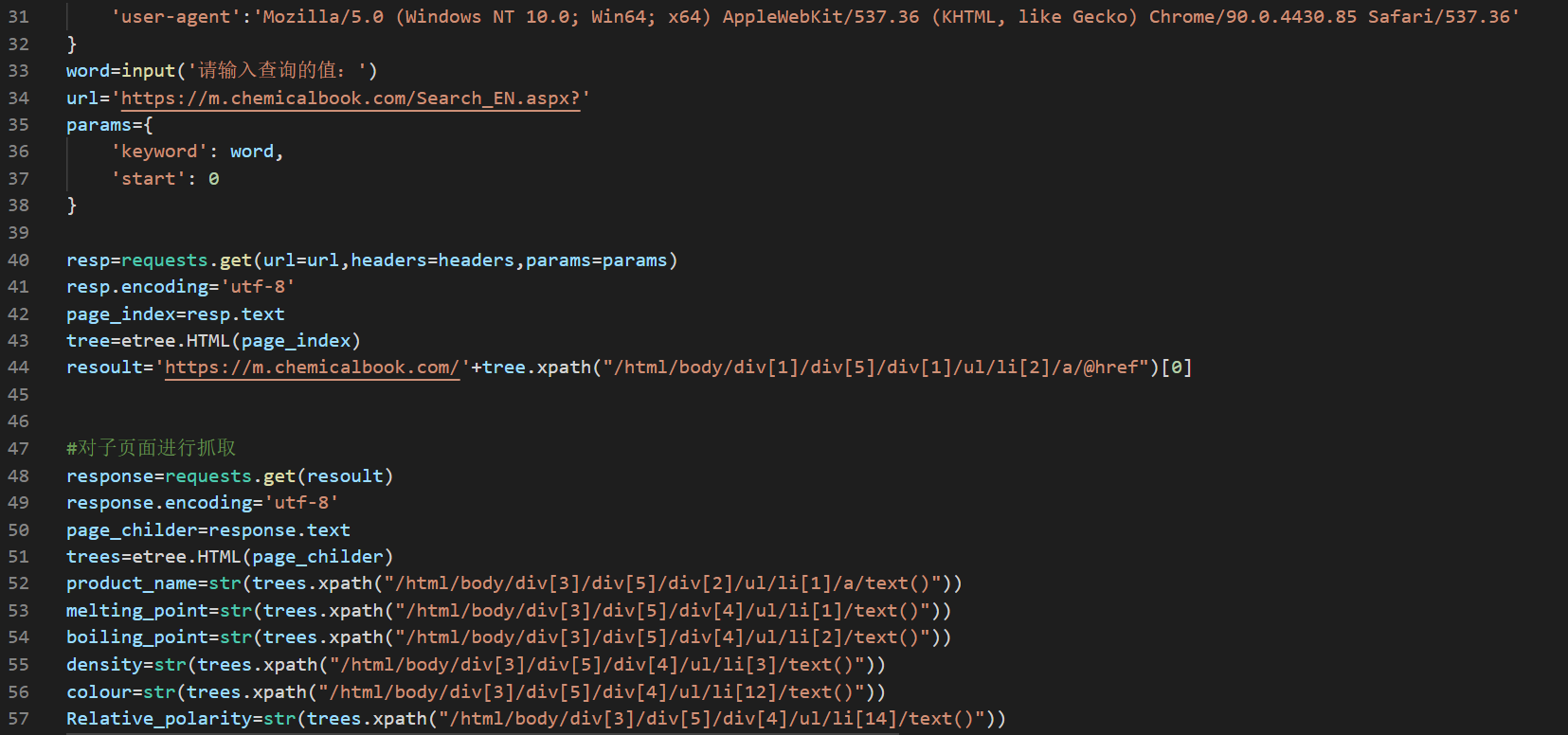
****

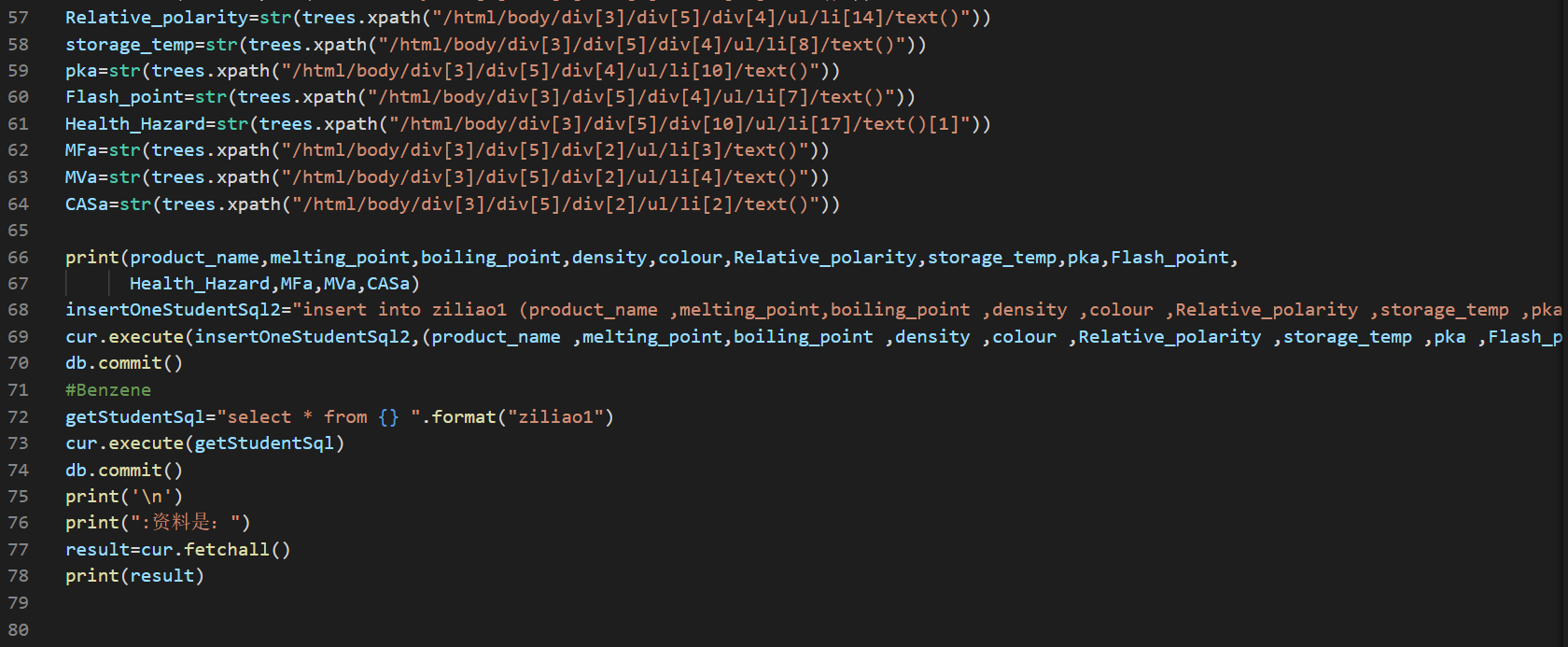
****

****

**6.2.1Python与My SQL连接后的爬虫部分**

****

****

****

1. **测试（Test）**

程序主入口，点击相关按钮可进行跳转。

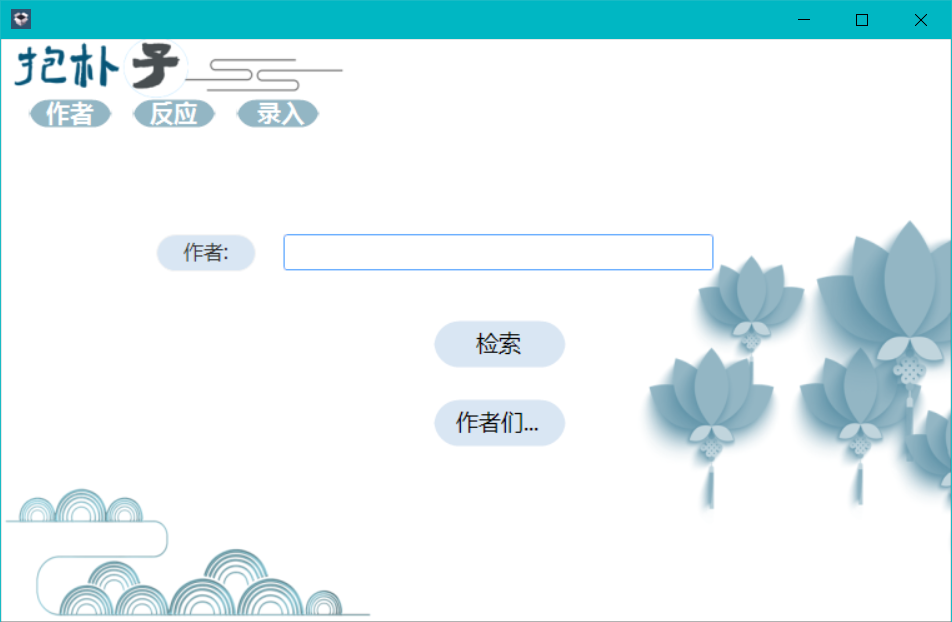


登录/注册界面，完成输入后即可跳转。





工作主页面：左上三个按钮可跳转至相应功能的窗口。





点击相应按钮或是输入科研工作者再进行检索后会跳转：



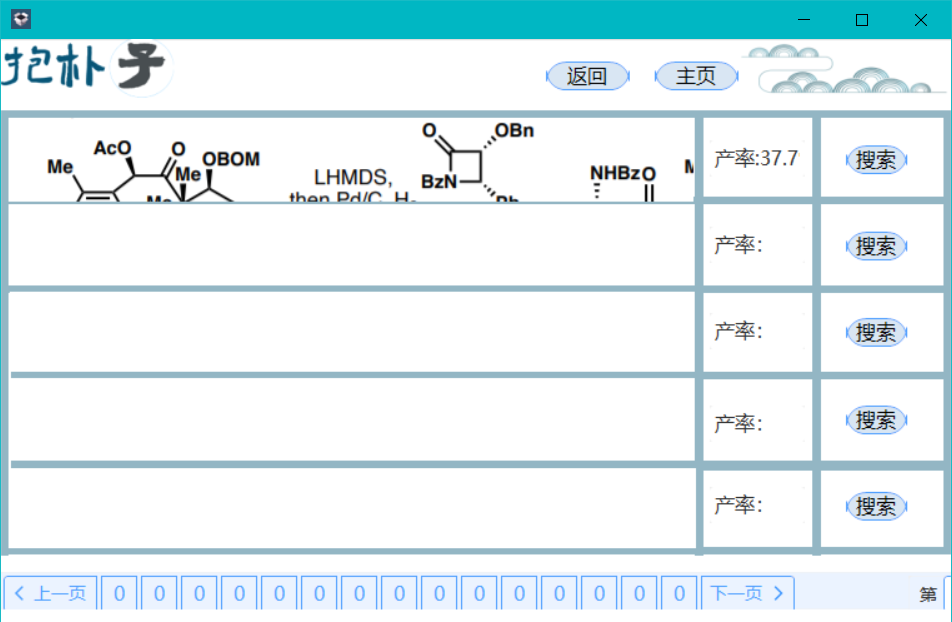
若进行物质检索，则有：



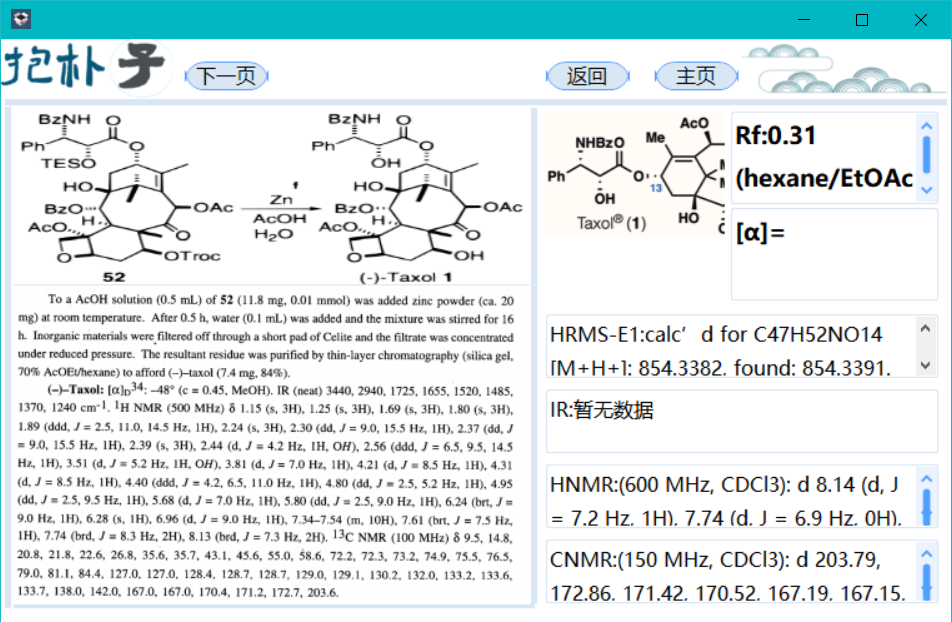
点击检索：



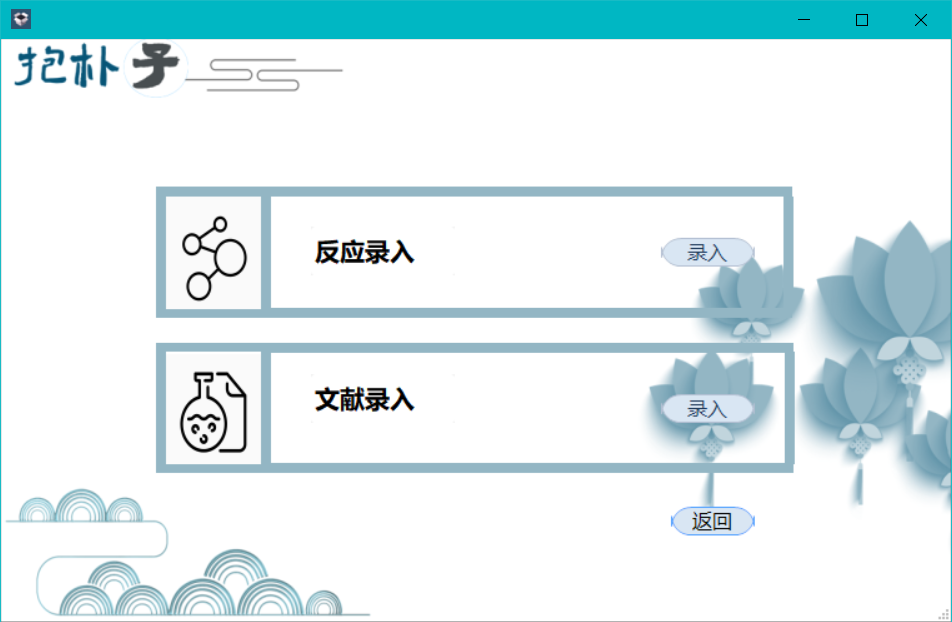




后续内容较为相似，故只截取一个画面。



录入部分情况和之前较为类似，只是添加“完成”和“上传图片”两个按钮，textbox也为可编辑状态而不是只读。



选取图片后的结果，更多窗口情况大地类似，故只选用这一部分窗口。



爬虫部分：

