基于JAVA的SQL语句学习平台的设计与实现

计算机与信息科学学院 计算机科学与技术 2017级 周艳明

指导教师 陈 勇

**摘 要：**

SQL是目前应用最广泛的关系数据库语言，也是高校计算机及相关专业学生必学的一门语言之一。而学习该语言的重要途径是专项练习,在针对性的学习平台对SQL以往传统的练习方式由于学生得不到及时反馈使得学生的学习进度和学习积极性受到较大影响。

学生在学习SQL语言时,必须通过不断练习才能提高实践操作能力。而评判学生练习的正误和给予及时反馈是提高其学习积极性和学习能力的重要手段。文章基于Java的SQL语言自学习平台.主要针对SQLSERVER数据库设计一个SQL语言自动练习系统,主要难点是SQL语句正确性的判断。

当前系统针对个人博客的特性和在线编程的特性进行了整合，采用Spring、SpringMVC、MyBatis框架整合的一套B/S系统。该系统主要提供在线学习和在线编程等功能，能有效利用线上资源和平台无关等特点丰富个人知识体系。目前对传统个人博客进行了深入了解和研究发现了互联网带来的便利的同时促进了个人博客发展，发现存在一些美中不足。为了解决不相容问题，开发的当前系统不仅能让用户轻松学习到理论知识，还能立即上手实践，让枯燥的理论变得有趣，以此弥补传统博客的不足。

**关键词：**Spring；SpringMVC；MyBatis；B/S系统

**Abstract：**The current system for personal blog features and online programming features of the integration, the use of Spring, SpringMVC, MyBatis framework integration of a set of B/S system. The system mainly provides online learning and online programming functions, and can effectively utilize online resources and platform independent features to enrich personal knowledge system. At present, we have an in-depth understanding and research on the traditional personal blog and found that the convenience brought by the Internet has promoted the development of personal blog. In order to solve the problem of incompatibility, the current system developed not only makes it easy for users to learn theoretical knowledge, but also makes it easy for users to get started and make boring theories interesting, which makes up for the shortcomings of traditional blogs.

**Key words：**Spring；SpringMVC；MyBatis；B/S system

1. 项目概述
   1. 项目背景

一,引言

SQL是结构化查询语言(Structured Query Lan-guage)的缩写,目前应用最广泛的关系数据库语言。它具有功能齐全、方便实用、语法简洁等特点，包括数据查询、数据操纵、数据定义和数据控制四个功能，被用户广泛接受并成为国际标准。

SQL语言是高校计算机及相关专业学生必学的一门语言之一。而学习该语言的重要途径是练习,以往传统的练习方式由于学生得不到及时反馈使得学生的学习进度和学习积极性受到较大影响。本文尝试通过设计一个针对SQL SERVER的SQL语言自动练习系统, 在学生输入答案时,系统能马上评判该题,判断题目的正误,并提示出错原因。本系统较为关键的问题包括:

<https://www.ixueshu.com/document/214c9856c46110076f94dd9662cf01b3318947a18e7f9386.html>

<https://www.ixueshu.com/document/219dd97a879319e5.html>

互联网下的学习资源日渐丰富，在网络的环境下可以随处可以学习。丰富却也容易让人眼花缭乱，学习以后的东西如果不能及时消化，也会根据遗忘规律忘记于脑海深处。

编程是一件非常有趣的事情，却也存在枯燥在其中，学习过程异常的艰难，对于一些算法的理解更是让人谈之色变。如果能够在学习过程中及时自我实现，哪么将会让其变得有趣，用自己的算法去测试既定数据的正确性，更能激发人的学习欲望，同时也对这个算法有更加深入的理解，从害怕到喜欢的过程就是这么简单实在。

社区仍然是编程爱好者交流的天堂，在这里能看见一些人优秀的思维方式和简单有效的问题解决方案，供更多人去学习和分享自己对于不同算法的理解，从他人的思维中整理出更多自己能进行理解的思路，从而让自己的编程之路更顺利。

这个系统的设计，不仅仅是作用于知识学习，更多的是对自己已经学到的知识一个整体架构架设与实现，让自己能够更加充分理解语言之中的计算机玄学，也加深自己对整体架构的思考与深入，从而让自己的学习不仅能提高自己，也能辅以他人治学。

* 1. 项目目的

算法是一道难关，对于21世纪的互联网职业来说，单体时代已经过去，海量数据显现了时下互联网发展的趋势，普通的编程技术已经不能很好解决海量数据的处理，计算机硬件也遇到了瓶颈，只有利用好仅有的硬件资源通过提高算法能力的方式去处理海量数据的运算与分析已解决系统性能瓶颈。

这个系统以相关专业知识为铺垫，从数据结构入手，提升编程爱好者对语言的基础的认知，更加熟练使用一些已经封装好的数据结构。同时，辅以数据结构实现原理，从学习中更加深入理解一些数据结构的特性，让编程爱好者从表面应用到实现原理掌握，更好理解程序在计算机中的运行。随着学习的深入，可以自行练习算法，提升解决实际问题的能力。

1. **理论基础**
   1. 软件开发工具

互联网发展进入快速时代，开发工具也数不胜数，近年和Java相关的开发工具主要有Eclipse、InteliJ IDEA，其中IDEA因为更智能、布局更舒适获得广大Java程序员的喜爱，但是不像Eclipse是一个免费的开发工具，却可以提高开发效率、错误率低、视图更清晰以及集成开发工具多等特点，使得当前系统选择IDEA作为首选开发工具。程序员难免和数据库打交道，在业内使用最多的依然是Navicat，虽然这些工具都是收费应用，但是用起来错误率低还是非常友好的。

* 1. 软件开发技术
     1. B/S 简介

B/S结构是WEB时代最为流行的一种网络结构模式。浏览器成了主要的客户端，降低了系统维护成本，使得系统开发更趋于简单化，同时让系统更易于使用。虽然这种结构优点很多，却也存在不可避免的缺陷。在用户请求，服务器响应的模式下会在通信上造成巨大开销，对于用户的恶意请求容易导致系统和数据安全性问题。随着WEB飞速发展和人们网络依赖程度加深，B/S架构依然非常活跃。

* + 1. Java 简介

Java是当前TIOBE排行榜第一的语言，是一门纯面向对象的服务端开发语言，处理服务请求有着非常出色的表现，其生态在国内也比较完善，第三方开源框架丰富，具有较强的生命力。Java语言功能强大简单易用，利用JVM自动垃圾回收解决了困扰C++开发者手动回收垃圾的困扰，也是一门平台无关的开发语言，有着“一次编译，到处运行”的强悍移植能力。

* + 1. Spring 简介

Spring[1]是轻量级的J2EE开发框架，为解决企业应用开发复杂问题而生，有着高内聚低耦合、面向切面编程、声明式事务、降低J2EE开发难度等优点。其IOC容器更利于对象创建和依赖关系维护，AOP可以非常方便从横切面切入程序较好控制程序权限、运行监控等。极大降低了J2EE开发难度，使其由重变轻，更易于程序维护。

* + 1. SpringMVC 简介

SpringMVC[2]是WEB开发中MVC分层很重要的部分，核心实现是以Servlet为入口，简化了传统MVC对控制器、模型独享、过滤器以及处理程序对象的操作，让程序员更注重核心实现，使得分层更加简单和独立。还提供RESTful网络应用程序设计风格和开发方式，让请求更加隐蔽，提取核心数据更加变得不容易猜测，提升了网络服务的安全性。

* + 1. MyBatis 简介

MyBatis[3]作为一个开发者定制SQL的持久层框架，包含有动态SQL的特性，让条件可变的查库操作变得简单。封装了原生JDBC操作，避免了JDBC和手动设置参数等操作和结果集获取，从而更加方便。由传统的CRUD变为对象映射完成数据库操作。其和Spring整合非常简单，降低了开发难度，提高了开发效率，是一款非常简单易学、灵活的开源免费框架。

* 1. 系统架构

当前系统采用目前WEB主流B/S架构完成整体开发，有助于减少项目开发、维护成本。B/S架构的优点是可以减少软件的安装，使用浏览器就能完成请求和响应，可以避免客户端不必要的性能消耗。能够在浏览器里面通过请求网址完成相关的操作，方便了用户操作和浏览。程序跑在服务器上，管理员只需要管理好服务器就行了，提高服务器的承载能力从而减轻用户负载。

* 1. 数据库系统

MySQL[4]是一个免费开源的关系型数据库，体积小、性能好、成本低、多操作系统支持等优点，可视化工具支持多。MySQL在速度、可靠性、适应性都能满足企业级需求。在存储、维护和应用上，MySQL都是比较简单，满足标准的情况下提供了一些简单的SQL支持，从而使得书写SQL也是一件简单的事情，语义上非常适合开发者。

1. **系统应用分析**
   1. 系统需求分析

该系统是一个个人博客和在线编程结合学习的B/S系统，具有个人博客和在线编程的优点。让用户在线及时学习知识还能随时随地进行实践巩固，节约了用户真机实操的成本，也降低了时间成本。当前系统主要围绕开发人员进行的系统开发，简单高效突出重点，让知识更易获取、实践吸收。当前系统前后端完全分离采用AJAX发送JSON数据，支持独立接口，可扩展性强；采用了平台无关语言，增加了系统可移植性。

* 1. 业务需求分析

互联网发展越来越迅速，学习资源越来越广，对学习的掌握程度越来越难，只有实践加上理论才能学到更多,掌握越牢固对职业生涯越有帮助。可以结合理论和实践的需求越来越迫切。移动的编程也成为了当前开发者进步的重要过程，随处可编程可以充分利用业余时间实现一些可能混淆的地方，巩固知识的理解。

* 1. 功能需求分析

进入互联网的时代，越来越多办公都希望能够网上完成，协同完成一个任务，非常及时和方便。伴随着知识上网的动作，博客越来越多，分享的知识越来越深，对于消化理解的能力不断升级，如果能将理论和实践都能同时解决，将是顺应发展趋势的一个重大变革。现有需求如下：

1. 在线学习理论知识。
2. 在线完成理论知识实现。
3. 本地开发环境开发配置工具。
   1. 技术可行性分析

当前系统部署在服务器上，对于客户端要求不需要很高，多终端运行无压力，主要从硬件资源和软件资源进行分析。

1. 软件环境

软件环境从服务器和终端进行分析，正常运行的环境包括：1）操作系统；2）数据库；3）语言环境；4）开发工具；5）部署环境。当前系统可运行软件环境如表3.1所示。

表3.1 可运行软件环境

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 操作系统 | 数据库 | 开发语言 | 开发工具 | 环境部署 |
| CentOS 7 / 8 | MySQL 8 | Java | 无 | Nginx + Tomcat |
| Windows 10 | MySQL 8 | Java | Intelij IDEA | IDEA Tomcat |

1. 硬件环境

当前系统开发硬件和实际运行环境相关性不是很大，仅作为参考。开发过程中使用到的硬件资源如表3.2所示。

表3.2 可运行硬件环境参考

|  |  |
| --- | --- |
| CPU | Intel CORE i5 / 高通骁龙 652 |
| 硬盘 | 1T / 32G |
| 内存 | 8G / 3G |

* 1. 经济可行性分析

当前系统部署环境需要一台独立的云服务器，简单部署就能运行系统，用户访问当前系统只需要记录网址就能够访问系统资源，获取资源内容。从软件方面对于系统成本就能非常容易控制，降低了客户端维护、版本迭代更新的成本。平台可移植性好，借助浏览器就能多终端运行，可以说成本控制已经非常友好。从硬件方面对于系统运行最低配置要求不高，所以服务器硬件只要能够稳定运行就能满足日常需求，如果访问量过高可能需要配置更高的硬件环境。相对于单独的客户端已经控制好了成本问题。

1. **系统设计**
   1. 系统流程
2. 新用户注册

新用户需要注册一个账户进行登录才能访问，系统内部资源。系统内部设置了拦截器去询问当前是否有用户登录，如果用户登录以后则能访问，否则给出提示信息。

1. 用户登录

用户登录以后刷新整个页面，访问主界面，可以通过导航栏进入指定的模块获取模块内容。

1. 动态实现分享知识

在每个模块内部，有代码编辑区域，可以在当前区域完成编码工作以及代码结果返回显示等信息，增加理论和实践的契合度，填充学习理论知识的空虚。用户还能对当前模块分享的理论进行评价，或者给出自己的理解，一千个人心中有一千个哈姆雷特，能够快速组建起一个强大的知识体系。

1. 访问环境配置小工具

万事开头难，这些小工具能够快速将本机环境搭建起来，非常有利于初学者快速搭建环境投入编码学习工作，从而减轻了第一步压力，能够将时间充分利用到学习新知识上。

1. 系统优化反馈

一个系统的稳定运行少不了用户的建议和提议，这能够使开发者能够快速抓取用户的喜好，做出用户喜欢的东西，从而长久稳定的运行下去，因此这部分是倾听用户心声的支持模块。

1. 系统工作流程如图4-1所示。

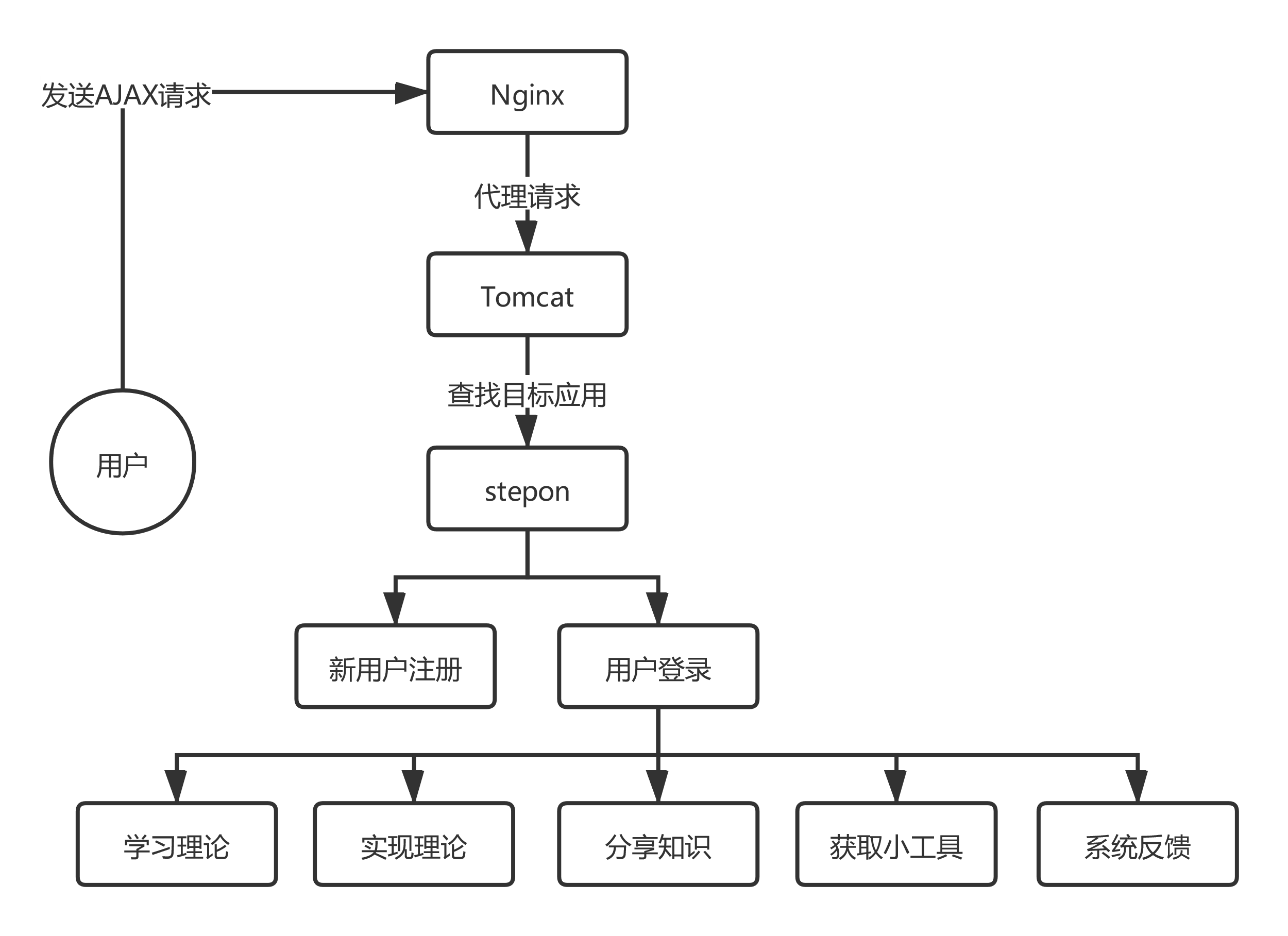


图4-1 系统工作流程

1. JVM编译原理[5]如图4-2所示。

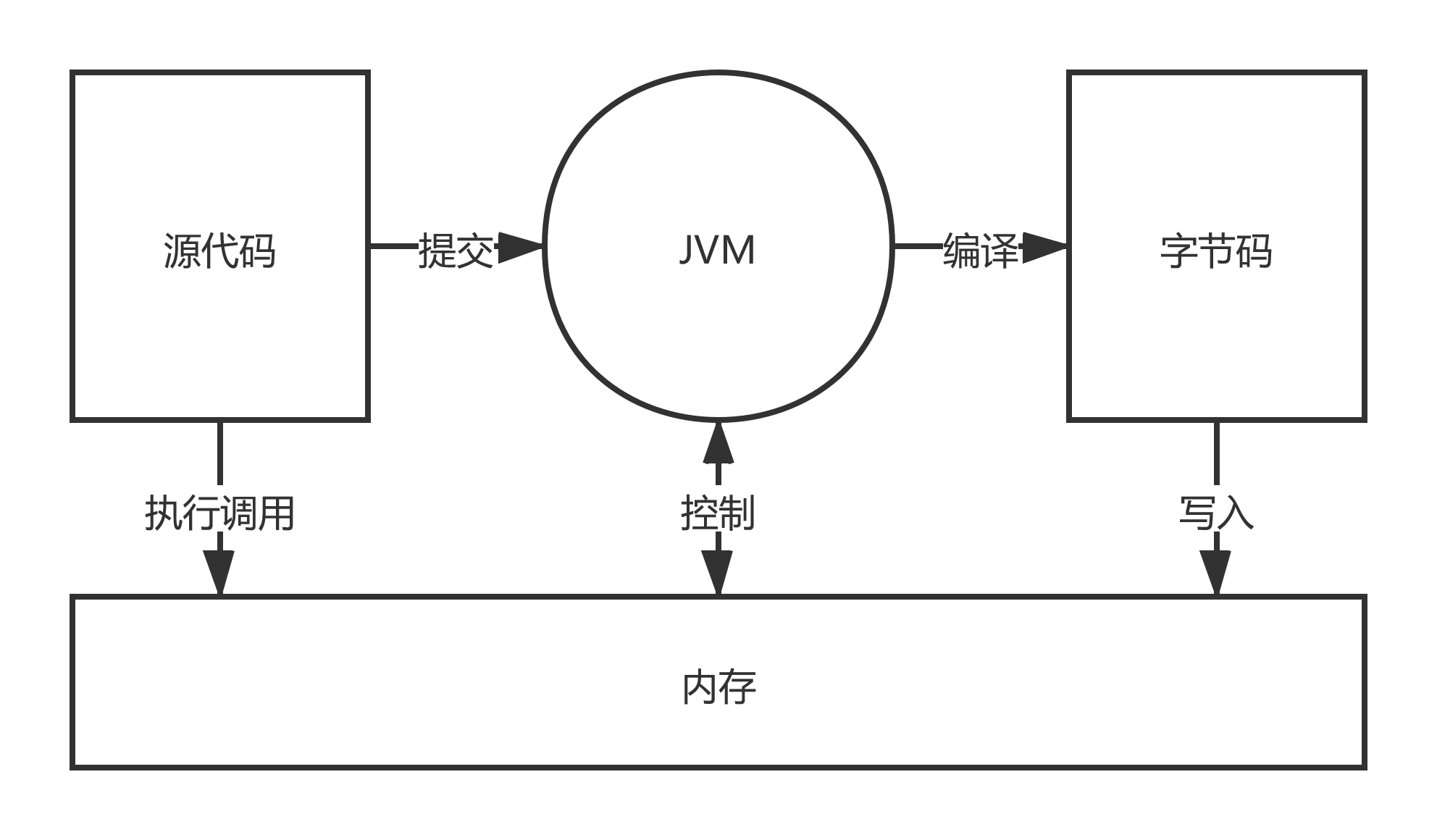


图4-2 JVM执行流程

* 1. 系统结构设计

1. 代理服务器：Nginx

将用户请求通过Nginx[6]代理后发送给后端真实WEB容器有利于程序的安全性，不会直接暴露真实的后端WEB容器。Nginx的并发量非常高，能有效解决高并发场景的发生，是一个比较高性能的代理服务器。代理情况如图4-3所示。

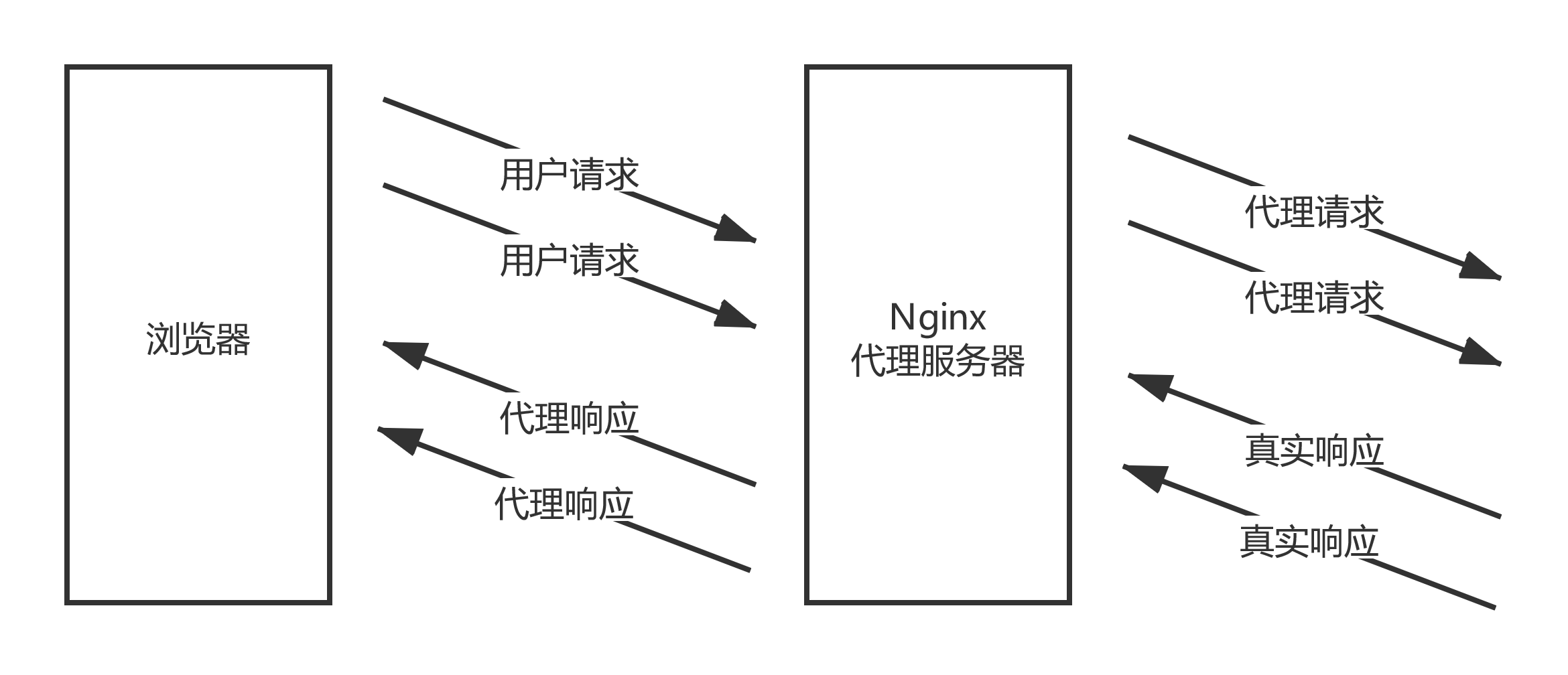


图4-3 Nginx代理流程

1. WEB容器：Tomcat

真实处理用户请求的容器，将代理发送的请求经过逻辑处理后再通过代理反馈给用户，使得当前容器功能相对单一和独立，处理性能有所上升，还能进行分布式集群以此减轻单个容器的负担。处理情况如图4-4所示。

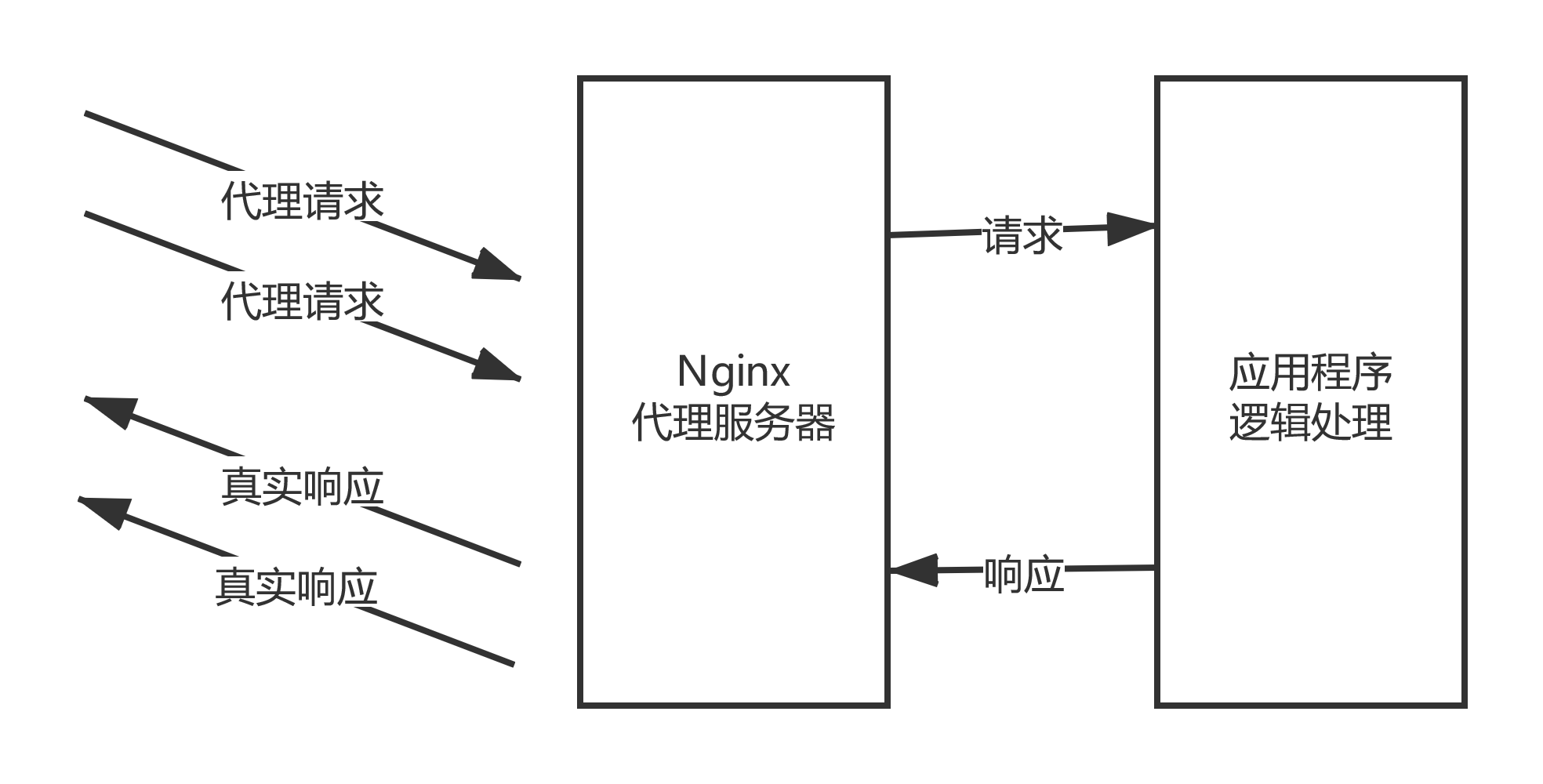


图4-4 Tomcat处理流程

* 1. 系统模块设计

1. 数据结构模块

这一部分主要进行一些数据结构理论知识分享，包含多语言数据结构（目前仅包含Java有关的数据结构）。通过新增数据结构节能点，丰富当前数据库内容，点击目标内容查看相关数据结构的具体内容。整体流程如图4-5所示。

1. 算法练习模块

算法是未来程序员应当掌握的知识，这些灵活的理论不仅能练习开发者逻辑思维，还能精简冗余代码，养成良好编码习惯。能通过新增算法练习丰富数据库，点击目标内容可以查看详细内容。整体流程如图4-5所示。

1. 小工具模块

快速辅助开发者搭建相关软件系统环境，快速投入正式开发中，能有效减轻开发者负担，一步到位不再繁琐。整合百度多种方法集合到批处理命令中，让文字代码化，步骤更清晰，辅以手动实践模式快速理解环境配置。整体流程如图4-5所示。

1. 反馈模块

距离用户最近的地方，能够让用户的声音直接到达开发者眼前。反馈信息能够通过点赞数进行排序，让更多用户反馈的问题能够及时得到解决，满足大多数需求的情况下能够兼顾小问题。整体流程如图4-5所示。

用户登录

开始

登录成功？

No

数据结构

算法练习

小工具

反馈

Yes

新增目标内容

显示目标条目

显示目标详细内容

点击目标 ？

No

Yes

添加到目标数据表

结束

验证表单 ？

No

Yes

图4-5 数据结构/算法练习/小工具/反馈处理流程图

* 1. 数据库设计

1. 根据系统设计所需数据库实体-属性图，如图4-6、图4-7、图4-8、图4-9、图4-10、图4-11、图4-12，以及系统总体E-R图4-13所示：

user

datastructure

图4-6 用户表实体-属性图 图4-7 数据结构表实体-属性图

tool

datastructure\_history

图4-8 数据结构提交历史表实体-属性图 图4-9 小工具表实体-属性图

feedback

comment

图4-10 用户评论表实体-属性图 图4-11 反馈表实体-属性图

practice

图4-12 练习表实体-属性图

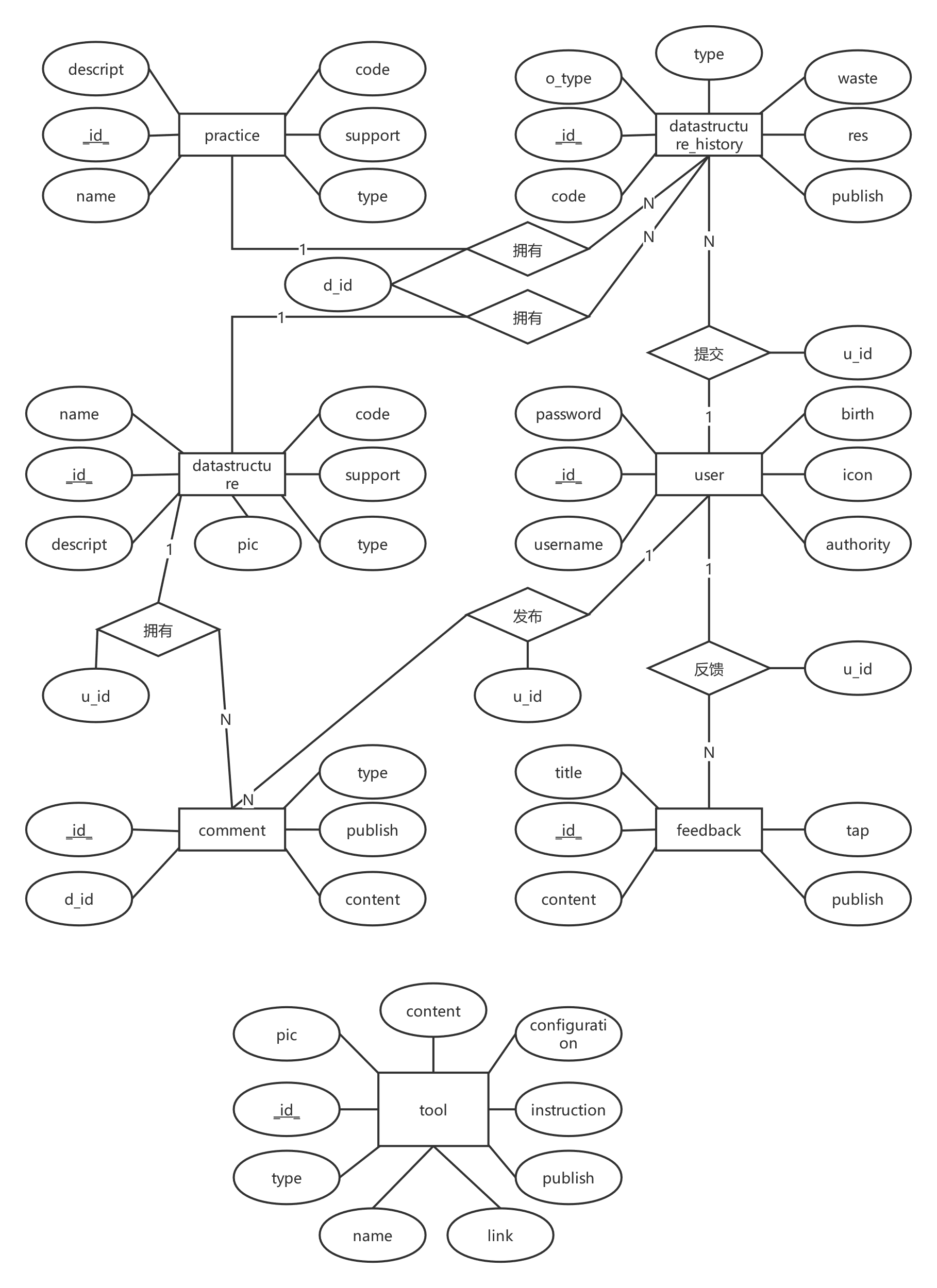
****

图4-13 系统总体E-R图

1. 用户表设计，如表4.1所示。

表4.1 用户表

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 字段名 | 类型 | 长度 | null | 主键 | 外键 | 候选码 | 注释 |
| 1 | user\_id | int | 11 | false | √ |  |  | 用户id |
| 2 | username | varchar | 255 | false |  |  |  | 用户姓名 |
| 3 | user\_head | varchar | 255 | true |  |  |  | 用户头像 |
| 4 | user\_introduction | varchar | 1000 | true |  |  |  | 用户简介 |
| 5 | password | varchar | 255 | true |  |  |  | 用户密码 |
| 6 | role\_id | int |  | false |  |  |  | 用户角色 |
| 7 | create\_time | datetime |  | false |  |  |  | 注册时间 |

1. 技能点表设计，如表4.2所示。

表4.2 技能点表

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 字段名 | 类型 | 长度 | null | 主键 | 外键 | 候选码 | 注释 |
| 1 | skill\_id | int | 11 | false | √ |  |  | 技能id |
| 2 | skill\_topic | varchar | 255 | true |  |  |  | 技能主题 |
| 3 | simple\_skill\_content | varchar | 3000 | true |  |  |  | 简化内容 |
| 4 | skill\_content | varchar | 3000 | true |  |  |  | 技能内容 |
| 5 | skill\_type | varchar | 255 | true |  |  |  | 技能类型 |

1. 角色表设计，如表4.3所示。

表4.3 角色表

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 字段名 | 类型 | 长度 | null | 主键 | 外键 | 候选码 | 注释 |
| 1 | role\_id | int | 11 | false | √ |  |  | 角色id |
| 2 | role\_name | varchar | 255 | false |  |  |  | 角色名字 |
| 3 | create\_time | datetime |  | false |  |  |  | 创建时间 |

1. 评论表设计，如表4.4所示。

表4.4 评论表

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 字段名 | 类型 | 长度 | null | 主键 | 外键 | 候选码 | 注释 |
| 1 | comment\_id | int | 11 | false | √ |  |  | 评论id |
| 2 | skill\_id | int | 11 | true |  | √ |  | 技能id |
| 3 | user\_id | int | 11 | true |  | √ |  | 用户id |
| 4 | comment\_content | varchar | 3000 | true |  |  |  | 评论内容 |
| 5 | comment\_time | datetime |  | true |  |  |  | 评论时间 |

1. 收藏表设计，如表4.5所示。

表4.5 收藏表

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 字段名 | 类型 | 长度 | null | 主键 | 外键 | 候选码 | 注释 |
| 1 | collect\_id | int | 11 | false | √ |  |  | 收藏id |
| 2 | user\_id | int | 11 | false |  | √ |  | 用户id |
| 3 | skill\_id | int | 11 | false |  | √ |  | 技能id |
| 6 | collect\_time | datetime |  | true |  |  |  | 收藏时间 |

1. 反馈信息表设计，如表4.6所示。

表4.6 反馈信息表

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 字段名 | 类型 | 长度 | null | 主键 | 外键 | 候选码 | 注释 |
| 1 | id | int | 31 | false | √ |  |  | 自增主键 |
| 2 | title | int | 31 | false |  |  |  | 反馈标题 |
| 3 | content | int | 31 | false |  |  |  | 反馈内容 |
| 6 | publish | date | 0 | false |  |  |  | 反馈时间 |
| 7 | tap | int | 31 | false |  |  |  | 点击次数 |
| 8 | u\_id | int | 31 | false |  | √ |  | 反馈用户id |

1. 练习表设计，如表4.7所示。

表4.7 练习表

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 字段名 | 类型 | 长度 | null | 主键 | 外键 | 候选码 | 注释 |
| 1 | id | int | 10 | false | √ |  |  | 自增主键 |
| 2 | name | varchar | 255 | false |  |  |  | 题目名 |
| 3 | descript | varchar | 2000 | false |  |  |  | 题目描述 |
| 4 | code | varchar | 10000 | false |  |  |  | 源代码 |
| 5 | type | varchar | 255 | false |  |  |  | 题目类型 |
| 6 | support | int | 31 | false |  |  |  | 点赞数 |

1. **系统实现预览**
   1. 简单主界面

主界面采用大方简介的界面直接指明主要模块，让用户清楚明白这个网站究竟有什么作用，可以快速定位到达指定模块功能。如图5-1所示。

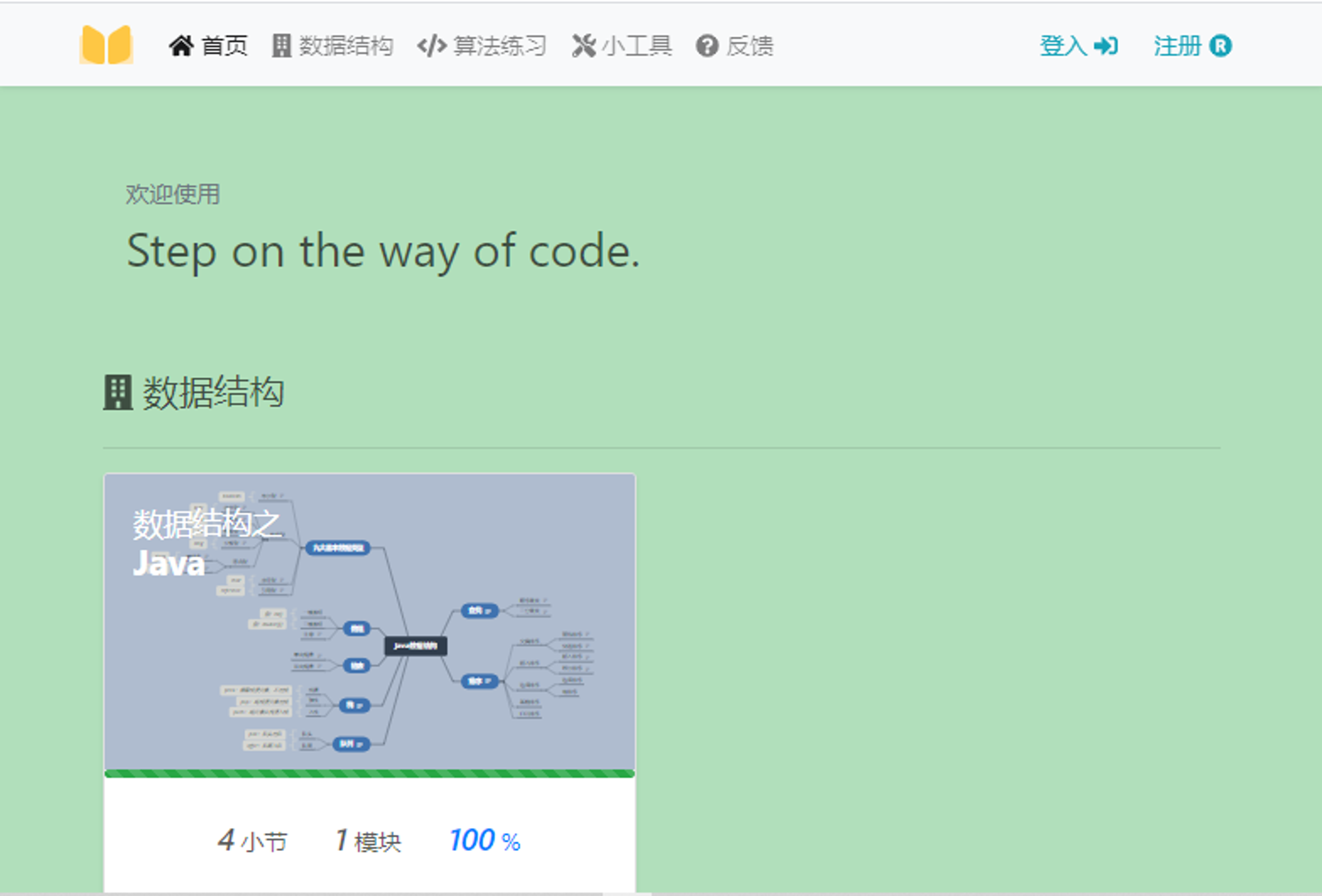


图5-1 主界面效果

* 1. 登录/注册界面

登录/注册界面采用模态框进行展示，能减少独占页面从而丰富页面内容。系统设计超级管理员和普通用户群，超级管理员能够轻松完成所有操作，普通用户能分享自己的思想和所学。如图5-2，5-3所示。



图5-2 登录界面效果



图5-3 注册界面效果

* 1. 数据结构界面

数据结构分为多语言模块，目前仅支持Java模块，其他语言尚未涉及，能后续完善。整体简单大气有条理，目录更清晰。如图5-4所示。



图5-4 数据结构效果

* 1. 页面详情界面

详情页面显示目标的内容描述，用户评论情况，用户提交历史记录，预览源代码和编辑源代码，最后提交执行源代码。这是一个理论和实践集合的网页，将理论知识再次代码体现，能加深开发者印象，增加用户体验。如图5-5所示。



图5-5 页面详情界面效果

* 1. 小工具界面

小工具可以辅助开发者快速搭建本机环境，并提供手动搭建步骤，主要以Windows小工具为主，后续可能会增加Linux小工具（服务器一般使用Linux）。当前小工具只包含了Java开发相关工具，用更多时间投入开发，更多压力给操作系统处理。并且在GitHub提供源代码可以参照改改写写能够更适合自己。如图5-6所示。



图5-6 小工具界面效果

* 1. 反馈界面

距离用户最近的是能听见用户的心声，这里就是一个用户和开发者交流的平台，显示结果是根据点赞排序的，越前面的越是迫切，就能让开发者先看见然后加以完善，处理好用户和开发者关系，能稳住用户，也能让程序向着用户期望的方向去发展。如图5-7所示。

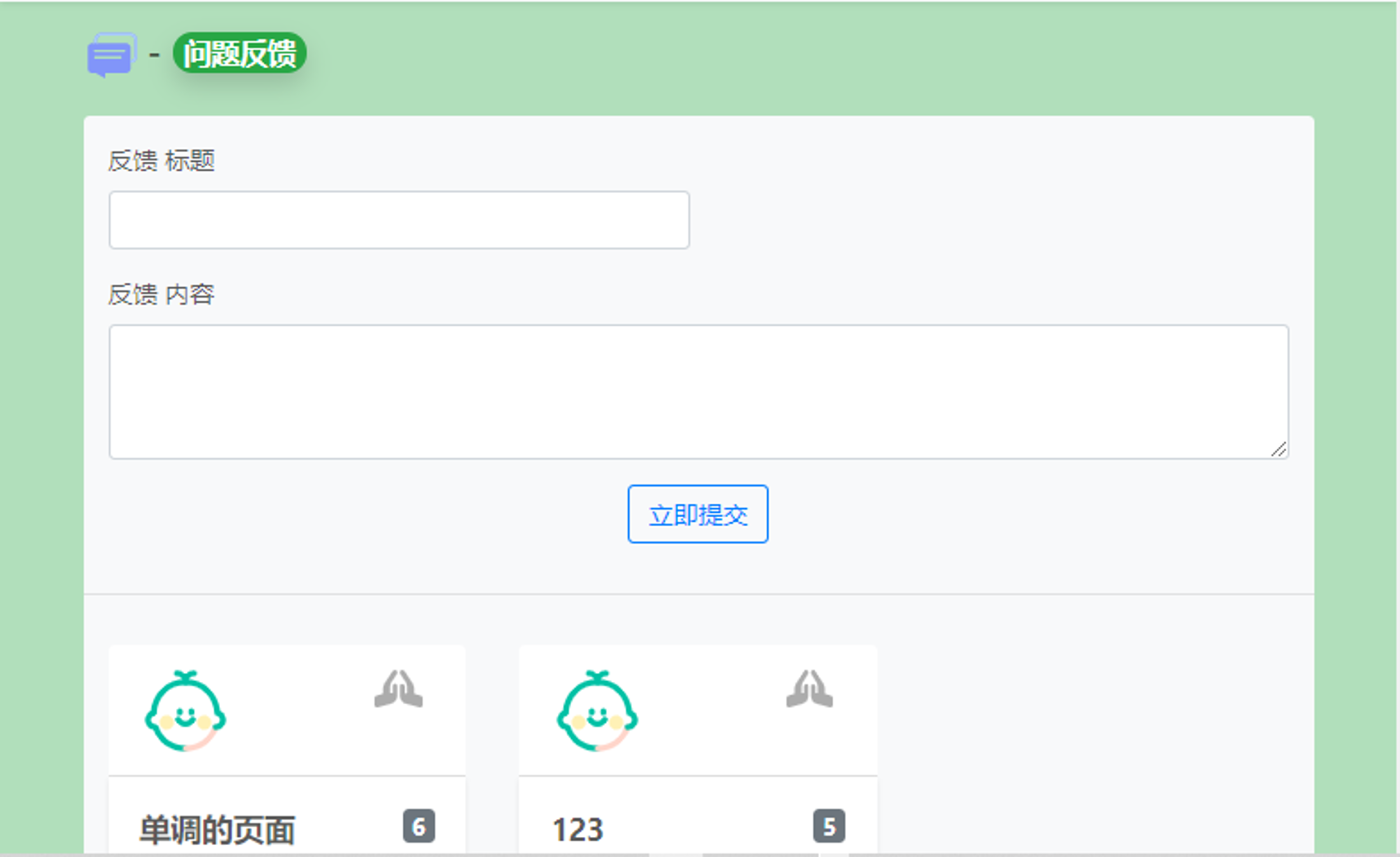


图5-7 反馈界面效果

1. **结束语**

当前系统已经能够完成用户云学习的期望，弥补了传统博客只携带理论干货，缺少动手实践的功能，已经将现有趋势向前推进。当然系统还有很多可扩展的功能，在后续可以继续完善。

1. 优化方面

系统整体是以模块化为主扩展方便，当前缺少更多语言支持，后续提供更多语言支持。数据增多也可能会导致高并、海量数据等问题，可以在后续继续关注板块进行升级。

1. 框架升级

当前页面布局采用Bootstrap[7]、Jquery[8]渲染而成，这样直接操作dom树是一件非常消耗性能的工程，可以在后续使用Vue[9]进行前端页面渲染。后端虽然使用SSM框架进行处理请求，但是随着当前业务需求，后面可能会使用整合型框架SpringBoot简化开发，加上微服务[10]将各个模块单独运行。

参考文献:

[1] 柳伟卫. Spring5开发大全[M]. 北京:北京大学出版社, 2018:200-255.

[2] 沃尔斯，布雷登巴赫，等. Spring in Action[M]. 北京:人民邮电出版社, 2008:59-78.

[3] K. Siva Prasad Reddy. Java Persistence with MyBatis 3[M]. UK:PACKT PUBLISHING, 2013:60-90.

[4] Baron Schwartz，Peter Zaitsev，Vadim Tkachenko，等. 高性能MySQL[M]. 北京:电子工业出版社, 2013(3):35-256.

[5] 周志明. 深入理解Java虚拟机[M]. 北京:机械工业出版社, 2011:90-99.

[6] 陶辉. 深入理解Nginx[M]. 北京:机械工业出版社, 2016:20-50.

[7] 赵丙秀，张松慧，等. Bootstrap基础教程[M]. 北京:人民邮电出版社, 2018:25-100.

[8] 单东林，张晓菲，魏然，等. 锋利的jQuery[M]. 北京:人民邮电出版社, 2012:90-150.

[9] 张耀春. Vue.js权威指南[M]. 北京:电子工业出版社, 2016:35-155.

[10] 沙鲁巴·夏尔马. Java微服务[M]. 北京:电子工业出版社, 2017:100-236.