



毕业设计（论文）

题 目： 基于 J2EE 的
 OJ 系统的设计与实现

学 生： 朱建华

指导老师： 鲍春波 副教授

院 系： 信息科学与工程学院

专 业： 软件工程

班 级： 软件工程 1202

学 号： 3120305222

2016 年 6 月

福建工程学院本科毕业设计（论文）作者承诺保证书

本人郑重承诺：本篇毕业设计（论文）的内容真实、可靠。如果存在弄虚作假、抄袭的情况，本人愿承担全部责任。

学生签名：

年 月 日

福建工程学院本科毕业设计（论文）指导教师承诺保证书

本人郑重承诺：我已按有关规定对本篇毕业设计(论文)的选题与内容进行了指导和审核，且提交的毕业设计（论文）终稿与上传至“大学生论文管理系统”检测的电子文档相吻合，未发现弄虚作假、抄袭的现象，本人愿承担指导教师的相关责任。

指导教师签名：

年 月 日

目 录

1.	选题背景.....	1
1.1	选题来源	1
1.2	选题主要研究和解决的问题及技术要求	1
1.3	选题的目的和意义	1
2	方案论证.....	2
2.1	系统平台环境	2
2.1.1	硬件平台	2
2.1.2	软件平台	2
2.2	相关技术	2
3	系统分析.....	4
3.1	系统架构分析	4
3.2	需求分析	7
3.2.1	数据流图 DFD.....	7
4	系统设计.....	9
4.1	数据库设计	9
4.1.1	概念模型设计	9
4.1.2	逻辑结构设计	9
4.2	总体设计	14
4.2.1	系统模块层次结构图	14
4.2.2	系统功能模块描述表	15
4.3	详细设计	16
4.3.1	评测系统	16
4.3.2	比赛系统	18
4.3.3	讨论系统	20
4.3.4	挑战模式系统	20
4.3.5	考试系统	21

5	系统实现.....	23
5.1	MVC 的设计模式.....	23
5.2	使用 JAVA 代码生成 HTML 代码.....	24
5.3	使用 JQuery 和 JSON 动态调整 HTML 代码.....	26
5.4	对 JDBC 的封装.....	27
	总结与展望.....	28
	致谢语.....	29
	参考文献.....	30

基于 J2EE 的 OJ 系统的设计与实现

摘要：本文介绍了一种利用 J2EE 的技术架构实现的一个具有完整功能的 OJ 系统。本系统使用 J2EE 的技术架构，MVC 的框架模式，并介绍了不同于传统 MVC 框架的实现方式。使用 Mysql 数据库存储，利用面向对象的方式实现了 oj 的判题、比赛、讨论等功能，并实现了 OJ 的虚拟判题功能。本系统旨在帮助 ACM 集训队更好地进行日常训练，指引每年的新生训练方向，还能方便地举办每年学校举办的 ACM 竞赛，并且为校内的 ACM 推广贡献一份力量。

关键字：在线评测系统；OJ；ACM

The Design and Development of Online Judge System Based on J2EE

Abstract: This paper introduces a kind of OJ system, which is realized by the technology of J2EE. The system uses the J2EE technology architecture, MVC framework model, and describes the implementation of different from the traditional MVC framework. Using Mysql database storage, the use of object-oriented approach to achieve the OJ of the sentence title, competition, discussion and other functions. And realize the function of virtual judge. The system aims to help ACM training team daily training, guidelines for the annual orientation direction, but also easily held every year the school held the ACM contest and to contribute to the promotion of ACM on campus.

Key words: Online Judge System; OJ; ACM

1. 选题背景

1.1 选题来源

近几年来，我校参与 ACM 竞赛（国际大学生程序设计竞赛，以下简称 ACM 竞赛）越来越频繁，为此，我校组建了校 ACM 集训队，准备每年的 ACM 竞赛，而 ACM 竞赛的主要比赛内容是利用编程在短时间内解决一些问题，通过解决问题的效率来排名。比赛过程中的题目都是用机器判断的。因此在平时训练时，需要一个和比赛环境类似的训练环境，通过机器判断来更好的模拟比赛环境。这样的系统，我称之为 Online Judge 系统（简称 OJ 系统）

虽然市面上已经有其他大学的 OJ 系统了，但是我校在 ACM 竞赛上尚处于起步阶段，对题目难度的要求比较低，其他 OJ 系统上的题目难度区间大，并且没有归类，不适合我校学生的训练，因此，集训队迫切地需要一个适合我校情况的 OJ 系统。

1.2 选题主要研究和解决的问题及技术要求

本系统需要解决的问题主要是集训队的日常训练，以及承办校级的程序设计竞赛。另外，需要能够有效地帮助集训队的管理者方便使用。所以本 OJ 系统基于 J2EE，利用 MVC 思想的方式，后端使用 JAVA 语言，前端使用 HTML5，并通过 javascript 等语言进行页面渲染等方式。并将平时训练成绩，刷题结果等存入 MySQL 数据库保存。并提供一些数据统计等操作给集训队管理者方便地查看训练效果。

1.3 选题的目的和意义

我校的 ACM 处于刚起步阶段，需要一个更完善，更适合本校集训的 OJ 系统，针对本校的基本情况，可以实现更多更适合的功能系统，并且一定程度上激励起学生的兴趣，给集训队员们入门和初始的学习方向。还有对校内其他不了解 ACM 的学生进行宣传和推广 ACM 竞赛，使学习更能发现人才，在 ACM 竞赛中获得好成绩。

2 方案论证

2.1 系统平台环境

2.1.1 硬件平台

硬件平台是 WEB 服务器的搭建平台，只需要一个普通的 PC 机，使用 windows 系统的服务器即可。

2.1.2 软件平台

软件平台是系统需要的开发环境和运行环境，合理的环境可以使系统开发更加迅速，使用更稳定。经过研究和比对，本系统的使用的软件平台如下：

运行环境：JDK1.8 及以上、JavaEE 5.0 及以上、Tomcat 6.0 及以上、mysql 5.0 及以上、mingw32-gcc-4.7 及以上

开发环境：windows、mysql、tomcat

开发工具：Intellij Idea、Chrome

运行环境主要涉及的是 Web 服务器的环境，包括 mysql 和 tomcat。gcc 是用来评测代码的工具。开发工具中，Intellij Idea 是一款非常友好的 Java 开发工具，Chrome 是带有开发者工具的浏览器，主要用于调试 web 前端。

2.2 相关技术

名词定义：

本 oj：即本系统。

目标 oj：有的题目的测试数据不是在本系统，而是在网上的其他 oj 系统上，则题目实际是在测试数据所在的 oj 上评测的，该题目的测试数据所在的 oj 称为题目的目标 oj。

本地评测：题目的测试数据在本系统的 web 服务器上，用户提交的代码直接在本系统的 web 服务器上完成评测。

虚拟评测：题目的测试数据在目标 oj 上，web 服务器通过转发提交到目标 oj 进行评测的过程称为虚拟评测。

进行调用，来实现本地判题。

北大的评测机核心程序是一个.exe 程序，使用 java 的 Runtime 类来执行这个.exe 程序，按照指定格式设定好这个.exe 程序的运行参数，就可以实现本地判题功能。运行参数可以指定评测机的时限、空间限制、评测路径、标准输入输出文件列表，然后该评测机核心程序会返回评测结果。

3 系统分析

3.1 系统架构分析

本系统的核心功能即为判题功能，由用户提交代码，由服务器去评测。

由于 java web 服务器每收到一个 http 请求，就使用一个线程来单独处理这个请求。假如是由处理 http 请求的线程去执行虚拟判题，则是如图 3-1 这样的模型。

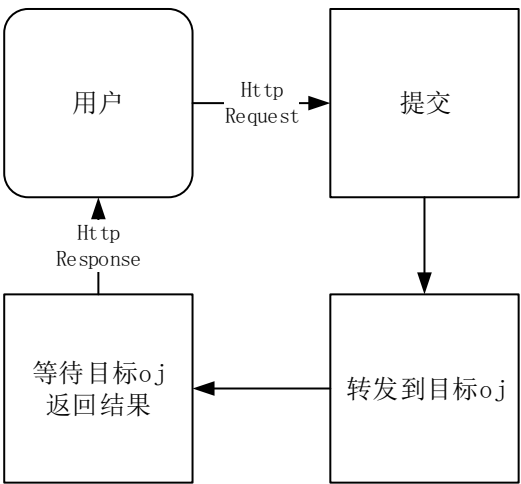


图 3-1 基本模型

这种模型有一个致命的缺点，就是用户从发送 request 请求到服务器返回 response 经过了非常长的时间，这个时间可能达到几秒到几分钟的时间，这个延迟是不能忍受的，因此，这样的模型是不可行的。

改进方案 1: 发送 http request 请求之后，另外创建一个线程去执行虚拟判题，

然后直接返回 response。如图 3-2 所示：

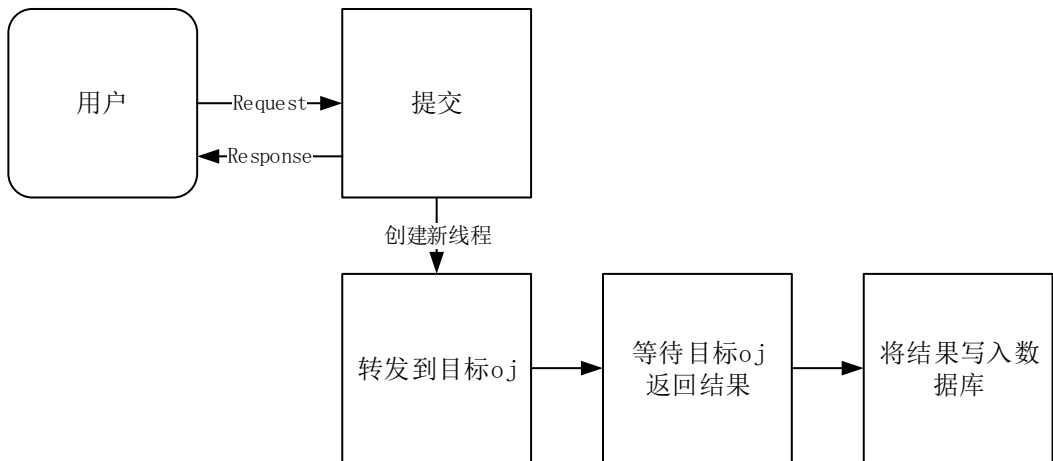


图 3-2 改进方案 1 模型

这个模型解决了基本模型中的问题，可以在极短的时间内返回 response。但是在进行虚拟判题时必须使用目标 oj 的用户名密码来提交，而是本 oj 创建的目标 oj 用户的数量是有限的。我们不妨将这里创建的新线程称为“提交器”，当短时间内有大量提交的时候，必然会在同一时间创建了大量的提交器，这样的话，提交器的数量就有可能超过用户数量，那么就有的提交器无法获得账号密码。并且每次提交都创建一个新的提交器也是非常浪费系统资源的，因此可以使用生产者——消费者模型来处理这个问题。

由于评测不能马上能获得结果，并且由于比赛等实际情况中，可能有短时间内的大量提交，因此，设计出生产者——消费者模式来处理。使用任务队列存放每个人的提交。系统预先创建好多个提交器，提交器不断去取任务队列中的提交执行虚拟判题。这样就可以得到改进方案 2 的模型，如图 3-3 所示：

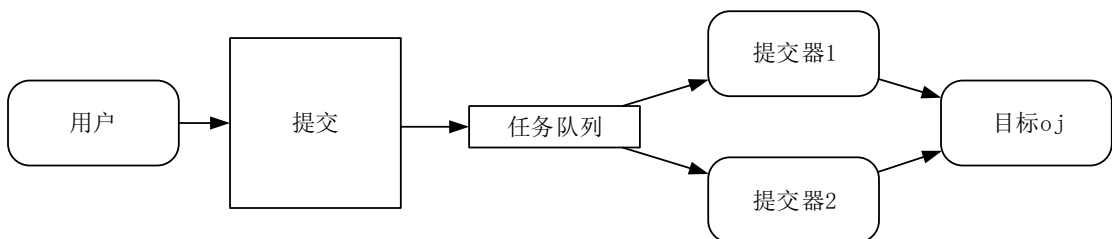


图 3-3 改进方案 2 模型

模型 2 已经基本实现了生产者——消费者模型，但是仍然有一个问题。由于目标 oj 是多样化的，假如在短时间内发生了多次目标 oj1 的提交，此时目标 oj1

的用户已经用完，任务队列处在积压的阶段，但是任务队列中可能还有一些不是目标 oj1 的评测请求，由于它们不是队首元素，因此它们也是处于等待阶段的，但是其他目标 oj 的用户名却是处于空闲状态，所以这样的模型仍然存在资源浪费的弊端。

根据实际情况，每道题目的目标 oj 是不同的，因此可以在用户提交的时候给该提交分类，不同的 oj 使用不同的任务队列，给每个提交器分配一个专用的目标 oj 用户密码，这样创建的提交器的数量等于目标 oj 用户的数量，这样就指定了提交器是属于哪个 oj 的，然后由提交器去它对应的队列中取得提交记录，然后再执行虚拟评测。

最终的总体架构的设计如图 3-4 所示：

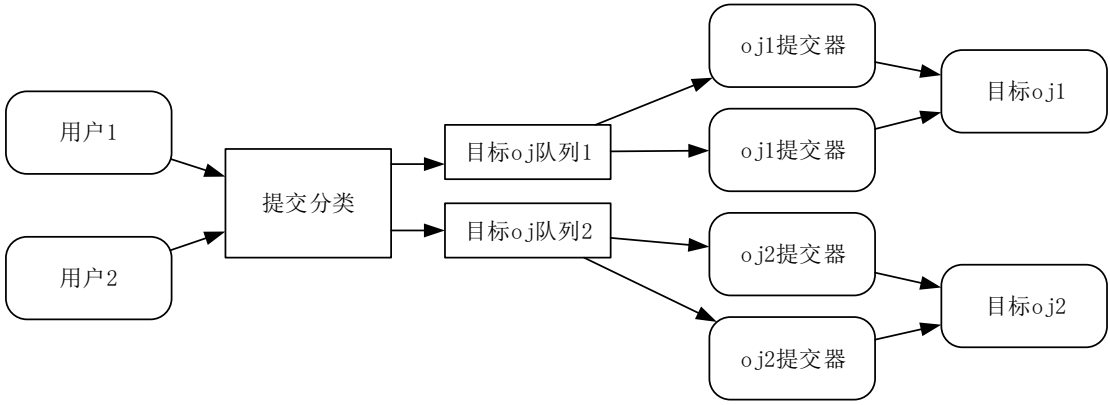


图 3-4 最终方案模型

这样的模型较之前的模型已经具有非常大的优势了，而且 oj 提交器也可以替换成本地 oj 的评测机。因此最终决定以本模型来控制本 oj 的判题功能流程。

3.2 需求分析

3.2.1 数据流图 DFD

本系统的评测模块的顶层数据流图，如图 3-2 所示

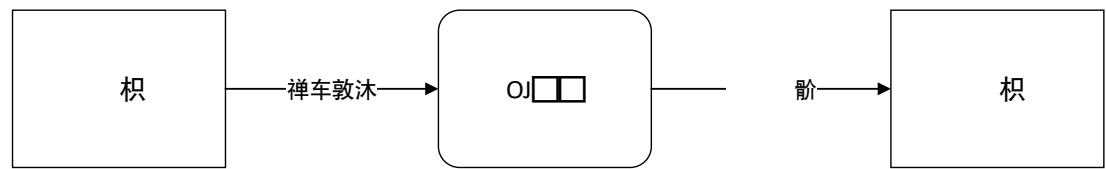


图 3-5 顶层数据流图

本系统的评测模块的零层数据流图，如图 3-3 所示

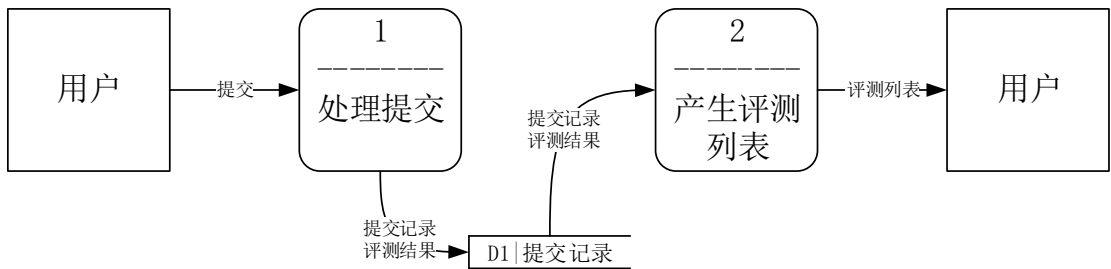


图 3-6 零层数据流图

本系统的评测模块的一层数据流图，如图 3-4 所示

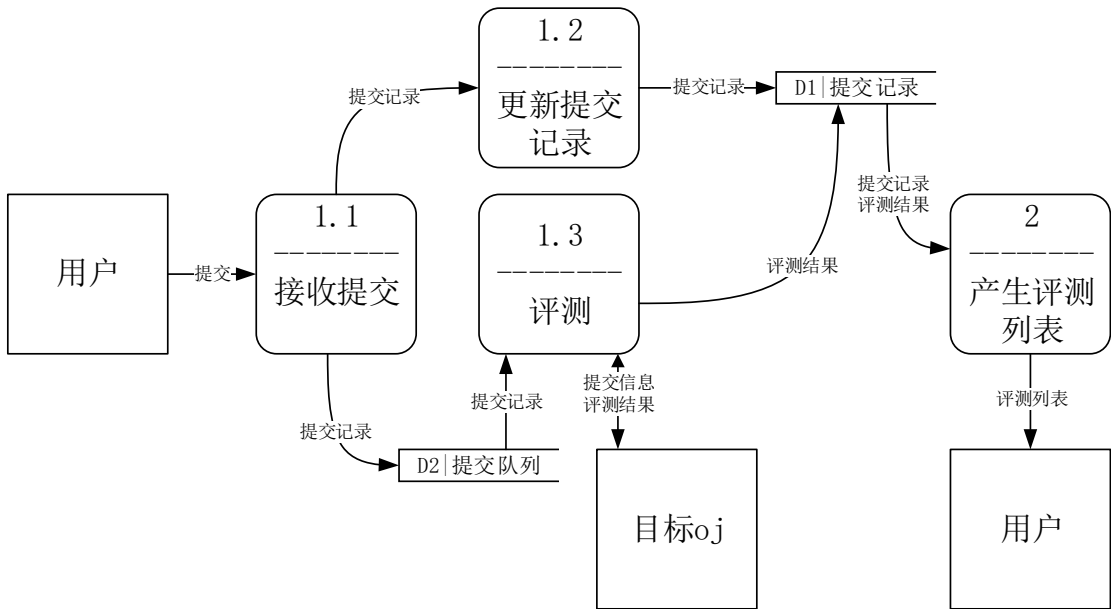


图 3-7 一层数据流图

本系统模块的二层数据流图，如图 3-5、3-6 所示。

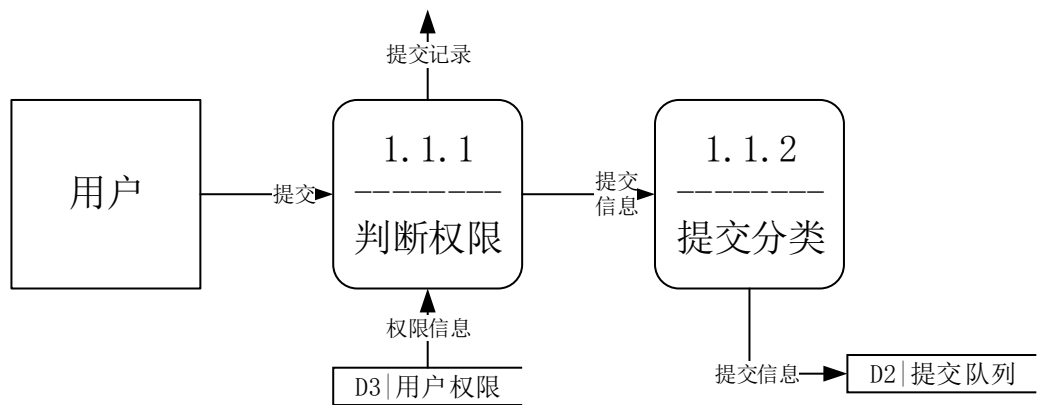


图 3-8 二层数据流图（1.1 分解）

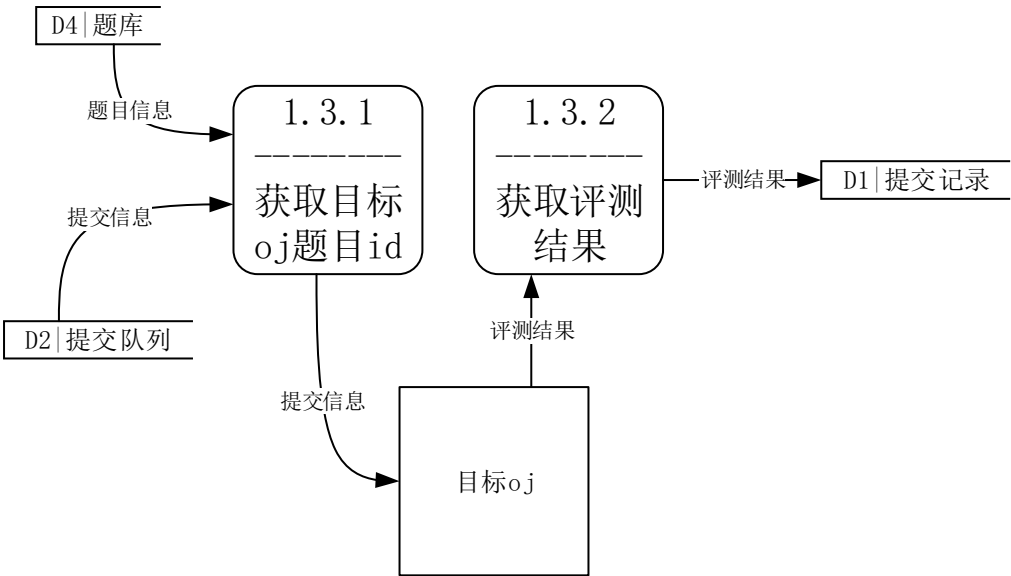


图 3-9 二层数据流图（1.3 分解）

4 系统设计

4.1 数据库设计

4.1.1 概念模型设计

系统的基本模型 ER 图如图 4-1 所示：

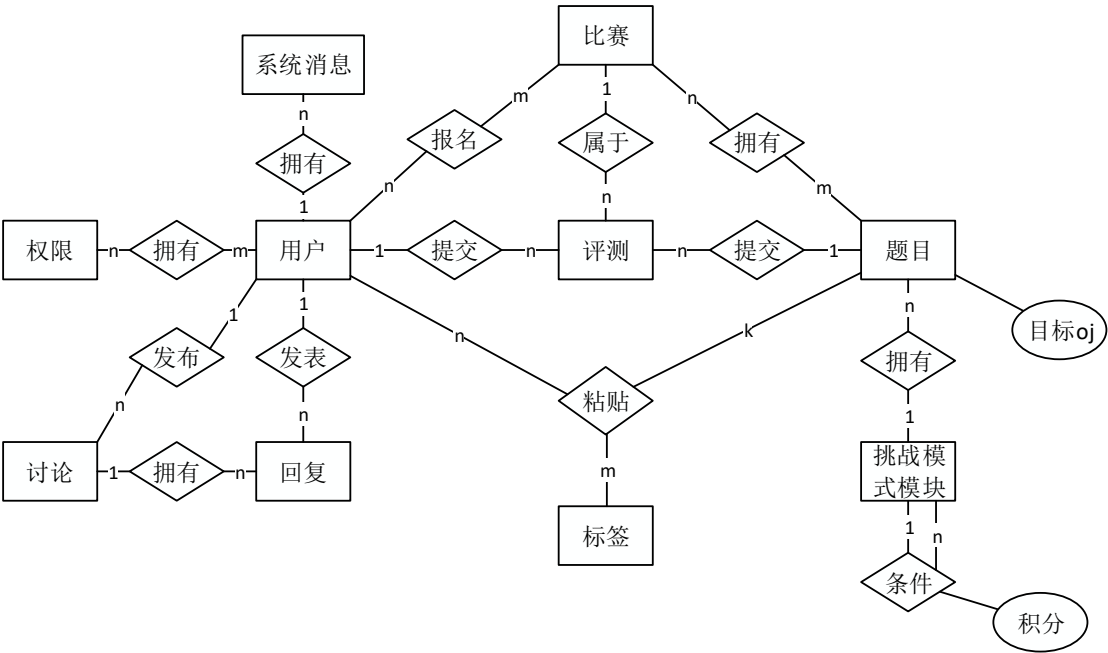


图 4-1

4.1.2 逻辑结构设计

基于对访问速度的考虑，实际建表时的表结构并不是完全按照 ER 图所示的那样，但是基本的概念模型是不变的。

用户表：

表 4-1 用户

字段名	类型	大小	主键	外键	描述
username	varchar	30	是		用户名
password	varchar	32			密码，用 MD5 加密
nick	varchar	50			昵称
gender	int	11			性别
school	varchar	50			学校
Email	varchar	50			学校
motto	varchar	100			签名
registertime	datetime				注册时间
type	int	11			
Mark	varchar	100			标记
rating	int	11			积分
ratingnum	int	11			参加积分赛的场次
acb	int	11			
name	varchar	30			姓名
faculty	varchar	30			学院
major	varchar	30			专业
cla	varchar	30			班级
no	varchar	30			学号
phone	varchar	30			联系电话
acnum	int	11			通过的题目数量
inTeamStatus	int	11			集训队员状态
inTeamLv	int	11			集训队员等级

权限表：

表 4-2 权限

字段名	类型	大小	主键	外键	描述
id	int	11	是		
name	varchar	20			

用户权限表

表 4-3 用户权限

字段名	类型	大小	主键	外键	描述
username	varchar	30	是	是	
perid	int	11	是	是	

题目表

表 4-4 题目

字段名	类型	大小	主键	外键	描述
pid	int	11	是		题目 id
ptype	int	11			题目类型 (本地/其他 oj)
title	varchar	80			标题
ojid	int	11			目标 oj 的 id
ojspid	varchar	10			题目对应 目标 oj 的 题目 id
visiable	int	11			可见性
author	varchar	300			作者

比赛表

表 4-5 比赛

字段名	类型	大小	主键	外键	描述
id	int	11	是		比赛 id
name	varchar	100			比赛名称
beginTime	datetime				开始时间
endTime	datetime				结束时间
rankType	int	11			排名类型
ctype	int	11			比赛类型
password	varchar	30			比赛密码
registerstarttime	datetime				报名开始 时间
registerendtime	datetime				报名结束 时间
info	text				比赛说明
computerating	int	11			是否计算 rating
createuser	varchar	30		是	创建人
kind	int	11			比赛分类

比赛题目表

表 4-6 比赛题目

字段名	类型	大小	主键	外键	描述
cid	int	11	是	是	比赛 id
pid	int	11	是		题目 id
tpid	int	11		是	题目在题 库中的 id

评测表

表 4-7 评测

字段名	类型	大小	主键	外键	描述
id	int	11	是		
ruser	varchar	20		是	提交人
pid	int	11		是	题目 id
cid	int	11			比赛 id
lang	int	11			代码语言
submitTime	datetime				提交时间
result	int	11			评测结果
timeUsed	varchar	10			运行时间
memoryUsed	varchar	10			使用内存
code	text				代码
codelen	int	11			代码长度

比赛报名表

表 4-8 比赛报名

字段名	类型	大小	主键	外键	描述
cid	int	11	是	是	比赛 id
username	varchar	30	是	是	用户名
status	int	11			报名状态
info	varchar	50			提示信息
time	datetime				报名时间

其他表结构在此不一一赘述，诸如挑战模式模块、讨论模块等。

4.2 总体设计

4.2.1 系统模块层次结构图

本系统模块间层次分明，如图 所示，包括评测系统、比赛系统、讨论系统、挑战模式系统、考试系统等模块构成。

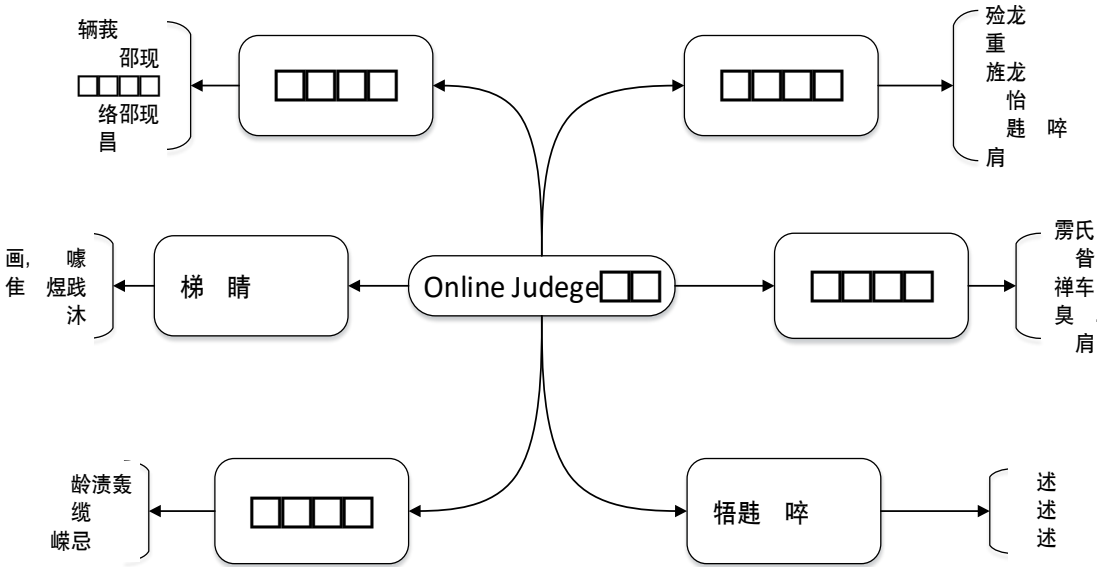


图 4-2 系统模块层次结构

4.2.2 系统功能模块描述表

系统模块分成六大块，如下表所示：

表 4-9 各模块功能说明

序号	系统名称	说明
1	基础功能系统	一般用户可以注册、登录，管理员用户可以对其他用户进行权限操作。
2	评测系统	一般用户可以在 OJ 上提交任意公开的题目，系统会自动对新提交进行评测，用户可以看到所有自己提交的代码和评测结果。管理员可以对已经评测过的代码进行重判。
3	比赛系统	一般用户可以参加公开的比赛，也可以报名其他需要报名的比赛。管理员可以审核比赛报名。管理员可以创建比赛、给比赛添加题目、更改比赛的排名方式等。
4	讨论系统	一般用户可以发起讨论，可以回复公开的讨论。管理员可以将某些讨论置于首页显示。
5	挑战模式系统	一般用户可以查看挑战模式的各个模块，包括模块说明、模块题目等。可以查看自己的挑战进度。管理员可以更改挑战模式的各个模块的说明、开启条件、题目。
6	考试系统	教师用户可以创建比赛，可以在考试中导入学生账号，学生账号可以参加考试，教师可以看到考试结果。

4.3 详细设计

详细设计将针对除了基础功能模块外的其他五个模块一一进行详细解释，其中着重说明评测系统和比赛系统。

4.3.1 评测系统

生产者——消费者模式。生产者即为所有要提交评测的用户。用户通过填写网页上的表单提交到 web 服务器，服务器自动获取该提交应该转发到哪个目标 oj，将改评测数据存入数据库，返回评测数据的 id，并将该提交放入指定的 oj 对应的任务队列中。

用户提交流程图：

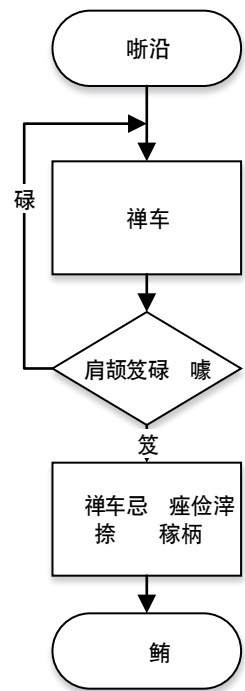


图 4-3 用户提交流程

提交器根据目标 oj 的数量被分为几类，每一类不断地去读取指定目标 oj 对应的任务队列中的数据，然后转发给目标 oj。提交的过程中，一般按照以下几个步骤进行：

- ①目标 oj 提交前都要进行登录操作。服务器模拟 http 的 post 提交对指定 oj 执行登录操作。
- ②提交代码，服务器模拟 http 的 post 请求对指定的 oj 执行提交操作。

③提交成功后解析目标 oj 的评测结果页面，每隔一段时间访问一次，判断评测是否结束。如果结束则获取评测结果返回。

提交器流程图：

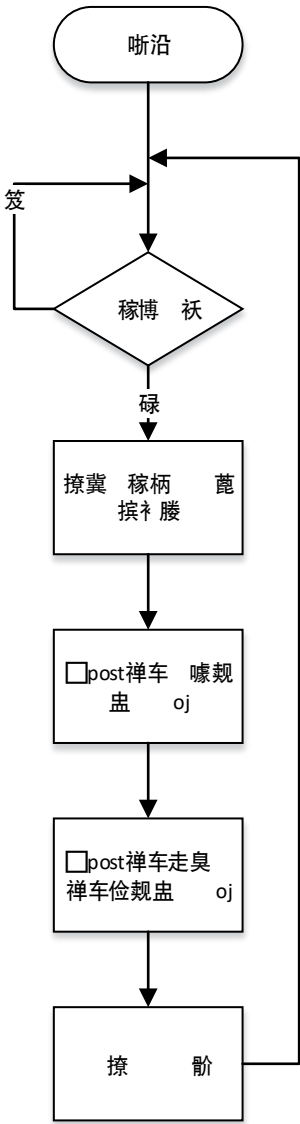


图 4-4 提交器流程

任务队列利用 JAVA 中的 `LinkedBlockingQueue` 类来实现。每个提交器是一个单独的线程，不断地执行提交操作。

这只是一个最简单的提交模型，还有很多其他的业务逻辑需要处理。比如判断权限、观战模式等。这里就将这些细枝末叶都舍弃，还原一个最原始的模型呈现。

4.3.2 比赛系统

比赛系统是用于满足平时训练、训练赛、正式赛而开发的一个模块。比赛主要有

由于需求繁多，比赛被分为如下几类：

- 1、练习：满足队员们平时练习，可以将题库中的某一类题目放在一场练习赛中。本模式下，所有属于比赛题目的提交都会被计入排名，包括在其他比赛中提交的，和直接在题库中提交的。
- 2、积分：集训队一般情况下每周会进行周赛，这种比赛比较正式，是集训队内部定期举行的训练赛。在比赛结束后，管理员可以根据每个人的比赛排名计算出积分（rating）。
- 3、娱乐：一些比较具有娱乐性的比赛，如短码赛、正式比赛热身赛（一般题目比较搞怪，或者是压力测试的题目）
- 4、正式：正式比赛在需要报名的情况下，必须完整地填写个人信息。这个比赛主要是用于正式的个人赛，对应我校每年举办的新生赛。
- 5、组队：组队赛在报名时需填写组队信息，对应我校每年举办的校赛。

比赛权限被分为以下几类：

- 1、公开：比赛进行时所有用户都可以进入比赛。
- 2、注册：比赛前有一段报名时间，在报名时间内报名了，比赛时即可参加比赛。
- 3、正式：比赛前有一段报名时间，在报名时间内报名后，需要由管理员审核，审核通过的用户在比赛时可以参加比赛。

关于 rating 的计算：

Rating 的设计是参考了国际象棋的 Elo rating system 的算法。具体证明在此将不做过多的叙述。

Elo rating system 算法只针对一对一的输赢而计算的算法，为了适应 oj 上比赛是多人同时进行的，因此在 Elo rating system 算法的基础上进行了简单的改进。

以下公式是用于计算当两个用户的积分分别为 R_a 和 R_b 时，进行一次比赛，应该扣除或增加的积分。

当多人同时参加一场比赛后，根据排名规则可以得到本场比赛的排名。此时假定参加比赛的每个人都与其他所有人进行了一次比赛。假定某场比赛共有 n 个人参加，则每个人与其他人所有人都可以得到一个 $rating$ 的变化值，则每个人都会得到 $n-1$ 个变化值。将这些变化值取平均数，即得到了某个用户在这场比赛中 $rating$ 的总变化值。再根据本场比赛的人数和全局参数计算一个线性变化的参数 k 值。最后将 $rating$ 的总变化与 k 相乘得到最终实际的变化值。

另外，每个人的初始 $rating$ 分为 1500 分。

这样做的一个不利因素是当场次较少时，水平较低的用户 $rating$ 分数较高，因此在实际的 $rating$ 基础上，增加一个和场次有关的参数做为乘数，实际显示的 $rating$ 为实际 $rating \times \text{乘数}$ 。

$Showrating$ 是显示的积分， $rating$ 是实际积分， num 是参加比赛的场次，则公式写成：

关于判重：

比赛中需要由服务器来自动检测代码相似度，这主要是为了管理员方便地检查是否有不诚信的用户以抄袭、传阅代码的方式来提高比赛名次。

服务器可以利用判重来检测两份代码的相似度。

两份代码有一个重复率表示两份代码的相似度。

定义：

代码 1 的字符串是 s_1 ，代码 2 的字符串是 s_2

len_1 表示 s_1 的长度， len_2 表示 s_2 的长度

重复率表示为 $(\text{最少的操作次数}) / \max(len_1, len_2)$ 。

最少的操作次数：是指通过三种操作使 s_1 变成 s_2 的最少操作次数。

三种操作分别是：

- 1、在 s_1 中插入一个字符
- 2、在 s_1 中删除一个字符

3、修改 s1 中的一个字符

为了检测更准确，s1 和 s2 中的空白字符要预先处理掉。

这使用如下动态规划算法：

定义表示 s1 的前 i 个字符变成 s2 的前 j 个字符需要的最少操作次数。则可以得到

最后求的即为所需要的操作次数。

4.3.3 讨论系统

讨论系统是为了用户之间讨论题目算法、或者管理员发布通告等功能。另外管理员可以设置将某个讨论放置于首页显示，来控制首页的显示内容。

管理员可以控制每个讨论的优先级来决定讨论的显示顺序。

管理员可以隐藏某些讨论。

发布讨论时使用百度的 UEdit 插件进行富文本编辑。

另外还可以简单地控制一些讨论的显示样式。

4.3.4 挑战模式系统

挑战模式是为了能够给初步学习算法的新人们提供一套较为完整的入门模块，新用户可以通过挑战模式的指引逐步深入学习各个算法。

挑战模式内，一个知识点被单独分到一个模块中，每个模块都有它的开启条件。整个挑战模式的模型即为一个拓扑图，每个模块都是图上的一个节点，每个条件是图上的一个带权有向边，表示开启某模块需要先在前一个模块中获得多少积分。整体模型如图 4-5 所示：

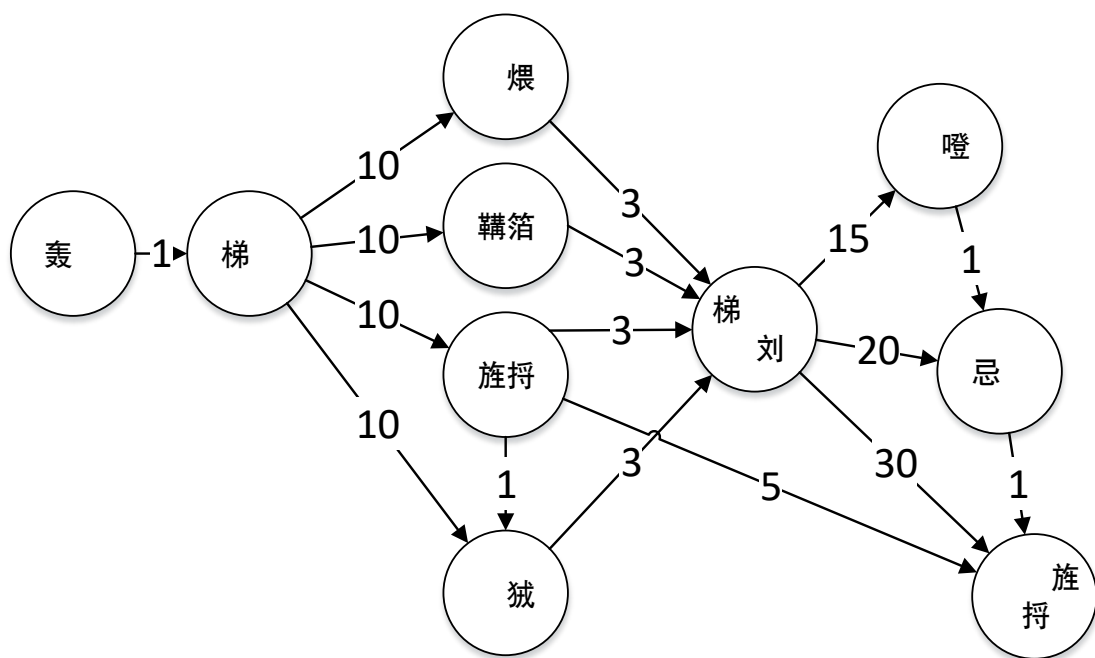


图 4-5 挑战模式拓扑图

当模块满足开启条件时才会呈现给用户。假设要开启“分治模块”，则需要要在“基础题集合”模块中获得 20 分，在递归中获得 1 分。

若某模块的所有前驱模块都开启了，则会以红色边框显示该模块。如果“递归”模块和“基础题集合”模块都已经开启了，则“分治”模块就会显示出来。

用户点击会显示本模块的开启条件。若某模块的任意前驱模块没有开启，则不显示该模块。

由于模块较多，服务器会预先将模块的所有配置从数据库读入内存中，可以提高访问速度。

4.3.5 考试系统

为了推广 ACM 竞赛，让更多人参与进来，开发了适合于相关编程课程的考试系统。一方面可以推广 OJ 系统和 ACM 竞赛的知名度，一方面可以帮助授课教师简单地进行科目考试的开展和统计。

考试题目分为选择题、代码填空题、结果填空题、编程题。其中选择题有单选和多选两种。教师用户通过核实可以获得出卷和出题权限，可以添加这四种题型。这几种题型有比较显著的特点：都可以由系统自动评卷。

选择题：多选或者单选。学生参加考试时通过单选或复选框来选择答案。

代码填空题：出卷时预先写好一段代码，将其中一部分挖去，由学生来补充挖去的部分。出卷时还要上传测试数据。自动评测时将学生填上的部分和题面的代码合并，并利用 oj 的判题功能进行评测。

结果填空题：只要填写结果，并且结果必须和标准答案一模一样才能通过本题。

编程题：和 oj 的其他题目一样，出题人要写上题面并上传测试数据。

个人题库：

出题用户可以在个人题库上查看到自己出的题。

公共题库：

出题用户可以将自己出的题目公开到公共题库上。

只有拥有出题权限的用户才能看见公共题库。

组卷：

出题用户在组卷的时候，可以从个人题库或者公共题库选定一些题目组成一张完整试卷，可以给每道题目定义分值。

统计：

考试结束后，会产生一个考试结果报表，显示每个人的得分情况，并且可以任意查看每个用户提交的答案。

结果报表还会显示每道题的通过情况，以便教师查看学生的知识点掌握情况。

5 系统实现

这部分将着重说明代码细节上的实现。包括贯穿全项目的面向对象思想，以及一些利用 JAVA 的高级语法实现的一些封装。

5.1 MVC 的设计模式

市面上有大家所熟知的各种 MVC 框架，如 SSH(Struts+Spring+ Hibernate)、SSM (spring+springMVC+Mybatis) 等，但是本系统并没有使用这些框架，仅使用了 struts2 用做控制器，其他都使用了自己封装的内容。

在 View 层，本系统主要使用了两种方式：

一种是使用 JAVA 代码来生成 HTML 页面代码。这样做的目的是可以将相似的界面部分合并在一起，减少代码冗余。主要使用的方式是使用 java 的多态，封装好常用的几种显示方式，主要是表格的显示。这部分将在 5.2 章节中详细介绍。

另外一种是在前端使用 jquery 动态修改 HTML 页面，并使用 JSON 传输数据。这么做的好处是可以减少页面刷新，避免过多的重复网络传输。这部分将在 5.3 章节中详细解释。

在表格 5-1 作出了本系统中的 MVC 实现和传统 SSH 的 MVC 实现上的比较：

表 5-1 MVC 实现方式比较

层	传统 SSH 框架	本 oj 系统
M（模型层）	使用 hibernate，直接将数据转化成 java 实体类	使用自己实现的 jdbc 封装+javaBean 来操作
V（视图层）	使用 JSTL、struts 标签、或者 JSP 标签来控制动态页面的产生。	使用 JSP 调用 java 类直接生成较为完整的 HTML 代码。
C（控制层）	使用 struts2 控制处理请求，用 spring 完成一些注入和解耦	同样使用 struts2 来控制处理请求，但是没有使用 spring。

5.2 使用 JAVA 代码生成 HTML 代码

这部分将本系统中最典型的例子展现出来。

有以下界面：

图 5-1 和 5-2 分别是比赛列表和评测结果页面。

比赛列表							New
<div><< < 1 2 3 4 5 ... > >></div> <div>名称 <input type="text"/></div> <div>权限 全部</div> <div>状态 全部</div> <div>筛选</div>							
#	名称	开始时间	结束时间	权限	状态	类型	
120	福建工程学院第十一届ACM程序设计大赛——补题场	2016-05-15 12:00	2016-05-21 00:00	公开	可观战	练习	
113	快进来一起玩——双指针 or 尺取法篇	2016-04-25 10:00	2016-05-20 12:00	公开	可观战	练习	
112	快进来一起玩——杂题篇	2016-04-23 22:00	2016-05-20 12:00	公开	可观战	练习	
59	计算几何基础	2016-04-21 12:00	2016-05-20 12:00	公开	可观战	练习	
27	KMP算法	2016-04-20 20:00	2016-05-20 00:00	公开	可观战	练习	
111	验题场(自己出的一些周赛题目)	2016-04-20 12:00	2016-05-20 12:00	私有	可观战	隐藏	
25	搜索入门和进阶	2016-04-18 00:00	2016-05-20 00:00	公开	可观战	练习	
108	第三届福建省大学生程序设计竞赛	2016-04-07 22:00	2016-05-20 12:00	私有	可观战	隐藏	
107	第六届福建省大学生程序设计竞赛	2016-04-07 21:50	2016-05-20 12:00	公开	可观战	练习	
104	第一周——第五周周赛补题场	2016-03-29 12:00	2016-05-20 12:00	公开	可观战	练习	
19	字典树	2016-03-28 16:00	2016-05-20 00:00	公开	可观战	练习	
18	优先队列	2016-03-20 23:00	2016-05-20 00:00	公开	可观战	练习	
48	最小生成树基础题	2016-03-16 21:00	2016-05-20 00:00	公开	可观战	练习	
16	并查集	2016-03-11 17:00	2016-05-20 00:00	公开	可观战	练习	
98	基础数据结构 (简单树的应用+简单图论)	2016-03-01 12:00	2016-05-20 00:00	公开	可观战	练习	

图 5-1 比赛列表

评测列表								
<div><< < ... 3 4 5 6 7 ... > >></div> <div>用户 <input type="text"/></div> <div>题目 <input type="text"/></div> <div>结果 All</div> <div>语言 All</div> <div>筛选</div>								
#	用户	题目	评测结果	语言	耗时	使用内存	代码长	提交时间
22586	chenshuiji(陈学军)	1939	Wrong Answer	C	0MS	1404KB	230	2016-05-18 16:04:05
22585	Hboss (霸气小鸽)	2124	Accepted	C++	497MS	1896KB	421	2016-05-18 16:02:55
22584	chenshuiji(陈学军)	1939	Wrong Answer	C	15MS	1404KB	269	2016-05-18 16:01:29
22582	Open_Seana (任性，可以，很强势【疯狂WA】)	1895	Wrong Answer	C++	639MS	11804KB	754	2016-05-18 15:58:44
22581	Open_Seana (任性，可以，很强势【疯狂WA】)	1895	Wrong Answer	C++	327MS	11804KB	634	2016-05-18 15:52:04
22580	FJUTACM1408 (我是谁)	1443	Accepted	C++	47MS	676KB	793	2016-05-18 00:34:09
22568	admin(萌萌哒管理员)	2145	Accepted	C++	529MS	2196KB	328	2016-05-17 23:42:12
22563	3151301111 (兰家旺)	1939	Accepted	C++	0MS	1404KB	215	2016-05-17 23:14:14
22561	cyz1998(JIAYOU)	1849	Accepted	C++	15MS	1804KB	237	2016-05-17 23:11:05
22555	Mysorrow (不会玩)	1147	Accepted	C	0MS	1420KB	191	2016-05-17 23:05:14
22553	chenshuiji(陈学军)	1939	Time Limit Exceeded	C	1000MS	1380KB	216	2016-05-17 22:14:56
22552	3151301111 (兰家旺)	2128	Accepted	C++	295MS	1892KB	310	2016-05-17 22:08:26
22544	cqh1315475391 (ILJX)	1204	Wrong Answer	C++	-	-	419	2016-05-17 20:47:42
22543	chenshuiji(陈学军)	1827	Wrong Answer	C	0MS	1832KB	544	2016-05-17 20:35:39
22542	chenshuiji(陈学军)	1827	Wrong Answer	C	93MS	2216KB	528	2016-05-17 20:33:37
22541	chenshuiji(陈学军)	1827	Wrong Answer	C	202MS	2220KB	466	2016-05-17 20:25:20
22540	cqh1315475391 (ILJX)	1203	Accepted	C++	0MS	1404KB	395	2016-05-17 20:05:59
22538	chenshuiji(陈学军)	2000	Accepted	C	0MS	1244KB	393	2016-05-17 19:40:59

图 5-2 评测结果

可以很明显地看出两个界面的共同之处：

整体是一个 panel，panel 有一个标题栏，panel 整体分为两部分，前面一部分是翻页按钮和一个筛选表单，后一部分是一个表格。

```
public abstract class pageBean {
    private String cl="table";
    TableHTML table = new TableHTML();
    List<String> Colname = new ArrayList<>();

    protected String ERROR_CELL_TEXT="=ERROR=";
    public abstract String getTitle();
    public abstract int getPageSize(); //当前页大小
    public abstract int getPageNum(); //总页数
    public abstract int getNowPage(); //当前页
    public abstract String getCellByHead(int i,String colname); //第i行的每列的内容

    public abstract String getLinkByPage(int page);
    public abstract String rightForm();

    public String getTableClass(){return cl;}
    public void setCl(String cl){this.cl=cl;}
    protected void addTableHead(String... ss) { Collections.addAll(Colname,ss); }
    protected String getColname(String colname) { return colname; }

    public String tableHTML(){...}
    public void addClass(int r,int c,String cls) { table.addCl(r,c,cls); }

    public String btn(String href,String text) { return HTML.abtn("sm",href,text,""); }
    public String btn_disabled(String href,String text) { return HTML.abtn("sm",href,text,"disabled"); }
    public String btn_active(String href,String text) { return HTML.abtn("sm",href,text,"btn-primary"); }
    public String page(){...}
    public String head(){...}
    public String foot(){...}
    public String HTML() { return HTML.panelnobody(getTitle(),head()+tableHTML()+foot()); }
    public static int getPageNum(int Num,int everyPageNum){...}
}
```

图 5-3 pageBean

如图 5-3 所示，由一个叫“pageBean”的虚类来处理一个需要翻页的显示页面，其中的 TableHTML 是一个用于产生一个表格的类。

PageBean 中有几个重要的虚方法：

String getTitle();

子类重写此方法可以修改显示的标题。

int getPageSize();

子类重写次方法可以修改每页显示的表格行数

int getPageNum();//总页数

此方法返回总页数

int getNowPage();//当前页

此方法用于返回当前是第几页

String getCellByHead(int i, String colname);

最关键的是这个方法，参数 i 是行数，colname 是列名。此方法返回第 i 行

名为 colname 列的单元格内要显示的内容。

```
String getLinkByPage(int page);
```

此方法返回指定页码跳转的链接

```
String rightForm();
```

此方法返回位于翻页按钮组右边的表单。

当我需要一个新的翻页展示的界面，我只需要新建一个 PageBean 的子类，然后实现以上几个方法即可。大大减少了代码冗余，这比用 jsp 页码和 JSTL 标签或者 Struts 标签都更有灵活性。代码量也非常少，只留下了少数页面之间不同的部分的编写，这部分完全可以体现出面向对象编程的优势所在。

使用此方法不足的地方在于产生 HTML 代码时对 web 服务器无形增加了处理压力，代码写的不好可能会导致服务器响应变慢。

5.3 使用 JQuery 和 JSON 动态调整 HTML 代码

为了解决上面的缺点，使用 JQuery 和 JSON 可以大大缓解这种压力。在前端页面使用 JQuery 动态生成 HTML 代码也不失为一种很好的办法。而主要的难点在于如何用 JavaScript 实现面向对象编程。

我们知道 JavaScript 并不是非常强大的面向对象语言，要实现和上述类似的面向对象设计，需要更深入的学习 JavaScript。

在本系统中，挑战模式的部分是使用 JQuery+JSON 的模式开发的，但是并没有用 Javascript 来模拟实现多态，因此在这里就不多叙述。不过在使用的过程中，学习了很多关于 JavaScript 的闭包概念，利用闭包可以模拟出类的共有成员、私有成员，可以方便地控制类的访问权限。

这么做的优点是将 HTML 代码的生成过程放到了浏览器执行，减轻了服务器压力。而且在传输过程中使用 JSON 传输数据也比直接传输页面的 HTML 代码用的网络量小。

但是，用这种方式比上一种方式的缺点是多了一次转换，必须先将数据转化为 JSON 格式，然后在前端再将 JSON 格式的数据转化到界面呈现。比第一种方式代码量是有所增加的。

5.4 对 JDBC 的封装

JDBC 虽然实现了对多种数据库的统一连接方式，但是他的 `Statement`、`ResultSet`、`Connection` 还是写起来非常复杂。虽然可以用 `Hibernate` 框架对 JDBC 进行更强大的操作，但是在本系统中我还是自己写了一个可以简化 JDBC 操作的框架。

有如下写法：

```
User u = new SQL("SELECT * FROM users WHERE username=?  
",username ).queryBean(User.class);
```

使用这一句话便可以获得指定 `username` 的用户 `JavaBean` 了。虽然还是需要自己写 SQL 查询语句，但是相对于 JDBC 最原始的写法已经方便了许多。

仔细分析这句语句，`SQL` 是一个类，他的构造函数是可变参数函数，这样可以方便地指定 sql 语句内的参数而不会造成 SQL 注入的 bug。

`SQL` 的实体类的 `queryBean` 方法的定义如下：

```
public <T extends IBeanResultSetCreate> T queryBean(Class<T> cls);
```

`IBeanResultSetCreate` 是一个接口，实现了这个接口的类可以用 `queryBean` 方法直接查询出来。实现 `IBeanResultSetCreate` 接口需要提供一个方法用于 `ResultSet` 和 `JavaBean` 的转化规则。

此外 `SQL` 类还提供了 `queryBeanList()` 等方法用来将 sql 查询结果的内容每一行对应一个 `JavaBean` 然后将它们放到一个 `List` 容器中返回。还有 `queryString()`、`queryNum()` 等方法用于查询出一个字符串或者是一个整数。

`SQL` 类的内部也已经把资源释放整合进去，开发时就可以不用考虑资源释放的问题了。

总结与展望

本 oj 系统与其他的 oj 系统相比，有以下几个优势：

- 1、本 OJ 不仅仅可以自己上传题目，也可以拉取其他 oj 的题目进行虚拟评测，这在校 ACM 集训队发展的初级阶段是非常有利的。由于我校 ACM 集训队刚刚起步，OJ 上的题目是有限的，而且自己原创的题目也不能满足集训队日常训练的需要，因此通过本 oj 拉取其他 oj 的题目可以获得更加丰富的题库，将其他各个 oj 的题目整合在一起，使得队员们平时训练更加方便。以前我们训练的时候，做一个类型的题目要到好几个不同的 oj 上去做题，现在，只要把相关的题目都拉取到本 oj 就可以了。
- 2、本 oj 的一大特色是挑战系统模块，本模块主要是为新人的入门而设计的。我校的 ACM 竞赛尚处于起步阶段，有许多有兴趣的同学苦于难以入门而只能望洋兴叹。挑战模式的存在使得更多有兴趣但摸不着方向的同学有了更加明确的方向。
- 3、比赛的模式多样，满足各种需要。比赛模式有不需要报名的公开赛，有需要简单报名的积分赛，也有需要详细信息的正式赛，和组队参与的组队赛，不仅满足了平时训练的需要，也可以满足校级大赛的需要。这是在原本我校使用的开源 oj 所不具有的功能。
- 4、丰富的统计，用户可以更清楚地看到自己的训练成果，可以知道自己的做题倾向，可以知道题目的难易度，题目的分类等等。有助于用户把握自己的训练节奏，让训练效果更上一层楼。
- 5、内嵌的考试系统，教师可以利用 OJ 系统上安排期中或期末考试，由机器自动评测并评分，一方面可以简化教师的批改和出题操作，一方面也可以对 OJ 系统起到推广的效果。

除了以上几个优点外，本系统仍然有许多可以优化的地方

- 1、管理员功能可以更加丰富。目前的后台管理系统比较简单，界面比较简陋，还有很多优化的空间。
- 2、可以开发更多可以提高队员训练积极性的内容，比如一些奖励机制。
- 3、本系统或许可以走出校园，走向全国。提高知名度，让更多的 ACMer 参与进来。

致谢语

大学四年，学习到很多很多。从大一开始加入 ACM 集训队开始，就对 ACM 竞赛种下了浓厚的兴趣，也取得了一些成绩。大学四年对我影响最深刻的莫过于 ACM 竞赛了，因此我首先要感谢的是 ACM 竞赛，和带我进入集训队的集训队教练。正是 ACM 竞赛教会了我如何坚持，如何利用时间去做有意义的事情。也要感谢四年来不断努力的自己，正是有了以前的努力，才有了现在的我。

从 ACM 集训队退役后，我就不断地思考，我感觉这点成绩并不是什么值得提起的事，我觉得我应该还可以走的更高。也正是因为我们是第一届正式集训队，我也更加希望我校的 ACM 集训队能走的更远，我愿意为 ACM 集训队的发展贡献自己的一份力量，所以才有了这套 OJ 系统。这套系统的开发完全让我从算法过渡到了项目开发上，从面向过程的思维方式过渡到面向对象的思维方式。因此我要也要感谢这套 OJ 系统，正是这个系统让我从跨出了从学校走向社会的第一步。

从开始毕业设计到现在，我学会了很多，不仅仅是在写程序方面，在面对业务流程方面的能力也得到了显著的提高。因此我还要感谢带领我做毕业设计的指导老师，正是老师孜孜不倦地教诲，我才能从 0 开始一直坚持到完成毕业设计。

在开发的过程中，碰到许许多多的问题，由于我刚开始开发 oj 系统的时候 java 基础不扎实，开发理念不完全，正是有老师和同学的帮助，我才能一路走到现在，少走了很多弯路。从 0 开始建立起来的编程思想、项目的架构理念，都要感谢老师和很多帮助过我的同学。

参考文献

- [1] 冯燕奎, 赵德奎. JSP 实用案例教程[M] 清华大学出版社, 2004, 5: 70-100
- [2] 飞思科技产品研发中心, 《JSP 应用开发详解（第二版）》, 北京: 电子工业出版社, 2004
- [3] 蔡剑, 景楠.Java Web 应用开发: J2EE 和 Tomcat[M].北京: 清华大学出版社, 2004.
- [4] Matt Bishop. Computer Security: Art and Science. New York, 2002
- [5] Matt Bishop. Vulnerabilities Analysis. Proceedings of the Second International Symposium on Recent Advances in Intrusion Detection. Los Angeles 2006
- [6] Vivek Chopra, Jon Eaves, Rupert Jones 著, 张文静, 林琪 等译. Beginning JavaServer Pages JSP 程序设计. 北京: 人民邮电出版社, 2006
- [7] Deepak Alur, John Crupi, Dan Malks 著, 刘天北, 熊节 等译. J2EE 核心模式 (第二版). 北京: 机械工业出版社, 2005
- [8] HDU Online Judge 系统: <http://acm.hdu.edu.cn/>
- [9] Bootstrap3 中文文档: <http://v3.bootcss.com/>
- [10] jquery API 1.12 中文文档: <http://jquery.cuishifeng.cn/>