**\*\*\*\*\*概要设计规约（说明书）**

**软件需求规格(说明书)模板解释及说明：**

修订历史：描述修订人、时间及在哪个版本上修订。如果是迭代开发每个版本对应一次迭代。非迭代开发每次正式的修改标记为一个版本。可以用以下的表来体现：

修订历史记录

|  |  |  |  |
| --- | --- | --- | --- |
| **编写日期** | **版本** | **说明** | **作者** |
| 2021-12-13 | 1.0 | 第一次迭代 | 全体人员 |
|  |  |  |  |
|  |  |  |  |

评审历史记录

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **评审日期** | **版本** | **评审参与人员** | **评审后修改批准日期** | **确认人员签字** |
| 2021-12-13 | 1.0 | 全体人员 | 2021-12-13 | 全体人员 |
|  |  |  |  |  |
|  |  |  |  |  |

账户子系统 @田溪

社区子系统 @邹嘉祥

OJ子系统 @王凌 @yx

在线工具子系统 @田溪

1. 引言

## 1.1. 概要设计依据

我们概要设计的主要依据是需求分析规约以及需求规约。

根据需求分析规约中的体系结构，我们决定将总系统分为四个子系统再细化实现——账户子系统、OJ子系统、社区子系统和在线工具子系统，因此，我们在这四个子系统的基础上设计了接口。

我们还根据四个子系统涉及的相关属性以及相关操作，设计了code, comment, question, reference, sample, solution, user, hibernate\_sequence这八张表，并详细设计了每个表的属性、属性类别、主码和外码。

根据需求规约中的分析，我们决定要利用硬件条件为内存1G的客户端、Tomcat5.0或Weblogic所支持的应用服务器以及支持Mysql的数据库服务器来搭载我们的系统。

根据需求规约中关于技术的说明，我们决定才用Vue.js, AXIOS, ElementUI的前端技术，与Java, Spring Boot, Spring Cloud, Mysql, Redis, Mybatis等后端技术来完成我们的项目。

## 1.2. 参考资料

《Software Engineering: A Practitioner’s Approach 8th》

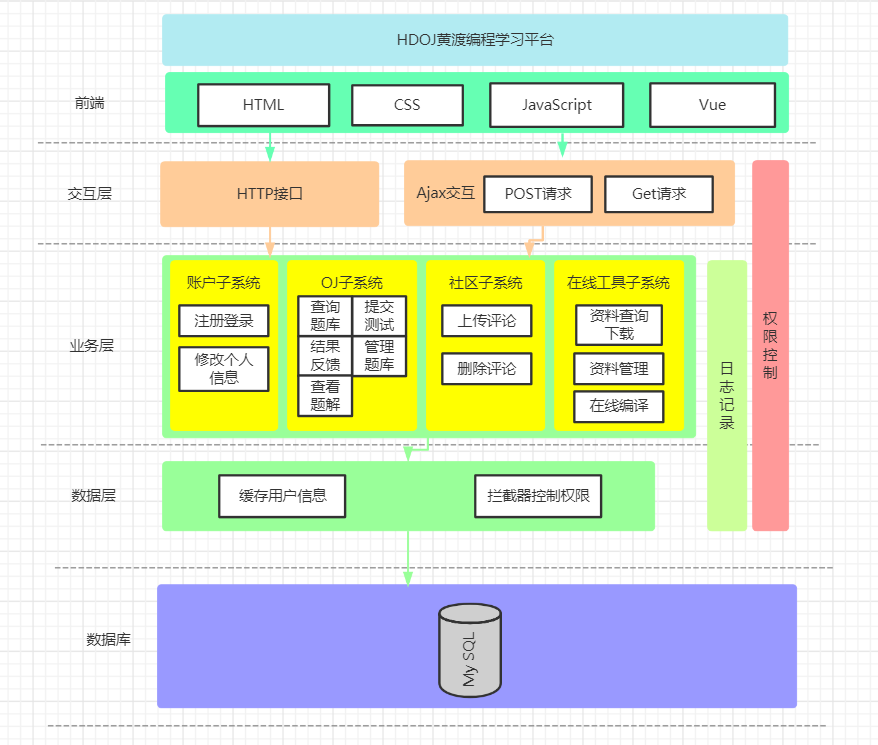
《软件工程：实践者的研究方法》

《国家标准软件开发文档规范》

《软件开发流程》

# 2. 概要设计

## **2.1. 系统总体架构设计**



#### 本系统的四个子系统功能细化之后主要实现的功能有： 1。账户子系统

#### 注册

#### 登录

#### 账户刷题信息

#### 修改个人信息

#### 获取个人信息

#### 社区子系统

#### 查看评论

#### 点赞评论

#### 添加评论

#### OJ子系统

#### 执行代码

#### 提交代码

#### 获取题库

#### 查询题目

#### 增加题解

#### 获得题解个数

#### 点赞题解

#### 代码保存

#### 代码获取

#### 获取样例

#### 在线工具子系统

#### 在线编译

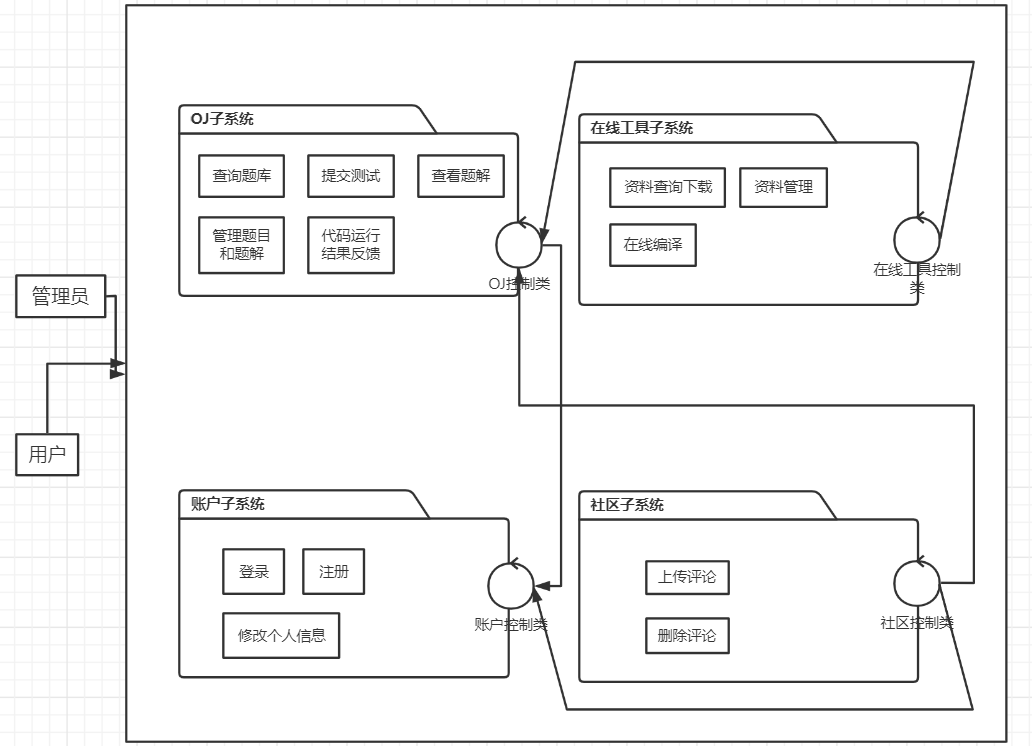
#### 资料获取

#### 资料查询

#### 资料下载

## 2.2. 系统软件结构设计

在该系统中，存在用户和管理员两个使用者，其中用户使用本系统的基本功能，管理员负责管理本系统的一些信息和资料。一共存在账户子系统、OJ子系统、在线工具系统、社区子系统共四个子系统，其中我们子系统之间各有函数调用，OJ子系统中存在查询题库，提交测试，查看题解，管理题目和题解，代码运行结果反馈功能，在线工具子系统包括资料查询下载，资料管理，在线编译功能，账户子系统包括登录注册，修改个人信息功能，社区子系统包括上传评论，删除评论功能。



## **2.3. 接口设计** @王凌

### 2.3.1 内部接口

#### 1. 账户子系统

##### 1.1注册

传输参数：

|  |  |  |  |
| --- | --- | --- | --- |
| **字段** | **类型** | **是否必需** | **说明** |
| username | String | 是 | 用户名，本地存储的token是用户名对应的用户id |
| email | String | 是 | 电子邮箱，后续有利用电子邮箱进行用户验证，活动宣传以及找回密码等功能。 |
| password | String | 是 | 密码，经过加密之后存进数据库 |
| university | String | 是 | 大学 |
| age | String | 是 | 年龄 |

返回参数：

|  |  |  |
| --- | --- | --- |
| **字段** | **类型** | **说明** |
| code | Int | 可根据返回的code判断是否成功以及错误类型进行提示。 |

##### 1.2 登录

传输参数：

|  |  |  |  |
| --- | --- | --- | --- |
| **字段** | **类型** | **是否必需** | **说明** |
| username | String | 是 | 用户名，本地存储的token是用户名对应的用户id |
| password | String | 是 | 密码，经过加密之后存进数据库 |

返回参数：

|  |  |  |
| --- | --- | --- |
| **字段** | **类型** | **说明** |
| code | Int | 可根据返回的code判断是否成功以及错误类型进行提示。 |
| id | String | 存储在本地token之中，既保证了用户信息安全，又能根据本地的token信息在数据库找到用户 |
| username | String | 用户名，保存在token之中，便于调用 |
| avatar | String | 用户头像，用字节码保存 |
| email | String | 电子邮箱 |

##### 1.3 账户刷题信息

传输参数：

|  |  |  |  |
| --- | --- | --- | --- |
| **字段** | **类型** | **是否必需** | **说明** |
| userId | Long | 是 | 用户Id，取自本地存储的token内容，安全方便 |

返回参数：

|  |  |  |
| --- | --- | --- |
| **字段** | **类型** | **说明** |
| code | Int | 可根据返回的code判断是否成功以及错误类型进行提示。 |
| map | Map | 该用户的题目列表 |

##### 1.4 修改个人信息

传输参数：

|  |  |  |  |
| --- | --- | --- | --- |
| **字段** | **类型** | **是否必需** | **说明** |
| user | User | 是 | 存储了更新信息的用户类，传输之后直接覆盖可以更新原用户信息 |

返回参数：

|  |  |  |
| --- | --- | --- |
| **字段** | **类型** | **说明** |
| code | Int | 可根据返回的code判断是否成功以及错误类型进行提示。 |

##### 1.5 获取个人信息

传输参数：

|  |  |  |  |
| --- | --- | --- | --- |
| **字段** | **类型** | **是否必需** | **说明** |
| userId | Long | 是 | 用户Id，取自本地存储的token内容，安全方便 |

返回参数：

|  |  |  |
| --- | --- | --- |
| **字段** | **类型** | **说明** |
| code | Int | 可根据返回的code判断是否成功以及错误类型进行提示。 |
| username | String | 用户名 |

#### 社区子系统

##### 2.1 查看评论

传输参数：

|  |  |  |  |
| --- | --- | --- | --- |
| **字段** | **类型** | **是否必需** | **说明** |
| solutionid | Long | 是 | 题解Id，每个题解之后都可以被评论 |

返回参数：

|  |  |  |
| --- | --- | --- |
| **字段** | **类型** | **说明** |
| code | Int | 可根据返回的code判断是否成功以及错误类型进行提示。 |
| comments | List<Comment> | 返回评论列表 |

##### 2.2 点赞评论

传输参数：

|  |  |  |  |
| --- | --- | --- | --- |
| **字段** | **类型** | **是否必需** | **说明** |
| commentId | Long | 是 | 评论Id，用户可以对评论进行点赞 |

返回参数：

|  |  |  |
| --- | --- | --- |
| **字段** | **类型** | **说明** |
| code | Int | 可根据返回的code判断是否成功以及错误类型进行提示。 |

##### 2.3 添加评论

传输参数：

|  |  |  |  |
| --- | --- | --- | --- |
| **字段** | **类型** | **是否必需** | **说明** |
| comment | Comment | 是 | 传输评论类 |

返回参数：

|  |  |  |
| --- | --- | --- |
| **字段** | **类型** | **说明** |
| code | Int | 可根据返回的code判断是否成功以及错误类型进行提示。 |

#### 3. OJ子系统

##### 3.1 执行代码

传输参数：

|  |  |  |  |
| --- | --- | --- | --- |
| **字段** | **类型** | **是否必需** | **说明** |
| data | Map<String, String> | 是 | data中包含String类型的input的样例和String类型的code代码 |

返回参数：

|  |  |  |
| --- | --- | --- |
| **字段** | **类型** | **说明** |
| code | Int | 可根据返回的code判断是否成功以及错误类型进行提示。 |
| result | String | 返回的结果 |

##### 3.2 提交代码

传输参数：

|  |  |  |  |
| --- | --- | --- | --- |
| **字段** | **类型** | **是否必需** | **说明** |
| data | Map<String, String> | 是 | data中包含String类型的code代码和Long类型的的questionId。 |

返回参数：

|  |  |  |
| --- | --- | --- |
| **字段** | **类型** | **说明** |
| code | Int | 可根据返回的code判断是否成功以及错误类型进行提示。 |

##### 3.3 获取题库

传输参数：

无

返回参数：

|  |  |  |
| --- | --- | --- |
| **字段** | **类型** | **说明** |
| code | Int | 可根据返回的code判断是否成功以及错误类型进行提示。 |
| questions | List<Question> | 返回所有题目列表 |

##### 3.4 查询题目

传输参数：

|  |  |  |  |
| --- | --- | --- | --- |
| **字段** | **类型** | **是否必需** | **说明** |
| questionId | Long | 是 | 传输题目Id |

返回参数：

|  |  |  |
| --- | --- | --- |
| **字段** | **类型** | **说明** |
| code | Int | 可根据返回的code判断是否成功以及错误类型进行提示。 |
| question | Question | 返回题目类 |

##### 3.5 获取题解

传输参数：

|  |  |  |  |
| --- | --- | --- | --- |
| **字段** | **类型** | **是否必需** | **说明** |
| questionId | Long | 是 | 传输题目Id |

返回参数：

|  |  |  |
| --- | --- | --- |
| **字段** | **类型** | **说明** |
| code | Int | 可根据返回的code判断是否成功以及错误类型进行提示。 |
| solutions | List<solution> | 返回solution类的列表 |

##### 3.6 增加题解

传输参数：

|  |  |  |  |
| --- | --- | --- | --- |
| **字段** | **类型** | **是否必需** | **说明** |
| solution | Solution | 是 | 传输题解类 |

返回参数：

|  |  |  |
| --- | --- | --- |
| **字段** | **类型** | **说明** |
| code | Int | 可根据返回的code判断是否成功以及错误类型进行提示。 |

##### 3.7 获得题解个数

传输参数：

无

返回参数：

|  |  |  |
| --- | --- | --- |
| **字段** | **类型** | **说明** |
| code | Int | 可根据返回的code判断是否成功以及错误类型进行提示。 |
| result | List<Integer> | 返回题解个数的列表 |

##### 3.8 点赞题解

传输参数：

|  |  |  |  |
| --- | --- | --- | --- |
| **字段** | **类型** | **是否必需** | **说明** |
| solutionId | Long | 是 | 传输题解Id |

返回参数：

|  |  |  |
| --- | --- | --- |
| **字段** | **类型** | **说明** |
| code | Int | 可根据返回的code判断是否成功以及错误类型进行提示。 |

##### 3.9 代码保存

传输参数：

|  |  |  |  |
| --- | --- | --- | --- |
| **字段** | **类型** | **是否必需** | **说明** |
| code | Code | 是 | 传输代码类 |

返回参数：

|  |  |  |
| --- | --- | --- |
| **字段** | **类型** | **说明** |
| code | Int | 可根据返回的code判断是否成功以及错误类型进行提示。 |

##### 3.10代码获取

传输参数：

|  |  |  |  |
| --- | --- | --- | --- |
| **字段** | **类型** | **是否必需** | **说明** |
| userId | Long | 是 | 传输用户Id |
| questionId | Long | 是 | 传输题目Id |

返回参数：

|  |  |  |
| --- | --- | --- |
| **字段** | **类型** | **说明** |
| code | Int | 可根据返回的code判断是否成功以及错误类型进行提示。 |
| result | Code | 返回最新存储的code类 |

**3.11获取样例**

传输参数：

|  |  |  |  |
| --- | --- | --- | --- |
| **字段** | **类型** | **是否必需** | **说明** |
| questionId | Long | 是 | 传输题目Id |

返回参数：

|  |  |  |
| --- | --- | --- |
| **字段** | **类型** | **说明** |
| code | Int | 可根据返回的code判断是否成功以及错误类型进行提示。 |
| samples | List<Sample> | 返回样例类列表sample |

#### 在线工具子系统

##### 4.1 在线编译

传输参数：

|  |  |  |  |
| --- | --- | --- | --- |
| **字段** | **类型** | **是否必需** | **说明** |
| data | Map<String, String> | 是 | data中包含String类型的code |

返回参数：

|  |  |  |
| --- | --- | --- |
| **字段** | **类型** | **说明** |
| code | Int | 可根据返回的code判断是否成功以及错误类型进行提示。 |
| result | String | 返回答案 |

##### 4.2 资料获取

传输参数：

无

返回参数：

|  |  |  |
| --- | --- | --- |
| **字段** | **类型** | **说明** |
| code | Int | 可根据返回的code判断是否成功以及错误类型进行提示。 |
| references | List<Reference> | 返回资料列表 |

##### 4.3 资料查询

传输参数：

|  |  |  |  |
| --- | --- | --- | --- |
| **字段** | **类型** | **是否必需** | **说明** |
| searchKey | String | 是 | 传输资料搜索码 |

返回参数：

|  |  |  |
| --- | --- | --- |
| **字段** | **类型** | **说明** |
| code | Int | 可根据返回的code判断是否成功以及错误类型进行提示。 |
| result | List<Reference> | 返回资料列表 |

### 2.3.2 外部接口

#### 2.3.2.1 资料下载

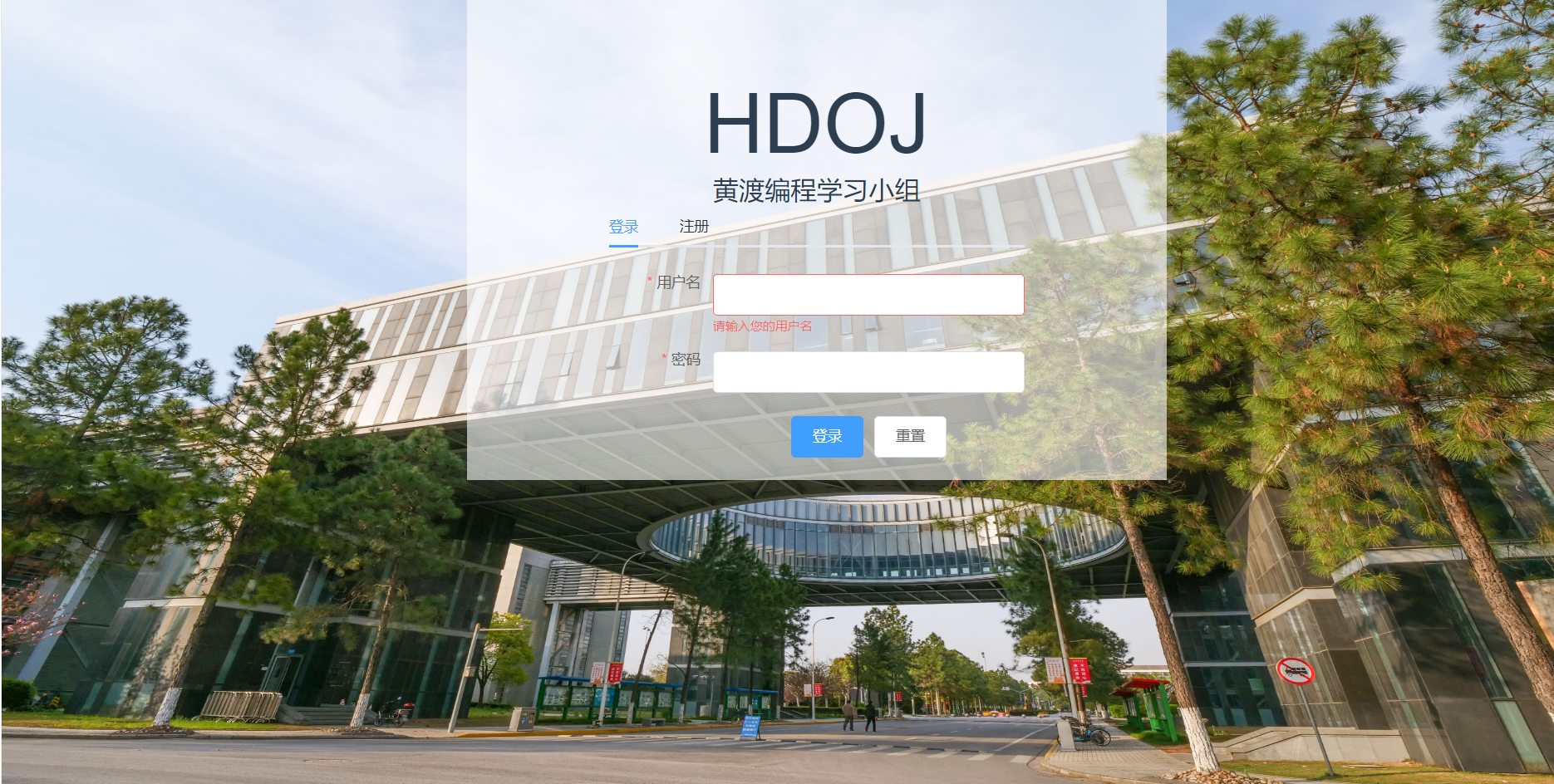
我们在资料界面枚举了许多对学习编程有用的资源，这些资源都没有部署在我们自己的服务器上，而是调用了资料下载的外部接口。

#### 2.3.2.2 外部页面跳转

为了让丰富学习内容，我们还设置了leetcode, codeforces, 洛谷等多个编程学习和练习网站的外部接口，方便我们进行外部界面的跳转。

## 2.4. 界面设计 @田溪

#### 账户子系统：





上述是登录注册页面，在该页面中用户可以进行登录和注册操作，其中注册时会要求写上用户名，密码，邮箱，大学名称等，这些信息中的用户名和密码可以在登录中输入，以及其他信息可以在之后的个人信息页面展现。

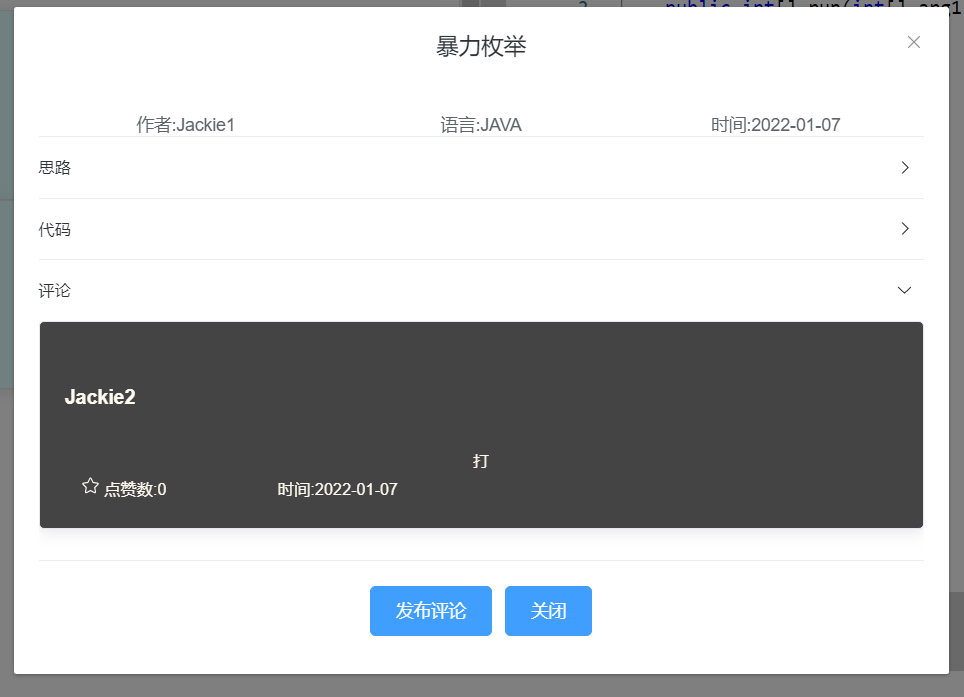


这是个人信息页面，在该页面，用户可以查看自己的个人资料，以及之前所提交的代码运行记录，同时用户可以点击编辑个人资料来编辑个人资料。



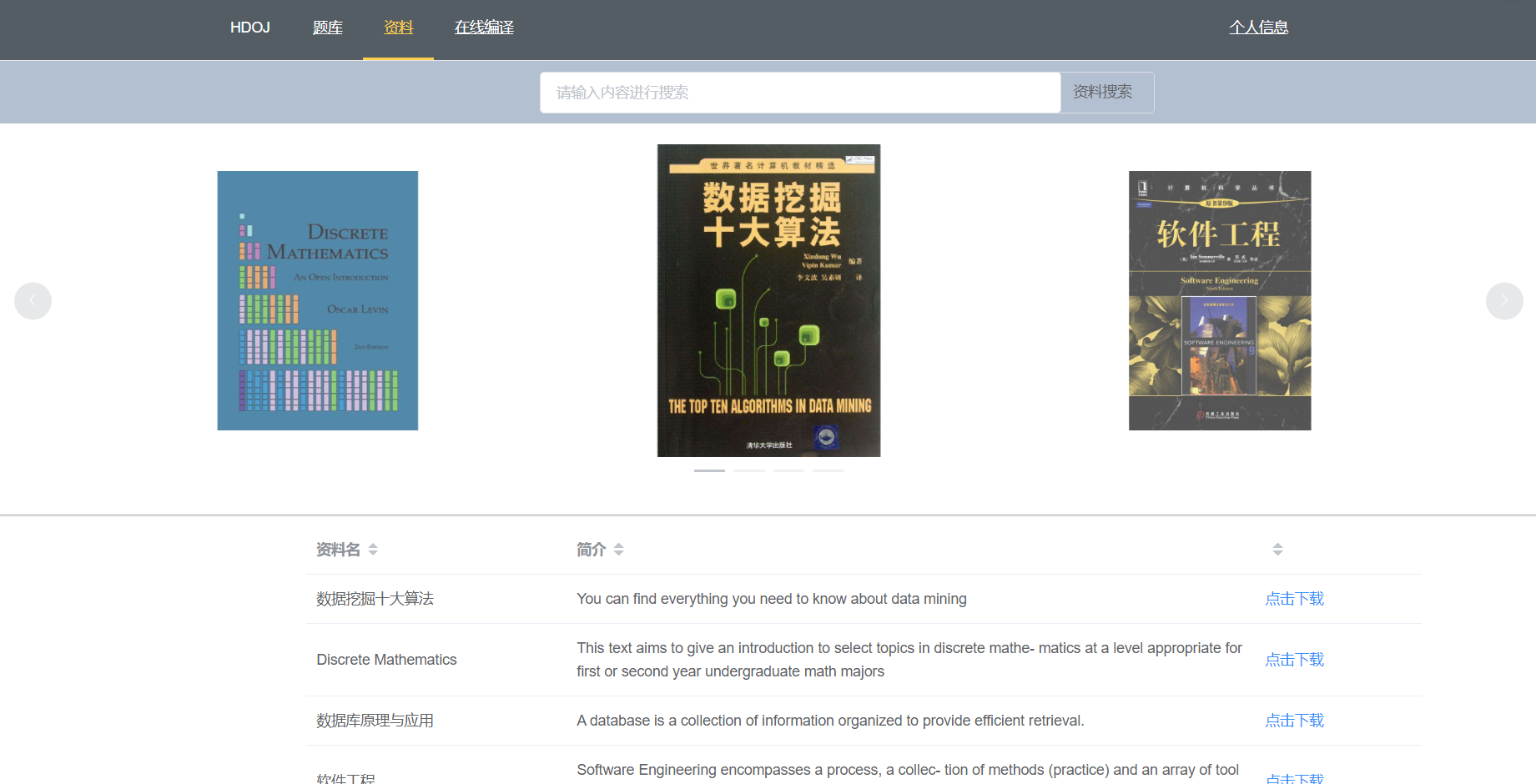
这是编辑个人信息页面，在该页面，用户可以修改自己的个人信息，用户在点击修改个人资料按钮后，会出现上述界面，界面上会先显示用户之前的信息，如有需要，用户可以修改并选择确认修改，将成功修改自己的个人信息。

#### 社区子系统：

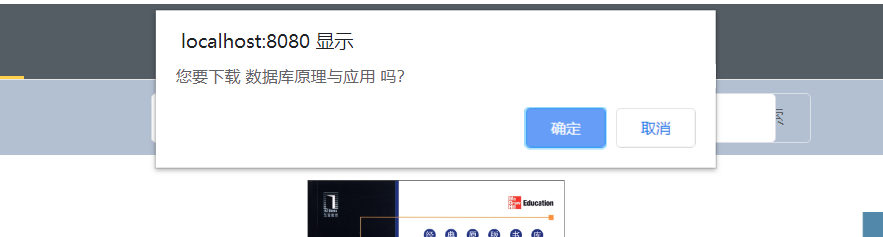


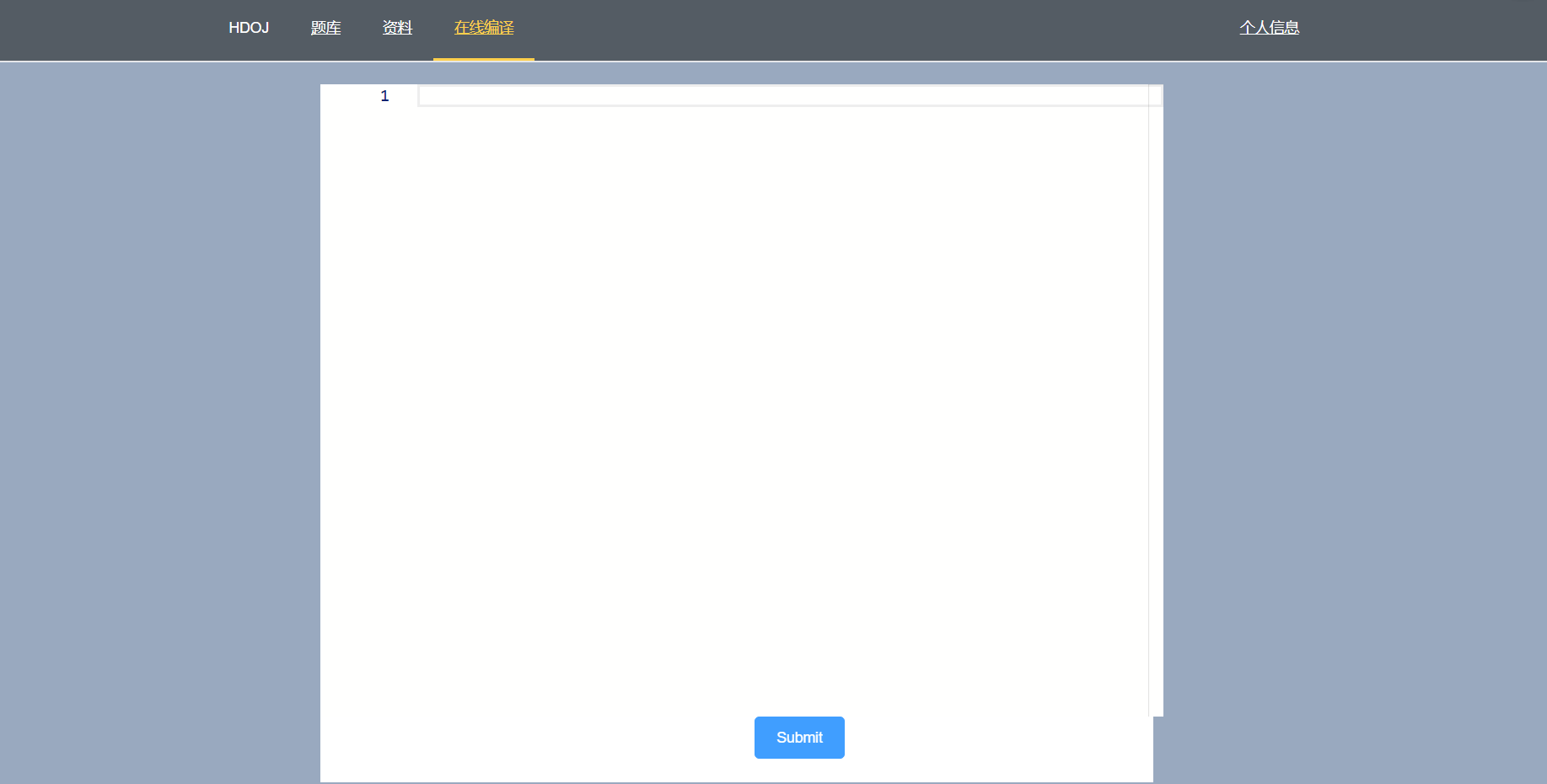
这是某一个题解的详细页面，用户可以在该页面查看其他人对于这道题的评论，也可以发布自己关于这道题的评论，也可以对某一个评论进行点赞。

#### 在线工具子系统：



这是资料界面，用户可以在该界面看到一些资料的信息，同时可以选择下载，将资料下载到本地。如下图所示



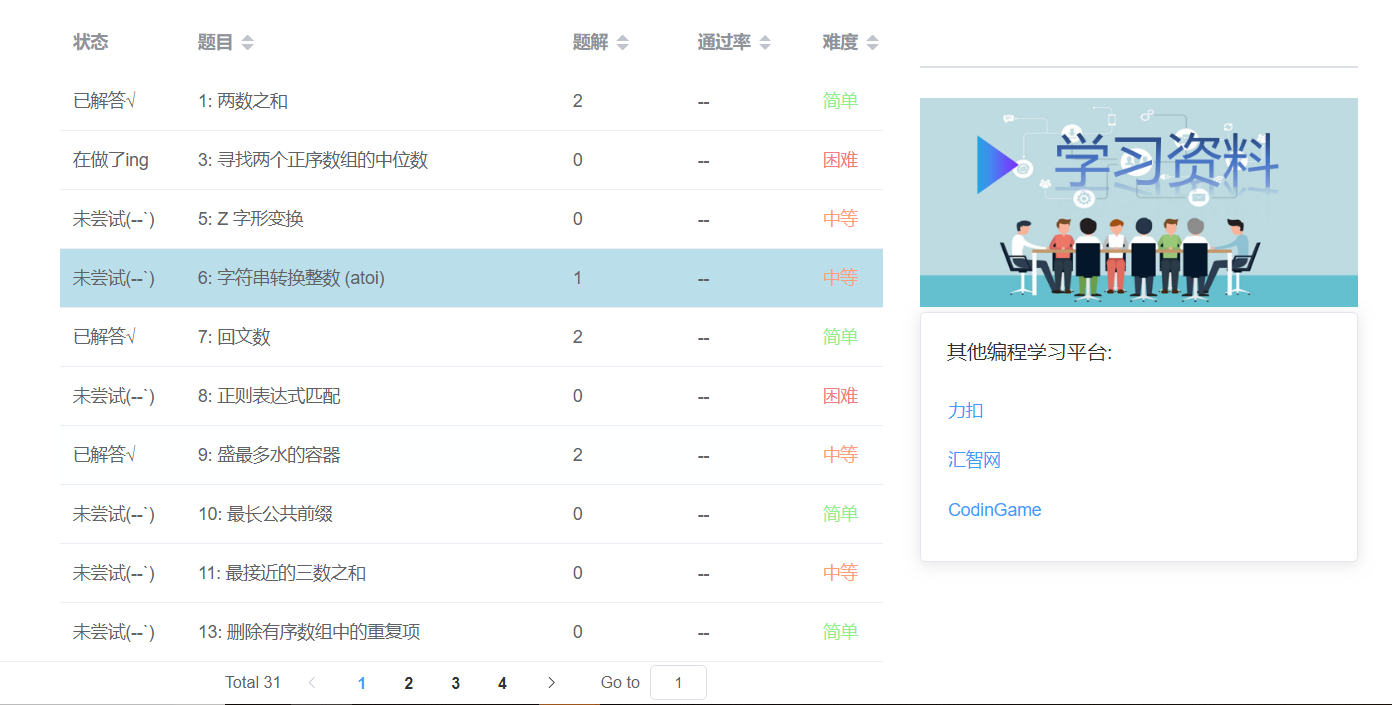


这是在线编译界面，用户可以在该页面进行代码编辑，通过Submit提交代码，后端可以运行代码后给出反馈。

#### OJ子系统：



这是题库主页面，用户可以在该页面查询题库，查看日历信息以及其他功能。用户可以在给出的筛选器中选择题目标签用来筛选题目，也可以通过搜索功能搜索自己想要的题目，二者可以兼容。用户可以点击某一道题目进入到题目的详细信息界面。



如上图所示，题库表格提供了跳转分页功能，同时右侧边栏给出了一些其他变成学习平台的跳转链接，以及点击学习资料图片可以跳转至在线工具子系统中的资料页面。



这是题目详细信息界面，用户可以在该页面查看题目的给定信息已经查看题解，查看提交记录，同时右边框中是用户编辑代码的页面，用户可以在该页面进行代码的编辑，会提前给出代码的一些初始内容以供用户继续书写。底部有一些按钮，其中随机一题是跳转到随机一道题目的页面，上一题和下一题可以跳转至对应的题目详细信息页面。而执行代码则会让用户手动输入参数来测试自己的代码，而不会提交至数据库作为提交记录保留。保存代码按钮可以让用户在点击之后保存当前书写的代码，在下次进入该题目时，记录会覆盖初始代码，让用户继续书写。提交按钮可以让用户提交代码并通过已有的样例进行测试，结果会记录在提交记录中以供用户查看。



这是题解界面，用户可以在该页面查看这道题的所有用户提交的题解，用户可以在该页面点击某一个题解来查看更详细的题解信息，包括思路和代码，同时用户可以点赞一个题解。用户也可以点击右上角的上传题解按钮来上传自己的解题方式。



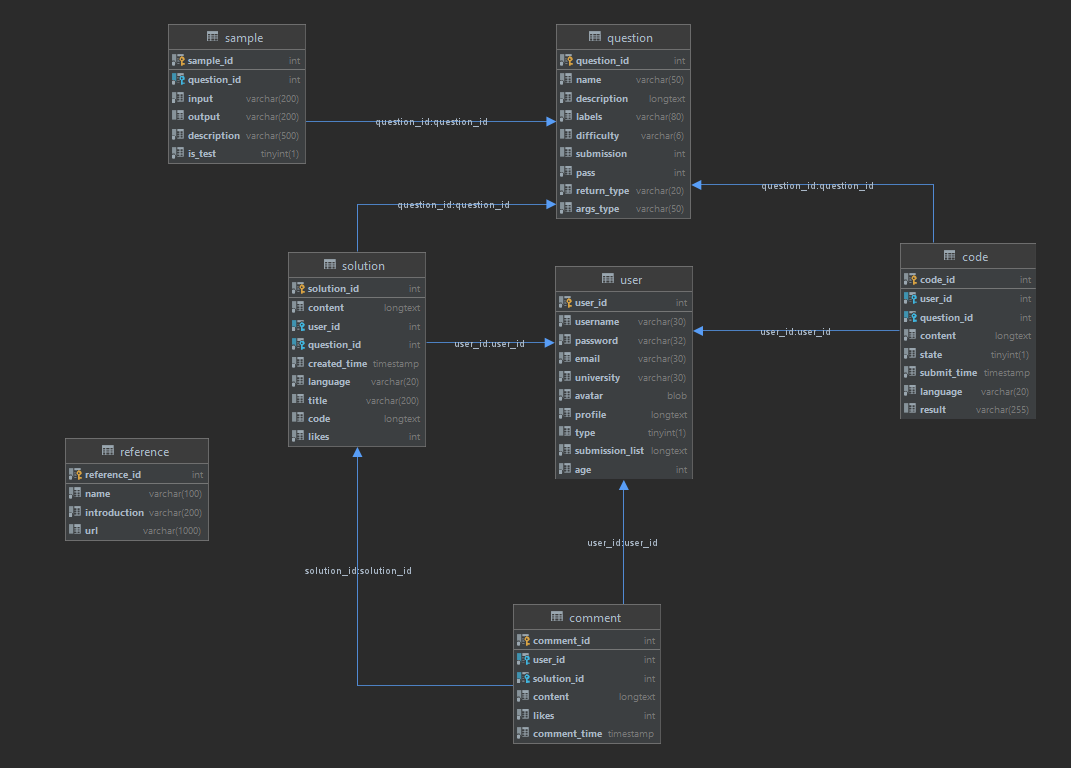
这是提交记录页面，用户可以在该界面查看自己的提交记录，同时用户提交代码的反馈也会在该页面显示。

## 2.5 数据库设计 @zjx

**编写目的：**

数据库的表结构设计在项目开发中至关重要。撰写数据库设计说明将详细描述HODJ—黄渡编程学习平台的数据库设计规划，并指导该系统的数据库各方面的内容，为开发人员、系统管理人员提供基准文档。同时，通过写数据设计说明，可以规范数据名称、数据范围、数据代码等，提高软件开发工作的效率。

**数据库总ER图：**

****

**详细说明：**

1.

表名： code 功能说明：保存提交代码的相关信息 表内容：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **列名** | **类型** | **属性** | **长度** | **描述** |
| **code\_id** | INT | PK,Not Null | / | 代码编号 |
| **user\_id** | INT | Not Null | / | 用户编号 |
| **question\_id** | INT | Not Null | / | 题目编号 |
| **content** | LONGTEXT | Not Null | / | 代码内容 |
| **state** | TINYINT | Not Null | 1 | 代码提交状态 |
| **Submit\_time** | TIMESTAMP | Not Null | / | 代码提交时间 |
| **Language** | VARCHAR | Not Null | 20 | 代码语言 |
| **Result** | VARCHAR | / | 255 | 代码运行结果 |

外键：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **字段** | **参考表** | **参考字段** | **删除时** | **更新时** |
| **user\_id** | user | user\_id | CASCADE | CASCADE |
| **question\_id** | question | question\_id | CASCADE | CASCADE |

2.

表名： comment 功能说明：保存题解评论的相关信息 表内容：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **列名** | **类型** | **属性** | **长度** | **描述** |
| **comment\_id** | INT | PK,Not Null | / | 评论编号 |
| **user\_id** | INT | Not Null | / | 用户编号 |
| **solution\_id** | INT | Not Null | / | 题解编号 |
| **content** | LONGTEXT | Not Null | / | 评论内容 |
| **likes** | INT | Not Null | / | 点赞数量 |
| **comment\_time** | TIMESTAMP | Not Null | / | 评论时间 |

外键：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **字段** | **参考表** | **参考字段** | **删除时** | **更新时** |
| **user\_id** | user | user\_id | CASCADE | CASCADE |
| **solution\_id** | solution | solution\_id | CASCADE | CASCADE |

3.

表名： question 功能说明：保存题目的相关信息 表内容：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **列名** | **类型** | **属性** | **长度** | **描述** |
| **question\_id** | INT | PK,Not Null | / | 题目编号 |
| **name** | VARCHAR | Not Null | 50 | 题目标题 |
| **description** | LONGTEXT | Not Null | / | 题目描述 |
| **labels** | VARCHAR | Not Null | 80 | 题目标签，描述题目类型 |
| **difficulty** | VARCHAR | Not Null | 6 | 题目难度 |
| **submission** | INT | / | / | 提交次数 |
| **pass** | INT | Not Null | / | 通过次数 |
| **return\_type** | VARCHAR | Not Null | 20 | 题目返回类型 |
| **args\_type** | VARCHAR | Not Null | 50 | 题目输入参数类型 |

4.

表名： reference 功能说明：保存学习资料的相关信息 表内容：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **列名** | **类型** | **属性** | **长度** | **描述** |
| **reference\_id** | INT | PK,Not Null | / | 资料编号 |
| **name** | VARCHAR | Not Null | 100 | 资料标题 |
| **introduction** | VARCHAR | Not Null | 200 | 资料相关介绍 |
| **url** | VARCHAR | / | 1000 | 资料的外部链接（可下载） |

5.

表名： sample 功能说明：保存测试/展示用例 表内容：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **列名** | **类型** | **属性** | **长度** | **描述** |
| **sample\_id** | INT | PK,Not Null | / | 用例编号 |
| **question\_id** | INT | Not Null | / | 题目编号 |
| **input** | VARCHAR | Not Null | 200 | 用例的输入 |
| **output** | VARCHAR | / | 200 | 用例的运行结果 |
| **description** | VARCHAR | Not Null | 500 | 用例描述 |
| **is\_test** | TINYINT | Not Null | 1 | 判断是测试还是展示用例 |

外键：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **字段** | **参考表** | **参考字段** | **删除时** | **更新时** |
| **question\_id** | question | question\_id | CASCADE | CASCADE |

6.

表名： solution 功能说明：保存题解的相关信息 表内容：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **列名** | **类型** | **属性** | **长度** | **描述** |
| **solution\_id** | INT | Not Null/PK | / | 题解编号 |
| **content** | LONGTEXT | Not Null | / | 题解内容 |
| **user\_id** | INT | Not Null | / | 用户编号 |
| **question\_id** | INT | Not Null | / | 题目编号 |
| **likes** | INT | Not Null | / | 点赞数量 |
| **created\_time** | TIMESTAMP | Not Null | / | 题解发布时间 |
| **language** | VARCHAR | Not Null | 20 | 题解所使用的编程语言 |
| **title** | VARCHAR | / | 200 | 题解的标题 |
| **code** | LONGTEXT | / | / | 题解的代码 |

外键：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **字段** | **参考表** | **参考字段** | **删除时** | **更新时** |
| **user\_id** | user | user\_id | CASCADE | CASCADE |
| **question\_id** | question | question\_id | CASCADE | CASCADE |

7.

表名： user 功能说明：保存用户的相关信息 表内容：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **列名** | **类型** | **属性** | **长度** | **描述** |
| **user\_id** | INT | Not Null/PK | / | 用户编号 |
| **username** | VARCHAR | Not Null | 30 | 用户昵称 |
| **password** | VARCHAR | Not Null | 32 | 账户密码 |
| **email** | VARCHAR | / | 30 | 用户邮箱地址 |
| **university** | VARCHAR | / | 30 | 用户所在的学校 |
| **avatar** | BLOB | / | / | 用户头像 |
| **profile** | LONGTEXT | / | / | 用户个人简介 |
| **type** | TINYINT | Not Null | 1 | 是否为管理员 |
| **submission\_list** | LONGTEXT | / | / | 用户提交的题解历史 |
| **age** | INT | Not Null | / | 用户的年龄 |

**数据库安全规则：**

1.对数据库中的数据定期备份以保障系统安全

2.数据库系统和所在的服务器设置用户的标识以鉴定是否是合法用户，保证数据安全。

3.密码传输皆采用md5 算法加密以代替明文传输，保证账户密码的传输安全

## 2.6 系统出错处理设计

### 2.6.1 出错信息



|  |  |  |
| --- | --- | --- |
| 报错方法 | 返回HTTP错误代码 | 报错内容 |
| UNKNOWN\_ERROR | 201 | 未知错误 |
| EMPTY\_RESPONSE | 202 | 没有信息 |
| INVALID\_ROLE | 204 | 无效角色 |
| INVALID\_PASSWORD | 205 | 无效密码 |
| INVALID\_SERVICE | 501 | 无效服务 |
| INVALID\_USER | 503 | 无效用户 |

### 2.6.2 补救措施

#### 2.6.2.1 无效用户，无效角色，无效密码问题

当用户使用无效账户、无效角色以及无效密码后，系统给出提示：“对不起，需要重新登录！”因为本系统不允许用户的匿名登录。之后返回到login/register界面，清空本地存储的token，需要用户重新登录并重新认证身份。

#### 突然断网，网页缓存清空

当用户正在网页上编写代码的时候，突然发生了断网问题，网页缓存清空，丢失了原先的代码。但是由于本系统有代码提交机制，所以数据库内存储了最后一次保存的代码信息。可以直接从数据库内调用，返回最近一次保存的代码。

#### 无效服务问题

当用户正在网页上提交代码的时候，突然发生了无效服务问题，说明服务器由于多人同时进行提交出现了卡顿问题，此时服务器端会线保存提交的代码，逐个运行并在5秒内返回初运行结果。

#### 没有信息问题

当用户进行操作之后返回信息为Null之后，系统会返回没有信息的警示，并提示用户是否提交了空代码等原因，需要用户重新刷新界面并进行操作。

#### 2.6.2.5 系统崩溃

由于数据在数据库中已经有备份，故在系统出错后可以依靠数据库的恢复功能，并且依靠日志文件使系统再启动，就算系统崩溃用户数据也不会丢失或遭到破坏；但有可能占用更多的数据存储空间，因此我们选择在夜间凌晨两点进行统一备份。

#### 2.6.2.6 数据丢失

由于固态硬盘，机械硬盘本身的特性，难以避免发生数据丢失的问题。因此我们统一在每周进行数据的二次备份。尽量避免了用户数据的丢失，保护了用户的合法权益，可以能够在用户与我们系统的沟通下进行进一步的丢失数据搜索。

### 2.6.3 系统维护设计

为了系统维护的方便，我们在程序内部设计中设置了在线编译模块类维护在线运行模块。这主要是为了如果没有编译成功的代码直接传输到服务器运行之后，有较大的可能会导致服务器崩溃、卡顿等多个问题。

因此我们设置了在线编译功能，在线检测传输的代码是否编译成功吗，如果编译失败，则直接先返回编译错误的信息，直到编译成功之后才传输到服务器进行运行。

这个模块主要涉及了Java动态编译，Java类文件的结构，Java类加载器& Java类的热替换，Java反射和将一个类变为线程安全类等技术。