

北京交通大学 海滨学院

学士学位论文

基于 Vue 的前后端分离备考系统的设计 与实现

姓 名	李 郡
学 号	17851093
学 院	计算机与信息技术学院
专 业	计算机科学与技术
指导教师	王建
职 称	讲师

二零二一年 六 月 五 日

学位论文原创性声明

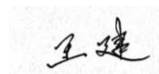
本人所提交的学位论文《基于 Vue 的前后端分离备考系统的设计与实现》，是在导师的指导下，独立进行研究工作所取得的原创性成果。除文中已经注明引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的研究成果。对本文的研究做出重要贡献的个人和集体，均已在文中标明。

本声明的法律后果由本人承担。

论文作者（签名）：李郡

2021 年 6 月 5 日

指导教师确认（签名）：



2021 年 6 月 5 日

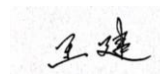
学位论文版权使用授权书

本学位论文作者完全了解北京交通大学海滨学院有权保留并向国家有关部门或机构送交学位论文的复印件和磁盘，允许论文被查阅和借阅。本人授权北京交通大学海滨学院可以将学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或其它复制手段保存、汇编学位论文。

论文作者（签名）：李郡

2021 年 6 月 5 日

指导教师（签名）：



2021 年 6 月 5 日

摘 要

在网络技术逐渐渗入社会生活各个层面的今天，传统的学习以及备考方式也面临着变革，线上学习进行备考则是网络技术一个很重要的发展方向。网上备考一方面可以突破时空的界限，另一方面用户在自己熟悉的环境中考试也会更自在，可以更好地发挥自己的真实水平，网络考试将是以后考试发展的趋势，所以迫切需要开发一个备考系统。

为了增加大学生计算机专业的知识储备量，提高考试通过率，本文设计了前后端分离备考的系统，首先对系统的数据库和功能模块进行了充分的分析以及设计，其次对系统的设计概要以及设计方法和功能做出相应的说明，最后利用 Vue, Element-ui 等开发框架实现了系统的四大模块，分别为分类管理，题目管理，文件管理，试题练习以及试卷练习。本系统采用 B/S 架构，数据库采用 MongoDB，服务器使用 Node.js 进行搭建，Express 作为服务器的基础框架，Vue 作为前端开发框架，Element-ui 作为第三方组件库，开发环境为 Visual Studio Code，前端通过 ajax 调用后端接口实现用户界面之间的数据交互，所完成的系统基本上满足了大学生备考的各种需求。

前后端分离备考系统是基于计算机专业的大学生进行的开发，系统功能较为合理，可用于在线学习，错题回顾，试卷形式练习的备考系统，可以有效地提高大学生计算机专业水平，对学生的知识存储有一定促进作用，能够让用户得到较好的备考体验。

关键词：备考系统；Vue；前后端分离

ABSTRACT

Today, when network technology has gradually infiltrated all levels of social life, traditional learning and exam preparation methods are also facing changes. Online learning for exam preparation is a very important development direction for network technology. On the one hand, online exam preparation can break through the boundaries of time and space. On the other hand, users will be more comfortable taking the exam in the environment they are familiar with, and can better display their true level. Online exams will be the trend of future exam development, so there is an urgent need for development. A test preparation system.

In order to increase the amount of computer science knowledge reserve of college students and improve the test pass rate, this paper designs a system with separate front and back test preparations. First, the database and functional modules of the system are fully analyzed and designed, and then the design outline and design methods of the system are fully analyzed and designed. Make corresponding explanations and functions, and finally use Vue, Element-ui and other development frameworks to realize the four major modules of the system, namely classification management, topic management, file management, test exercises and test paper exercises. The system adopts B/S architecture, the database adopts MongoDB, the server is built with Node.js, Express is the basic framework of the server, Vue is the front-end development framework, Element-ui is the third-party component library, the development environment is Visual Studio Code, the front-end The data interaction between the user interfaces is realized by calling the back-end interface through ajax. The completed system basically meets the various needs of college students for exam preparation.

The front-end and back-end separation test preparation system is based on the development of computer majors. The system functions are relatively reasonable. It can be used for online learning, wrong question review, and test preparation system for practice in the form of test papers. It can effectively improve the computer professional level of college students and store students' knowledge It has a certain stimulating effect and can give users a better test preparation experience.

Key words: Exam preparation system; Vue; Front and rear separation

目 录

第 1 章 绪论	1
1.1 项目背景及意义	1
1.2 国内外研究现状	1
1.3 主要研究内容	2
1.4 论文组织结构	2
第 2 章 关键技术介绍	3
2.1 B/S	3
2.2 MVVM 模式	3
2.3 数据库 MongoDB	3
2.4 框架以及运行环境介绍	4
2.4.1 Vue 框架	4
2.4.2 Node.js	4
2.4.3 Express	4
2.5 本章总结	5
第 3 章 系统需求分析	6
3.1 需求概述	6
3.2 功能需求分析	6
3.2.1 分类管理	6
3.2.2 题目管理	7
3.2.3 文件管理	8
3.2.4 试题练习	8
3.2.5 试卷练习	10
3.3 本章小结	11
第 4 章 系统设计	12

4.1	体系架构设计	12
4.1.1	整体架构设计	12
4.1.2	逻辑架构设计	12
4.1.3	网络结构设计	12
4.1.4	功能架构设计	13
4.2	功能详细设计	13
4.2.1	分类管理	13
4.2.2	题目管理	15
4.2.3	文件管理	16
4.2.4	试题练习	17
4.2.5	试卷练习	20
4.3	数据库设计	22
4.3.1	概念结构设计	22
4.3.2	逻辑结构设计	25
4.4	本章小结	28
第 5 章	系统实现	29
5.1	分类管理功能的设计与实现	29
5.2	题目管理功能的设计与实现	31
5.3	文件管理功能的设计与实现	38
5.4	试题练习功能的设计与实现	39
5.5	试卷练习功能的设计与实现	45
5.6	本章总结	48
第 6 章	系统测试	49
6.1	测试原则	49
6.2	功能测试	49
6.2.1	分类管理模块测试	49
6.2.2	题目管理模块测试	50

6.2.3	文件管理模块测试.....	51
6.2.4	试题练习模块测试.....	51
6.2.5	试卷练习模块测试.....	55
6.3	性能测试.....	56
6.4	本章总结.....	57
第 7 章	总结和展望.....	58
7.1	总结.....	58
7.2	展望.....	58
参考文献	59
致 谢	60

第 1 章 绪论

绪论主要对备考系统的开发背景及意义、研究现状、研究内容以及论文结构进行了详细分析。论述了前后端分离备考系统的意义以及整个项目要研究的内容和论文的整体结构。

1.1 项目背景及意义

本论文主要阐述了基于 Wap（Wireless Application Protocol，无线应用通信协议）技术的备考系统的设计与实现。在当前快节奏的生活方式下，传统的理论学习方式已经不能满足人们的需求，对于纸质学习资料与备考方式，用户体验效率不高、携带不便，其次，电子系统资源需要下载、安装，对用户来说不仅浪费资源，试题库得不到及时更新，导致用户不能体验实时更新的试题信息。

在这种情况下，开发基于 Wap 技术的备考系统的设计与实现，为大学生提供知识定位，加深用户对知识点的印象，对实现教育管理部门考务工作的信息化，提高教育管理质量水平，具有重要的现实意义。除此之外，该系统关键技术的研究对于其它类型的备考管理系统具有重要的参考价值，对推动考试产业的发展意义深远。

1.2 国内外研究现状

随着多媒体技术和网络通信技术的发展，基于网络的计算机远程教学手段越来越受到人们的关注。目前，在国外，题库系统逐步扩展应用于各个教学阶段。特别是中、高等学校的考试和作业以及正迅速发展的远程教育等方面的应用，起到了良好的效果。

我国自 80 年代后期，出现了多所学校或行政机构联合开发试题库的局面，同时，也出现了很多学科的试题库。90 年代，政府部门在全国范围内有计划、有组织、系统地开展了计算机辅助教学与国家题库的建设。但真正囊括了试题库的各种优点和综合了各学科特点的试题库系统并不多，特别是已经开发的题库大多存在专用性较强、网络功能较弱、交互界面不丰富等问题。针对此问题，开发一个实用型题库系统，并广泛开展题库系统的应用，必将能够促进我国计算机网络教学模式在应用中的迅速发展，从而提高计算机在各种教学管理中的应用。

前后端分离的备考系统主要面向大学生计算机专业的科目考试，对于用户来说主要包括高频错题，随机练习，分类练习，我的错题以及我的试卷模块等，通过手机网页版方式更为方便高效地提高知识储备，对当代大学生计算机水平的提高具有深远意义。

1.3 主要研究内容

随着社会的发展,网络系统资源虽有成熟的学习备考系统,但由于功能庞大,操作不便。因此开发一种操作简便适用的 Wap 系统,本系统可以根据高频错题、分类练习、随机练习、我的错题以及我的试卷提供用户需求,管理员可以通过分类管理、题目管理以及文件上传的方式对题库进行操作。另外,其采用大量的标准化试题,对题目进行统计、排序等操作,将题目以不同形式展示给用户,提高广大学子的备考效率。

本论文首先对前后端分离备考系统设计所涉及的相关技术进行逐一介绍,并主要对该系统的设计概要以及系统的设计方法和功能做了详细说明,本系统按照软件工程的思想进行系统设计,在需求分析的基础上进行功能模块划分,然后对各个模块进行分析和详细设计,并用相关技术实现,最后对相应的功能模块进行测试。

1.4 论文组织结构

本文详细介绍了基于 Vue 的前后端分离备考系统的设计与实现,全文共涉及七个部分,各章节的具体内容如下:

第 1 章 绪论。进行前后端分离备考系统的现状研究和该系统的背景意义以及主要内容,介绍了如今备考系统的国内外现状,。

第 2 章 关键技术介绍。主要介绍了前后端分离备考系统所用到的开发工具和相关技术的功能与使用。

第 3 章 系统需求分析。对整个系统进行概括性总结,介绍了前后端分离备考系统的分类管理,题目管理,文件上传,高频错题,分类练习,随机练习,我的错题以及我的试卷八个模块的功能需求。

第 4 章 系统设计。在做了系统分析的基础上,随前后端分离备考系统进行体系架构设计、功能详细设计和数据库设计,以给出较详尽的设计方案。

第 5 章 系统实现。系统前端实现了各种模块;后台实现了对信息的增删改查,以及系统管理。

第 6 章 系统测试。系统完成后对系统的模块使用,系统管理模块进行整体测试,然后得出测试结论。

第 7 章 总结。对前后端分离备考系统进行相关的总结,并对前后端分离备考系统在未来的发展方向做了简要描述。

第 2 章 关键技术介绍

本章主要论述了开发前后端分离备考系统用到的一些相关技术和开发工具，主要包括 B/S 架构、Vue 前端框架及 MongoDB 数据库等。利用这些成熟的技术可以大大提高系统的开发效率。

2.1 B/S

B/S 结构（Browser/Server，浏览器/服务器模式），所谓的 B/S 架构是由基于客户端的 Web 浏览器和远程端的服务器共同构成的。该架构的特点是非常依赖于互联网络，并对网络的行能要求非常高^[1]。该结构模式可以在客户机安装一个浏览器，服务器安装 MongoDB 等数据库，浏览器通过服务端同数据库进行数据交互。该结构方便了系统的开发、维护以及使用，共享性亦是该系统开发的主要特点，对资源利用程度相对较高，是当今软件应用的主要流行结构模式。

该系统采用的 B/S 结构通过浏览器进行手机端访问，可以面向的范围更广阔，针对大学生计算机专业更为方便使用，作为 B/S 开发，在需求量增大时，可以通过添加服务器的数量进行数据输出，对大学生的备考体验意义深远。

2.2 MVVM 模式

MVVM 即 Model-View-View Model 是一种软件开发的架构模式。模型引用后端传递的数据。视图是指看到的页面。视图模型是连接 View 和 Model 的桥梁。有两个方向：一个是将模型转换为视图，该视图将传回的数据转换为可见页面；第二个是将视图转换为模型，即将可见的页面转换为后端的数据。实现方式称为数据的双向绑定^[2]。MVVM 促进了图形用户界面的开发，而区别于传统的 MVC 模式，MVVM 模式采用的是双向绑定，即视图的变动会自动更新的 View Model 而无需手动更改。

MVVM 由 Microsoft 架构师 Ken Cooper 和 Ted Peters 发明，旨在简化事件驱动的用户界面编程，最早被用于 WPF 中，但是 MVVM 却被广泛拥有用于现在的前端开发中，vue.js 就是其中的一个使用 MVVM 模式的框架。

2.3 数据库 MongoDB

数据库是按照数据结构来组织、存储和管理数据的仓库。它主要进行管理各种系统的数据，在科学研究和决策领域更为广泛使用。本系统使用 MongoDB 数据库进行存储，这

个分布式的文件存储数据库为 WEB 应用提供更高性能数据存储解决方案。与此同时, MongoDB 是面向文档的数据库,他是非关系型数据库,而不是关系型数据库,非结构化数据的存储在大数据的实现中起着重要的作用^[3]。他的数据模型可以将一条记录表示成非常复杂的层次关系。对于 SQL 数据库适合需求确定和对数据完整性要求严格的应用。MongoDB 数据库适用于对速度和可扩展性比较看重的那些不相关的,不确定和不断发展的需求^[4]。对于面向集合存储,更易存储对象类型的数据,为获取数据渲染页面提供方便。

2.4 框架以及运行环境介绍

备考系统主要使用 Vue 框架,引入第三方 Element-ui 的 UI 库,使用 Node.js 搭建服务端,利用 Express 框架进行后台环境开发,通过 ajax 调取接口实现页面渲染。以下针对主要框架以及运行环境进行相应介绍。

2.4.1 Vue 框架

Vue.js 是一个当前热门的基于 MVVM 架构模式的 JavaScript 前端框架,它的设计主要用于组织和简化网页开发。该备考系统项目主要是手机版网页开发。核心库专注于声明性渲染和组件组合,通过交互进行数据页面的渲染。并且通过官方提供的库(如路由,状态管理和构建工具)来实现所需的功能模块。

Vue.js 是一套用于构建用户界面的,可以自底向上逐层应用的渐进式前端框架^[5]。它想比于其他主流的 JavaScript 框架例如 AngularJS 或 React 都具有运行效率高、语法简洁、自身占空间小、上手容易等特点,他的核心只关注视图层,是一个允许采用简介的模板语法来声明式的将数据渲染进 DOM 的系统,并且所有的这些都是响应式的^[6]。

2.4.2 Node.js

现阶段 Node.js 技术是用来做高并发是一种很好的选择。Node.js 是一个基于 Chrome V8 引擎的 JavaScript 运行环境。Node.js 使用了一个事件驱动、非阻塞 I/O 的模型,使其轻量又高效^[7]。Node.js 允许开发人员使用 JavaScript 进行服务器端脚本运行脚本并且在服务器端生成动态网页内容,然后将页面发送到用户的 Web 浏览器。大多数服务器端 Web 应用程序都基于 Node.js 构建,Node.js 是高度依赖模块的环境^[8]。因此,Node.js 代表了一种“无处不在 JavaScript”范式,围绕单一编程语言统一 Web 应用程序开发,而不是用于服务器端和客户端脚本的不同语言。

2.4.3 Express

Express 是一个基于 Node.js 平台的 Web 应用开发框架,可以轻松创建各种 web 或者移动端应用。Express 作为 Node.js 的一个 Web 开发框架已经被工程师们所熟知。使用该

框架的各种特性可以更加方便、快速地开发出一个比较完整的 Web 应用程序。和其他 Web 开发框架一样，Express 隐藏了代码背后的繁琐，开发者可以将工作的重心放在编写代码上，但 Express 更为优秀的是为开发者们提供了一种路由机制，分析了 Express 路由机制的特性，并研究该路由机制在程序开发中的应用方式^[9]。

2.5 本章总结

本章主要从前后端分离备考系统的主要开发工具，数据处理方式等方面对开发中用到的技术以及工具进行了简单介绍，最后对开发中将用到的框架进行了简要说明。

第3章 系统需求分析

前后端分离备考系统开发定义时期的最终步骤是对整个项目进行需求分析。需求分析是软件计划阶段的重要活动，也是软件生存周期中的一个重要环节，该阶段是分析系统在功能上需要“实现什么”，而不是考虑如何去“实现”^[10]。需求分析阶段主要是根据备考系统用户的需求，实现对备考系统功能的准确、清晰的描述，并确定要完成的工作。需求分析做的好坏直接影响到系统开发的成功与否，以及系统质量的好坏，需求分析在整个项目开发中也是相当重要的一部分。

3.1 需求概述

本系统包括系统管理员用户，系统管理员所拥有的管理权限可以进行题目分类管理、题目管理以及文件管理。在题目分类，题目管理和文件管理部分，系统管理员可操作的具体功能如下：添加分类、修改分类、删除分类、设置前置分类、题目分页、添加题目、修改题目、删除题目以及文件录入等。

大学生可以通过本系统选择不同错题频率进行作答，同时可以选择题目的分类，对不同科目进行选择，亦可选择随机练习、当前用户的错题以及动态组卷的形式进行操作。用户可操作的具体功能如下，选择不同错题频率的单选题，多选题，判断题和论述题，随机练习中可以对每个题型进行随机练习；对分类练习可以选择不同科目类型，并可以操作每个类型中的单选题，多选题，判断题和论述题，对于论述题在当前用户必须作答之后进行查看答案；当前用户可以对自身的错题进行重新作答，实现我的错题模块；我的试卷是实现以试卷形式作答；用户可以修改个人资料，错题清空，帮助与反馈，将此链接分享好友以及查看关于备考。

3.2 功能需求分析

功能需求分析是对系统整体进行分模块的需求阐述，针对系统管理员以及大学生计算机专业人群，为此系统设计了不同的模块，本章对各个模块进行了分析，并给出了各个模块的用例图。

3.2.1 分类管理

分类管理模块：系统管理员通过系统管理可以控制分类添加以及分类列表，系统管理员的具体功能包括：新建分类对科目类型进行添加，同时可以设置前置分类的知识点，以

及添加分类的名称，同时系统管理员可以分页查看、修改以及删除分类。题目分类用例如图 3-1 所示。

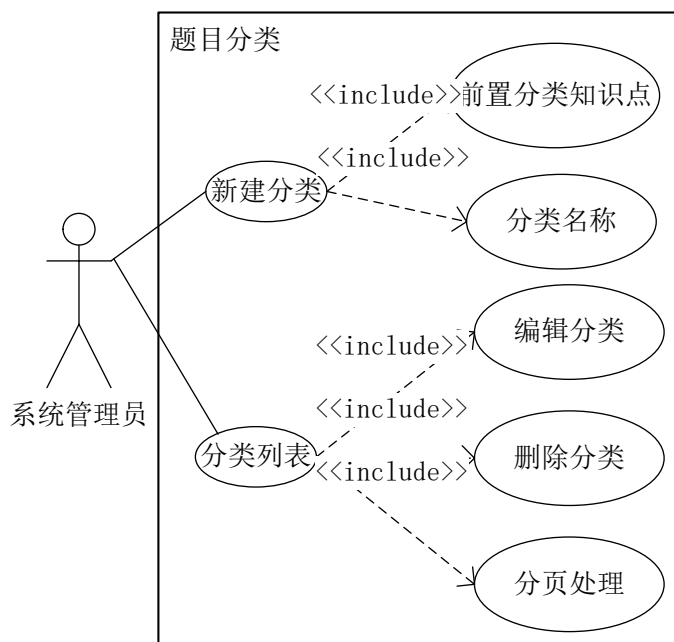


图 3-1 题目分类用例图

3.2.2 题目管理

题目管理模块：系统管理员通过系统管理可以对题型进行分类，可以分为单选题、多选题、判断题以及论述题，系统管理员可以操作的具体功能包括：录入题目，录入答案选项，设置题目答案，选择所属分类，设置难度系数，进行提交保存，对题目进行编辑以及删除，设置题目分页处理。题目管理用例图如 3-2 所示。

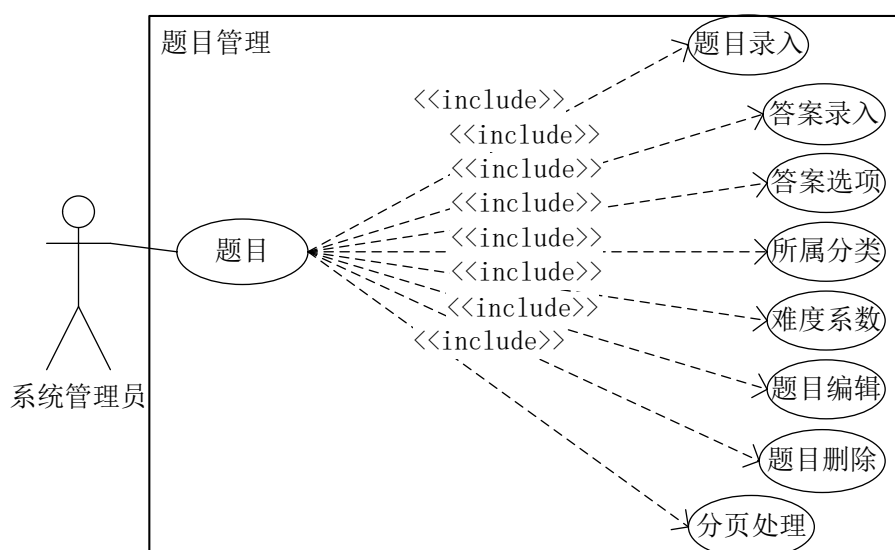


图 3-2 题目管理用例图

3.2.3 文件管理

文件管理模块：系统管理员通过系统管理可以对文件进行上传，管理员对文件的具体操作包括：选择需要的题目分类，点击上传，提交保存。文件管理用例图如 3-3 所示。

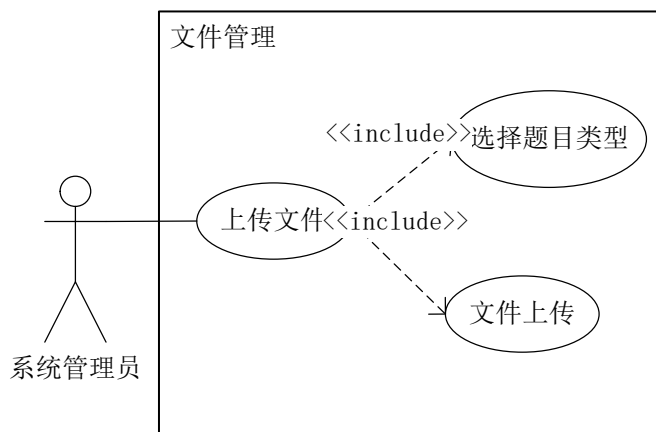


图 3-3 文件管理用例图

3.2.4 试题练习

本系统的试题练习模块主要分为高频错题，分类练习，随机练习以及我的错题，以下是对四个模块的分析，并给出了相应的用例图。

(1) 高频错题模块：用户登录后可以操作高频错题，在高频错题中可以选择易错率较高，易错率特高以及易错率极高，用户可以操作的具体功能包括：切换单选题，多选题，判断题以及论述题，点击所选题目的答案，实时显示用户作答的当前的题号以及总题号的比，当前用户作答正确个数以及当前用户作答错误个数。高频错题模块用例图如图 3-4 所示。

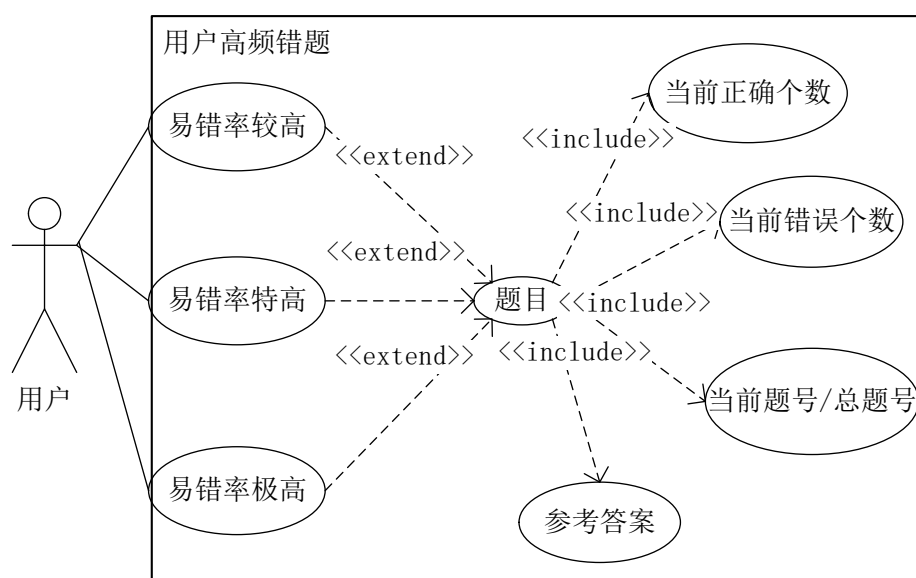


图 3-4 高频错题模块用例图

(2) 分类练习模块：用户登录后可以对分类练习进行操作，在分类练习中可以选择所需类型，用户操作的具体功能包括：类型选择，切换单选题，多选题，判断题以及论述题，用户作答题目之后显示当前题目答案，实时显示用户作答的当前题号以及总题号，当前用户作答正确个数以及当前用户作答错误个数。分类练习模块用例图如图 3-5 所示。

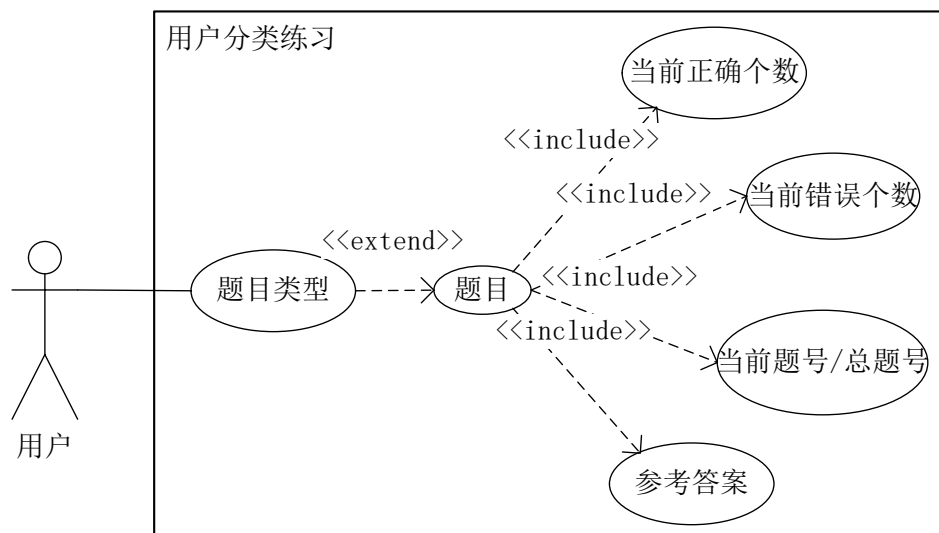


图 3-5 分类练习模块用例图

(3) 随机练习模块：用户登录后可以对随机练习进行操作，用户操作的具体功能包括：切换单选题，多选题，判断题以及论述题，用户作答题目之后显示当前题目答案，实时显示用户作答的当前题号以及总题号，当前用户作答正确个数以及当前用户作答错误个数。随机练习模块用例图如图 3-6 所示。

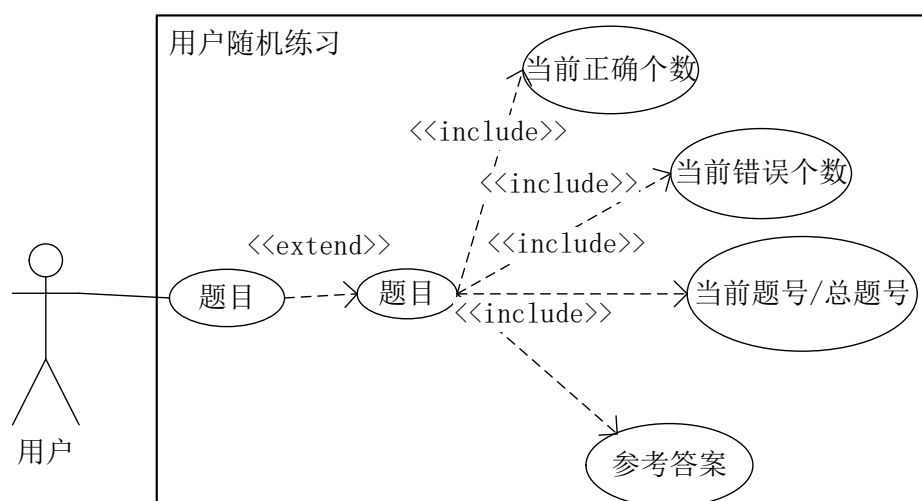


图 3-6 随机练习模块用例图

(4) 我的错题模块：用户登录后可以对我的错题进行操作，在我的错题中可以选择所需类型，用户操作的具体功能包括：类型选择，切换单选题，多选题，判断题以及论述题，

用户作答题目之后显示当前题目答案，实时显示用户作答的当前题号以及总题号，当前用户作答正确个数以及当前用户作答错误个数。我的错题模块用例图如图 3-7 所示。

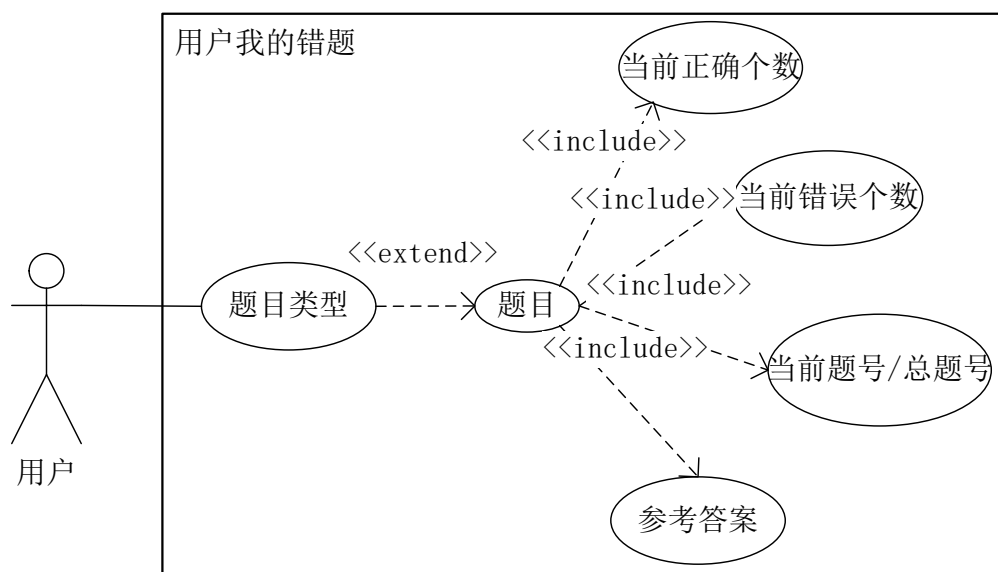


图 3-7 我的错题模块用例图

3.2.5 试卷练习

我的试卷模块：用户登录后可以对我的试卷进行操作，在我的试卷中可以选择所需类型，用户操作的具体功能包括：试卷类型选择，实时显示用户作答的当前的题号以及总题号，提交试卷，查看答案，在每套试卷作答完成可以查看各个题型的正确个数与总个数的比。我的试卷模块用例图如图 3-8 所示。

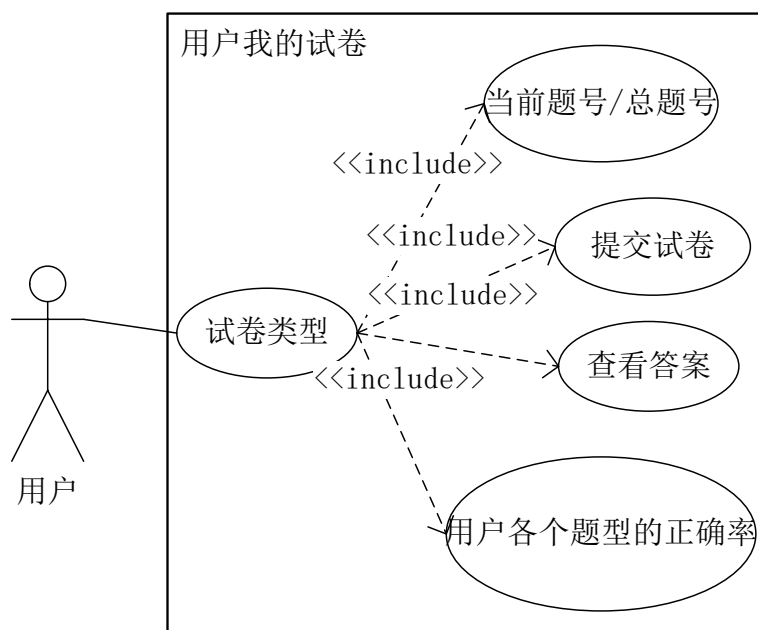


图 3-8 我的试卷模块用例图

3.3 本章小结

本章节从前后端分离备考系统功能需求方面进行了详细的分析。根据划分出的两大模块，系统管理员模块以及用户模块，细分出其中每个模块所包含的功能，通过用例图对每一个模块中的功能进行了详尽描述。

第4章 系统设计

本章主要根据需求分析阶段的准备工作对前后端分离备考系统设计与实现进行了详细设计。详细规划了备考系统设计的各个模块，介绍了备考系统设计实现的设计思想和各个功能模块的实现思路。

4.1 体系架构设计

体系架构是包括一组部件以及部件之间的联系的一个模型。体系架构设计的主要目标是开发一个模块化的程序结构，并给出各个模块之间的控制关系。另外，体系架构设计融合了程序结构和数据结构，接口定义使数据流经过程序。

4.1.1 整体架构设计

系统设计工作应该自顶向下地进行。首先设计整体架构，然后再逐层深入，直至进行每一个模块的设计。整体设计主要是指在系统分析的基础上，对整个系统的划分（子系统）、机器设备（包括软、硬设备）的配置、数据的存贮规律以及整个系统实现规划等方面进行合理的安排。

在备考系统的设计与实现的过程中，首先规划了备考系统的整个需求分析，然后对每个功能进行细化设计，将该系统进行页面交互，通过不断的完善，实现一个具体的结构设计系统。

4.1.2 逻辑架构设计

逻辑架构关心的是如何将系统分为不同部分以及各部分之间如何交互。但系统划分为不同单元后，必须考虑这些逻辑单元之间是如何协作的。逻辑架构设计的工具有：子系统划分，系统功能逻辑层次图，类与对象模型，包图，以及时序图、协作图、状态图和活动图等。

4.1.3 网络结构设计

网络架构（Network Architecture）是为设计、构建和管理一个通信网络提供一个构架和技术基础的蓝图。网络架构典型地有一个分层结构。分层是一种现代的网络设计原理，它将通信任务划分成很多更小的部分，每个部分完成一个特定的子任务和用小数量良好定义的方式与其它部分相结合。网络架构设计图如图 4-1 所示。

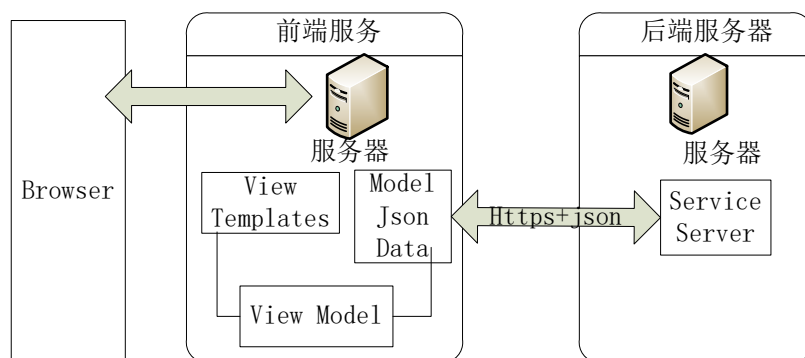


图 4-1 网络架构设计图

4.1.4 功能架构设计

功能架描述的是系统详细的功能、接口和时序特征，覆盖系统级、分系统级和单机各层级的功能实现、性能分析、内在关联与接口分析，是完成既定任务目标所需要的全部功能集合，对应于系统级、分系统级、产品级、甚至更小单元的功能、性能和相互间的逻辑关系，对系统的理解和交流具有重要意义，为保证功能和接口的准确，功能架构随着系统需求的演进而进一步定义^[1]。功能架构设计图如图 4-2 所示。

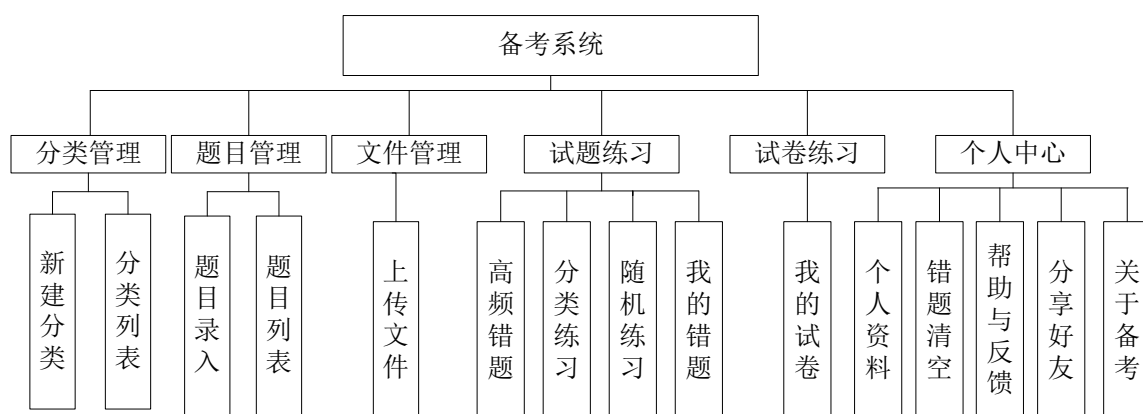


图 4-2 功能架构设计图

4.2 功能详细设计

系统的具体功能主要分为分类管理，题目管理，文件管理，高频错题，分类练习，随机练习，我的错题以及我的试卷八大部分，以下是对各个模块通过业务流程图以及时序图进行的功能详细说明。

4.2.1 分类管理

该系统的后台操作主要管理员人是系统管理员，对分类的管理直接对数据库的操作对于使用者是不方便的。创建了如图 4-3 所示的分类管理业务流程图。

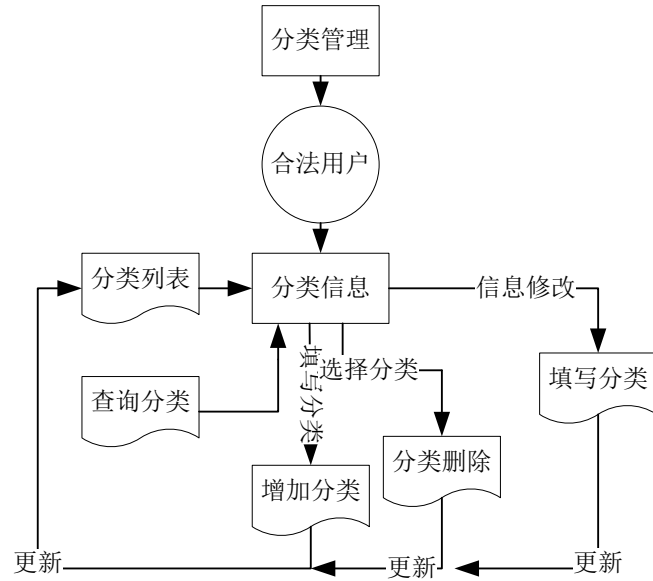


图 4-3 分类管理业务流程图

系统设计了分类管理的可视化业务，方便管理员进行相关操作，因此创建了对分类管理的时序图如图 4-4 所示。

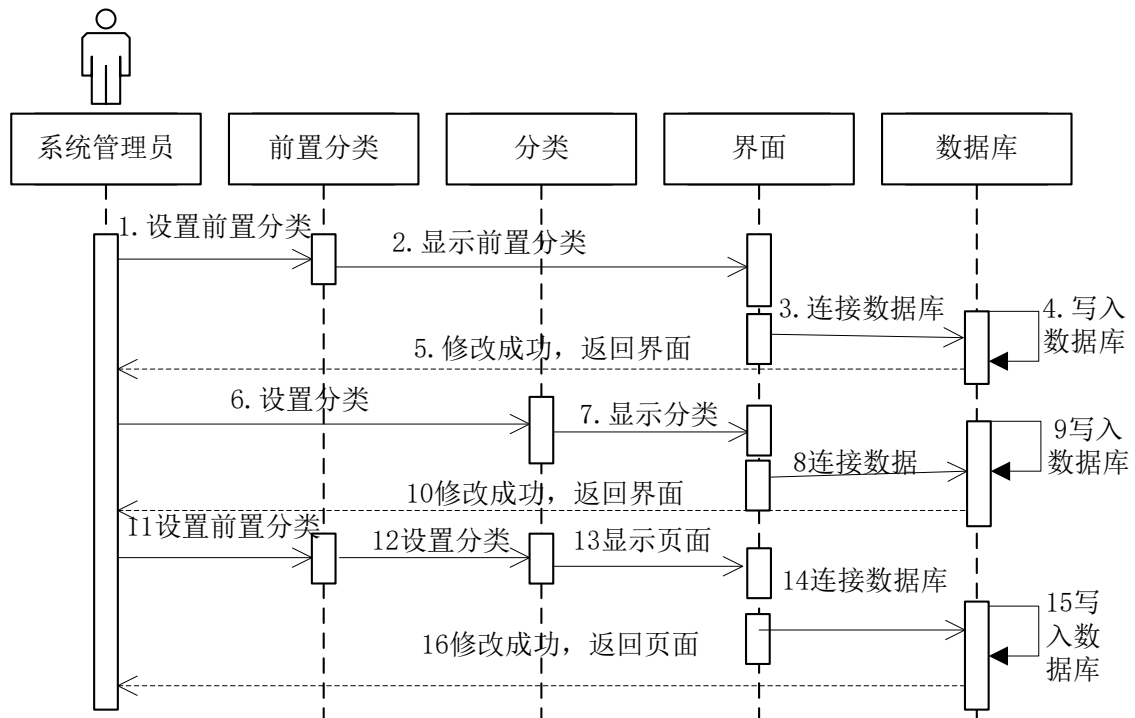


图 4-4 分类管理时序图

分类管理主要功能有添加修改分类，系统管理员可以对题目分类进行添加，也可以设置前置分类。通过数据对页面渲染将分类进行列表显示，处理分页，编辑分类以及删除分类返回给系统管理员。

4.2.2 题目管理

该系统的对题目的管理直接对数据库的操作对于使用者是不方便的。如图 4-5 所示的题目管理业务流程图。

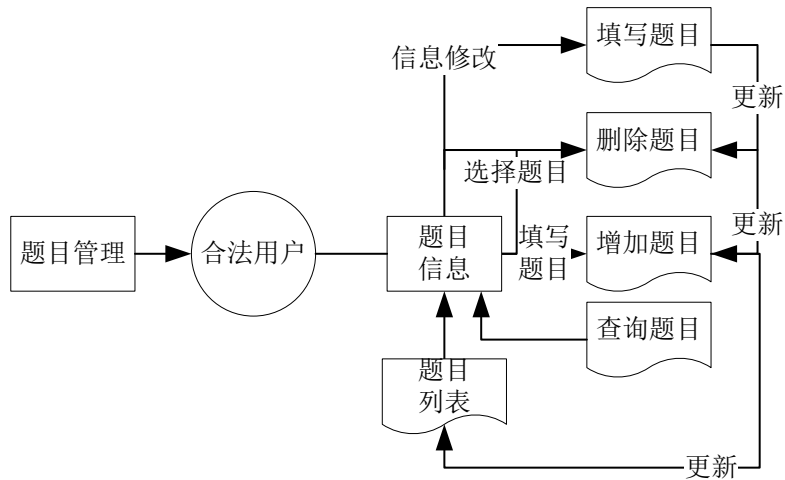


图 4-5 题目管理业务流程图

系统设计了单选题，多选题，判断题以及论述题，方便管理员进行相关操作，因此创建了题目管理的时序图如图 4-6 所示。

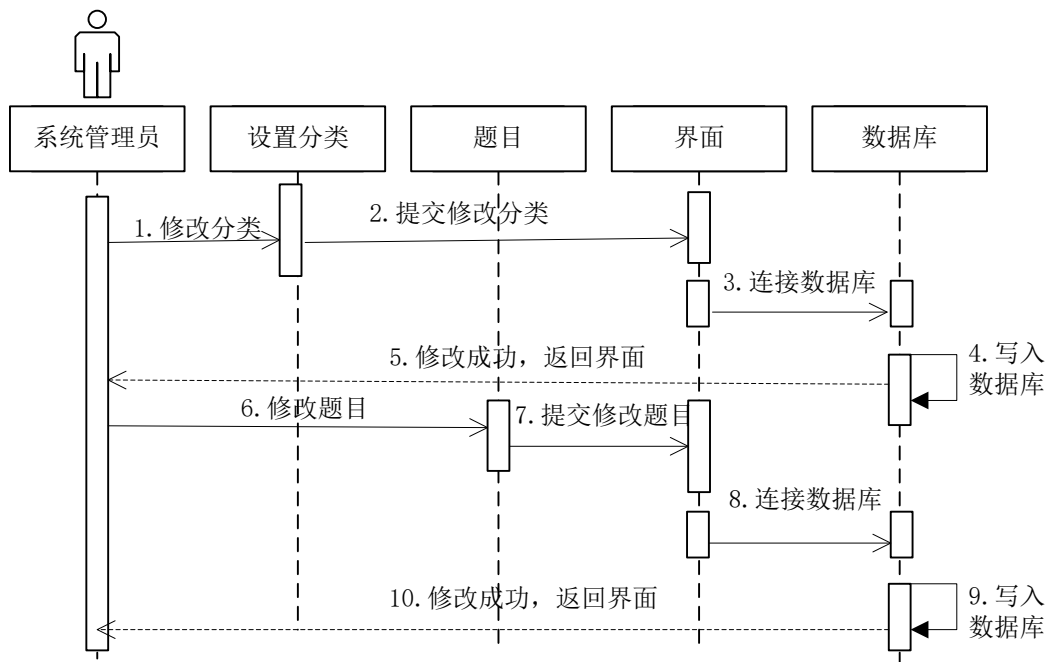


图 4-6 题目管理时序图

题目管理主要有对题目进行录入，设置题目选项答案，设置正确答案，选择难度系数，选择所选分类，查看录入题目，编辑题目，删除题目，分页处理等功能。

4.2.3 文件管理

该系统的对文件的管理直接对每个题目分别录入是不方便的。系统设计了文件上传批量题库录入,方便管理员进行相关操作,因此创建了如图 4-7 所示的文件管理业务流程图。

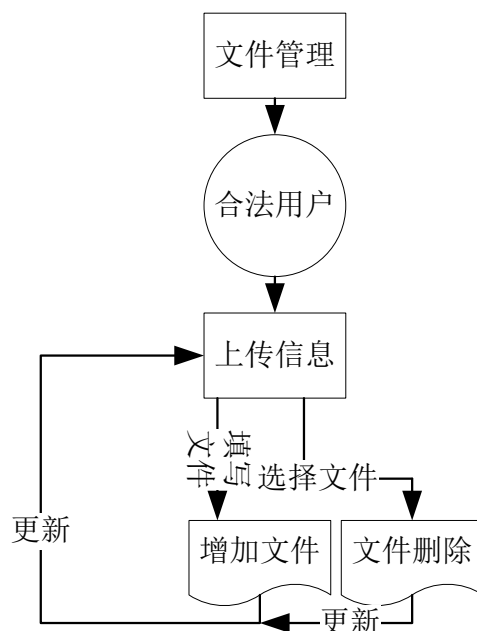


图 4-7 文件管理业务流程图

文件管理主要有对文件上传以及对文件删除的功能。对文件管理的时序图如图 4-8 所示。

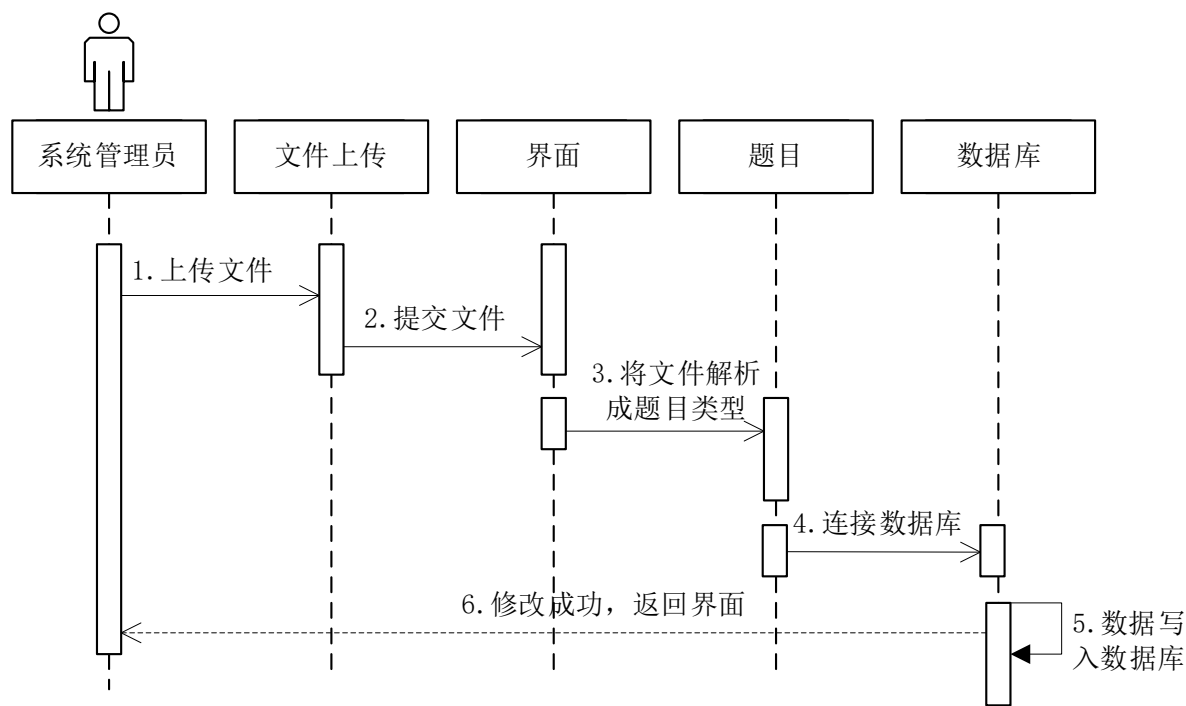


图 4-8 文件管理时序图

4.2.4 试题练习

试题练习主要针对用户的试题练习模块进行详细设计，主要包含高频错题，分类练习，随机练习以及我的错题四大模块。

(1) 高频错题模块：该系统对高频错题模块的管理是在用户体验角度进行的完善。因此创建了如图 4-9 所示的高频错题业务流程图。

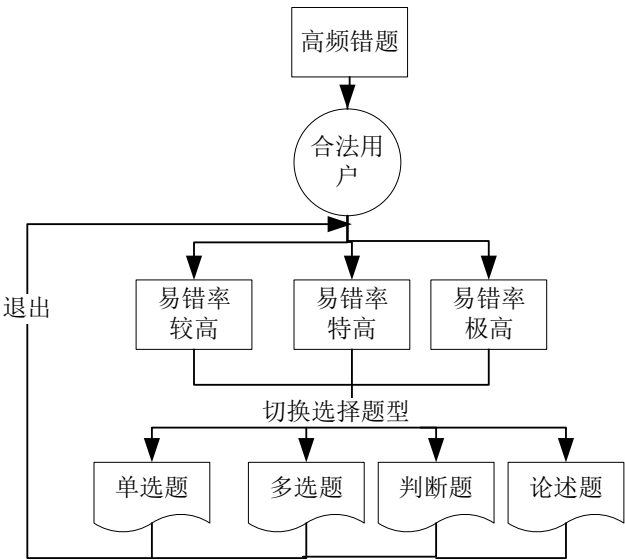


图 4-9 高频错题业务流程图

系统设计了不同难度系数的易错率较高、易错率特高、易错率极高三个错题等级，方便用户进行相关操作。对高频错题的时序图如图 4-10 所示。

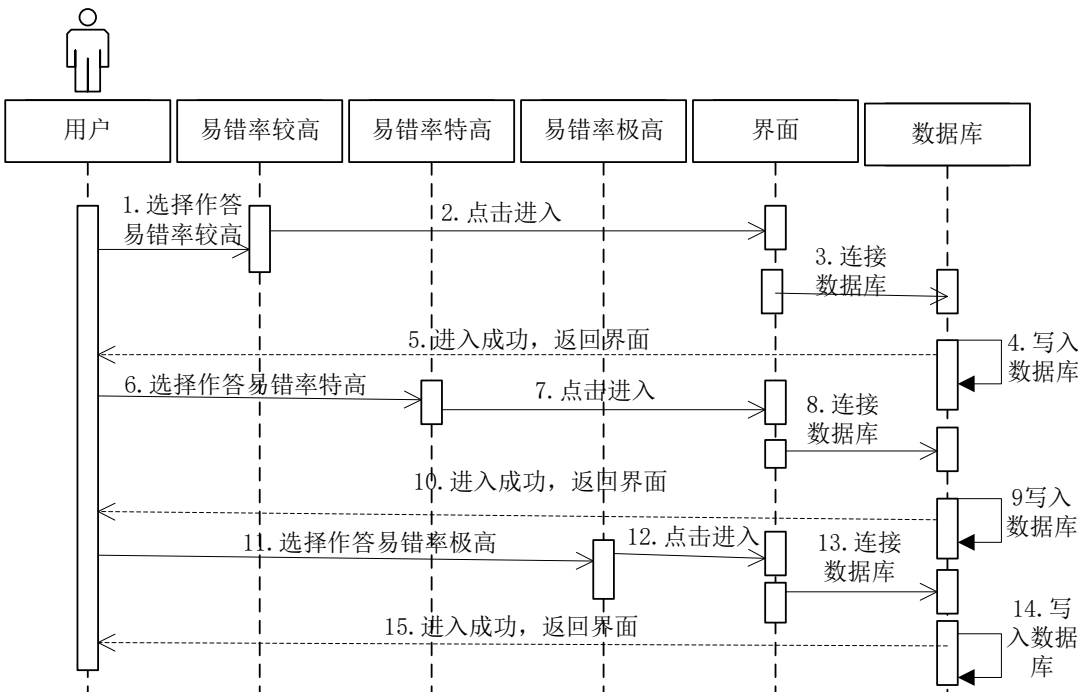


图 4-10 高频错题时序图

高频错题模块主要对题目的难度系数不同的题目划分为三个等级，实现用户对每个等级的单选题，多选题，判断题以及论述题的作答功能。

(2) 分类练习模块：该系统对分类练习模块的管理是在用户体验角度进行的完善。因此系统设计了不同类别的题型，针对类别分类实现用户对单选题，多选题，判断题以及论述题的作答，方便用户在不同科目类别进行相关操作，因此创建了如图 4-11 所示的分类练习业务流程图。

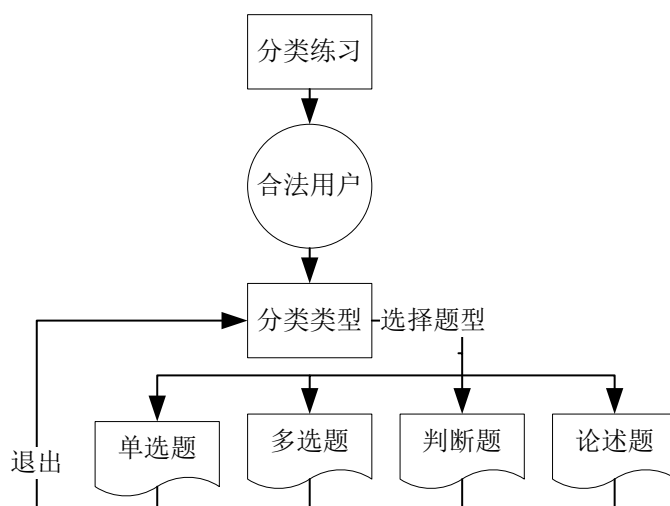


图 4-11 分类练习业务流程图

分类练习模块主要对题目的不同的类别划分的不同形式的题目作答，实现用户对每个类别的单选题，多选题，判断题以及论述题的作答以及切换功能。对分类练习的时序图如图 4-12 所示。

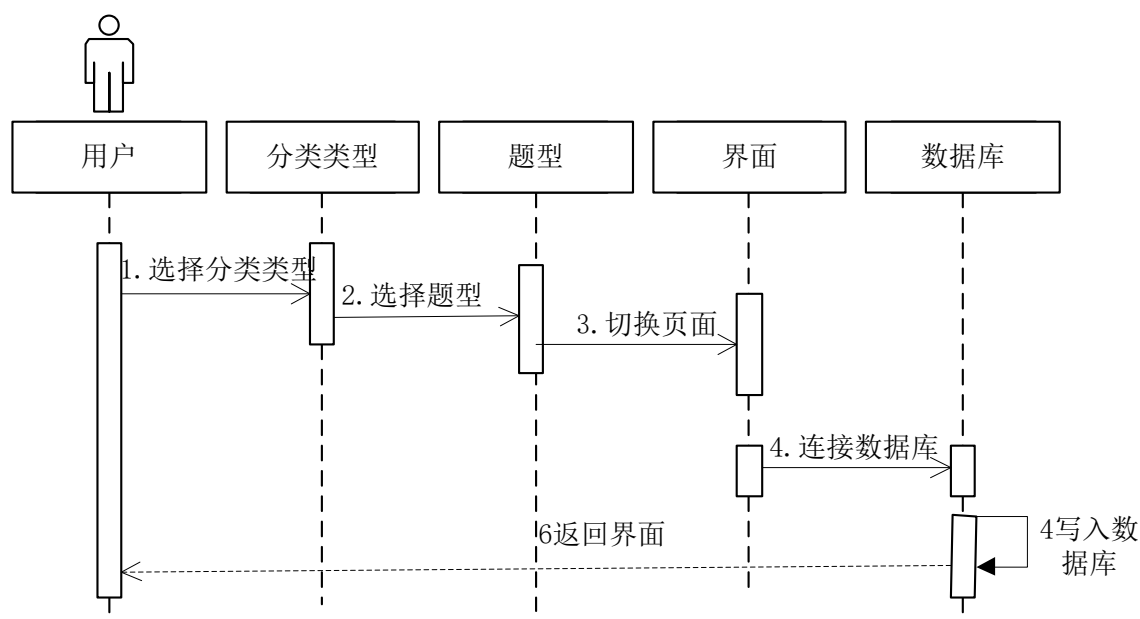


图 4-12 分类练习时序图

(3) 随机练习模块：该系统对随机练习模块的管理是在用户体验角度进行的完善。因此系统设计对单选题，多选题，判断题以及论述题进行不同题型切换和题目随机练习。方便用户在不同题型类别进行相关操作。

如图 4-13 所示的随机练习业务流程图。

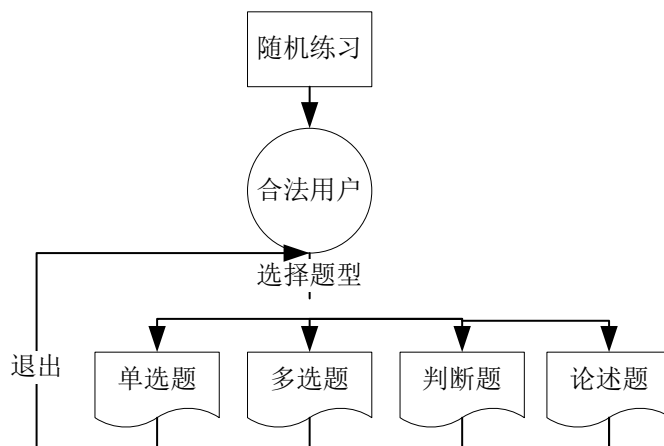


图 4-13 随机练习业务流程图

随机练习模块主要对题目的不同的题型划分的随机形式的题目作答，实现用户对每个题型的单选题，多选题，判断题以及论述题的作答以及切换功能。

对随机练习的时序图如图 4-14 所示。

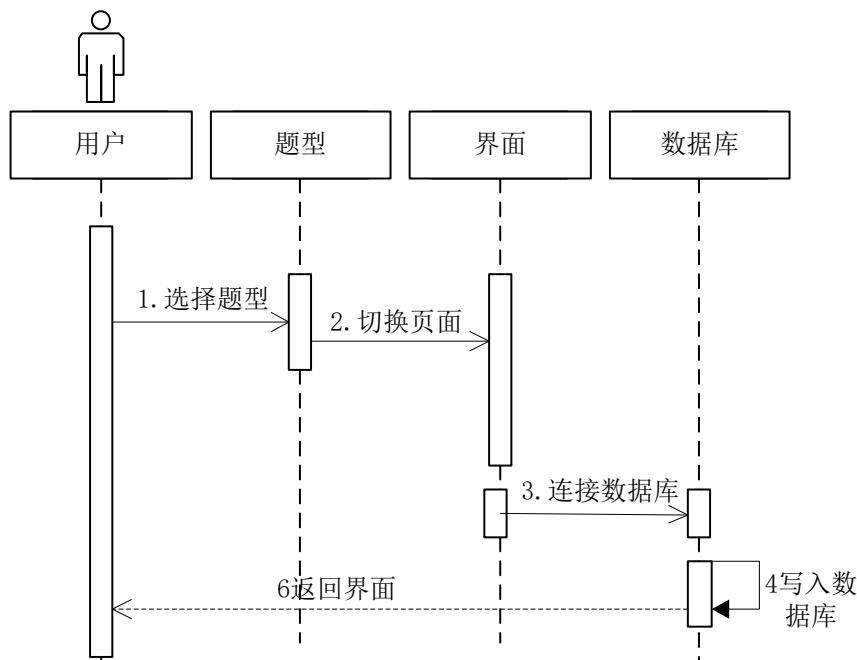


图 4-14 随机练习时序图

(4) 我的错题模块：该系统对我的错题模块的管理是在用户体验角度进行的完善。因此系统设计了不同类别的题型，在用户作答分类练习以及随机练习后，对用户的错题进行集中处理，针对当前用户所选分类的错题实现用户对单选题，多选题，判断题以及论述题

的作答。方便用户在不同科目类别中对当前错题的题目进行相关操作，因此创建了如图 4-15 所示的我的错题业务流程图。

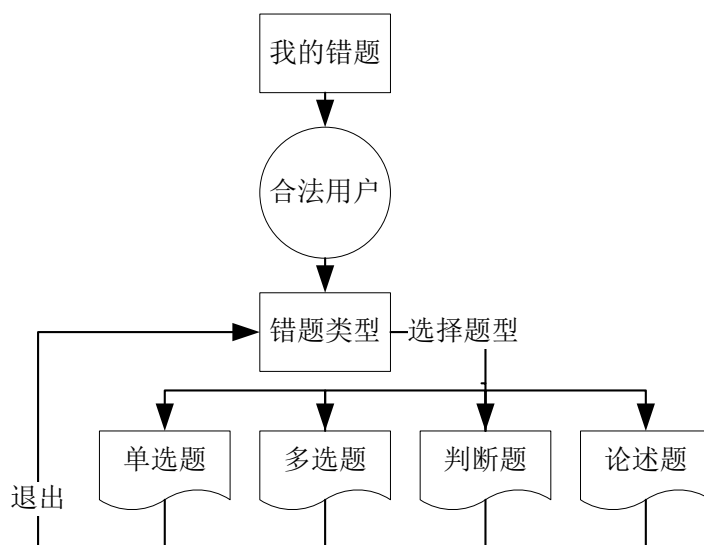


图 4-15 我的错题业务流程图

我的错题模块主要对题目的不同的类别生成的错题划分的不同形式的题目作答，实现用户对每个类别中的错题的单选题，多选题，判断题以及论述题的作答以及切换功能。对我的错题的时序图如图 4-16 所示。

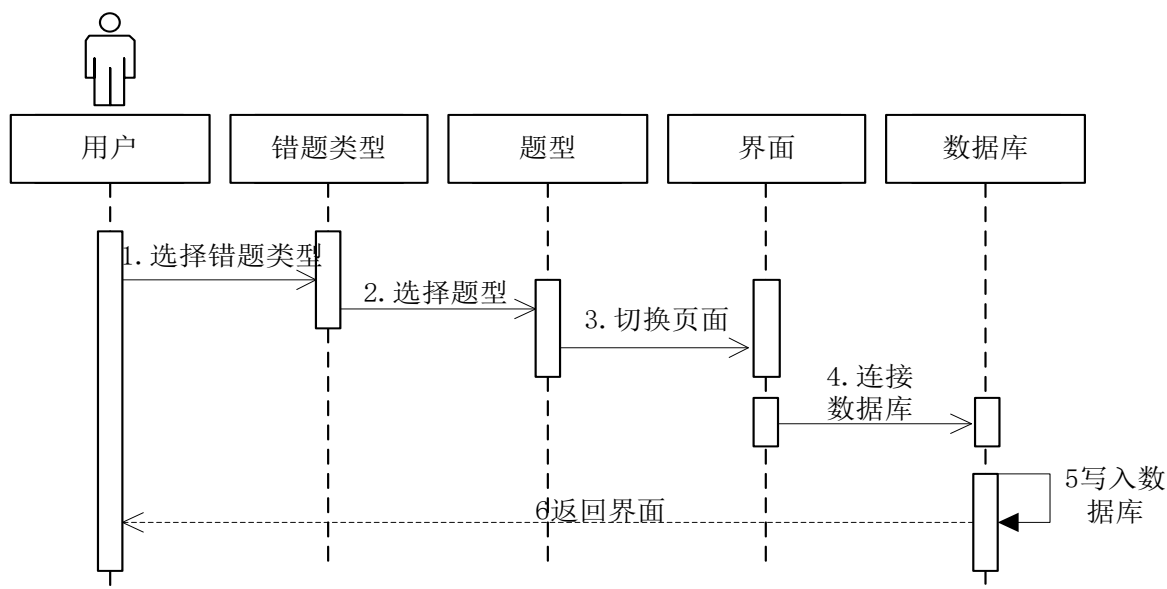


图 4-16 我的错题时序图

4.2.5 试卷练习

该系统对我的试卷模块的管理是在用户体验角度进行的完善。如图 4-17 所示的我的试卷业务流程图。

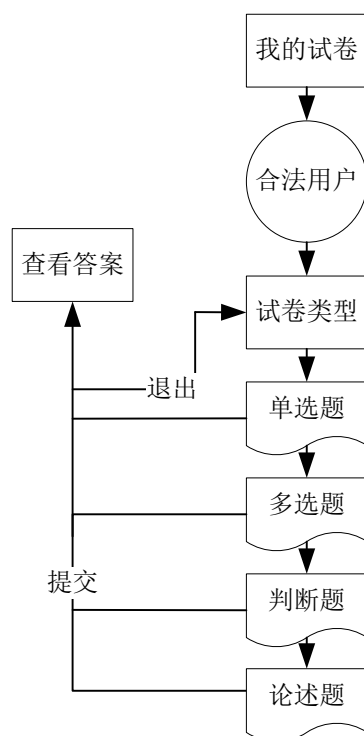


图 4-17 我的试卷业务流程图

系统设计了不同类别的试卷，针对类别分类实现用户对每个类型的试卷进行作答，方便用户在不同科目类别进行相关操作，因此创建了我的试卷的时序图如图 4-18 所示。

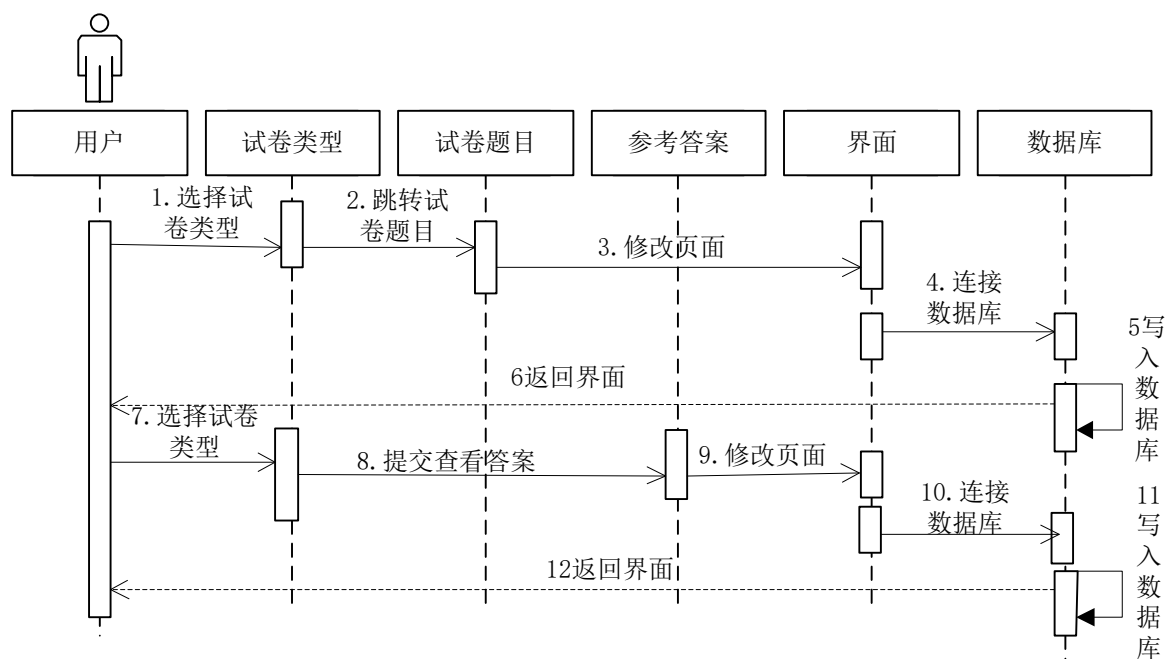


图 4-18 我的试卷时序图

我的试卷模块主要对题目的不同的类别划分的不同形式的试卷作答，实现用户对每个类别的试卷作答功能。

4.3 数据库设计

数据库设计是指在相应确切数据库管理系统中，依据用户的相应需求，对数据进行设计以及构建的过程，数据库管理包括对已经给出的数据结构进行相应的数据库构建，对已完成数据库中的数据管理的统一，通过已经搭建的数据库生态，进行修改、分析以及添加等简单问题的处理，通过应用系统对数据做出进一步的分析^[12]。

4.3.1 概念结构设计

根据需求分析阶段得出的结论，对备考系统进行数据库的详细设计。整体可以划分出十个实体，包括用户实体、分类实体、单选题实体等。

(1) 用户实体属性包括姓名和密码。用户实体属性图如图 4-19 所示。

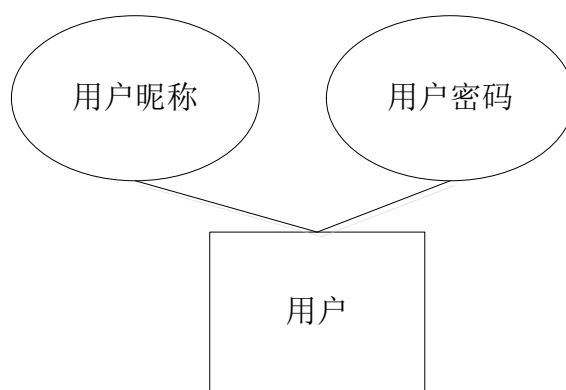


图 4-19 用户实体属性图

(2) 管理员实体属性包括管理员的姓名和管理员设置的密码。管理员实体属性图如图 4-20 所示。

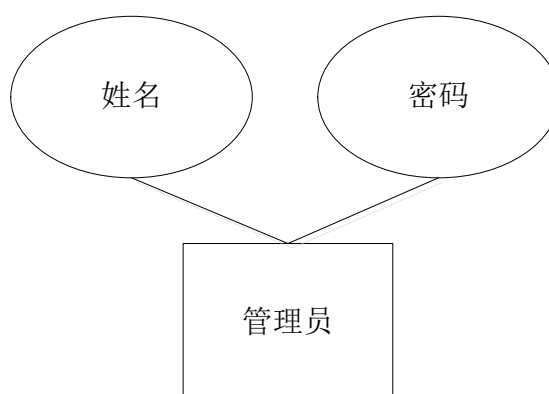


图 4-20 管理员实体属性图

(3) 题目分类实体属性包括分类名称以及前置分类知识点。题目分类实体属性图如图 4-21 所示。

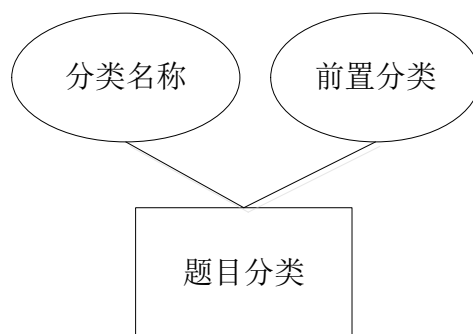


图 4-21 题目分类实体属性图

(4) 单选题实体属性包括前置分类、题目分类、题目难度系数、题目名称、题目 A、题目 B、题目 C、题目 D 以及正确答案。单选题实体属性图如图 4-22 所示。

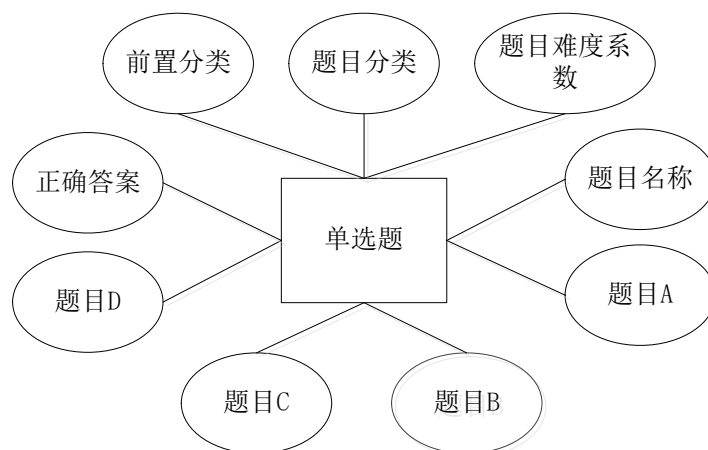


图 4-22 单选题实体属性图

(5) 判断题实体属性包括前置分类、题目分类、题目难度系数、题目标题以及正确答案。判断题实体属性图如图 4-23 所示。

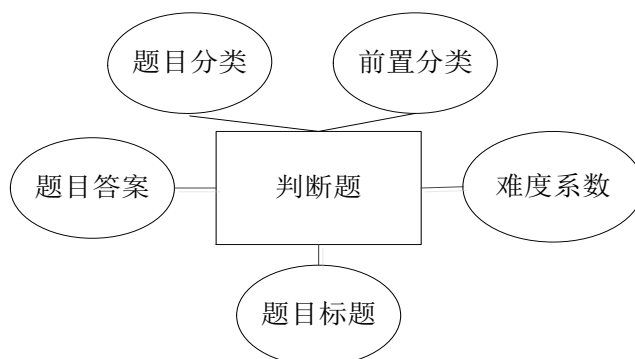


图 4-23 判断题实体属性图

(6) 多选题实体属性包括前置分类、题目分类、题目难度系数、题目名称、题目 A、题目 B、题目 C、题目 D 以及正确答案。多选题实体属性图如图 4-24 所示。

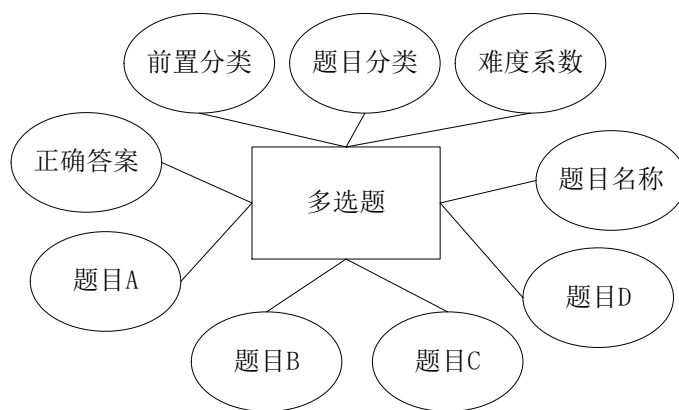


图 4-24 多选题实体属性图

(7) 论述题实体属性包括前置分类、题目分类、题目难度系数、题目标题以及正确答案。论述题实体属性图如图 4-25 所示。

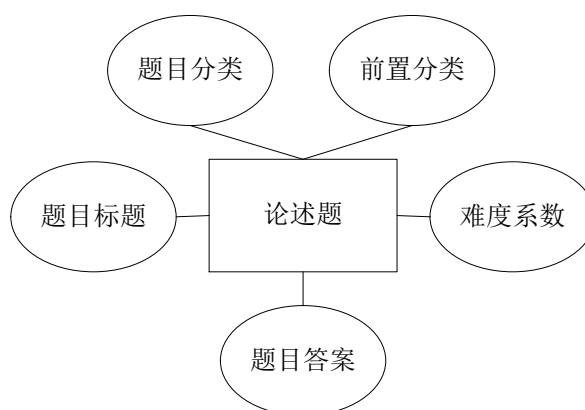


图 4-25 论述题实体属性图

(8) 用户错题实体属性包括用户名、题目的 id 以及题目类型的 id。用户错题实体属性图如图 4-26 所示。

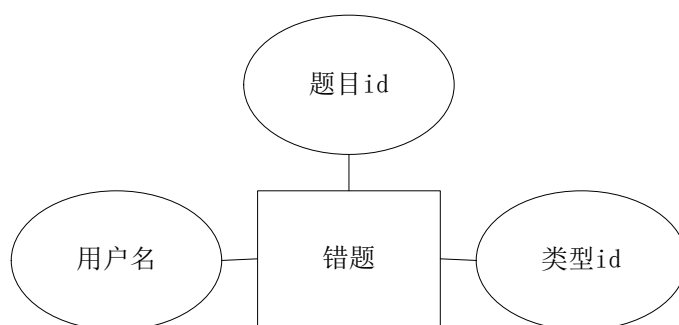


图 4-26 用户错题实体属性图

(9) 概念模型就是从现实世界到信息世界的第一层抽象，以确定领域实体属性关系，用 E-R 图表示。E-R 图主要是由实体、属性和联系三个要素构成的^[13]。几个实体的 E-R 图如图 4-27 所示。

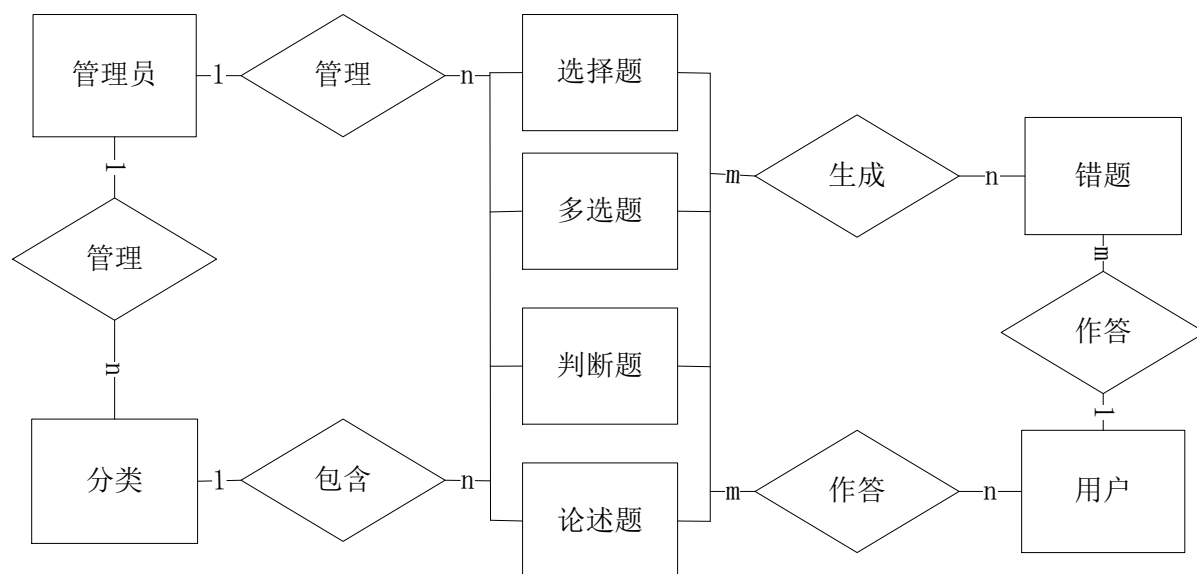


图 4-27 E-R 图

4.3.2 逻辑结构设计

数据库应用系统的逻辑设计是建立数据库应用系统过程中最为重要的一环^[14]。根据对备考系统的概念模型设计，在数据库中创建相应的数据库表，并对表中的每一个属性做相应的介绍。创建的表分别为管理员表、用户表、分类表、单选题表、多选题表、判断题表、论述题表、文件表以及错题表。以下为每一个数据库表的详细介绍。

(1) 管理员表：定义了管理员的姓名、密码这两个字段，具体结构如表 4-1 所示。

表 4-1 管理员表

字段名	字段类型	允许为空	主键
_id	ObjectId	否	是
username	String	是	否
password	String	是	否

(2) 用户表：定义了用户的姓名、密码这两个字段，具体结构如表 4-2 所示。

表 4-2 用户表

字段名	字段类型	允许为空	主键
_id	ObjectId	否	是
username	String	是	否
password	String	是	否

(3) 分类表：定义了分类的 ID、前置分类以及分类名称这几个字段，具体结构如表 4-3 所示。

表 4-3 分类表

字段名	字段类型	允许为空	主键
_id	ObjectId	否	是
pre	ObjectId	是	否
name	String	是	否

(4) 单选题表：定义了分类 id、分类名称、难度系数、题目标题、答案 A、答案 B、答案 C、答案 D 以及正确答案这几个字段，具体结构如表 4-4 所示。

表 4-4 单选题表

字段名	字段类型	允许为空	主键
_id	ObjectId	否	是
categoryId	String	是	否
categoryName	String	是	否
rate	Int32	是	否
title	String	是	否
answerA	String	是	否
answerB	String	是	否
answerC	String	是	否
answerD	String	是	否
rghtAnswer	String	是	否

(5) 多选题表：定义了分类 id、分类名称、难度系数、题目标题、答案 A、答案 B、答案 C、答案 D 以及正确答案这几个字段，具体结构如表 4-5 所示。

表 4-5 多选题表

字段名	字段类型	允许为空	主键
_id	ObjectId	否	是
categoryId	String	是	否

续表 4-5 多选题表

字段名	字段类型	允许为空	主键
rate	Int32	是	否
title	String	是	否
answerA	String	是	否
answerB	String	是	否
answerC	String	是	否
answerD	String	是	否
rghtAnswer	String	是	否

(6) 判断题表：定义了分类 id、分类名称、难度系数、题目标题以及正确答案这几个字段，具体结构如表 4-6 所示。

表 4-6 判断题表

字段名	字段类型	允许为空	主键
_id	ObjectId	否	是
categoryId	String	是	否
categoryName	String	是	否
rate	Int32	是	否
title	String	是	否
rightAnswer	String	是	否

(7) 论述题表：定义了分类 id、分类名称、难度系数、题目标题以及正确答案这几个字段，具体结构如表 4-7 所示。

表 4-7 论述题表

字段名	字段类型	允许为空	主键
_id	ObjectId	否	是
categoryId	String	是	否
categoryName	String	是	否
rate	Int32	是	否

续表 4-7 论述题表

字段名	字段类型	允许为空	主键
title	String	是	否
rightAnswer	String	是	否

(8) 错误题表：定义了分类 id、题目 id 以及用户名这几个字段，具体结构如表 4-8 所示。

表 4-8 错误题表

字段名	字段类型	允许为空	主键
_id	ObjectId	否	是
categoryId	String	是	否
questionId	String	是	否
username	String	是	否

4.4 本章小结

此章节主要从分类管理、题目管理、文件上传、试题练习以及试卷练习几个模块对备考系统进行了设计。界面主要从合理布局以及易学性两方面进行了阐述，系统功能设计主要对备考系统中具体要实现的功能以及数据库进行了详细设计。

第 5 章 系统实现

本章节主要论述了备考系统各模块的业务逻辑功能的实现。首先概述各个模块要完成的功能，然后结合流程图和伪代码详细介绍了实现细节，且每个功能都给出了相应的运行效果图。

5.1 分类管理功能的设计与实现

分类管理，主要功能有设置前置知识点，分类的名称、对分类的前置知识点以及分类名称进行编辑和删除、显示分类列表以及列表分页。具体的分类管理流程图如图 5-1 所示。

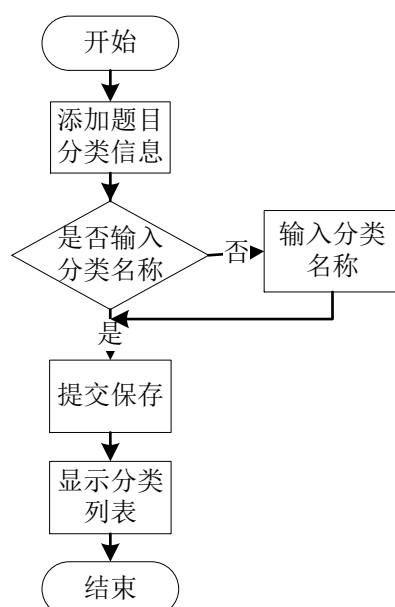


图 5-1 分类管理流程图

对分类管理的信息核心代码如下：

```

if (this.id) {
    res = await this.$http.put(`/category/${this.id}`, this.model);
} else {
    res = await this.$http.post("category", this.model); }
this.$router.push("/admin/category/list");
    
```

在分类管理界面，系统管理员在新建分类中设置前置分类知识点以及分类名称，在提交保存时，判断分类名称是否为空，如图 5-1 中分类管理流程图中，如果系统管理员未添加分类名称，则提示系统管理员题目分类名称不为空，重新输入进行添加分类，调取分类的添加接口，通过提交保存可以将系统管理员输入的分类名称以及前置分类知识点添加到

数据库同时跳转到分类列表页面。在列表页面系统管理员可以进行编辑和删除，同时列表页中设置分页处理。通过判断分类的 ID 存在与否进行添加分类或编辑分类，如果已经存在分类 ID 则调取编辑分类接口，将编辑的信息回显。如果管理员进行删除分类，则调取删除分类接口，删除当前分类 ID 对应的记录。分类管理的新建分类运行效果图如图 5-2 所示。



图 5-2 分类管理新建分类运行效果图

分页则是指定当前页和每页的记录条数，通过数组的截取方法显示当前页的信息。分类管理的分类列表运行效果图如图 5-3 所示。



图 5-3 分类管理分类列表运行效果图

5.2 题目管理功能的设计与实现

本节主要论述本备考系统题目管理模块逻辑功能的实现，其中包括单选题、多选题、判断题和论述题。

(1) 单选题：题目管理的单选题，主要功能有录入单选题题目，设置答案选项，设置正确答案，选择所属分类，设置难度系数，显示单选题列表，对单选题题目进行编辑以及删除，设置分页等功能。具体的题目管理的单选题流程图如图 5-4 所示。

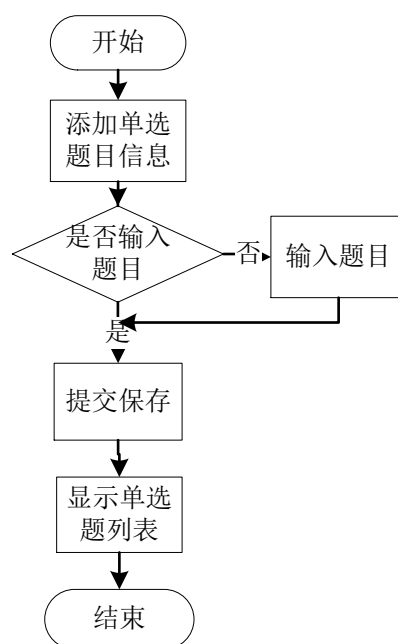


图 5-4 题目管理单选题流程图

对题目管理中单选题的信息核心代码如下：

```
if (this.id) {  
    res = await this.$http.put(`/choiceQuestion/${this.id}`, this.question)  
} else {  
    res = await this.$http.post(`/choiceQuestion`, this.question)  
}  
this.$router.push(`/admin/choiceQuestion/list`)  
const res = await this.$http.get(`/choiceQuestion/${this.id}`)  
this.question = res.data.question
```

在题目管理的单选题界面，系统管理员在单选题录入中录入题目标题，录入选项答案以及设置正确答案，选择所属分类和设置当前题目的难度系数，在提交保存时，判断录入题目标题是否为空，如图 5-4 中题目管理的单选题流程图中，如果系统管理员未添加单选题题目，则提示系统管理员题目不为空，重新输入单选题题目标题，调取单选题的添加接

口，同时调用当前题目的所属分类的接口，通过提交保存可以将系统管理员输入的单选题题目信息添加到数据库同时跳转到单选题列表页面。

在列表页面系统管理员可以进行编辑和删除，同时列表页设置分页处理。通过判断题目的 ID 存在与否进行添加单选题或编辑单选题，如果已经存在题目 ID 则调取编辑接口，将编辑的信息回显。如果管理员进行删除题目，则调取删除接口，删除当前题目 ID 对应的记录。

题目管理中的单选题的录入选择题运行效果图如图 5-5 所示。

The screenshot shows a web application interface for managing questions. The left sidebar contains a menu with categories like '题目分类' (Question Classification), '题目管理' (Question Management), and '文件管理' (File Management). Under '题目管理', there are sub-items for '选择题' (Choice Questions), '单选题录入' (Single Choice Question Entry), '单选题列表' (Single Choice Question List), '多选题录入' (Multiple Choice Question Entry), '多选题列表' (Multiple Choice Question List), '判断题' (True/False Questions), '判断题录入' (True/False Question Entry), '判断题列表' (True/False Question List), '论述题' (Essay Questions), '论述题录入' (Essay Question Entry), and '论述题列表' (Essay Question List). The main content area is titled '录入单选题' (Enter Single Choice Question). It contains the following form elements: a text input field for '录入题目' (Enter Question); a section for '录入答案选项' (Enter Answer Options) with four radio buttons and corresponding input boxes labeled '请输入选项A内容', '请输入选项B内容', '请输入选项C内容', and '请输入选项D内容'; a dropdown menu for '所属分类' (Select Category) with the placeholder '请选择'; a '难度系数' (Difficulty Coefficient) section with five star icons; and a blue '提交保存' (Submit and Save) button at the bottom right.

图 5-5 题目管理单选题录入运行效果图

分页是指当前页和每页的记录条数，通过数组的截取方法显示当前页的信息。题目管理中的单选题的单选题列表运行效果图如图 5-6 所示。

ID	题目	所属分类	操作
60782a9eef0a4f1ee83985a4	从器件角度看, 计算机经历了五代变化。但从系统结构看, 至今绝大多数计算机仍属于 () 计算机。	1.计算机组成原理	编辑 删除
60782aeef0a4f1ee83985a5	以下有关运算器的描述, () 是正确的。	1.计算机组成原理	编辑 删除
60782b2cef0a4f1ee83985a6	常用的虚拟存储系统由 () 两级存储器组成, 其中辅存是大容量的磁表面存储器。	1.计算机组成原理	编辑 删除
60842f43726e674670be67d0	下列关于虚设备技术的叙述中, 哪一个错误的?	2操作系统	编辑 删除
608eb0573b44be02ec291b90	操作系统	2操作系统	编辑 删除

图 5-6 题目管理单选题列表运行效果图

(2)多选题主要功能有录入多选题目, 设置正确答案, 选择所属分类, 设置难度系数, 对多选题题目进行编辑以及删除等功能。具体的题目管理的多选题流程图如图 5-7 所示。

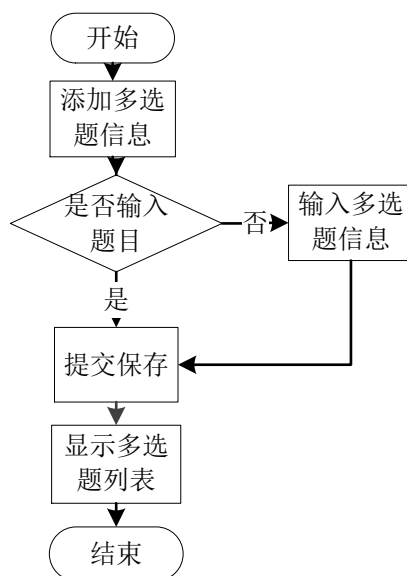


图 5-7 题目管理的多选题流程图

对题目管理中多选题的信息核心代码如下：

```
res = await this.$http.post('/multipleChoiceQuestion', this.question)}
const res = await this.$http.get(`/multipleChoiceQuestion/${this.id}`)
this.question = res.data.question
const res = await this.$http.get('category')
this.categories = res.data
this.categories.forEach((item, index, array) => {
if (this.question.category.categoryId === item._id) {
    this.value = index
    return false
}
```

在题目管理的多选题界面，在提交保存时，判断录入题目标题是否为空，如果系统管理员未添加多选题题目，则提示系统管理员题目不为空，重新输入多选题题目标题，调取多选题的添加接口，同时调用当前题目的所属分类的接口，通过提交保存可以将系统管理员输入的多选题题目信息添加到数据库同时跳转到多选题列表页面。题目管理中的多选题的运行效果图如图 5-8 所示。

The screenshot shows a web browser window with the URL `localhost:8080/admin/multipleChoiceQuestion/create`. On the left is a sidebar menu with categories like '题目分类' (Question Classification) and '题目管理' (Question Management). The '题目管理' section is expanded, showing options like '多选题录入' (Multiple Choice Question Entry), which is highlighted in blue. The main content area is titled '录入多选题' (Enter Multiple Choice Question). It contains the following fields:

- 录入题目** (Enter Question Title): A text input field.
- 录入答案选项** (Enter Answer Options): Four rows, each with a checkbox and a text input field labeled '请输入选项A内容' through '请输入选项D内容'.
- 所属分类** (Select Category): A dropdown menu currently showing '请选择' (Please Select).
- 难度系数** (Difficulty Coefficient): A star rating system with five stars.
- 提交保存** (Submit and Save): A blue button at the bottom right.

图 5-8 题目管理多选题录入效果图

(3) 判断题：题目管理的判断题，主要功能有录入判断题题目，设置正确答案，选择所属分类，设置难度系数，对判断题题目进行编辑以及删除，设置分页等功能。具体的题目管理的判断题流程图如上图 5-9 所示。

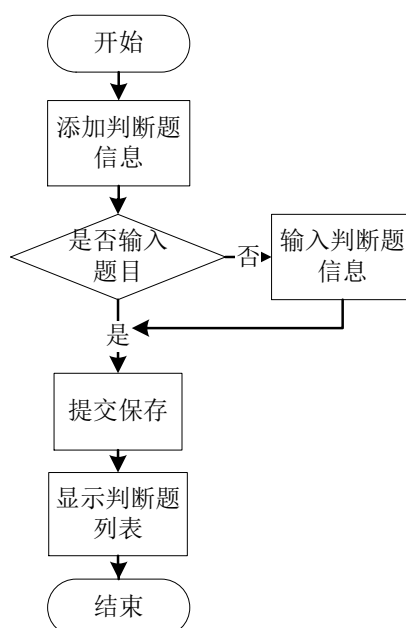


图 5-9 题目管理的判断题流程图

对题目管理中判断题的信息核心代码如下：

```
res = await this.$http.post('/judgedQuestion', this.question)
this.$router.push('/admin/judgedQuestion/list')
const res = await this.$http.get('category')
this.categories = res.data
this.categories.forEach((item, index, array) => {
  if (this.question.category.categoryId === item._id) {
    this.value = index
    return false
  })
})
```

在题目管理的判断题界面，在列表页面系统管理员可以进行编辑和删除，同时列表页设置分页处理。在题目管理的判断题界面，系统管理员在判断题录入中录入题目标题，设置正确答案，选择所属分类和设置当前题目的难度系数，在提交保存时，判断录入题目标题是否为空，如图 5-9 题目管理的判断题流程图中，如果系统管理员未添加判断题题目，则提示系统管理员题目不为空，重新输入判断题题目标题，调取判断题的添加接口，同时调用当前题目的所属分类的接口，通过提交保存可以将系统管理员输入的判断题题目信息添加到数据库同时跳转到判断题列表页面。题目管理中的判断题的录入运行效果图如图 5-10 所示。



图 5-10 题目管理判断题录入效果图

（4）论述题：题目管理的论述题，主要功能有录入论述题题目，设置参考答案，选择所属分类，设置难度系数，显示多选题列表，对论述题题目进行编辑以及删除，设置分页等功能。具体的题目管理的论述题流程图如图 5-11 所示。

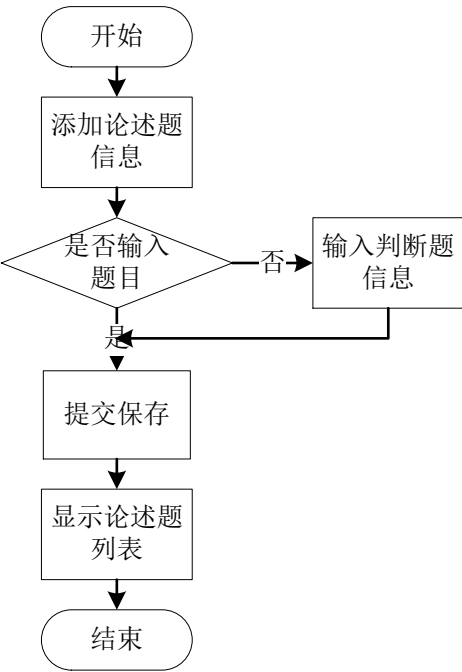


图 5-11 题目管理的论述题流程图

对题目管理中论述题的信息核心代码如下：

```
if (this.id) {
```

```
res = await this.$http.put(`/discussQuestion/${this.id}`, this.question)
} else {
  res = await this.$http.post(`/discussQuestion`, this.question)
}
const res = await this.$http.get(`/discussQuestion/${this.id}`)
this.question = res.data.question
const res = await this.$http.delete(`/discussQuestion/${row._id}`)
```

在题目管理的论述题界面，系统管理员在论述题录入中录入题目标题，设置参考答案，选择所属分类和设置当前题目的难度系数，在提交保存时，判断录入题目标题是否为空，如图 5-11 题目管理的论述题流程图中，如果系统管理员未添加论述题题目，则提示系统管理员题目不为空，重新输入论述题题目标题，调取论述题的添加接口，同时调用当前题目的所属分类的接口，通过提交保存可以将系统管理员输入的论述题题目信息添加到数据库同时跳转到论述题列表页面。

分页是指当前页和每页的记录条数，通过数组的截取方法显示当前页的信息。题目管理中的论述题的运行效果图如图 5-12 所示。

← → ↻ ⓘ localhost:8080/admin/discussQuestion/create

目 题目分类

✉ 题目管理

选择题

单选题录入

单选题列表

多选题录入

多选题列表

判断题

判断题录入

判断题列表

论述题

论述题录入

录入论述题

录入题目

录入参考答案

所属分类

难度系数 ☆ ☆ ☆ ☆ ☆

提交保存

图 5-12 题目管理论述题录入效果图

5.3 文件管理功能的设计与实现

题目管理的文件管理，主要功能有文件上传。具体的题目管理的文件上传流程图如图 5-13 所示。

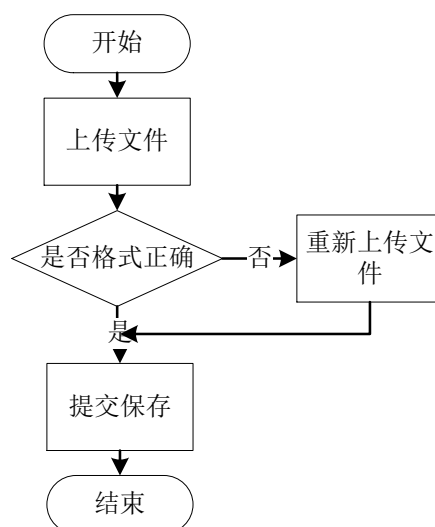


图 5-13 文件上传流程图

对题目管理中文件上传的信息核心代码如下：

```
var icon = req.body.icon
await File.create({ 'model': req.body })
fs.readFile('../server/upload/' + icon.split('/').pop(), function(err, data)
const multer = require('multer')
const upload = multer({
  dest: __dirname + '/../../upload'
})
router.post('/upload', upload.single('file'), async (req, res) => {
  const file = req.file
  file.url = `http://localhost:6500/upload/${file.filename}`
  res.send(file)
})
```

在文件上传界面，系统管理员在文件管理中进行文件上传，设置选择的题目类型，点击文件上传可以进行上传文件，在提交保存时，判断上传文件是否为空，如图 5-13 文件上传流程图中，如果系统管理员未添加上传的文件，则提示系统管理员文件格式错误，重新上传文件，在文件上传接口中需要设置文件存储路径，将文件路径进行拼接为绝对路径，同时获取当前文件所属的题型接口，通过提交保存可以将系统管理员上传的文件解析成所属题目的类型添加到相应的题目类型数据集合中。文件管理的运行效果图如图 5-14 所示。



图 5-14 文件上传效果图

5.4 试题练习功能的设计与实现

(1) 高频错题主要功能有易错率较高、易错率特高和易错率极高，通过用户切换选择题目类型，单选题，多选题，判断题和论述题进行题目作答，实时统计当前用户作答正确的个数以及错误的个数，显示当前用户作答的题数与总数量的比值，用户作答完当前题目给出相应的正确答案。具体的高频错题流程图如图 5-15 所示。

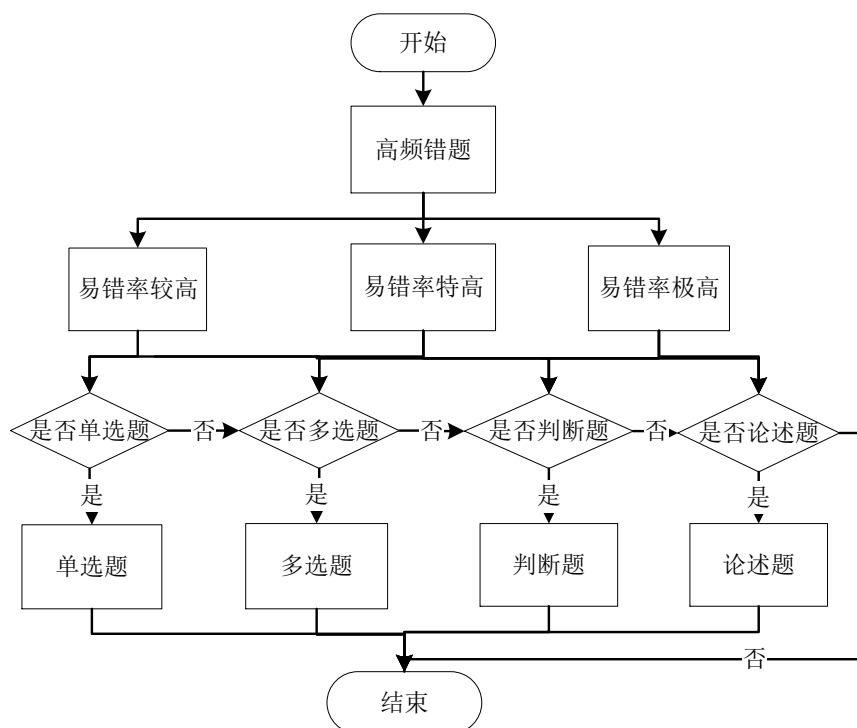


图 5-15 高频错题流程图

对高频错题的信息核心代码如下：

```
if (this.$route.params.id === "1") {  
    let rateOne = this.wrongdataArr.filter((item) => {return item.rate <  
3;});  
    rateOne.forEach((item) => {  
        let title = item.title, answerList = [item.answerA, item.answerB,  
item.answerC, item.answerD,], rate = item.rate, right = item.rightAnswer,  
category = item.category;  
        let qObj = {title,answerList,rate,right,ategory};  
        this.question.push(qObj);  
    })  
}
```

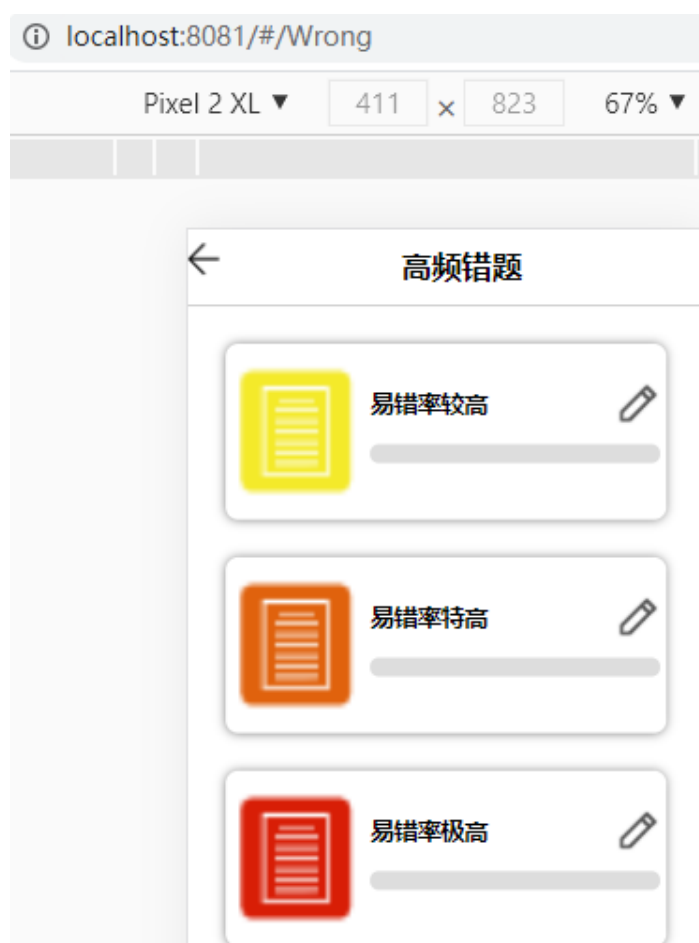


图 5-16 高频错题运行效果图

在高频错题界面，通过系统管理员设置的题目难度系数可以分为易错率较高、易错率特高和易错率极高三个等级，对不同题型单选题、多选题、判断题和论述题中分别调用各自的接口进行判断，如果页面所传参数为易错率较高，将难度系数小于 3 的题目记录添加

到当前数组中，如果页面所传参数为易错率特高，将难度系数小于 4 大于等于 3 的题目记录添加到对应的易错率特高的数组中，如果页面所传参数为易错率极高，将难度系数小于 5 大于等于 4 的题目记录添加到当前数组中，由此形成三个不同级别的错题频率。选择不同频率类型后进入题型的切换，对单选题，多选题，判断题以及论述题进行作答。具体的高频错题的运行效果图如上图 5-16 所示。

(2) 分类练习，主要功能有用户选择不同类型的科目，通过用户切换选择题目的类型，单选题，多选题，判断题和论述题进行题目作答，实时统计当前用户作答正确的个数以及错误的个数，显示当前用户作答的题数与总数量的比值，用户作答完当前题目给出相应的正确答案。

具体的分类练习流程图如图 5-17 所示。

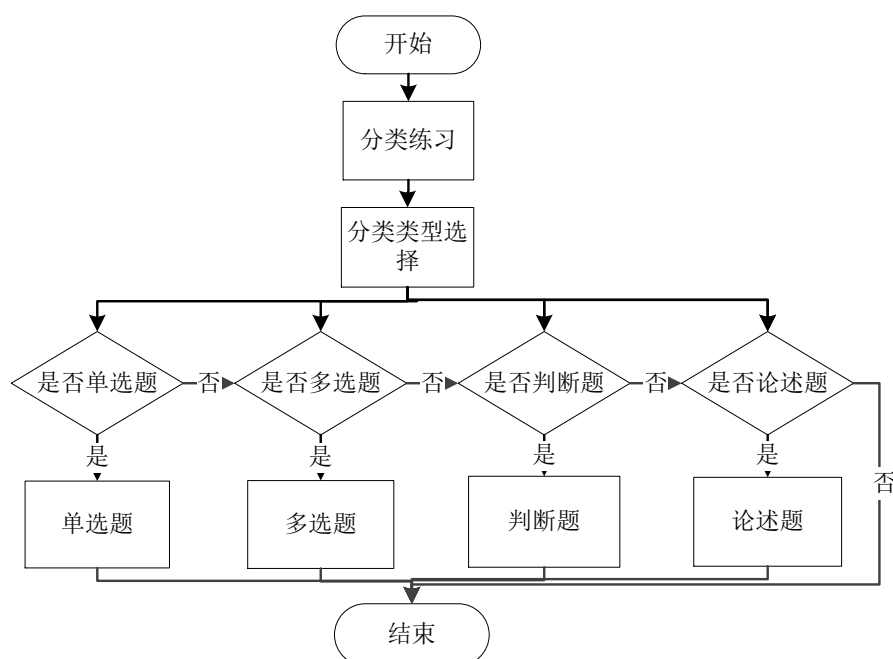


图 5-17 分类练习流程图

对分类练习的信息核心代码如下：

```

if (this.$route.params.id && typeof this.$route.params.id === "string") {
    const categoryRes = await
this.$http.get(`/category/${this.$route.params.id}`);}
if (categoryRes.data.pre) {
    const categoryPreRes=await
this.$http.get(`/choiceQuestion/${categoryRes.data.pre}`);}
const preData = categoryPreRes.data;
preData.forEach((item) => {
    this.RecomandArr.push(item.question);

```


});

在分类练习界面，首先获取前置分类的 ID，如果前置分类 ID 存在，则先调用前置分类 ID 的接口，加入到获取题目的数组中，如果不存在前置分类的 ID 则获取当前类型的题目，调用当前类型的接口，加入到获取题目的数组中。选择题目类型后进入题型的切换，对单选题，多选题，判断题以及论述题进行作答。具体的分类练习的运行效果图如图 5-18 所示。



图 5-18 分类练习运行效果图

(3) 随机练习，最要功能是对不同题型单选题、多选题、判断题和论述题进行随机排序练习，实时统计当前用户作答正确的个数以及错误的个数，显示当前用户作答的题数与总数量的比值，用户作答完当前题目给出相应的正确答案。

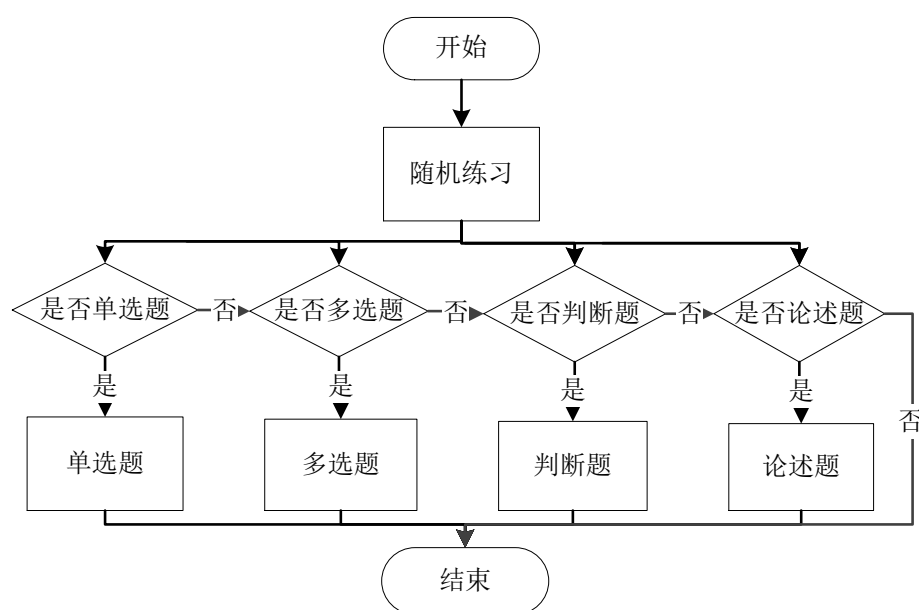


图 5-19 随机练习流程图

随机练习，最要功能是对不同题型单选题、多选题、判断题和论述题进行随机排序练习，实时统计当前用户作答正确的个数以及错误的个数，显示当前用户作答的题数与总数量的比值，用户作答完当前题目给出相应的正确答案。具体的随机练习流程图如上图 5-19 所示。

对随机练习的信息核心代码如下：

```
res = await this.$http.get("/choiceQuestion");
let random = res.data;
let randomArr = shuffle(random);
let arr = [];
randomArr.forEach((item) => {
    item.question.questionId=item._id;
    arr.push(item.question);
});
export function getRandom (max, min) {
    return Math.floor(Math.random() * (max - min) + min)
}
export function shuffle (arr) {
    return arr.sort(() => Math.random() - 0.5)
}
```

随机练习中通过导入随机模块进行随机排序相应的题目，在模块中使用 **Math** 方法生成随机数，利用 **sort** 排序将题目进行随机乱序排序。调用模块将题目数据渲染页面。

在随机练习的答题界面中，单选题通过用户的点击事件，可以显示用户作答答案与正确答案，同时判断当前用户的当前题目是否正确，如果当前用户作答正确，则在错误列表删除当前题目，在正确列表中增加当前题目，反之亦然。

判断题通过用户的点击事件，同样可以显示用户作答答案与正确答案，同时判断当前用户的当前题目是否正确，如果当前用户作答正确，则在错误列表删除当前题目，在正确列表中增加当前题目，反之亦然。

多选题则是当用户点击题目答案时进行保存当前用户的作答信息，通过用户左右滑动的距离超过一定距离之后，进行判断当前用户作答与正确答案对比，如果用户答案与正确答案一致则会继续进行题目作答，如果答案不一致则保留当前题目页面，判断当用户再次滑动再进行以下的题目作答。具体的随机排序的多选题运行效果图如图 5-20 所示。

论述题则需要在作答空间判断用户是否输入答案，在用户作答之后系统才允许进行查看答案并且可以多次点击显示隐藏答案信息。

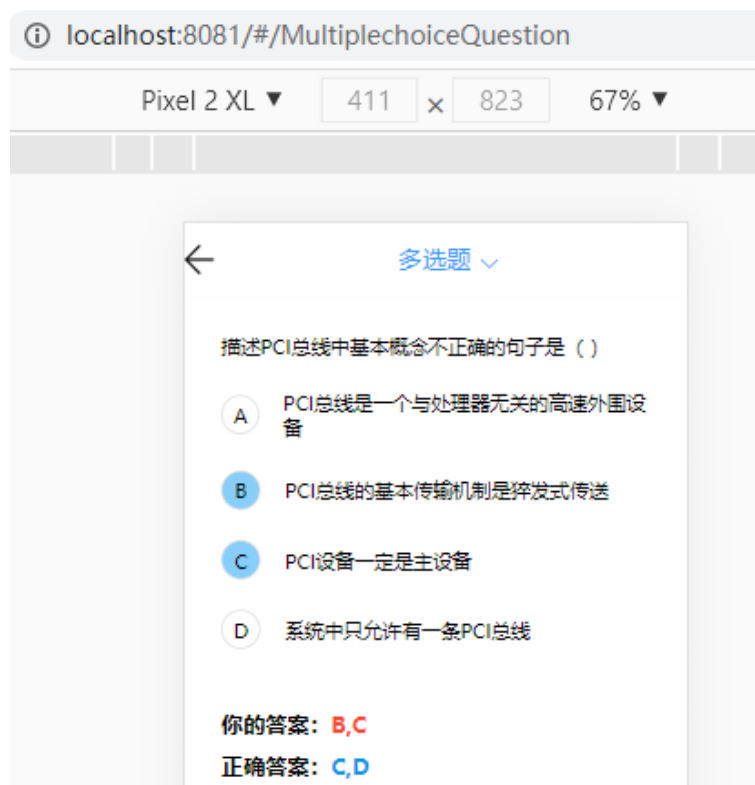


图 5-20 随机练习多选题运行效果图

(4) 我的错题，主要功能有用户选择不同类型的科目，对当前用户的错题通过用户切换选择题目的类型，单选题，多选题，判断题和论述题进行题目作答，实时统计当前用户作答正确的个数以及错误的个数，显示当前用户作答的题数与总数量的比值，用户作答完当前题目给出相应的正确答案。具体的我的错题流程图如上图 5-21 所示。

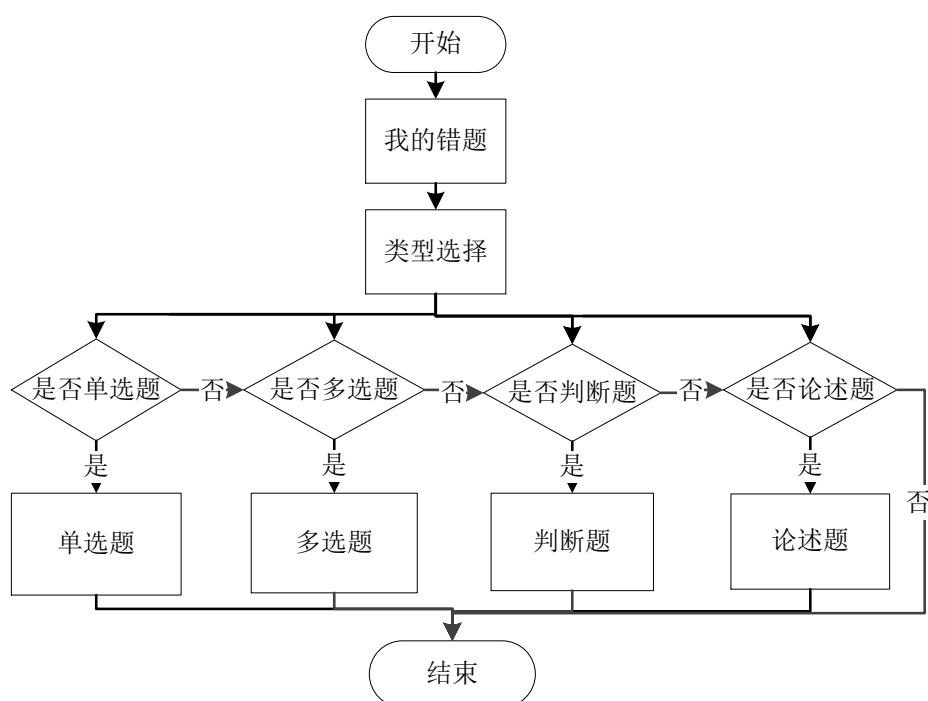


图 5-21 我的错题流程图

对我的错题的信息核心代码如下：

```
this.$router.push(`/ChoiceQuestion/Wrong/${categoryId}`)  
const res = await this.$http.get(`/category`)  
this.category = res.data  
async postWrongQuestion() {  
    let res = await this.$http.post("/wrongquestion", this.wrongQuestion)}
```

在我的错题界面，首先获取分类的 ID，调用分类的接口进行显示题目的类型，点击每个类型进行传递相应的参数调用每个题型中存入错题的接口。选择题目类型后进入题型的切换，对单选题，多选题，判断题以及论述题进行当前用户的错题作答。具体的我的错题的运行效果图如图 5-22 所示。



图 5-22 我的错题运行效果图

5.5 试卷练习功能的设计与实现

我的试卷，主要功能有用户选择不同类型的科目，对当前用户在不同科目的基础上随机组卷，每份试卷由单选题、多选题、判断题和论述题组成，可以显示当前用户作答的题

数与总数量的比值，在用户作答过程中。可以随时提交，查看错误答案。具体的我的试卷流程图如图 5-23 所示。

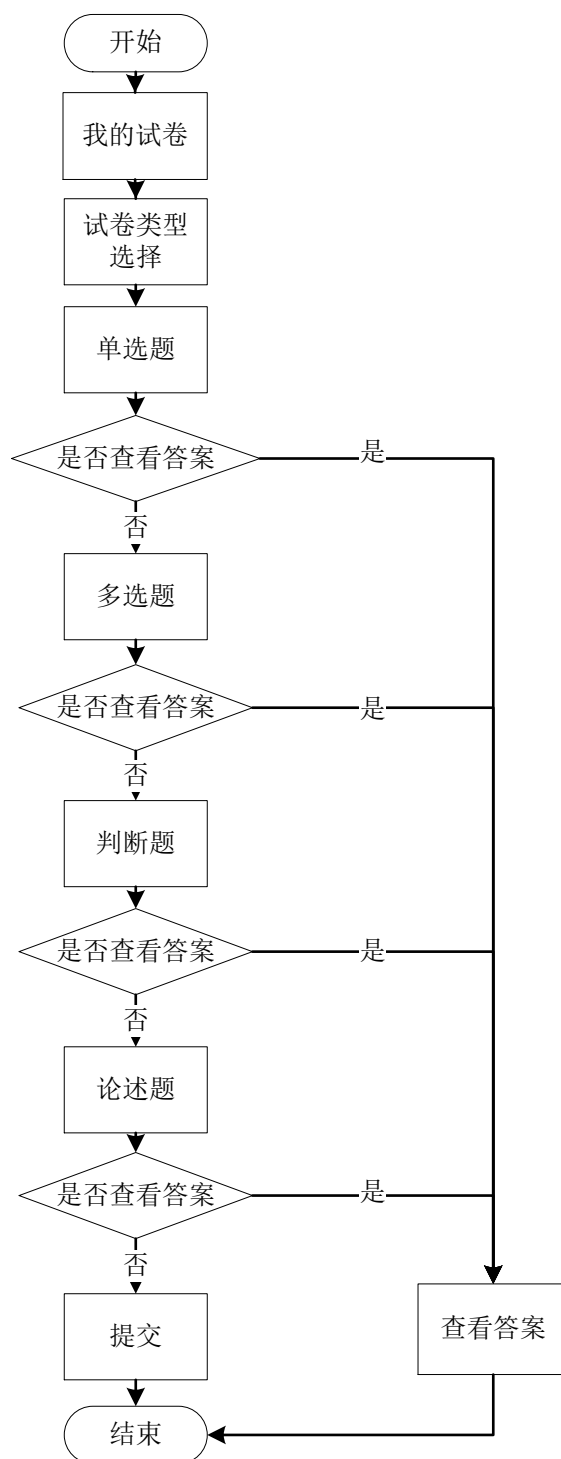


图 5-23 我的试卷流程图

对我的试卷的信息核心代码如下：

```
const res = await this.$http.get('/exam/detail', {  
  params: {id: this.$route.params.id}});
```

```
this.exam_detail = res.data;  
this.user_answer = new Array(this.exam_detail.length);  
for (var detail of this.exam_detail) {  
    if (detail.questionType == 0) {  
        detail.question.answers = [detail.question.answerA,  
        detail.question.answerB, detail.question.answerC,  
        detail.question.answerD];}}}
```

在我的试卷界面，首先获取分类的 ID，调用分类的接口进行显示题目的类型，点击每个类型进行传递相应的参数进入相对应试卷界面，通过判断题目类型值与单选题、多选题、判断题以及论述题进行匹配，如果匹配成功则获取相应的题目以及正确答案。系统设定每套试卷题量为 100，判断每当用户滑过则进行当前题目累加，与总题数对比进行渲染。在用户作答过程中可以进行提交，提交直接跳转到当前用户作答正确率的界面，此时可以点击查看错题，遍历每个题型进行显示答案。具体的我的试卷的运行效果图如图 5-24 所示。



图 5-24 我的试卷题型运行效果图

5.6 本章总结

此章节主要对每个模块中的具体功能进行了详细设计与实现。每一个功能都介绍了备考系统的实现流程以及说明了实现的核心代码，从多方面论述了每一个功能的详细实现步骤以及实现原理，最后附上运行效果图加以补充与体现。

第 6 章 系统测试

在资源有限和程序版本频繁发布的情况下，测试自动化变得至关重要，这是软件开发人员的典型情况^[15]。本章主要是对备考系统进行的一个测试。

6.1 测试原则

系统测试原则可分为以下几点：

- (1) 对每一个功能模块都要制定严格的测试计划。
- (2) 从开发需求出发，对每一个功能需求都要分解出独立的测试项，这些测试项应包括具体的功能性测试和非功能性测试。
- (3) 在项目开发过程中最早且不断的对各个功能进行测试。
- (4) 测试用例的设计应该考虑到各种各样的因素。
- (5) 对每一个功能的测试结果都应做出详细的检查，并对测试出的错误结果进行验证。

6.2 功能测试

功能测试主要是针对功能方面的八模块（分类管理模块、题目管理模块、文件管理模块、高频错题模块、分类练习模块、随机练习模块、我的错题模块以及我的试卷模块）进行功能项的测试，并且列出测试用例表。

6.2.1 分类管理模块测试

分类管理模块测试的主要功能特性是对分类管理菜单下的两个页面进行功能测试，目的是查看是否可以按照设置前置分类知识点，是否添加分类名称，分类名称为空是否有提示，提交保存是否跳转到分类列表，分类列表的信息编辑以及删除是否实现，列表的分页功能是否实现。

测试结果：各项功能均能正确运行操作，无出错报错，可以设置前置分类知识点、分类的添加、查询、分页、编辑和删除等功能均可正常实现。测试用例表：如表 6-1 所示。

表 6-1 分类管理功能测试用例表

数据	操作描述	预期效果	实际效果
1 计算机组成原理	在分类名称一栏输入，提交保存	保存数据，跳转到分类列表	与预期相同

续表 6-1 分类管理功能测试用例表

数据	操作描述	预期效果	实际效果
/	不添加分类，提交保存	系统提示，分类不可为空	与预期相同
1 计算机组成原理	设置前置分类知识点	前置分类知识点选中	与预期相同
2 操作系统	点击编辑修改分类名称，提交保存	分类名称修改成功	与预期相同
3 计算机网络原理	点击删除，删除分类记录，是否确定删除，点击确定	删除对应的记录	与预期相同
4 软件工程	添加分类五条记录以上	列表下方出现分页	与预期相同
5 编译原理	添加分类五条记录以上	列表下方出现第二页以及更多页	与预期相同

6.2.2 题目管理模块测试

题目模块测试主要功能特性是针对每个题型的录入题目，题目为空是否系统提示，录入答案选项，录入正确答案，设置所属分类，选择难度系数，提交保存是否跳转到对应题目列表，对每个题型信息进行编辑以及删除，设置题型列表的分页等功能是否实现。

测试结果：各项功能均能正确运行操作，无出错报错，可以录入题目，录入答案选项，设置正确答案，设置所属分类，设置难度系数，每个题型的添加、查询、分页、编辑和删除等功能均可正常实现。测试用例表：如表 6-2 所示。

表 6-2 题目管理功能测试用例表

数据	操作描述	预期效果	实际效果
有关运算器的描述 正确的选项	在录入题目中输入题目标题	保存标题，可以进行提交保存	与预期相同
/	不添加录入题目标题，提交保存	系统提示，题目不可为空	与预期相同
并行、冯·诺依曼、智能、串行	添加答案选项，提交保存	题目答案添加完成	与预期相同
智能	设置正确答案，提交保存	题目正确答案添加完成	与预期相同

续表 6-2 题目管理功能测试用例表

数据	操作描述	预期效果	实际效果
1 计算机组成原理	设置所属题目， 提交保存	所属题目添加成功	与预期相同
有关运算器的描述 正确的选项	点击编辑，编辑 题目记录，提交保存	修改对应的记录	与预期相同
有关运算器的描述 正确的选项	点击删除，删除 题目记录，是否确定 删除，确定	删除对应的记录	与预期相同
冯·诺依曼主要三个 思想是什么	添加题目五条记 录以上	列表下方出现第 二页以及更多页	与预期相同

6.2.3 文件管理模块测试

文件管理模块测试的主要功能特性是对文件管理菜单下的页面进行功能测试，目的是查看是否设置题目类型，上传文件格式是否正确，提交保存是否解析为对应的题型题目等。

测试结果：各项功能均能正确运行操作，无出错报错，可以设置前置分类知识点、可上传文件等功能均可正常实现。测试用例表：如表 6-3 所示。

表 6-3 文件管理功能测试用例表

数据	操作描述	预期效果	实际效果
单选题	选择题目类型	可以选中题目类型	与预期相同
	不添加文件，提 交保存	系统提示，文件 格式错误	与预期相同
Choice.csv	点击上传文件， 提交保存	文件上传成功	与预期相同
计算机内部信息包 括哪两大信息	文件上传保存成 功	上传文件内容信 息解析成对应的题目	与预期相同

6.2.4 试题练习模块测试

(1) 高频错题模块测试的主要功能特性是对高频错题菜单下的页面进行功能测试，目的是查看是否将题目分为易错率较高、易错率特高以及易错率极高三个不同级别的难度，是否进入页面之后可以切换单选题、多选题、判断题和论述题，是否可以记录当前用户作答题目的正确个数以及错误个数，是否显示当前题目和总题目的比，单选题、多选题和判

断题，是否在用户作答之后显示用户作答答案以及正确答案，论述题中是否在用户作答之后点击参考答案生效等功能是否实现。

测试结果：各项功能均能正确运行操作，无出错报错，可以实现不同频率错题的选择，可以切换单选题、多选题、判断题和论述题，可以实时判断当前正确以及错误题数，显示当前题目的题数和总题数的比，用户作答完可以显示正确答案以及用户作答答案等功能均可正常实现。测试用例表：如表 6-4 所示。

表 6-4 高频错题功能测试用例表

数据	操作描述	预期效果	实际效果
易错率较高	点击易错率较高 跳转题目页面	显示难度系数 1-3 的题目	与预期相同
易错率特高	点击易错率特高 跳转对应页面	显示难度系数 3-4 的题目	与预期相同
易错率极高	点击易错率极高 跳转对应页面	显示难度系数 4-5 的题目	与预期相同
单选题	点击单选题，切 换单选题题型	当前题目为单选 题题目	与预期相同
多选题	点击多选题，切 换多选题题型	当前题目为多选 题题目	与预期相同
判断题	点击判断题，切 换判断题题型	当前题目为判断 题题目	与预期相同
论述题	点击论述题，切 换论述题题型	当前题目为论述 题题目	与预期相同
并行	点击当前题目选 项进行作答	显示用户作答答 案以及正确答案	与预期相同
冯·诺依曼	当前题目作答与 正确答案一致	正确个数累加 1	与预期相同
并行	当前题目作答与 正确答案不一致	错误个数累加 1	与预期相同
	用户左右滑动作 答	显示用户当前的 题数与总题数比	与预期相同

(2) 分类练习模块测试的主要功能特性是对类型选择菜单下的页面进行功能测试，目的是查看是否将题目根据题目类型进行区分，是否进入页面之后可以切换单选题、多选题、判断题和论述题，是否可以记录当前用户作答题目的正确个数以及错误个数，是否显示当

前题目和总题目的比，单选题、多选题和判断题中是否在用户作答之后显示用户作答答案以及正确答案，论述题是否在用户作答之后点击参考答案生效等功能是否实现。

测试结果：各项功能均能正确运行操作，无出错报错，可以实现不同题目分类类型的选择，实时判断当前正确以及错误题数，显示当前题目的题数和总题数的比，用户作答完可以显示正确答案以及用户作答答案等功能均可正常实现。测试用例表：如表 6-5 所示。

表 6-5 分类练习功能测试用例表

数据	操作描述	预期效果	实际效果
1 计算机组成原理	点击计算机组成原理进入对应的题目页面	显示计算机组成原理的题目	与预期相同
2 操作系统	点击操作系统跳转对应页面	显示操作系统的题目	与预期相同
3 计算机网络原理	点击计算机网络原理跳转对应页面	显示计算机网络原理的题目	与预期相同
单选题	点击单选题，切换单选题题型	当前题目为单选题题目	与预期相同
多选题	点击多选题，切换多选题题型	当前题目为多选题题目	与预期相同
判断题	点击判断题，切换判断题题型	当前题目为判断题题目	与预期相同
论述题	点击论述题，切换论述题题型	当前题目为论述题题目	与预期相同
并行	点击当前题目选项进行作答	显示用户作答答案以及正确答案	与预期相同
冯·诺依曼	当前题目作答与正确答案一致	正确个数累加 1	与预期相同
并行	当前题目作答与正确答案不一致	错误个数累加 1	与预期相同
	用户滑动作答	显示题数比	与预期相同

(3) 随机练习模块测试的主要功能特性是对不同题目类型进行随机的功能测试，目的是进入页面之后可以切换单选题、多选题、判断题和论述题进行随机练习，是否实现记录当前用户作答题目的正确个数以及错误个数，是否显示当前题目和总题目的比，单选题、多选题和判断题是否在用户作答之后显示用户作答答案以及正确答案，论述题是否在用户作答之后点击参考答案生效等功能是否实现。

测试结果：各项功能均能正确运行操作，无出错报错，可以实现不同题目随机练习，可以切换单选题、多选题、判断题和论述题，可以实时判断当前正确以及错误题数，显示当前题目的题数和总题数的比，用户作答完可以显示正确答案以及用户作答答案等功能均可正常实现。测试用例表：如表 6-6 所示。

表 6-6 随机练习功能测试用例表

数据	操作描述	预期效果	实际效果
单选题	点击单选题，切换单选题题型	当前题目为单选题题目	与预期相同
多选题	点击多选题，切换多选题题型	当前题目为多选题题目	与预期相同
判断题	点击判断题，切换判断题题型	当前题目为判断题题目	与预期相同
论述题	点击论述题，切换论述题题型	当前题目为论述题题目	与预期相同
并行	点击当前题目选项进行作答	显示用户作答答案以及正确答案	与预期相同
冯·诺依曼	当前题目作答与正确答案一致	正确个数累加 1	与预期相同
并行	当前题目作答与正确答案不一致	错误个数累加 1	与预期相同
	用户左右滑动作答	显示用户当前题数与总题数比	与预期相同

（4）我的错题模块测试的主要功能特性是对类型选择菜单下我的错题的页面进行功能测试，目的是查看是否将分类练习以及随机练习的错题题目根据题目类型进行区分，是否进入页面之后可以切换单选题、多选题、判断题和论述题，是否可以记录当前用户作答题目的正确个数以及错误个数，是否显示当前题目和总题目的比，单选题、多选题和判断题是否在用户作答之后显示用户作答答案以及正确答案，论述题是否在用户作答之后点击参考答案生效等功能是否实现。

测试结果：各项功能均能正确运行操作，无出错报错，可以实现不同题目分类类型的错题题目进行选择，可以切换单选题、多选题、判断题和论述题，可以实时判断当前正确以及错误题数，显示当前题目的题数和总题数的比，用户作答完可以显示正确答案以及用户作答答案等功能均可正常实现。测试用例表：如表 6-7 所示。

表 6-7 我的错题功能测试用例表

数据	操作描述	预期效果	实际效果
1 计算机组成原理	点击计算机组成原理进入对应的题目页面	显示计算机组成原理的错题题目	与预期相同
单选题	点击单选题，切换单选题题型	当前题目为单选题错题题目	与预期相同
多选题	点击多选题，切换多选题题型	当前题目为多选题错题题目	与预期相同
判断题	点击判断题，切换判断题题型	当前题目为判断题错题题目	与预期相同
论述题	点击论述题，切换论述题题型	当前题目为论述题错题题目	与预期相同
并行	点击当前题目选项进行作答	显示用户作答答案以及正确答案	与预期相同
冯·诺依曼	当前题目作答与正确答案一致	正确个数累加 1	与预期相同
并行	当前题目作答与正确答案不一致	错误个数累加 1	与预期相同
	用户左右滑动作答	显示用户当前题数与总题数比	与预期相同

6.2.5 试卷练习模块测试

我的试卷模块测试的主要功能特性是对类型选择菜单下我的试卷的页面进行功能测试，进入页面之后按照单选题、多选题、判断题以及论述题的顺序进行作答，是否显示当前题目和总题目的比，是否可以随时提交试卷，查看错题，提交试卷之后是否实现用户作答正确个数等功能。

测试结果：各项功能均能正确运行操作，无出错报错，可以实现不同题目分类类型作答，显示当前题目的题数和总题数的比，可以提交试卷之后查看错题，提交试卷之后显示用户作答正确的个数。测试用例表：如表 6-8 所示。

表 6-8 我的试卷功能测试用例表

数据	操作描述	预期效果	实际效果
1 计算机组成原理	点击计算机组成原理进入试卷页面	显示计算机组成原理的试卷题目	与预期相同

续表 6-8 我的试卷功能测试用例表

数据	操作描述	预期效果	实际效果
并行	点击当前题目选项进行作答	显示用户作答答案	与预期相同
	用户左右滑动作答	显示用户当前题数与总题数比	与预期相同
	点击提交	显示当前用户作答每个题型的正确个数	与预期相同
	点击查看错题	显示当前用户作答错误的题目以及答案	与预期相同

6.3 性能测试

性能测试的主要功能特性是对整个系统的各个运行时间和细小的功能进行测试，以验证该系统的性能是否满足大学生在备考中的使用和找出不足。以下是对备考系统的测试细节进行说明。

(1) 测试指标：系统启动时间，各项界面运行时间，小功能细节能否实现。

(2) 测试工具：Visual Studio Code。

(3) 测试环境：Node.js。

(4) 测试结果：系统启动时间偏慢，各项界面运行响应时间很快，小功能细节部分实现，仍可改进。测试用例表：如表 6-9 所示。

表 6-9 性能测试用例表

数据	操作描述	预期效果	实际效果
系统后台启动时间	点击执行后台	记录从点击启动到出现界面的时间	358ms
系统前台启动时间	点击执行前台	记录从点击启动到出现界面的时间	773ms
用户个人资料：修改昵称密码	点击昵称和密码	昵称和密码可修改，功能可设定	功能均可成功，但功能设定可选选项偏少
页面执行操作响应时间	点击执行	记录界面执行操作的响应时间	网络正常的情况下秒开，但服务器启动仍偏慢

6.4 本章总结

通过功能测试和性能测试两方面对备考系统进行的测试，在功能方面该系统已实现的功能都可正常正确运行；在性能测试方面该系统启动偏慢，页面响应时间速度可以，但小功能小细节没有落实，可以进行相应的改进。总结来说该系统可以满足大学生计算机专业在备考中的使用，还可以继续优化改进。

第 7 章 总结和展望

本章主要论述了对前后端分离备考系统开发过程的总结，并对备考系统在未来的发展方向做了简要描述。

7.1 总结

为了满足大学生计算机专业对备考系统的需求，在做了需求分析之后开发了前后端分离备考系统。本系统采用 B/S 结构开发，采用 Visual Studio Code 作为开发工具，使用 Vue 作为前端框架，用 Node.js 语言进行开发，用 MongoDB 作为后台数据支持，实现了系统管理员对题目的管理以及用户使用备考类别等功能。在独立完成本次项目的过程中，首先在文件上传过程中遇到格式解析转化的问题，通过搜索资料学习进行完善，其次，对于备考系统的设计与实现过程中，对系统的性能有待完善，在之后的学习中要对代码优化进行相应的学习，在备考系统的实现过程中得到提高。

前后端分离备考系统的分类管理、题目管理以及文件上传，方便了系统管理员对题目的录入以及各种功能的实现。高频错题、分类练习、随机练习、我的错题以及我的试卷方便了用户对题目的作答以及练习。在需求分析阶段主要对各个模块的主要功能进行了分析，通过用例图描述各个模块的主要功能；系统设计部分对每个功能模块进行详细设计并附每个模块的业务流程图和时序图，为具体实现阶段做了充足的准备，基本实现了前后端分离备考系统的主要功能。

7.2 展望

本次开发的基于 Vue 的前后端分离的备考系统的设计与实现已经告一段落，从前期的调研到系统的需求以及开发，这个过程并不是很顺利。虽然前后端分离备考系统的功能差不多可以实现，满足系统管理员以及用户对备考系统的使用，但是由于时间关系和能力有限，系统还存在一些问题，比如用户使用的头像上传，确保用户信息的完整性，添加视频类备考模式，管理员角色的权限太少以及管理员可以对更多类型的题目进行管理，这些问题都会给普通用户和管理员带来不便，希望在将来的学习中，可以将前后端分离备考系统进一步完善。

参考文献

- [1] 伍德雁. BS 架构在船舶设备管理系统开发中的应用[J]. 舰船科学技术, 2017 (12): 98-101.
- [2] 彭逸帆. 多用户管理系统的 web 前端框架研究[D]. 北京邮电大学, 2018: 17-18.
- [3] Kefa Zhou, Yinyi Cheng. Performance Analysis of PostgreSQL and MongoDB Databases for Unstructured Data[J]. Information Technology, 2019(8):11-13.
- [4] 李纪伟, 段中帅. 非结构化数据库 MongoDB 的数据存储[J]. 2018 (9): 7-9.
- [5] 麦冬, 陈涛. 轻量级响应式框架 Vue.js 应用分析[J]. 信息与电脑 (理论版), 2017 (7): 58-59.
- [6] 张蕙烽, 刘明杰. Vue.js 与 Lumen 组合框架的大创项目管理系统架构方案[D]. 2019: 34-35.
- [7] 钱立. WebSocket 与 Node.js 结合实现公交系统实时位置查询[J]. 2019 (4): 140-141.
- [8] Kyounggon Kim. Automatic detection and analysis of prototype pollution vulnerability in Node.js modules[J]. International Journal of Information Security, 2021(3):1-23.
- [9] 程桂花. Node.js 中 Express 框架路由机制的研究[J]. 浙江理工大学, 2018 (5): 198-201.
- [10] 余久久. 软件工程简明教程[M]. 北京: 清华大学出版社, 2015: 5-6.
- [11] 张贺, 魏强. 基于模型的卫星系统功能架构设计综述[D]. 科技创新导报, 2019: 15-18.
- [12] 马华, 张西学. 基于高校网络考试系统的数据库模型设计[J]. 中国教育信息化, 2019 (7): 90-93.
- [13] 梁利亭. 计算机软件数据库设计原则探讨[J]. 三门峡职业技术学院, 2020 (2): 116-117.
- [14] 王兵. 数据库应用系统逻辑结构设计初探[J]. 现代计算机 (专业版) 2019 (5): 14-15.
- [15] Kalugina I A. Automated Software Testing Technologies for Realistic Computer Graphics[J]. Programming and Computer Software, 2021(1):76-87.

致 谢

往事匆匆，一晃四年，如梦似幻，回首往事，历历在目，但人生路上有太多的离别，我们不得不说一声再见，然后各自珍重，珍惜当下。一段又一段的旅途，不断的收获成长与感悟，四年不仅是求学之路，更是成长之路上，一路走来更多的是感恩。

本学士学位论文的工作是在王建老师的悉心指导下完成的，王建老师严谨的治学态度和科学的工作方法给了我极大的帮助和影响。在此衷心感谢四年来王建老师对我的关心和指导。

在撰写学士学位论文期间，陈银娣同学对我学士学位论文中的备考研究工作给予了热情帮助，在此向他们表达我的感激之情。

另外对我的家人表示感谢，他们的理解和支持使我能够在学校专心完成我的学业。