

摘 要

在线编程评测系统已经发展多年，其普遍应用于编程竞赛和教学训练并且已取得了优异的效果。然而由于学生学习能力的差异，大部分测评系统不能达到个性化教学的目的。此外，对于课程中的大量知识点，学生不能很好的循序渐进学习，教师也很难精细化分析不同学生的学习情况。

本文针对上述问题，设计并实现了一款适用于教学的个性化在线编程系统。该系统基于 LAMP 开发框架，MVC 开发模式，设计了关于问题列表的独特数据结构以及个性化问题推荐算法，增设实境编程模块，便于学生实时根据所学知识进行编程测验。此外，教师可以根据意愿标注重点以及要求，可以快速获取学生学习状态、分析学生学习效果、批量导入学生、一键下载学生代码等。在判题机方面，使用轮询数据库的方式，将需要评测的题存入队列中并依次分发给空闲的判题机，判题机将问题运行在沙盒中并将结果写回数据库，整个过程保障了系统的高效、稳定、安全。经理论分析以及实验检验，本系统对教学应用方面有着一定的效果。

关 键 词：在线编程系统；个性化推荐；实境编程；LAMP 架构

ABSTRACT

Online programming evaluation system has developed for many years, which is widely used in programming competition and teaching training and has achieved excellent results. However, due to the differences of students' learning ability, most of the evaluation systems cannot achieve the purpose of personalized teaching. In addition, for a large number of knowledge points in the course, students cannot learn well in a step-by-step manner, and it is difficult for teachers to finely analyze the learning situation of different students.

Aiming at the above problems, this paper designs and implements a personalized online programming system for teaching. Based on the LAMP development framework and MVC development mode, the system designs a unique data structure about the problem list and personalized problem recommendation algorithm, and adds a real-time programming section, which is convenient for students to test real-time programming. In addition, teachers can mark the key points and requirements according to their wishes, quickly obtain students' learning status, analyze students' learning effect, import students in batches, and download students' codes with one click, etc. In the aspect of judging machine, we use the way of polling database to store the questions to be evaluated into the queue and distribute them to the idle judging machine in turn. The judging machine runs the questions in the sandbox and writes the results back to the database. The whole process ensures the efficiency, stability and safety of the system. Through theoretical analysis and experimental test, the system has a certain effect on teaching application.

KEY WORDS: online programming system; personalized recommendation; reality programming; LAMP architecture

目 录

1	需求分析	1
1.1	环境需求	1
1.2	功能需求	1
1.3	性能需求	2
2	推荐算法	3
2.1	概念和规则	3
2.2	方案和扩展	3
2.2.1	正向推荐	4
2.2.2	负向推荐	4
2.3	本章小结	6
3	系统设计	7
3.1	前台功能设计	7
3.1.1	用户系统	7
3.1.2	学习系统	9
3.1.3	作业系统	10
3.1.4	判题机	11
3.2	后台功能设计	12
3.2.1	课程管理	12
3.2.2	学生管理	13
3.2.3	参数配置	13
3.3	数据库设计	13
3.4	本章小结	14
4	系统实现	15
4.1	开发环境	15
4.2	用户系统	15
4.3	作业系统	18
4.4	学习系统	21
4.5	后台功能	22
4.6	判题机功能	23
4.7	本章小结	24
5	系统部署与测试	25
5.1	系统部署	25
5.1.1	前后端部署	25

5.1.2 判题机部署	25
5.2 系统测试	25
6 结论与展望	27
6.1 结论	27
6.2 展望	27

3066731463@qq.com

1 需求分析

需求分析是系统开发的首要以及关键环节，通过分析各个角色的需求来定义系统的相关功能，保证系统实现的完整性、合理性、安全性等。

1.1 环境需求

整个系统的开发技术框架主要基于 LAMP（即 Linux+Apache+Mysql+PHP），其中 PHP 开发使用了 ThinkPHP 框架，其是一款快速、兼容的轻量级国产 PHP 开发框架，使用了面向对象的开发结构和 MVC 模式（即 Model+View+Controller）。在判题机部分，参考 LPOJ 的判题模式，使用 python 语言编写脚本。本系统的开发成本很低，除了必要的人力成本以及服务器等硬件成本，Apache、ThinkPHP 等软件几乎都是开源免费的。此外，对于后期的二次开发以及维护上，都设计了相应文档，便于后续工作的开展。

1.2 功能需求

本系统的主要用户群体是教师、学生等与教育行业相关人员。学生主要的需求是账户注册、在线编辑代码、提交并获取结果、查看个人信息、找回密码、找回密码、修改密码、问题反馈等。教师主要的需求是批量导入学生、给学生选课、查看学生信息、查看反馈问题、导入课程信息、下载学生代码、设定要求等。图 1-1 是系统部分功能的用例图。

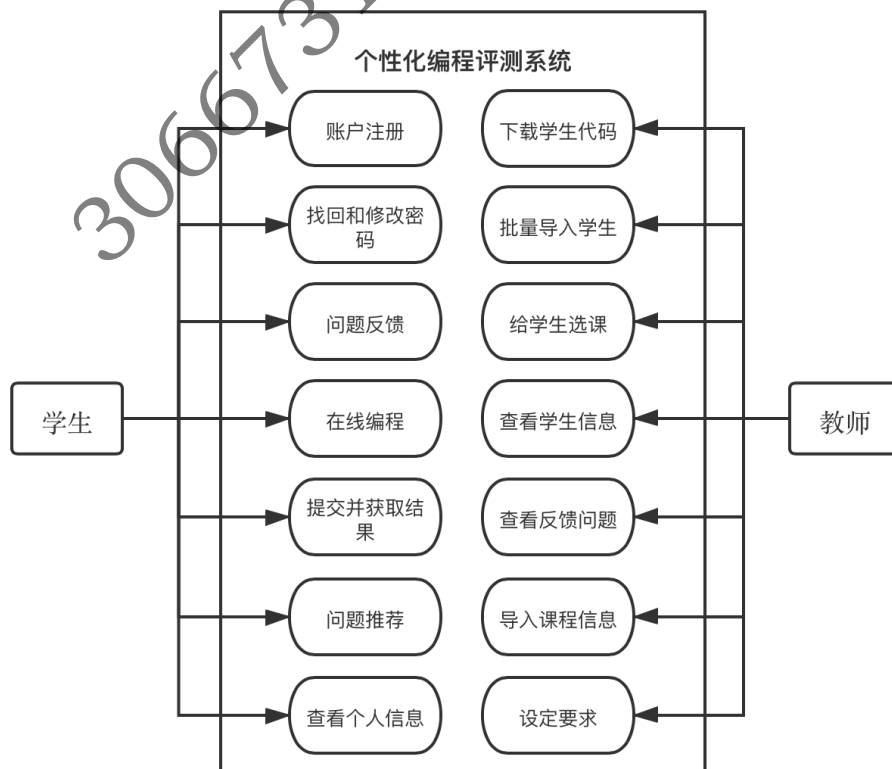


图 1-1 系统功能用例图

1.3 性能需求

由于系统只面向教师导入的学生，所以对于安全性能的考虑较为宽松，其主要分为用户账户安全、系统安全和用户程序安全。为避免用户账号丢失或者被他人窃取，一个合理的身份验证、有效的账户保护措施是非常重要的。对于访问不同的页面，每次系统都需要进行身份及权限验证。对于用户提交的程序，其运行在服务器上，特别容易导致安全问题，对每个程序都进行时间、内存以及环境限制是一个有效办法。

一个高效的系统不仅能带给用户良好的体验，也能节约非常多的资源。为了提升性能，系统需要不断的优化改进算法，减少对数据库的访问。在判题部分，也要做到并发执行，避免拥塞、提高效率。

3066731463@qq.com

2 推荐算法

推荐算法是整个系统的特色，设计一个合理有效的算法是整个系统设计的核心之一。本章将详细描述整个推荐算法的设计流程及原理，并对一些细节问题提出解决思路。

2.1 概念和规则

为了方便理解推荐算法的原理，首先对一些名词进行定义。

- 1) 知识点标号：学习的知识点的先后顺序；
- 2) 簇：包含知识点相同的问题的集合；
- 3) 簇级：包含知识点的最大标号；
- 4) 簇的大小：若簇 A 的簇级小于簇 B 的簇级，则认为簇 A 小于簇 B；若簇 A 和簇 C 簇级相等，但簇 A 包含的知识点数量小于簇 C，则认为簇 A 小于簇 C；若簇 A 和簇 D 簇级相等，且簇 A 包含的知识点数量等于簇 D，但簇 A 中知识点的第二大标号小于簇 D，则认为簇 A 小于簇 D；以此类推；
- 5) 簇链：将多个簇按从小到大排序成的一条链形结构；
- 6) 章节：具有不同的知识点的簇的集合；
- 7) 相关簇：包含的知识点是当前簇包含知识点的子集的簇的集合。

系统目前支持章节内簇间的推荐。图 2-1 为一个简单的簇链形成过程，规定学习知识点顺序为 a -> b -> c。可以看到簇 1 中仅含有一个知识点 a，则它的知识点的最大标号为 1，即簇级为 1；簇 2 中仅含有一个知识点 b，则它的知识点的最大标号为 2，即簇级为 2；簇 3 中含有两个知识点 b、a，则它的知识点的最大标号为 2，即簇级为 2；以此类推，簇 4、5、6、7 的簇级为 3。虽然簇 2 和簇 3 的簇级相等，但簇 2 包含的知识点数量少于簇 3，则簇 2 小于簇 3。同理，簇 5 和簇 6 簇级相等且知识点数量相等，但是簇 5 中，第二大的知识点标号小于簇 6，则簇 5 小于簇 6。按照簇链的形成规则，最终结果如图 2-1 所示。

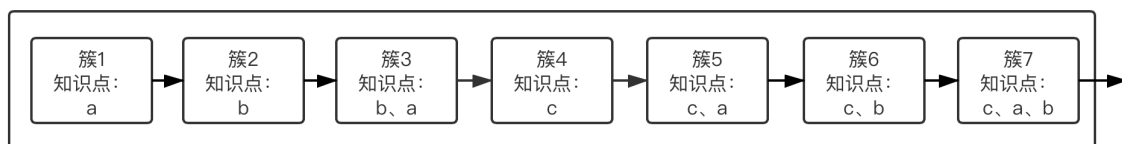


图 2-1 簇链

2.2 方案和扩展

实际中，学生的答题结果分为两种情况，正确和错误，接下来将分别对这两种情况的个性化推荐进行说明。

2.2.1 正向推荐

正确的情况较为简单，当学生对任意一个簇中的问题提交结果为正确时，老师可根据自己的要求，例如必须做 5 道题或者答对 3 道题，来决定是推荐这个簇的后继题目还是当前簇中的题目。如图 2-2 所示，其中箭头方向表示推荐方向。

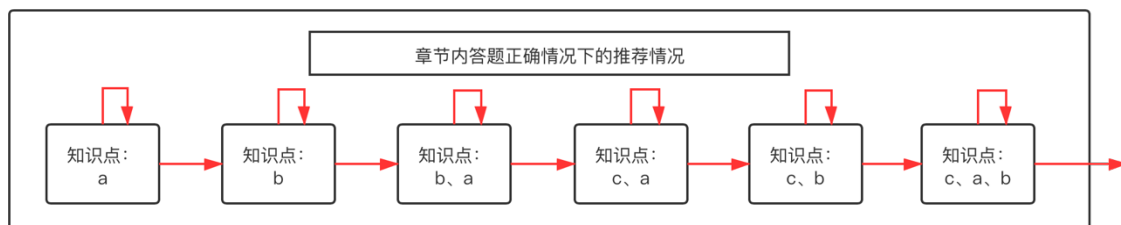


图 2-2 正向推荐示意图

2.2.2 负向推荐

错误的情况较为复杂。首先，如果学生对于当前的问题出现错误，那么规定不会推荐当前簇之后的簇中的题目。其次，考虑到当前簇中的问题涉及到多个知识点，可能存在对之前簇中的问题理解不透彻或者长时间过后遗忘的状况，那么就要考虑到回溯的问题。对于如何解决这些问题，具体分为以下几个方面：

1) 查找推荐题目

首先，需要确定当前簇的相关簇，它们只能是在当前簇之前的。具体方法是根据当前簇中的知识点得到其非空子集 ψ ，例如当前簇中知识点为(a, b, d)，那么其非空子集为 $\psi = \{(a), (b), (d), (a, b), (a, d), (b, d), (a, b, d)\}$ 。然后，从簇链的开始查找，若簇中知识点属于集合 ψ ，则认为其与当前簇相关，查找到当前簇结束，因为之后的簇不可能是其子集。例子如图 2-3 所示，第 4 个簇的知识点为(a, c)，那么其非空子集为 $\psi = \{(a), (c), (a, c)\}$ ，查找得到第 1、4 个簇符合，则推荐的题目将会从第 1 个簇和第 4 个簇中选择。

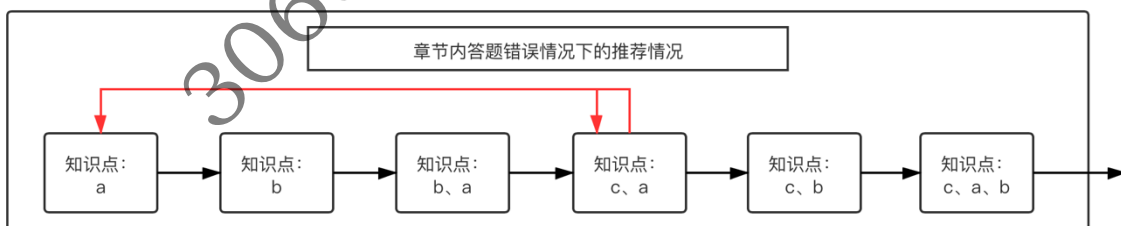


图 2-3 负向推荐示意图

2) 计算问题难度

首先，对于一个簇中的所有问题，其包含的知识点是完全一致的，难度差距不会很大。因此，簇中每个问题的难度可以用簇的难度 D_i 来表示。对于如何定义簇的难度考虑到三种方案。第一种就是老师直接定义簇的难度，其存在问题的是簇的结构会因为上传的问题而经常发生变化，老师很难定义难度且定义的难度在与其他簇相比时不具有说服力。第二种是直接认为知识点的难度相同，簇的难度只与知识点的数量有关，这种方式缺点很明显，当存在一个知识点的难度很高的时候，就不能有所体现。第三种是直接定义知识点的难度，这样在计算簇难度的时候可以取平均乘以一个增长系数来得到，

存在的缺点是老师需要给每个知识点定义难度标签，操作较多。结合三种方案的优缺点，本系统使用的是第三种难度定义方案。

3) 计算推荐概率

这里首先需要考虑选择推荐的簇与哪些因素有关。按照图 2-3 中的推荐进行考虑，首先若第 1 个簇中学生已经做了远比第 4 个多的题，那么可以认为其第 1 个簇中题目的掌握程度较好，推荐它的概率就会较小；其次，若第 1 个簇中学生做的题目的通过率远大于第 4 个，那么推荐它的概率也会较小；最后，考虑问题难度，显然，学生对于一个包含多个知识点的问题出现错误，那么更大的可能是在较难的知识点上。综上，推荐到簇 i 的概率与学生在簇 i 上做题数量、通过率成负相关，与簇 i 的难度成正相关，即

$$P_i = \frac{(\frac{x_i}{\sum_{j=1}^n x_j})^{-\alpha} \times (\frac{y_i}{x_i})^{-\beta} \times D_i^\gamma}{\sum_{k=1}^n (\frac{x_k}{\sum_{j=1}^n x_j})^{-\alpha} \times (\frac{y_k}{x_k})^{-\beta} \times D_k^\gamma} \quad (2-1)$$

令

$$W_i = (\frac{x_i}{\sum_{j=1}^n x_j})^{-\alpha} \times (\frac{y_i}{x_i})^{-\beta} \times D_i^\gamma \quad (2-2)$$

则

$$P_i = \frac{W_i}{\sum_{j=1}^n W_j} \quad (2-3)$$

式(2-1)中 P_i 表示簇 i 的推荐概率， x_i 表示学生在簇 i 中做题的数量， y_i 表示学生在簇 i 中通过的题的数量， n 表示当前簇的相关簇的数量， D_i 表示簇 i 的难度系数， α 表示做题数量的权重， β 表示通过率的权重， γ 表示难度系数的权重占比；式(2-2)中 W_i 表示簇 i 的推荐权重。

4) 轮盘赌法选择

对于已经求出的转移概率，为防止推荐时进入局部最优解，使用轮盘赌法进行选择。轮盘赌法是指通过系统生成 0~1 的随机数所在的概率区间来决定选择的对象。在图 2-3 中，若求出转移到第 1 个簇的概率为 0.2，第 4 个簇的概率为 0.8，这个时候，是存在转移到簇 1 的情况，如果只按照最大的概率来转移，可能会出现下一次推荐概率变为第 1 个簇 0.15，第 4 个簇 0.85，这样还是转移到第 4 个簇，这样一直下去就进入局部最优解了，轮盘赌法可以有效的避免此问题。

5) 相关参数配置

对于 α 、 β 、 γ 三个参数，其值根据教师的侧重点决定，例如教师认为通过率更加重要，则增大 β 的占比即可。对于公式 2-1，当学生做题数为 0 或者通过题数为 0 时，存在分母为 0 的情况，这里采用的方法是增加一个补偿量，随着做题数量的增加，补偿量对推荐结果的影响会越来越小。此外，实际中学生对答题错误分为两种情况，一种是清楚哪个知识导致的错误，这个时候若推荐的题目中含有相关知识点的问题，学生可以直接去选择；另一种情况是学生并不知道出错是什么原因，这个时候根据推荐的题目进行选择，若选择的题目刚好解决了问题，那么推荐是正确的；若选择的问题还是做

错了，那么转移概率就会发生变化，更加倾向于推荐导致错误的簇。

针对公式 2-2 和公式 2-3 进行扩展。假设 $\alpha = \beta = \gamma = 1$ ，则

$$W_i = \left(\frac{x_i}{\sum_{j=1}^n x_j}\right)^{-1} \times \left(\frac{y_i}{x_i}\right)^{-1} \times D_i = \frac{D_i \times \sum_{j=1}^n x_j}{y_i} \quad (2-4)$$

$$P_i = \frac{W_i}{\sum_{j=1}^n W_j} = \frac{\frac{D_i \times \sum_{j=1}^n x_j}{y_i}}{\sum_{k=1}^n \frac{D_k \times \sum_{j=1}^n x_j}{y_k}} = \frac{\frac{D_i}{y_i}}{\sum_{k=1}^n \frac{D_k}{y_k}} = \frac{D_i}{y_i \times \sum_{k=1}^n \frac{D_k}{y_k}} \quad (2-5)$$

可见此时推荐概率只与学生在簇 i 中通过的题的数量和簇 i 的难度系数有关。例子：图 2-3 中，学生在第 1 个簇中通过了 10 道题，第 1 个簇的难度系数为 1，在第 2 个簇中通过了 3 道题，第 2 个簇的难度系数为 2，则推荐第 1 个簇的概率 $P_1 = 0.1304$ ，推荐第 2 个簇的概率 $P_2 = 0.8696$ ，根据求得的概率进行轮盘赌法选择即可。

2.3 本章小结

本章介绍了基于簇结构的问题推荐算法，定义了“簇”“知识点标号”、“簇链”等词语的概念。设计了簇链的生成规则，针对簇链的结构推导出合理的转移概率公式。此外，对公式进行了简单的扩展，说明其具体应用，并对算法的一些细节问题进行了解决。该推荐算法结合了教师教学的实际情况，将每一章节的所有知识点按顺序进行归纳，符合学生的学习情况。其针对簇之间的关系，形成一个推荐网络，根据合适的因素计算出各个节点之间的转移概率，为不同学生的答题情况进行客观、有效的推荐，帮助学生更高效的学习。在实际部署中，算法简洁明了，容易实现；复杂度低，在推荐的过程中，快效、有效。总之，该算法基本上满足了目前的需求。

3 系统设计

本系统采用的是 B/S 架构，参考文献中的设计模式，如图 3-1，用户点开浏览器，并进行页面操作，浏览器向服务器发送请求，如有数据相关，服务器会向数据库发送 sql 语句，数据库将结果反馈给服务器，服务器再向浏览器发送界面，用户这时就收到了反馈，如此反复。此外，为了满足用户的需求和系统本身应具备的一些非功能需求，需要对数据库结构和一些相应功能进行详细的设计，以便在后续的实现中能更加高效、准确。本章将详细描述本系统的前台功能、后台功能以及数据库结构的设计情况，简要阐明这样设计的原因，同时将推荐算法与之结合，形成了一个初步完整的体系。

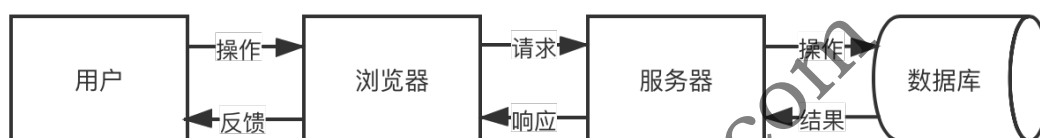


图 3-1 系统架构

3.1 前台功能设计

绪论 1.3.2 中已经对本系统的一些功能需求做出了简要说明。由于面向的用户群体是学生和教师，前台主要是为学生开发，所以只需分析学生的功能需求并进行设计即可。结合这些需求，现将前台功能划分为三大模块：

3.1.1 用户系统

用户系统是整个系统至关重要的一环，其要记载用户的信息、保障用户的账号安全等。一个完整的用户系统应包括：

1) 用户登录

用户登录重要的一点是要保证用户信息的真实性和安全性。目前，很多网站的登录都设置了手机号验证、验证码验证等。由于系统的学生用户只能是由教师导入的学生名单中的，所以对于真实性问题可暂不考虑。对于安全性，为防止有人恶意窃取他人密码，系统设置了账号安全级数。当在一段时间内连续输错密码达到限制次数时，账号将会被锁定 30 分钟，同时安全级数降低。当安全级数降低至 0 时，账号被冻结，这时需要联系管理员才能解冻。这样，当用户在不知情下遭到入侵时，一方面入侵者无法短时间内暴力破解，另一方面账号冻结时会给用户一个警示，用户联系管理员后可以查看登录记录，并且重新加密密码，保障账户安全。用户登录的服务器处理流程如图 3-2，系统根据用户提交的用户名和密码，先检验用户名是否存在，若存在则继续判断账号是否被冻结或锁定，进而判断密码的正确性，若均符合要求则顺利登录系统，否则按照相关错误进行信息提示。此外，整个系统对用户的所有输入都进行了校验，保证输入数据的合法性。

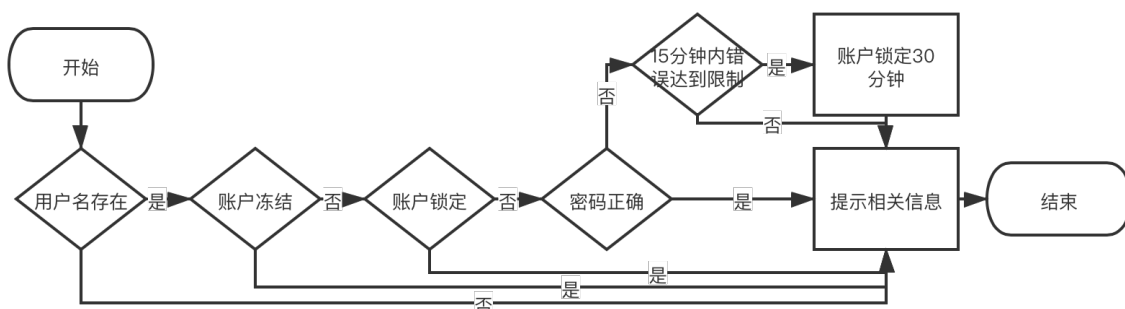


图 3-2 用户登录流程图

2) 注册账号

账户注册的操作较为简单,这里需要收集的信息有:用户名,为学生的学号;密码,用户自定义;真实姓名,用于校验用户身份,保证参与了课程;联系电话,方便教师联系学生;密保问题及答案,用于找回密码,需要谨慎设置。用户注册的服务器处理流程如图 3-3,需要根据数据库查询结果判断用户是否参与课程以及是否已经注册,并进行相关操作即可。

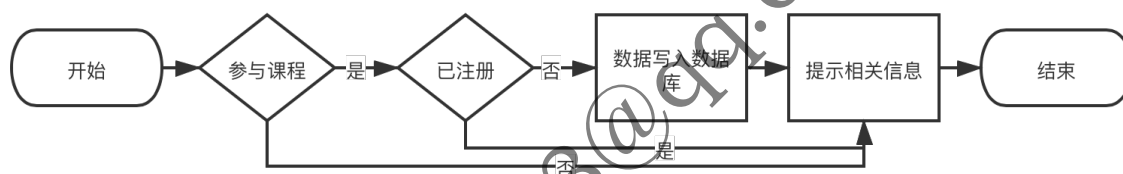


图 3-3 用户注册流程图

3) 找回密码

找回密码又一次涉及到账户安全性的问题,依旧使用安全级数,此外,为降低管理员账号的风险,对管理员用户禁用此功能。处理流程如图 3-4,用户输入用户名,系统会显示其相应的密保问题。用户输入密保答案以及新密码,如果答案错误,安全级数降低;如果正确,新密码则设置成功。同理,当安全级数降低至 0 时,账号被冻结,此时是无法继续找回密码的。

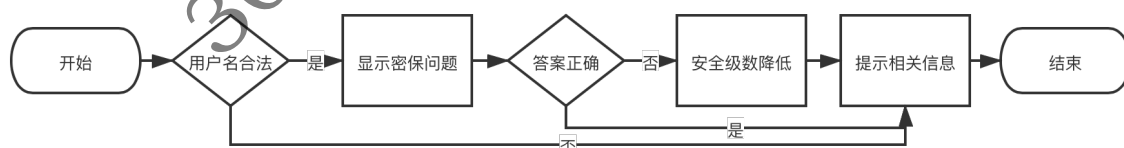


图 3-4 找回密码流程图

4) 修改密码

修改密码是用户登录之后才可以使用的功能,因此在安全性上有一定的保障。用户输入旧密码和新密码,当旧密码校验成功后,新密码存入数据库,立即生效。

5) 退出系统

用户退出系统,系统要保证将用户的一切缓存信息,如 Session、Cookie 中的数据清理干净,防止造成信息泄漏。

6) 问题反馈

为方便管理员收集用户建议和问题等,增设了问题反馈功能,当用户登录系统后,

可以针对系统 bug、课程问题等进行填卷反馈，后台系统会在同时收到反馈并显示给管理员。

7) 成绩评定

对于用户成绩，系统提供了一套算法，学习系统和作业系统各占一部分分，具体比例由教师决定。对于学习系统，所有课节分数相同，每完成一个课节学习则增加一课节分；对于作业系统，按照问题难度给问题打分，问题难度由其包含的知识点决定，分数计算方式为所有通过的题的难度除以所有问题的难度，这样更能显现越难的题分值越高的特点。用户可以查看自己已修课程信息和成绩，还可以查看自己的代码提交记录。

3.1.2 学习系统

学习系统即实境编程模块，其是一种基于音频实现师生代码互动的教学模式，有助于解决当前程序设计类课程中存在的学生缺乏编程实践、解决问题无头绪、教师无法面面俱到等问题，推动了计算机教学模式的创新。

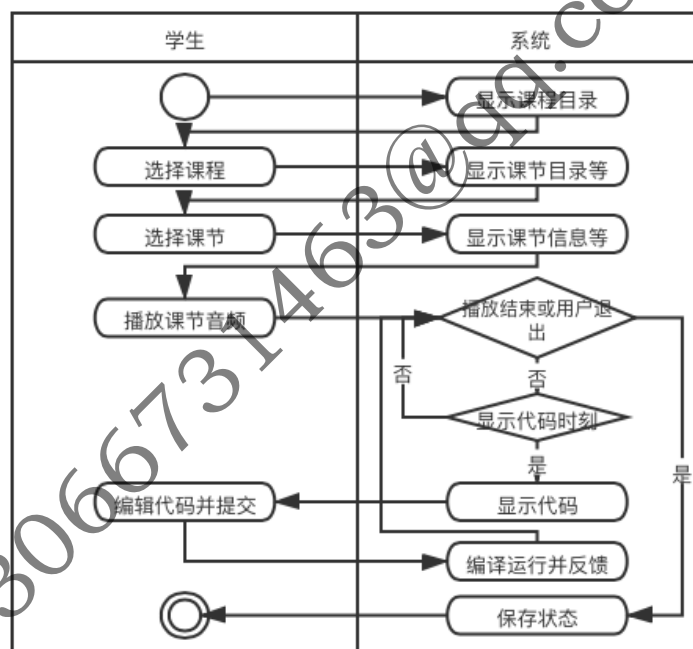


图 3-5 学习系统活动图

学习系统为学生提供了全新的编程环境。首先由教师录制好教学音频并安排好对应时刻提供的代码，学生可以随时登录系统，选择需要学习的课节，系统会播放相应的讲解音频并在特定时刻为学生提供代码，学生可以根据自身理解对代码进行更改、提交运行，系统会及时返回结果。整个学习过程摆脱了传统教学的局限性，学生可以随时、随地的学习，也可以根据自己的不足反复学习、练习，同时也大大减少了教师的工作量。此外，教师可以参考其他教师的教学，不断的完善自己的教学方式。也可以在提供的代码上进行改动，从而提高学生的独立编程能力。学习系统具有极高的自由度，而这正是传统教学模式所缺乏的，学生的自主学习、自我认知也在极大的促进着个性化教学。

学习系统的流程较为简单，如图 3-5，首先向学生展示课程目录界面，由学生选择

要学习的课程，系统会显示对应的课程信息、课节目录和学生学习时长、分数。学生再次选择要学习的课节，系统会显示课节信息、播放条、编辑框以及对应的输入输出框、编译结果框。其中课节信息由教师导入，播放条会播放要学习的音频，这里对播放进行了禁止快进的限制，每次离开页面会保存进度，进度以内是可以快进的。编辑框会在教师要求时刻显示相应的代码，学生可以进行更改，并且可以选择不同版本的编译器编译运行，系统会实时反馈结果。

3.1.3 作业系统

作业系统是整个系统的核心，个性化推荐也是在此体现。整个作业系统的活动如图 3-6。

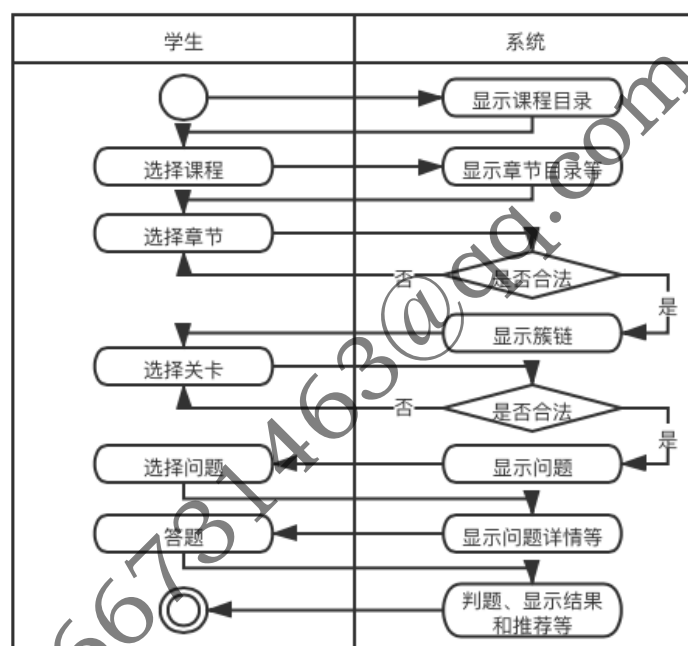


图 3-6 作业系统活动图

同样，首先向学生展示课程目录，由学生选择后，显示课程信息、章节目录、当前得分以及最近提交的题目、最近未通过的题目，学生可以快速选择继续做题。每个章节根据学生的答题情况显示通过和未通过，学生只能按照顺序选择，即必须先前章节通过才可以选择后续章节。学生选好章节后，推荐算法会根据章节内的问题形成簇链，并根据学生的提交情况和教师的要求统计出学生在每个簇上的状态，并将结果展示给学生。每个簇对学生来说都好像是个关卡，学生只有一关一关的闯关，才能通过这个章节。之后，学生选择对应的簇，系统就会显示相应的问题和状态，这些问题都是含有的知识点完全相同的。学生任意选择一个之后，系统会显示问题描述、测试样例、编辑框、该问题提交和通过数量、通过的平均代码行数、包含的知识点、提交状态、章节和课程完成度、上次提交结果以及推荐算法推荐的问题。这个时候推荐算法是按学生这个题错误进行计算的。当学生完成相应问题并提交之后，系统会进行判题，并将结果以及相应的推荐展示给学生。学生可以选择推荐的题目，也可以重新自由选择。

3.1.4 判题机

判题是整个系统的核心功能，由于在线评测系统是评测用户上传的程序，安全性的保障尤为重要。此外，由于多用户可能同时上传，高并发以及准确性也是必不可少的。针对此问题，已有很多非常完美的解决方案，例如 hustoj 轮询数据库、青岛大学 oj 创建沙盒等。本系统借鉴 LPOJ 的判题服务器和判题机部分，按照符合系统要求的方向进行了改动，下面分别介绍其设计原理。

1) 判题服务器设计

判题服务器工作流程如图 3-7 所示。后台服务器收到用户提交的代码后，将其保存到数据库中，此时判题机服务器每隔一段时间就访问一次数据库，获取未判题列表，使用队列进行保存。同时，另开线程与各个判题机通信，找到空闲的判题机，分发判题任务，判题机判好题后将结果写回数据库，此时后台服务器从数据库中获取判题结果并反馈给用户。

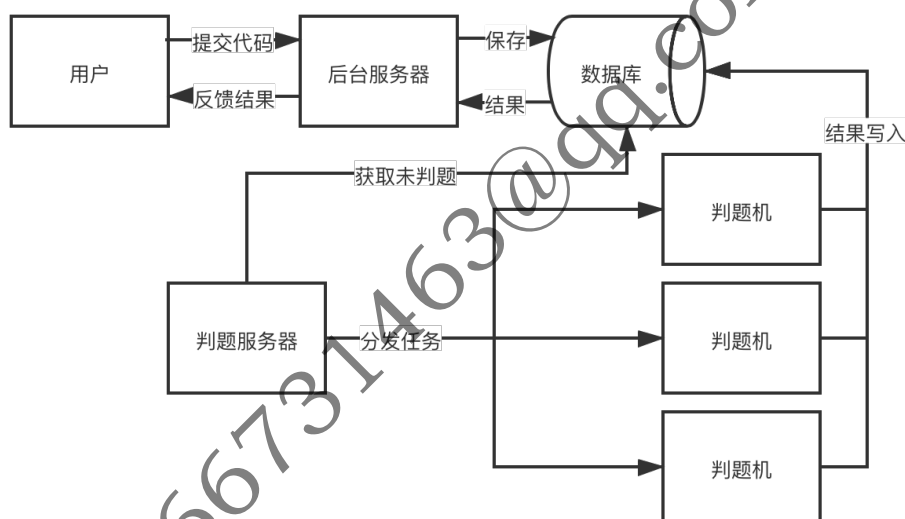


图 3-7 判题服务器工作示意图

整个判题过程中，判题服务器充当一个管理者，用户不断向数据库提交“任务”，判题服务器将这些“任务”用一个队列整合起来，分发给空闲的“机器”，“机器”收到“任务”后开始忙碌起来，结束后再次空闲起来等待新的“任务”。整个过程高并发、稳定、精确。

2) 判题机设计

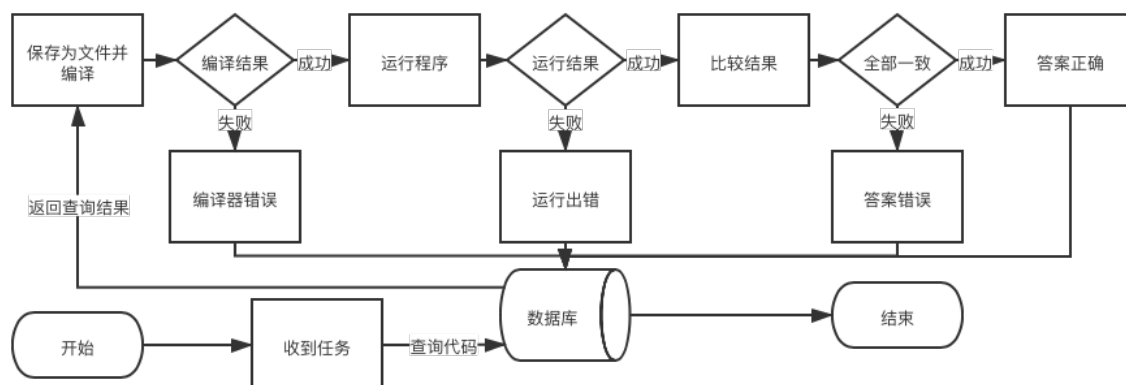


图 3-8 判题机工作流程图

图

判题机工作流程如图 3-8 所示。判题机收到来自判题服务器的任务后，根据相应 id 查到待判的程序，将其保存为文件形式并进行编译。若编译成功，则在沙盒中运行程序，否则将编译错误信息写入数据库。若运行成功，则与测试样例比较，按照结果正确的数量给分，否则将运行错误原因写入数据库。当判题结束后，判题机进入空闲状态。

3.2 后台功能设计

本系统的后台功能主要面向教师，所以只需分析教师的功能需求并进行设计即可，现将后台功能也分为三大模块：

3.2.1 课程管理

课程管理主要包括课程、章节、知识点、问题、课节及相关的增删改查操作。在进行删除操作时应输入密码，确保数据不会被误删或遭到他人恶意破坏。接下来分别对其要统计的数据和一些特殊操作进行说明。

1) 课程

教师首先需要添加课程信息，包括课程名称、任课教师、授课时间、编译语言、课程类型、课程公告和课程图片。其中课程类型指的是课程是否包含学习系统或作业系统，或者两者都。在进行删除操作时，会将课程相关联的数据同时删除，例如课程包含的章节、知识点、学生的成绩、答题情况等。此外，增设了课程复制功能，对于不同学期的相同课程，其间存在着极大的相似性，教师可以使用此功能复制相似课程，从而减少了添加章节、问题等操作。

2) 章节

接下来，教师需要对课程添加章节信息，包括所属课程、章节序号和章节名称。其中章节序号表示该章节在所属课程中的学习顺序，其禁止重复并且需要连续，系统会自动判断教师输入数据的合法性并进行提示。

3) 知识点

对于每章的知识点，教师需要添加所属课程、所属章节、知识点标号、知识点内容和知识点难度。其中知识点标号是指知识点学习顺序，在每个章节中禁止重复且需要连续，系统同样会对教师的输入进行检验。

4) 问题

同样，教师添加问题信息包括所属课程、所属章节、问题名称、问题内容、测试样例、限制时间、限制内存和问题模版。教师还可以对每个问题与所其属章节的知识点进行关联。此外，系统还统计出学生对该问题的解答情况，并增加一键下载学生代码功能，方便教师管理。

5) 课节

教师添加课节信息包括所属课程、所属章节、课节序号、课节名称、课节信息和课节音频。教师同样可以查看每个学生的学习情况。

3.2.2 学生管理

学生管理主要包括学生信息、学生成绩、学生登录情况及相关的增删改查操作。同样，删除操作时要输入密码。接下来分别对其要统计的数据和一些特殊操作进行说明。

1) 学生信息

学生信息包括班级、姓名、学号、权限、最近登录、状态、安全等级、联系方式、密保问题和密保答案，其中状态包括在线、离线、未注册和锁定。教师添加学生只需要填写班级、姓名、学号即可，其余信息均有默认值。为了方便教师导入学生，设定了上传 xls 文件方式。此外，学生还需与课程建立关联，即学生选课，以防未修读此门课程的学生随意提交代码。

2) 学生成绩

学生成绩根据课程类型来决定包含哪些方面，一个是作业成绩，一个是学习成绩。作业成绩即为学生答题情况，包括章节、问题、提交时间、状态、分数、答案、编译器和编译信息，其中状态包括等待加入队列、等待下发判题机、等待判题、通过、未通过和等待超时。学习成绩即为学习系统课程播放情况，包括章节、课节、学习时长、状态，其中状态包括通过和未通过。教师可以根据班级、学号或课程信息搜索相关学生，可以查看当前学生的学习状态、成绩等。此外，教师还可以下载此学生提交的全部代码。

3) 学生登录情况

教师可以查看的学生的登录情况，包括学生 id、学号、登录时间和状态，一方面可以保障学生的账户安全，另一方面通过学生登录情况可以判断学生的学习热情度。

4) 学生反馈问题

学生反馈问题包括学号、问题描述、反馈时间和是否阅读。在教师登录后台后，若有未阅读的反馈，系统会提示教师。当教师阅读后，反馈自动调整为已阅。

3.2.3 参数配置

在整个系统的运行中，有些重要参数是可变的，其根据教师的意愿而定。教师通过设定这些参数让学生达到自己的要求，进而实现教学的目的。这些重要参数包括学生每题最大存储数量、通过簇的最少做题数量、作业系统分数占比、做题数量在推荐算法中的权重、学习系统学生提交程序的内存限制等等。

综上，后台功能主要就是对数据的增删改查，当建立了一个完整的数据结构后，后台的实现也就非常容易了。本系统将课程与学生归纳为两个方向，并使用数据之间的关系将其关联起来，从而形成了一套完整、细致、精确的数据体系，为之后的数据库结构设计打下坚实基础。

3.3 数据库设计

结合系统功能并考虑到 Mysql 的主键外键关联特性，本系统共设计了 17 张数据表，主要分为课程相关、学生相关以及学生-课程关联三部分，部分数据表用途及结构如下：

-
- ```

 erDiagram
 que_ans ||--}| question : "N:1"
 question ||--}| que_know : "1:N"
 que_know ||--}| knowledge : "N:1"
 question ||--}| stu_test : "1:N"
 question ||--}| stu_que : "1:N"
 question ||--}| section : "1:N"
 section ||--}| chapter : "N:1"
 chapter ||--}| course : "N:1"
 chapter ||--}| stu_chapter : "1:N"
 course ||--}| stu_course : "1:N"
 chapter ||--}| sec_pro : "1:N"
 sec_pro ||--}| section : "1:N"
 chapter ||--}| setting : "1:1"
 setting ||--}| course : "1:1"
 user_info ||--}| user_problem : "N:1"
 user_info ||--}| load_log : "1:N"
 user_info ||--}| stu_test : "N:1"
 user_info ||--}| stu_que : "N:1"
 user_info ||--}| stu_section : "N:1"
 user_info ||--}| stu_chapter : "N:1"
 user_info ||--}| stu_course : "N:1"

```

数据结构的简要 ER 图如图 3-9，通过这 17 张数据表之间的相互关联以及 MySQL 的高并发性，系统的数据处理得到保障并且能够稳定运行。

本章介绍了整个系统设计的基本流程，从用户的需求分析出发设计相应的功能，根据功能进一步设计出数据库结构。此外，为保障系统判题的高效、准确以及安全，参考 LPOJ 设计了适用于自身系统的判题机，用户程序运行在沙盒中，避免因故意破坏或程序漏洞造成的损害。整个系统功能明确、体系完善、结构严谨，为后续的系统实现打下了坚实的基础。

## 4 系统实现

第三章已经将整个系统的结构设计进行了详细的描述，本章将具体介绍其实现过程及结果，针对一些功能给出了相应的代码和成果图。此外，对为提高系统运行效率、保障系统安全性做出的优化进行了简要说明。

### 4.1 开发环境

本系统开发环境是 Ubuntu18.04 操作系统，Apache 服务器以及 Mysql 数据库，开发语言是 PHP 和 Python，其中 PHP 开发占据主体，采用了 ThinkPHP 框架，其使用了面向对象的开发结构和 MVC 模式；Python 开发用于判题部分。

LAMP 是目前比较流行的架构，Apache 服务器开源免费，用户可以轻松的部署自己的配置；Mysql 安装方便、使用简单，是典型的关系型数据库；Ubuntu 是一个以桌面应用为主的 Linux 操作系统，其操作简单、运行稳定，且非常容易安装 Apache 和 Mysql；PHP 即超文本预处理器，是在服务器端执行的脚本语言，其成本低、速度快、容易学习；Python 是目前人工智能领域最火爆的语言，其包含了大量的库，便于一些功能的实现。

在前端页面开发上，即 View 层，采用了 Html+Css+JavaScript 模式，并为了美化界面使用了一些前端框架。后端开发上，即 Controller 层，采用 PHP 语言实现各种功能，并与 Model 层进行交互，实现对数据的增删改查。由于 ThinkPHP 框架本身丰富的组件和功能，在实际开发中，节约了很多的人力成本。

### 4.2 用户系统

结合 ThinkPHP 框架官方文档，首先开启服务器重写模块。由于框架本身面向对象开发和 MVC 模式，只需要创建相应的控制器并在其中定义功能函数，最终将数据渲染到视图上即可。

建立 Index 控制器，用于实现用户登录前的相关功能，包括用户登录、账号注册、找回密码；

---

#### Index 控制器

---

```
1 class Index extends Controller {
2 //用户登录、账号注册、找回密码等功能函数
3 }
```

---

#### 1) 用户登录

根据图 3-2，用户在前端输入账号和密码，点击登录按钮后，数据通过 Ajax 发送到后端，后端判断数据发送方式正确后，获取用户的数据并验证数据的合法性，若不合

法，则返回提示信息，否则确定当前时间和查询相关用户信息。若用户信息正确、账号状态正常以及密码正确，则登录成功，此时使用 session 存储用户信息，确保用户此时是在线状态，否则按相应的处理进行提示。如图 4-1。

用户登录功能

```
1 public function login() {
2 if 数据发送方式为 Ajax {
3 if 数据合法 {
4 if 用户信息正确 {
5 session 存储数据;
6 return 登录成功;
7 } else {
8 冻结、锁定等处理;
9 }
10 }
11 return 错误提示;
12 }
13 }
```

2) 账号注册

根据图 3-3，用户在前端完成个人信息输入后，后端同样先进行验证，然后判断用户是否参与了课程并没有注册过，若符合要求则注册成功，否则返回提示信息。如图 4-2。



图 4-1 用户登录界面



图 4-2 注册账号界面

3) 找回密码

根据图 3-4，用户先输入用户名，后端验证信息合法性后返回密保问题，用户输入新密码和密保答案后，后端判断答案的正确性后，返回修改成功，否则降低安全级数并提示错误信息。如图 4-3。

账户注册功能

```
1 public function get_user_question() {
2 if 验证合法 {
3 return 密保问题;
4 }
5 }
```

```
5 return 错误提示;
6 }
7
8 public function get_back_password() {
9 if 验证合法 {
10 if 答案正确 {
11 return 新密码修改成功;
12 } else {
13 安全级数降低;
14 }
15 }
16 return 错误提示;
17 }
```

建立 User 控制器，用于实现个人主页、问题反馈等登录后的功能。\_initialize 函数为控制器每次自动调用的函数，通过判断 session 内是否有数据进而判断用户是否处于在线状态中，防止非法入侵。

```
User 控制器
1 class User extends Controller {
2 public function _initialize() {
3 if (!session('username')) {
4 return $this->error('您没有登录或登录过期', url('index/index/login'), -1, 2);
5 }
6 }
7
8 //个人主页等功能函数
9 }
```

1) 个人主页

用户点击个人主页后，后端会根据 session 中的 userid 查询用户所修课程的详细信息 and 所有提交代码记录，将结果显示给用户，如图 4-4。

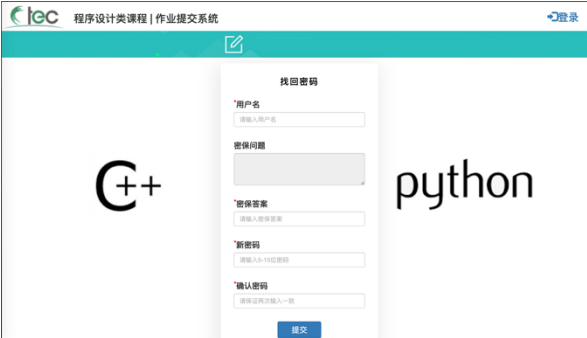


图 4-3 找回密码界面



图 4-4 个人主页界面

2) 修改密码

用户输入旧密码和新密码，后端验证合法性和正确性后，返回密码修改成功，如图

4-5。

### 3) 退出系统

删除用户 Session 和 Cookie 中全部数据, 如图 4-6。



图 4-5 问题反馈界面



图 4-6 退出系统界面

## 4.3 作业系统

作业系统实现的功能较多, 且还需保障高效性、稳定性等。仍建立 Homework 控制器, 与 User 控制器相似, 也包含\_initialize 函数。根据图 3-6, 用户有选课程、选章节、选关卡、选问题和答题 5 项操作, 为此, 设计了 5 个前端页面, 对于用户的每项操作, 后端都有对应的处理函数。

### 1) 选课程

数据库查询所有课程, 显示给用户, 如图 4-7。用户选择课程, 此时后端需先判断该课程是否存在, 避免用户通过 URL 输入造成数据不一致; 然后判断用户是否选修了此课程, 若没有, 则提示重新选课, 否则查询该课程详情、章节信息、用户得分、用户最近提交等, 并渲染页面, 如图 4-8。



图 4-7 选课程界面



图 4-8 选章节界面

### 2) 选章节

用户选择章节后, 后端先判断该章节是否属于用户已修课程, 然后判断先前章节是否通过, 保证用户按顺序完成。接下来检测 Cookie 中是否存在该章节的簇数据, 若存在, 则直接使用, 否则执行生成簇链算法。首先查询该章节包含的所有知识点并根据标号进行排序; 然后查询该章节包含的所有问题, 按照问题的知识点进行归类, 即生成簇; 最后使用递归比较簇的大小进行排序, 得到簇链。簇链生成结束后, 查询学生在簇

上的答题情况并根据管理员的要求判断是否通过，将最终结果显示给用户，如图 4-9。

簇链生成算法

```
1 public function cluster($chapter_id) {
2 if 章节合法 {
3 if cookie 含有该章节簇数据 {
4 $cluster = cookie($chapter_id); //获取 Cookie 中的簇数据
5 } else {
6 $knowledge = query($chapter_id, 'Knowledge'); //查询章节的所有知识点
7 $knowledge = sort1($knowledge); //按知识点标号进行排序
8 $questions = query($chapter_id, 'Question'); //查询章节的所有问题
9 $cluster = classify($questions, $knowledge); //将问题按包含的知识点进行分类
10 $cluster = sort2($cluster); //将簇按大小进行排序
11 $cluster = student($cluster); //将学生答题情况添加到簇链中
12 }
13 return $cluster;
14 }
15 return 错误提示;
16 }
```

3) 选关卡

用户选择关卡后，后端同样先检查章节是否合法和 Cookie 是否存在簇数据，然后判断关卡，即簇的选择是否按照顺序进行，最后将关卡中相应问题和用户的完成情况显示给用户，如图 4-10。



图 4-9 选关卡界面



图 4-10 选问题界面

获取簇链问题功能

```
1 public function question($chapter_id, $cluster_id) {
2 if 章节合法 {
3 $cluster = cluster($chapter_id); //获取簇链
4 if 选择簇合法 {
5 查询数据;
6 return 问题界面;
7 }
8 }
```

---

```

9 return 错误提示;
10 }

```

---

#### 4) 选问题

用户选择问题后，后端仍先检查数据合法性，然后查询用户对于该问题的解答情况和问题的详细信息，此时执行推荐算法，首先判断当前解答情况为正确还是错误，若为正确，从当前簇开始向后查找，直到找到未通过的簇，查找簇中未通过的题，推荐管理员设定的数量即可；若为错误，从第一个簇开始，查找相关簇，直到当前簇，统计相关簇中用户答题数量、用户通过数量、簇难度，根据管理员设定的权重计算推荐概率，使用轮盘赌法推荐即可。若用户并未尝试过此题，以答题错误处理。最后将所有数据显示给用户，如图 4-11。

---

#### 问题提交和推荐算法

---

```

1 public function submit($chapter_id, $cluster_id, $qid) {
2 if 章节合法{
3 $cluster = cluster($chapter_id); //获取簇链
4 $recommend = recommend($chapter_id, $cluster_id, $qid, $cluster);
5 查询数据;
6 return 答题界面;
7 }
8 return 错误提示;
9 }
10
11 public function recommend($chapter_id, $cluster_id, $qid, $cluster) {
12 if (state($qid, session('userid'))) { //学生答案正确，正向推荐
13 for ($i = $cluster_id; $i < len($cluster); $i++) {
14 if (!$cluster[$i]) {
15 $recommend_cluster = $i; //获取最近的未通过的簇
16 break;
17 }
18 }
19 } else { //学生答案错误，负向推荐
20 for ($i = 0; $i < $cluster_id; $i++) {
21 if (relate($cluster, $i, $cluster_id) {
22 array_push($related_cluster, get_info($cluster, $i); //获取相关簇
23 }
24 }
25 $result = get_question_from_related_cluster($related_cluster);
26 //从相关簇中计算转移概率并轮盘赌法推荐问题
27 }
28 return $result;
29 }

```

---



### 5) 答题

用户根据问题信息答题并提交代码，后端将用户代码保存到数据库，每隔一秒钟访问一次数据库，查看判题结果。若超过 5 秒仍处于等待加入队列状态，则判题服务器超时；若超过 10 秒仍处于等待判题状态，则判题机超时；若超过 30 秒仍没有判题结果，则判题超时。否则，后端根据判题结果完成相应操作，包括问题推荐。如图 4-12。

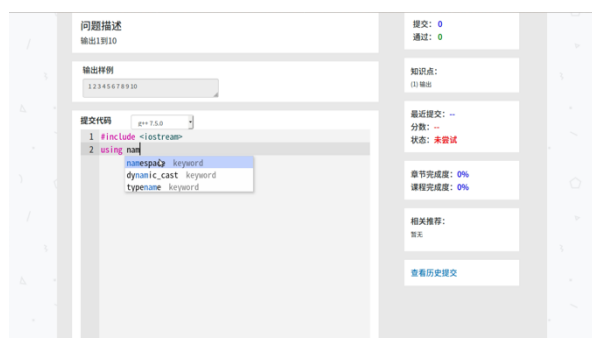


图 4-11 答题界面



图 4-12 结果界面

## 4.4 学习系统

学习系统和作业系统在功能上有很多相似之处。建立 Study 控制器，同 Homework 控制器。根据图 3-5 学习系统活动图，用户有选课程、选课节、播放音频、提交代码 4 项操作，用户播放音频和提交代码在同一界面上操作，因此设计了 3 个前端界面，同样都有对应的后端处理函数。

### 1) 选课程和选课节

选课程和作业系统几乎一致。选课节如图 4-13，用户选择课后，后端先检验课程与课节选择是否合法，然后查询课节详细信息并显示给用户。

### 2) 播放音频和提交代码

如图 4-14，进入了学习系统主页面，用户可以点击播放按钮学习课节内容，用户每次离开页面后端都会保存进度。对于进度内是可以快进的，进度外则禁止。此时，系统会根据管理员的设置在特定时刻显示提供的代码，用户可以随时更改并提交，后端判题处理同作业系统。



图 4-13 选课节界面



图 4-14 播放音频和提交代码界面

## 4.5 后台功能

后台功能主要是对数据表的增删改查操作，但是在处理过程中要保证数据的一致性和关联性。根据 3.3 后台功能设计，建立了课程相关的 Course、Chapter、Knowledge、Question 和 Section 五个控制器，学生相关的 Student、Score 和 Load 三个控制器，重要参数相关的 Setting 控制器以及管理员账户相关的 Index 和 Core 两个控制器，共计 11 个控制器。此外，开发使用了 Layui 框架美化管理界面，使用 JSZip 库将下载文件压缩成 zip 格式，使用 PHPExcel 处理上传的 xls 文件。

后台用户登录、修改密码和退出系统与前台实现几乎一致，但是要验证用户的权限。对于接收用户反馈问题，后台会先查询是否有未读反馈，若有，则在导航栏提示未读数量，管理员进入后立即标记已读。由于各功能间具有很高的相似性，因此接下来只对个别功能的实现进行介绍。

### 1) 添加课程

如图 4-15，由于 Layui 已经给出表格元素及相应的接口，按照官方文档操作即可。根据已经设计好的字段，管理员填写相应数据并提交后，后端检验数据提交方式和格式合法性，然后提交数据库即可。修改课程同理，删除课程需要管理员密码，验证通过后删除与课程关联的一切数据。

#### 添加课程功能

```
1 public function add_course() {
2 if 数据发送方式合法 && 数据格式合法 {
3 add($data, 'Course');
4 return 添加成功;
5 }
6 return 错误提示;
7 }
```

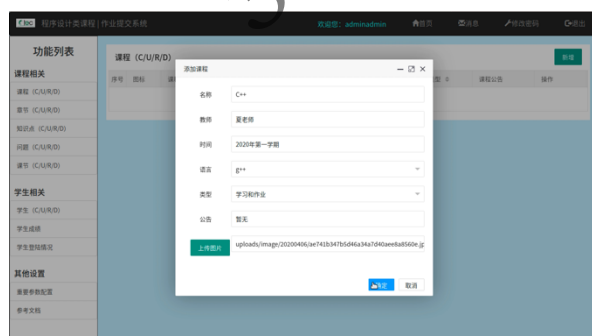


图 4-15 添加课程界面

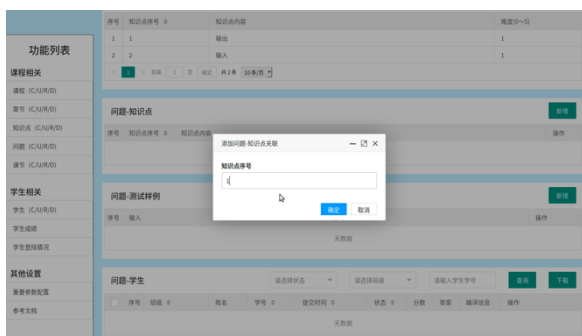


图 4-16 问题详情界面

### 2) 问题详情

如图 4-16，管理员添加问题后，系统会自动跳转到问题详情界面，或者管理员点击问题对应的详情按钮也可以进入。问题详情界面显示了问题的详细信息、问题所在章节的所有知识点、问题包含的知识点、问题包含的测试样例以及学生对该问题的解

答情况。这些即为与 Question 表关联表的数据，管理员可以随意更改。此外，增加下载学生代码功能，管理员可以有针对性或批量一键下载成压缩包形式，得到学生关于该问题的代码后，可以自行使用其他工具进行分析。

### 3) 导入学生

如图 4-17，由于大多数课程通常会有较多的学生参加，为了减少管理员的操作，增加了批量导入学生功能。管理员上传学生名单 xls 文件，文件内包含学生班级、姓名和学号，系统解析文件后，将数据保存的数据库。

### 4) 学生成绩

如图 4-18，管理员通过搜索查到指定学生和课程，系统根据课程类型显示学生学习与作业的完成数据。对于作业系统，会显示该学生的所有答题记录，管理员可以一键下载该学生的所有代码，系统会自动按章节分类。此外，在问题上与问题详情建立了链接，管理员可以快速查看问题的详细信息，即(2)问题详情中内容。对于学习系统，会显示该学生每个课节的学习时长和通过状态。

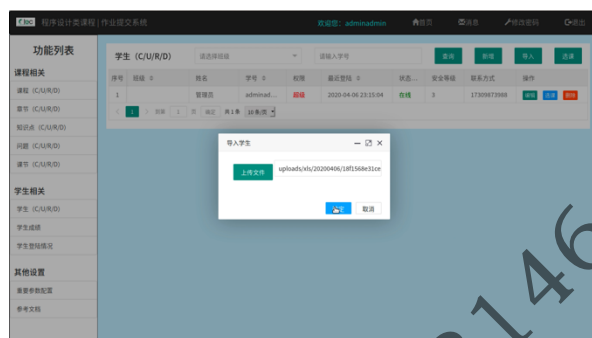


图 4-17 导入学生界面



图 4-18 学生成绩界面

## 4.6 判题机功能

判题机功能的实现是一大难点。首先，语言上选择了 Python 语言，其包含了 MySQLdb 等库，在实现过程中可减少很多不必要的麻烦。其次，进行环境配置，目前系统支持 C、C++ 和 Python 语言，在 Ubuntu 上安装不同版本的编译器即可。整个系统需要两个判题服务器，一个用于学习系统，一个用于作业系统。二者功能相近，只是查询的数据表不同。同理，两个系统判题机实现也是在数据表上存在差异。接下来具体说明实现原理。

### 1) 判题服务器

首先，定义全局队列；其次，开一个线程，连接数据库，每隔一秒钟获取一次等待加入队列提交 id，更新其状态为等待判题并存入队列；最后，服务器循环监听端口，每隔两秒钟获取一次判题机状态，若判题机为空闲状态，则提交 id 出队列，发送给判题机。

### 2) 判题机

首先，连接数据库、判题机服务器、初始化编译器路径等全局变量；其次，每隔一秒钟查看一次判题机服务器是否需判题；若需要判题，则更新当前判题机状态为忙，并

根据提交 id 获取用户代码、选用的编译器等数据，更新当前提交为判题中；然后将用户代码保存为文件形式并进行编译，若编译未通过，则将编译错误信息写入数据库，同时更新判题机状态为空闲，更新该提交为编译错误；否则根据题目要求的时间限制和内存限制在沙盒中运行程序，若运行失败，则写入相应的错误信息，同时更新状态，否则比较各个测试样例结果，若完全一致则答案正确，否则将出错的测试样例信息写入数据库，最后依旧是更新状态，等待下一轮判题。

## 4.7 本章小结

本章介绍了整个系统的实现过程，同时给出了部分功能的成果图和伪代码。总的来说，系统共分为用户系统、作业系统、学习系统、后台和判题机 5 大模块，它们相互关联却互不干扰，为后续更新提供了便利。此外，在实现过程中，为尽可能提高系统效率，使用了 Cookie 来作为缓存保存簇链结构；为了尽可能提升用户体验，使用了 Ace Editor 作为系统的编辑器。综上，整个系统的实现到此告一段落，但这并不是结束，随着用户需求的不断改变，系统也需要更新换代，而后续的道路将会更加漫长。

## 5 系统部署与测试

### 5.1 系统部署

系统部署分为两部分，一部分是基于 ThinkPHP 开发的前后端部署，另一部分是判题机部署。本系统在项目文件中已给出了部署文档，下面简要介绍一下具体流程。

#### 5.1.1 前后端部署

- 1) 配置 Apache 环境，开启 rewrite 模块，将 Apache 配置文件中 AllowOverride 后的 None 均替换为 All;
- 2) 配置 PHP 上传文件大小，找到 php.ini 文件，按照需求更改 upload\_max\_filesize 等参数大小;
- 3) 重启 Apache 服务器;
- 4) 将本系统文件移动到主机目录下，给文件添加权限;
- 5) 新建数据库，导入 psDB.sql，这个是为本系统设计好的数据库文件;
- 6) 修改本系统数据库配置文件 database.php，设置为刚建好的数据库名和密码;
- 7) 使用用户名为‘adminadmin’，真实姓名为‘管理员’进行注册，这个为预设定的管理员账号，注册后登录后台可以进行更改。

#### 5.1.2 判题机部署

- 1) 将判题机文件移到服务器任意目录下，修改 Homework 和 Study 目录下的 setting.json 文件中数据库账号和密码等;
- 2) 安装环境依赖，包括 gcc-7、gcc-8、g++-7、g++-8、python2.7、python3.6、libmysqlclient-dev 和 libseccomp-dev;
- 3) 安装 python3 依赖库 mysqlclient;
- 4) 编译必要库，进入 Judger 目录，执行 mkdir output && mkdir build && cd build && cmake .. && sudo make && sudo make install && cd .. && sudo rm -r output/ && sudo rm -r build/
- 5) 启动判题服务器和判题机，进入 Judger 目录，执行 sudo ./start.sh 1 1，其中第一个参数为作业系统判题机开启数量，第二个参数为学习系统判题机开启数量，均可任填。

### 5.2 系统测试

在系统部署成功后，需要对功能、性能、安全等方面进行全方位的测试，保障系统能正常投入使用。系统测试的工作量非常巨大，其需要结合系统各功能结构设计合理的测试用例，通过这些测试用例找出潜在的错误和缺陷。系统测试主要有黑盒测试和白盒测试两种方法。黑盒法是指不考虑被测程序的内部结构和处理过程，只关心它的

输入和输出是否能达到预期结果，也被称为功能性测试。白盒法是指利用程序内部的逻辑结构和有关信息，设计或选择测试用例，对程序所有逻辑路径进行测试，通过在不同点检查程序的状态，确定实际的状态是否与预期的状态一致，也被称为结构性测试。

受开发时间的限制，系统各功能模块在实现过程中均进行了简单的黑盒测试，具体结果见图 4-1~4-18。虽然实现过程中各功能、性能都满足了设计要求，但仍需投入到实际应用中以进一步确保系统的可用性、完备性等。

此外，对于白盒测试法，主要对 SQL 语句的逻辑关系进行了验证。由于整个系统中存在大量的表之间关联后的增删改查操作，因此验证 SQL 语句的正确性非常重要。在 ThinkPHP 框架中，提供了 BuildSql 函数用于将数据库操作转变为 SQL 语句，针对这些 SQL 语句，一方面直接逻辑分析其正确与否，另一方面在数据库中执行通过结果来进行验证。在测试过程中，确实发现了一些结果与预期不符，通过不断的更改测试，最终达到了要求。

后续测试版本将会投入使用，随着大量用户的参与，系统暗含的问题也将会暴露出来，届时会进一步完善系统，同时也会开发出更多的功能，提升用户的体验。

## 6 结论与展望

### 6.1 结论

在人工智能领域飞速发展的今天，人性化的软件服务越来越受到广大用户的喜爱与追捧，将这种服务与教育事业相结合也是我国古代“因材施教”理念的一种体现。分析了近年来在线编程系统的发展情况，本文设计并实现了一款个性化推荐编程系统。该系统目前支持 C、C++ 和 Python 语言编程课程，并为用户和管理员提供了丰富的功能。

整个系统的开发分为需求分析、系统设计、系统实现和系统测试四个阶段。在需求分析阶段，统计了系统的主要使用者学生和教师的需求并对安全和性能进行了简要说明；在系统设计阶段，将系统划分为前台功能、后台功能、判题机和数据库结构四部分，分别对各部分的相关功能的具体流程进行了详细描述；在系统实现阶段，介绍了实现的环境和语言，给出了五大模块对应的成果图和伪代码。在系统测试阶段，先是讲述了系统部署的具体流程，然后介绍了系统测试的两种常见方法并阐述了其在本系统上的体现。系统开发是一个漫长且艰巨的过程，从需求分析到系统测试，环环相扣，要保证每一步的正常执行。

整个开发过程虽然不易，但也是一种很好的锻炼方式。从最开始对网站开发的学习到对 ThinkPHP、Bootstrap、Layui 等的框架的尝试再到整个系统的实现部署测试，付出一份汗水，收获一份知识与成长。

### 6.2 展望

智慧化教学已有多年的发展历史，从早期的投影仪幻灯片发展到触控式黑板，科技在不断影响着教学模式。随着教学的传统理念与现代化技术的不断结合，为学生提供个性化服务必将是未来的一大潮流。

对于本系统，由于开发时间的限制，仍然有很多地方需要改进。为了丰富系统功能、提升用户体验，对未来的 2.0 版本提出了以下期许和意见：

- 1) 提高泛化能力。当前本系统仅支持 Linux 操作系统，在后续的开发中，可考虑扩展到 Windows、MacOS 等主流操作系统上。此外，还需支持更多的编程语言。当前本系统支持 C、C++ 和 Python 语言，后续可以加入 Java、PHP 等。
- 2) 增加题库功能模块。在实际情况中，对于不同的语言课程，很可能存在相同的问题，目前本系统只能重复添加或者复制问题，这样造成了数据的重复存储。考虑开发题库功能，对各章节的问题从题库选择添加即可。
- 3) 美化前端页面。相比于目前一些火爆的在线评测系统，本系统界面较为丑陋，后续应美化界面，让用户有个更好的体验。
- 4) 增设更新接口。本系统后续会在 GitHub 上开源，为便于随时更新，应增设更新接

口，在部署上可以更加容易、高效。

- 5) 代码重构，完成相关文档。完成变量命名规范，整理各功能函数，添加适当注释，书写代码文档。

3066731463@qq.com