**\*\*\*\*\*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》

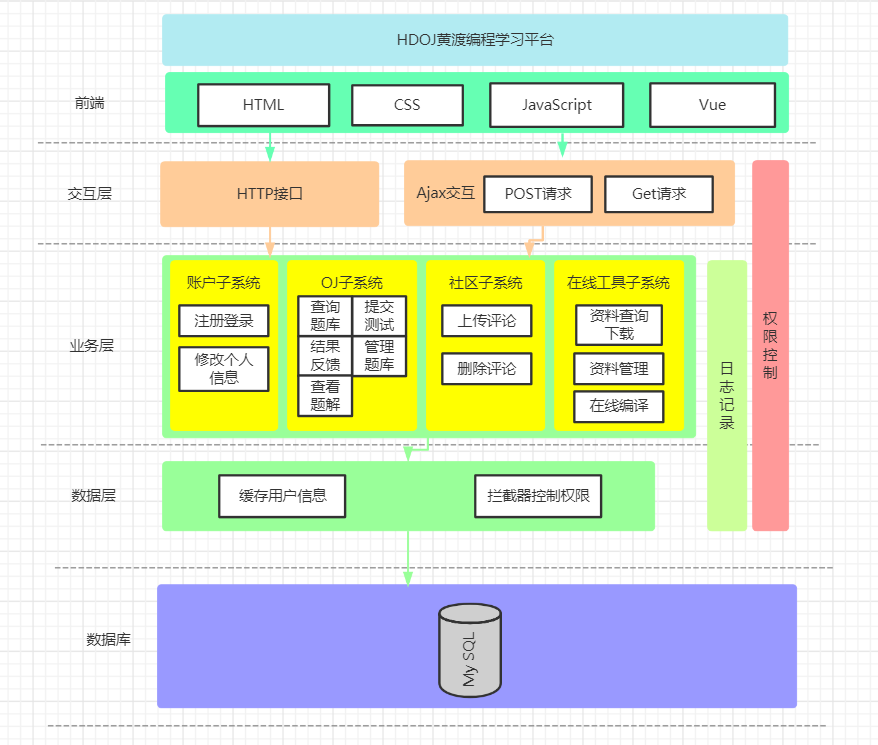
《软件工程：实践者的研究方法》

《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

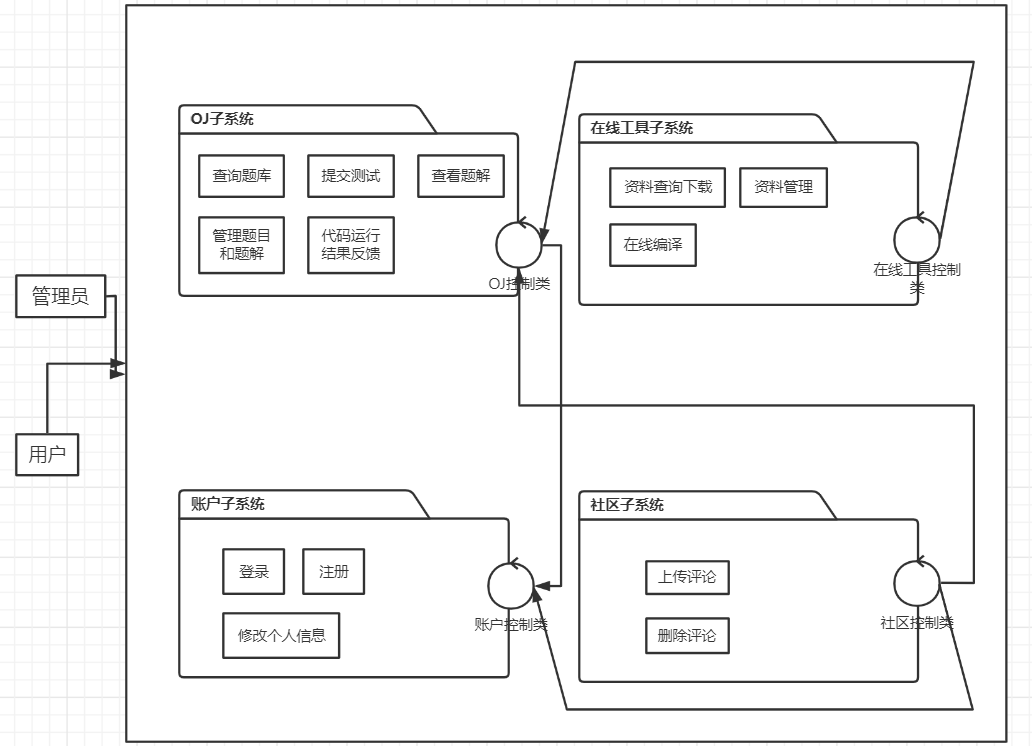
#### 资料下载

## 2.2. 系统软件结构设计

//对2.1中的系统总体架构设计图中的软件功能部分进行软件系统结构设计（或称软件//体系结构设计），设计的结果是子系统和或微服务和或类调用关系图，同时要进行文

//字描述。

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



## **2.3. 接口设计** @田溪 @王凌 @邹嘉祥 @yx

### 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 @邹嘉祥 @王凌 做界面，截图解释

对系统的界面进行设计，可以按照系统的层次来设计界面。界面设计可以使用原型模型设计方法。

## 2.5 数据库设计 @yx 不用画图，设计数据库，用文字描述信息

依据“软件需求（分析）规约（说明书）”的数据建模进行数据库逻辑设计和物理设计。指出本软件系统内部所使用的每个数据库结构的名称、标识符以及它们之中每个数据项、记录、文卷和系的标识、定义、长度以及它们之间的层次的或表格的相互关系。给出本系统内部所使用的每个数据库结构中的每个数据项的存储要求，访问方法、存取单位、存取的物理关系（索引、设备、存储区域）、设计考虑等。

## 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。