



# 广东工业大学

## 专项设计报告

### 基于 Node.js 全栈的考研在线刷题与错题管理系统的 设计与实现

学      院 \_\_\_\_\_ 国际教育学院 \_\_\_\_\_  
专      业 \_\_\_\_\_ 计算机科学与技术 \_\_\_\_\_  
年级班级 \_\_\_\_\_ 22 (2) \_\_\_\_\_  
学      号 \_\_\_\_\_ 3122009962 \_\_\_\_\_  
学生姓名 \_\_\_\_\_ 吴佳翮 \_\_\_\_\_  
指导教师 \_\_\_\_\_ 张灵 \_\_\_\_\_

2025 年 12 月

## 摘要

随着互联网技术与在线教育的持续发展，基于 Web 的考研刷题平台逐渐成为考生备考过程中重要的学习辅助工具，其通过网站形式贯穿考研复习的各个阶段，为平台与考生之间提供了高效的信息交互与学习支持环境。考研刷题平台依托互联网技术与网页系统，实现了题库资源与考生学习需求的精准匹配，借助系统的数据处理与反馈机制提升学习服务质量，进而促进考生备考效率的提升；对平台运营与管理方而言，基于网站的管理系统可集中维护题库资源、用户信息及学习数据，突破传统人工管理方式的局限，提高平台运行与维护效率。

当前，部分传统考研学习方式仍以纸质资料或零散线上资源为主，存在题目资源分散、学习过程缺乏系统管理、学习效果难以量化等问题，容易导致备考效率低下、学习计划执行困难。本文以考研刷题平台的管理与服务功能为研究核心，分析了平台中考生用户与管理员角色的功能需求，在此基础上研究了刷题平台的整体功能架构与运行环境，重点设计并实现了平台的管理员端系统。该系统基于 Web 技术构建，紧密结合考研刷题业务流程，可实现题库信息管理、用户信息管理、刷题数据统计与系统维护等功能，不仅有助于提升平台的管理规范性与运行效率，还能为考生提供稳定、可靠的学习服务支持。

**关键词：**在线教育，用户管理，Web 学习平台，系统设计

## **Abstract**

With the continuous development of Internet technology and online education, web-based postgraduate entrance examination practice platforms have gradually become important learning support tools in the process of exam preparation. Such platforms penetrate various stages of postgraduate exam review and provide an efficient online environment for interaction and learning services between the platform and examinees. By relying on Internet and web technologies, the practice platform realizes the effective integration of question bank resources and learners' study needs. Through data processing and feedback mechanisms, it improves learning service quality and promotes learning efficiency. For platform administrators, the web-based management system enables centralized maintenance of question banks, user information, and learning data, effectively improving platform operation and management efficiency.

At present, some traditional postgraduate exam preparation methods still rely on scattered learning resources and manual management, which leads to problems such as fragmented question resources, lack of systematic learning management, and difficulty in evaluating learning outcomes. These issues easily result in low preparation efficiency and increased learning costs. Focusing on the management and service functions of a postgraduate entrance examination practice platform, this paper analyzes the functional requirements of different user roles, including examinees and administrators. Based on this analysis, the overall functional architecture and operating environment of the platform are studied, with an emphasis on the design and implementation of the administrator-side management system. Built on a web-based architecture and aligned with the exam preparation process, the system supports question bank management, user information management, practice data statistics, and system maintenance. It not only enhances the standardization and efficiency of platform management but also provides stable and reliable learning support services for examinees.

**Keywords:** Online Education, User Management, Web-based Learning Platform, System Design

# 目录

|                     |           |
|---------------------|-----------|
| <b>1 绪论 .....</b>   | <b>1</b>  |
| 1.1 立项背景 .....      | 1         |
| 1.2 立项原因概述 .....    | 1         |
| 1.3 UML 概述 .....    | 错误！未定义书签。 |
| <b>2 项目背景 .....</b> | <b>3</b>  |
| <b>3 需求分析 .....</b> | <b>3</b>  |
| 3.1 功能需求 .....      | 错误！未定义书签。 |
| 3.2 性能需求 .....      | 7         |
| 3.3 项目开发要求 .....    | 错误！未定义书签。 |
| 3.4 系统安全性需求 .....   | 错误！未定义书签。 |
| 3.5 系统备份与恢复需求 ..... | 错误！未定义书签。 |
| 3.6 其他需求 .....      | 错误！未定义书签。 |
| 3.7 用例建模 .....      | 错误！未定义书签。 |
| 3.7.1 用户列表 .....    | 错误！未定义书签。 |
| 3.7.2 用户自动分类 .....  | 错误！未定义书签。 |
| 3.7.3 会员管理 .....    | 错误！未定义书签。 |
| 3.7.4 目标用户 .....    | 错误！未定义书签。 |
| <b>4 概要设计 .....</b> | <b>7</b>  |
| 4.1 系统总体概述 .....    | 7         |
| 4.2 功能模块概述设计 .....  | 8         |
| 4.3 面向对象设计方法 .....  | 10        |

|                       |           |
|-----------------------|-----------|
| 4.4 用户管理子系统结构设计 ..... | 10        |
| 4.5 动态行为建模 .....      | 11        |
| 4.5.1 查看用户列表 .....    | 11        |
| 4.5.2 用户自动分类 .....    | 12        |
| 4.5.3 会员管理 .....      | 13        |
| 4.6 交互建模 .....        | 14        |
| 4.6.1 用户列表 .....      | 14        |
| 4.6.2 用户自动分组 .....    | 14        |
| 4.6.3 删除会员资格 .....    | 14        |
| 4.6.4 审核会员 .....      | 15        |
| <b>结论及展望 .....</b>    | <b>16</b> |
| <b>参 考 文 献 .....</b>  | <b>19</b> |

# 1 緒論

## 1.1 立項背景

隨着互聯網技術與在線教育的深度發展，基於 Web 的考研刷題平臺已成為研究生備考過程中的重要學習載體，其覆蓋了學習資源獲取、在線刷題、學習反饋、數據統計與學習管理等多個環節，參與者主要包括考生與平臺管理員等角色，核心目標是通過信息化與數字化手段提升備考學習的系統性、效率性與科學性。考研刷題平臺作為在線教育平臺的重要應用形式，為考生提供了集中化、結構化的學習支持環境：對考生而言，平臺能夠統一展示題庫資源、試題解析與練習記錄，幫助其進行有針對性的刷題訓練與學習鞏固；對平臺管理方而言，系統可對題庫內容、用戶信息及學習數據進行集中管理，為平臺的穩定運行與持續優化提供支撐。

現階段，部分傳統考研備考方式仍以紙質資料或零散網絡資源為主，學習過程缺乏統一的管理與數據支撐。考生在學習過程中往往需要自行整理題目與學習進度，存在學習資源分散、刷題情況難以統計、學習效果難以評估等問題；同時，平臺在缺乏系統化管理工具的情況下，題庫維護、用戶管理與學習數據分析多依賴人工操作，效率較低，且容易出現信息更新不及时、管理不規範等現象。對於考生而言，這種模式容易導致學習計劃執行不力、復習方向不清晰；對於平臺管理方而言，也不利於對用戶學習行為的分析與平臺服務質量的提升，從而在一定程度上制約了刷題平臺的整體服務能力與教學效果。

## 1.2 立項原因概述

在現實生活中，電商賣家需要定期地向各類商品供應商進行採購以獲取豐富的商品。在此過程中，電商賣家不可避免地會遇到以下問題，以下是我列出的問題點：

1. 不同科目、不同章節的考研試題分散在各類教材、輔導資料及網絡資源

中，题目来源零散，考生往往需要在多个平台或资料之间切换，才能完成一次较为全面的刷题训练。

2. 考生需要自行寻找合适的刷题资源，由于信息获取渠道有限，部分考生只能接触到少量题库资源，难以对同类题型进行对比练习，也不利于选择更符合自身复习阶段的题目。
3. 在传统刷题方式下，考生完成练习后往往只能简单核对答案，缺乏系统化的错题整理与学习反馈机制，导致问题难以及时总结，影响学习效率。
4. 同时，考生与教学资源提供方之间的沟通渠道不够顺畅。试题更新、重点变化及学习建议多依赖人工通知或零散信息发布，存在信息传递滞后、覆盖不全面等问题。这种方式不仅效率较低，也难以及时满足考生的学习需求。
5. 为了提升考研刷题过程的系统性与效率，有必要构建一个基于 Web 的考研刷题平台，通过统一的平台对题库资源、学习过程及用户信息进行集中管理，从而为考生提供更加便捷、高效的刷题与学习支持服务。基于上述问题，平台的建设具有以下机会点和优势：
6. 平台可作为考生进行刷题与学习管理的统一入口，集中整合不同科目与题型资源，显著简化考生的刷题流程。
7. 平台能够对同类题目进行分类与对比展示，帮助考生更有针对性地选择适合自身水平与复习阶段的练习内容。
8. 平台可提供完善的错题管理与学习反馈功能，帮助考生及时发现薄弱环节，并进行针对性强化训练。
9. 平台支持对刷题记录与学习数据的统计与分析，帮助考生直观了解自身学习进度与效果。
10. 平台可通过管理员端对题库内容进行统一维护与更新，确保试题质量与内容时效性。
11. 平台为管理员提供便捷的题库发布与学习内容管理功能，便于平台持续完善教学资源与服务能力。
12. 平台的建设有助于形成规范化、可扩展的在线学习管理模式，为后续功能扩展与平台运营奠定基础。

## 2 项目背景

### 2.1 总体技术架构

本项目采用基于 Node.js 的全栈 Web 技术架构，前后端通过 HTTP 接口进行数据交互，整体遵循前后端分离的设计思想。系统主要由考生端 Web 前台、后台管理网页端以及服务器端三部分组成。服务器端基于 Node.js 运行环境，使用 Express 框架构建 Web 服务接口，负责业务逻辑处理、数据交互以及系统安全控制。数据库用于存储用户信息、题库数据、刷题记录等核心业务数据，从而支撑整个平台的稳定运行。

### 2.2 服务器端技术实现

服务器端基于 Node.js 平台进行开发，利用其事件驱动和非阻塞 I/O 的特性，提高系统在高并发访问场景下的响应能力。Express 作为服务器端的核心框架，用于构建 RESTful 风格的接口，实现前后端之间的数据通信。服务器端主要负责用户登录与权限校验、题库数据的增删改查、刷题记录统计、错题分析以及后台管理功能的统一调度。

在业务逻辑层中，通过模块化方式对不同功能进行拆分，例如用户模块、题库模块、刷题模块和统计分析模块，使系统结构清晰、易于维护与扩展。同时，服务器端会对请求参数进行校验，并对异常情况进行统一处理，提升系统的健壮性和安全性。

### 2.3 数据库与数据管理技术

系统数据库用于存储考生账户信息、题目及解析内容、刷题记录、错题数据以及系统日志等信息。通过合理设计数据库表结构，实现题库数据与用户刷题行为之

间的关联。刷题数据会按照时间、题型和正确率等维度进行记录，为后续的学习分析和统计功能提供数据支持。

在数据访问层中，服务器端通过封装数据库操作接口，完成数据的查询、插入、更新和删除，避免业务逻辑与底层数据操作的强耦合。同时，通过索引和合理的数据结构设计，提高系统在大量题目和用户访问场景下的查询效率。

## 2.4 考生端技术实现

考生端基于 Web 浏览器运行，采用 HTML、CSS 和 JavaScript 等前端技术进行页面开发。前端通过异步请求方式调用服务器端接口，获取题目数据、刷题结果以及学习统计信息，并将结果动态展示给用户。考生在刷题过程中产生的作答数据会实时提交至服务器端进行保存和分析，从而实现学习进度的持续跟踪。

在交互设计方面，前端注重操作的简洁性与响应速度，通过合理的页面布局和交互反馈，提高考生的使用体验。系统支持题目浏览、在线作答、解析查看、错题回顾和个人信息管理等功能，帮助考生高效完成复习任务。

## 2.5 后台管理端技术实现

后台管理网页端同样基于 Web 技术实现，与考生端共用服务器端接口体系，但在权限控制上进行区分。管理员登录后可通过管理界面对题库内容进行维护，包括题目的新增、修改、删除与审核，以及试题解析和分类信息的管理。同时，管理员可以查看平台用户数据和刷题统计信息，对系统运行情况进行监控和分析。

后台管理端通过统一的权限校验机制，确保只有具备管理员权限的用户才能访问相关功能，从而保障平台数据的安全性和管理操作的规范性。

## 2.6 系统安全与后台维护技术

在系统安全方面，服务器端通过身份认证与权限控制机制对不同角色进行区分，防止非法访问。用户敏感信息在存储和传输过程中采取必要的安全措施，降低数据泄露风险。同时，系统通过日志记录和异常监控，对运行状态进行持续跟踪，便于后期维护与问题排查。

通过采用 Node.js 全栈技术方案，本项目在保证开发效率的同时，也具备良好的扩展性和可维护性，能够满足考研刷题平台在功能扩展和用户规模增长方面的需求。

# 3 需求与项目设计

## 3.1.1 考生端（Web 用户端）功能需求

考生端基于 Web 网站实现，主要为考研考生提供在线刷题与学习管理服务，其功能需求包括以下几个方面。

### （1）首页展示功能

系统需要在首页展示学习公告及相关提示信息，并提供推荐试题及热门题目展示功能，帮助考生快速进入学习状态。

### （2）题库浏览与查询功能

系统应支持按科目、章节及关键字对试题进行查询，方便考生根据自身复习进度和学习需求快速定位目标试题。

### （3）在线刷题功能

系统应支持试题展示、答案提交及自动判题功能，考生在完成作答后可即时获取作答结果，并查看对应的试题解析内容。

### （4）错题与收藏管理功能

系统应具备错题自动记录功能，并支持错题查询与重复练习，同时允许考生对试题进行收藏与取消收藏操作，以便后续重点复习。

#### (5) 学习记录与统计功能

系统需要对刷题记录进行统计与展示，包括刷题数量、正确率及学习进度等信息，帮助考生直观了解自身学习情况。

#### (6) 个人信息管理功能

系统应支持考生进行注册与登录操作，并允许考生对个人账户信息及登录密码进行修改和维护。

---

### 3.1.2 管理员后台管理网页端功能需求

---

管理员后台管理网页端是本系统的重要组成部分，主要用于平台运行维护与数据管理，其功能需求如下。

#### (1) 用户管理功能

系统应支持管理员对考生用户信息进行查询与管理，包括用户状态的启用与禁用，以及用户学习数据的查看。

#### (2) 题库管理功能

系统应支持试题的新增、修改与删除操作，并提供试题分类管理功能，如按科目、章节及难度等级进行管理。

#### (3) 试题审核与维护功能

管理员应能够对试题内容及解析信息进行审核与维护，确保题库内容的准确性与规范性。

#### (4) 学习数据统计功能

系统应支持对刷题数量、正确率及用户活跃度等数据进行统计分析，为平台优化和教学改进提供数据支持。

#### (5) 系统管理功能

系统应支持管理员账号管理及权限分配功能，并提供系统公告发布功能，以保障平台运行的安全性与规范性。

### **3. 2 性能需求**

#### **1. 基本性能**

在100M局域网环境下查询操作的响应时间要求：3秒以内。

系统支持同时在线数大于200，并发用户数大于50。

#### **2. 数据查询**

查询单个数据主题(百万级)的响应时间不大于3秒。

目录检索的响应时间不大于 2秒。

查询统计报表(非实时统计)的响应时间不大于3秒。

#### **3. 精度要求**

软件的输入精度:小数点后保留 3 位有效数字

输出数据精度的要求:小数点后保留 3 位有效数字

传输过程中的精度:小数点后保留 3 位有效数字

#### **4. 时间特征**

按钮，输入输出等响应时间:0.5 秒以内

数据更新时间:0.5 秒钟以内

数据传送时间:1 秒以内

登录，注册等响应时间:1 秒以内

认证响应时间:2 秒以内

## **4 系统总体设计**

### **4.1 系统总体概述**

本系统为基于 Web 的考研刷题平台，采用浏览器访问方式运行，主要由两个功能端组成，分别为考生端和管理员后台管理网页端。系统通过对题库资源、用户信息及学习数据的统一管理，为考生提供在线刷题与学习支持服务，同时为平台管理人员提供高效、规范的后台管理功能。

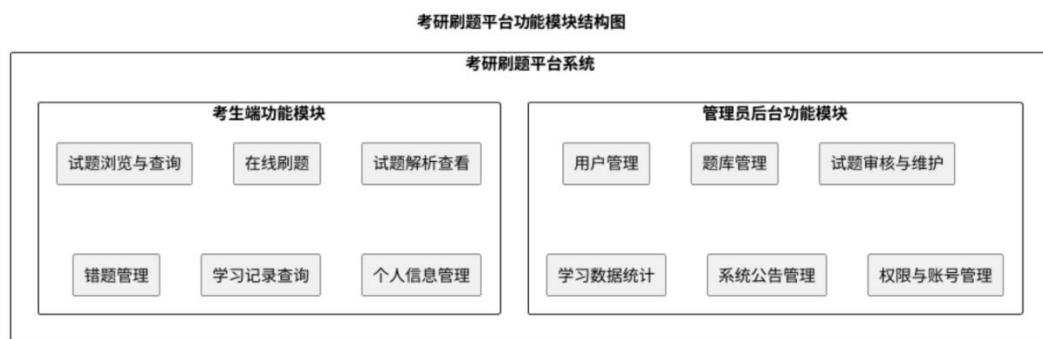
考生端为 Web 用户端，主要面向参加研究生入学考试的考生使用。考生通过浏览器访问系统，可完成在线刷题与学习管理等操作。考生端主要功能包括试题推荐、题库浏览、按科目或关键字查询试题、在线答题、查看试题解析、错题管理、学习记录查询以及个人账户信息管理等。通过系统对刷题数据的统计与分析，考生能够直观了解自身学习进度和复习效果，从而合理调整备考计划。

管理员后台管理网页端主要由平台管理员通过电脑浏览器访问，是系统运行与维护的重要组成部分。管理员通过该后台系统对平台用户、题库资源及学习数据进行集中管理。其主要功能包括用户信息管理、题库管理、试题审核与维护、学习数据统计分析、系统公告发布以及管理员权限配置等。通过后台管理系统，平台能够保障题库内容的准确性、系统运行的稳定性以及平台服务质量的持续提升。

通过考生端与管理员后台管理端的协同运行，系统实现了考研刷题业务流程的线上化与规范化，为考生提供良好的学习体验，同时提高了平台管理的效率与可靠性。

## 4.2 功能模块概述设计

### A. 考研刷题平台功能模块结构图



### B. 考生用户端

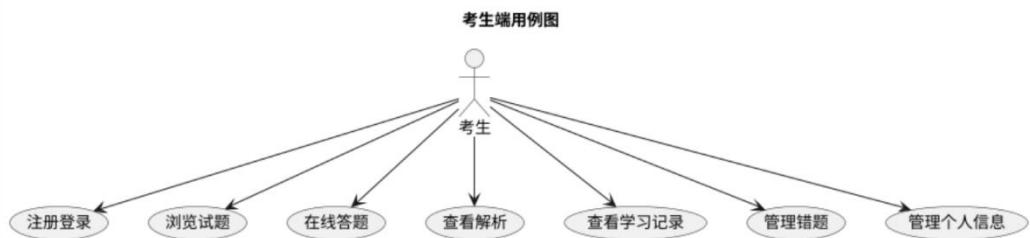
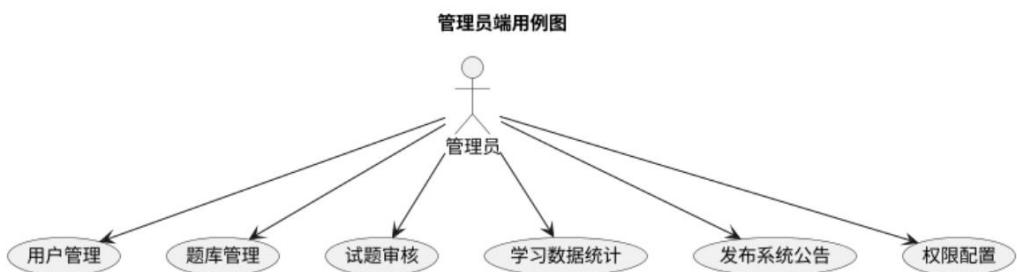
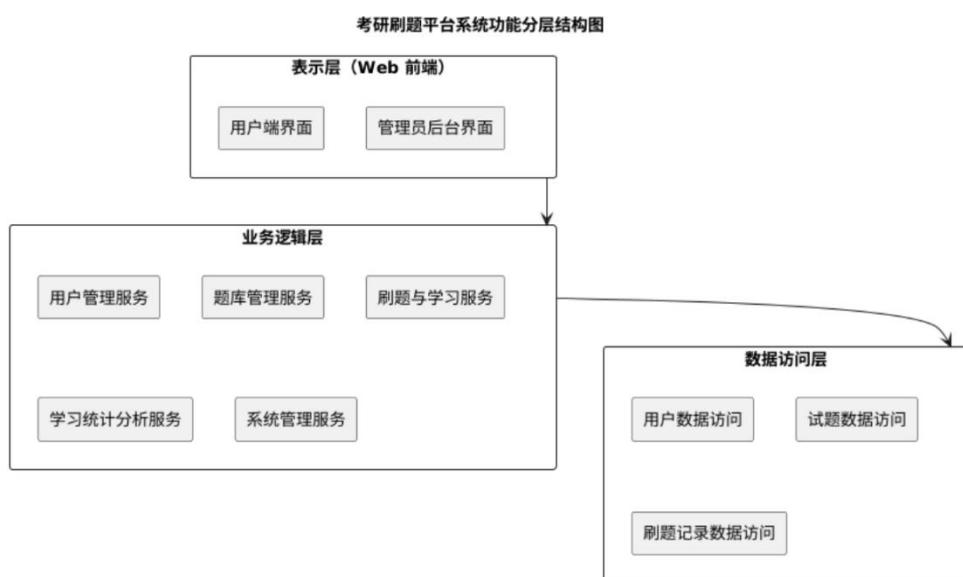


图 4.2 考生用户端



### C. 管理员后台管理网页面端

图 4.3 管理员后台管理网页面端



### D. 分层结构图

图 4.4 分层结构图

### 4.3 面向对象设计方法

在构建业务应用时，建议采用多层次的架构设计方案，以此将系统整体划分为表现层、业务逻辑层以及数据访问层这三个主要部分。这种分层策略的核心目标在于践行“高内聚、低耦合”的软件设计理念，从而提升系统的可维护性与可扩展性。

表现层作为系统与用户交互的直接窗口，负责向用户呈现信息并接收用户的操作指令，它决定了用户在使用系统过程中的直观体验和操作流程。

业务逻辑层处于表现层与数据访问层之间，主要承担对数据的处理和业务规则的实现，它根据表现层传递的需求，调用数据访问层获取相关数据，并按照预设的业务逻辑进行运算、判断和处理，最终将结果返回给表现层。

数据访问层则直接与数据库进行交互，专注于数据的基础操作，包括对数据的插入、删除、修改、查询等，它封装了数据访问的细节，为业务逻辑层提供稳定、高效的数据支持。

### 4.4 用户管理子系统结构设计

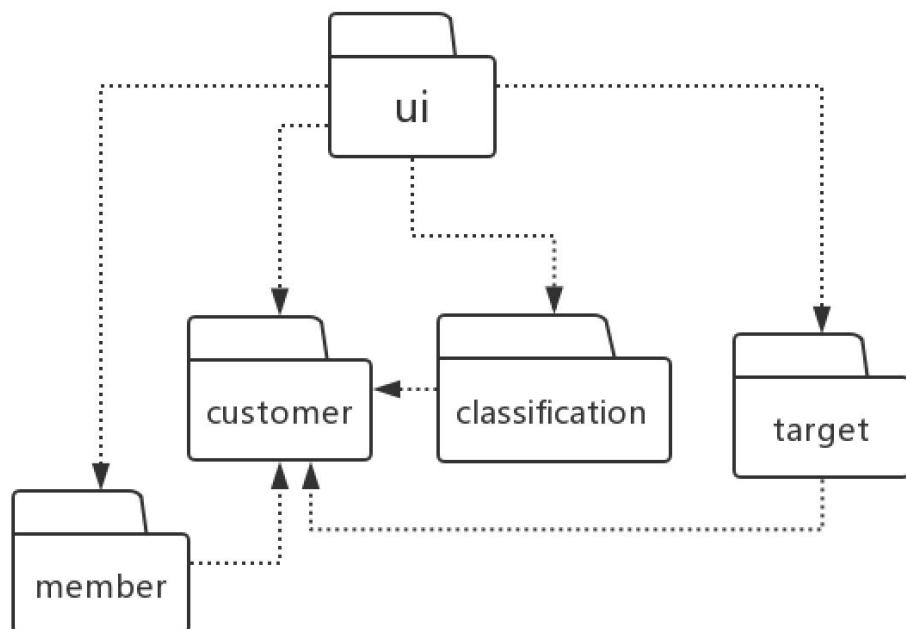


图 4.5 子系统结构

用户管理界面为业务人员提供了功能选择入口，涵盖用户列表查看、用户自动分类、会员管理、目标用户管理等多样化功能模块。

**用户管理功能模块：**主要用于实现用户信息的综合管理。支持查看用户的基础档案信息，可依据用户姓名、注册时间、所属区域等条件进行精准搜索。同时，提供用户删除以及将用户加入黑名单等操作功能。

**用户自动分类功能模块：**致力于实现用户的智能化分类管理。系统会基于用户的活跃度指标对用户进行自动划分。用户可自主选择活跃度范围来筛选并展示相应的用户列表，并且能够便捷地查找特定用户，进而查看其基础信息和历史活跃记录。

**会员管理功能模块：**专注于会员相关的管理工作。提供会员列表的查看功能，同时展示待审核的会员信息。支持通过搜索用户 ID 的方式，对申请升级为会员的用户进行审核操作。此外，还具备删除会员以及查看特定会员详细基本信息的功能。

**目标用户功能模块：**主要服务于目标用户的管理需求。用户可以查看当前所有的目标用户列表，也能通过搜索用户 ID、用户名称、所在区域等信息来精准定位特定用户，并且支持查看目标用户的详细基本信息。

## 4.5 动态行为建模

### 4.5.1 查看用户列表

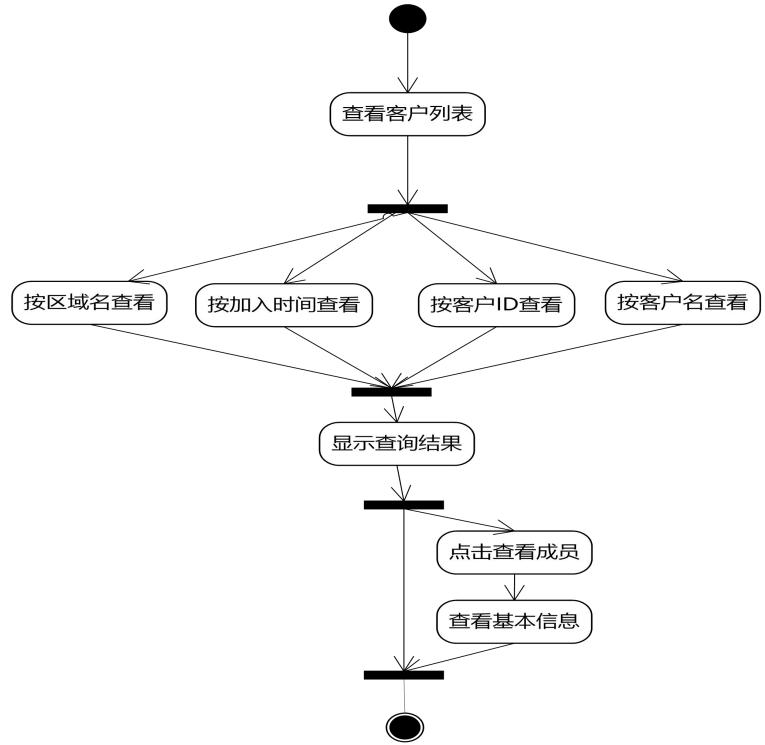


图 4.6 查看用户列表活动图

#### 4.5.2 用户自动分类

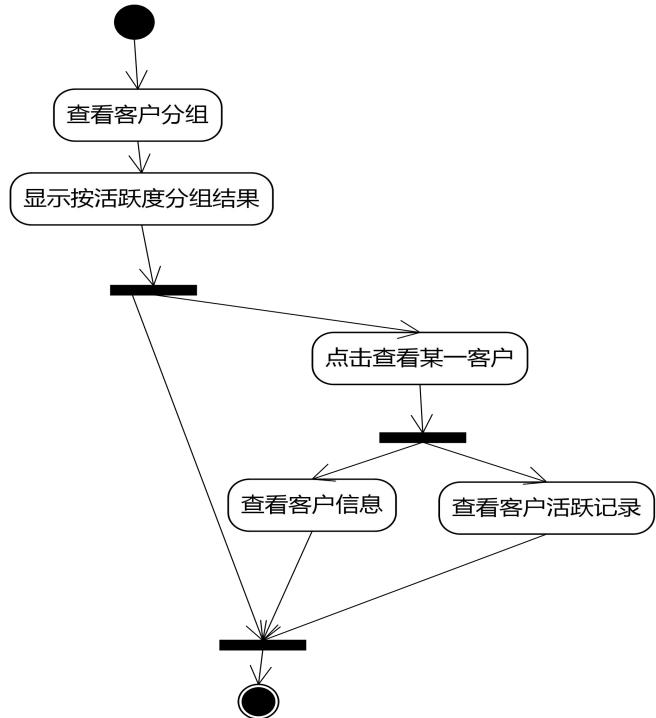


图 4.7 用户自动分类活动图

### 4.5.3 会员管理

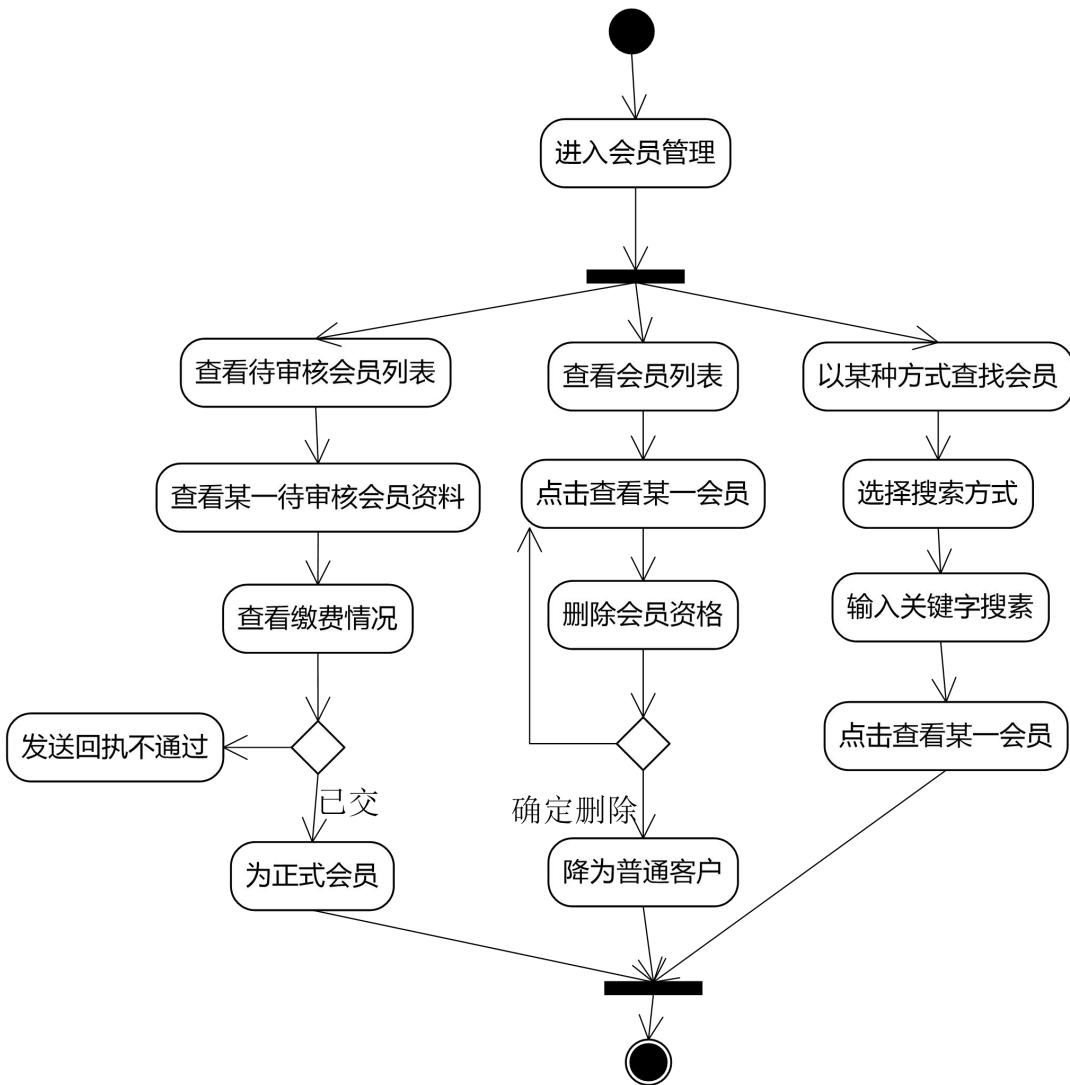


图 4.8 会员管理活动图

## 4.6 交互建模

### 4.6.1 用户列表

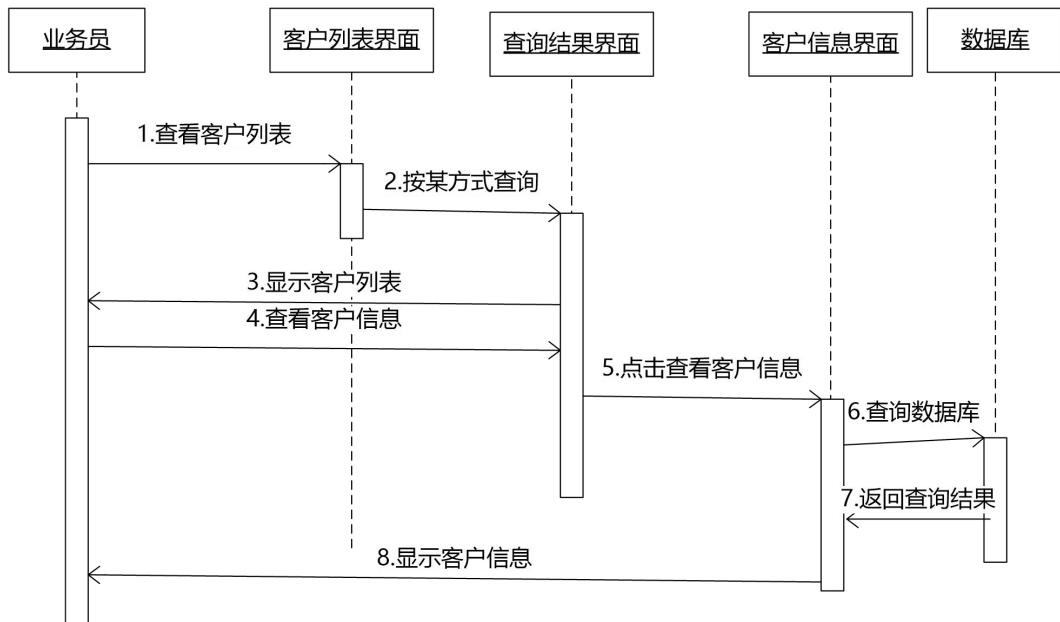


图 4.9 查看用户列表与信息时序图

### 4.6.2 用户自动分组

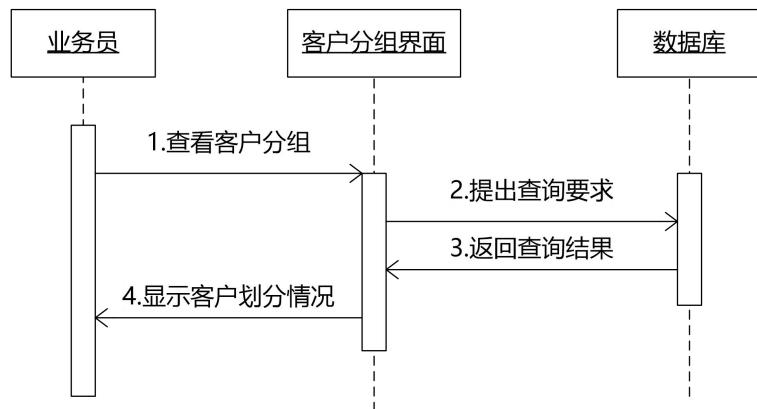


图 4.10 查看用户分组时序图

### 4.6.3 删 除会员资格

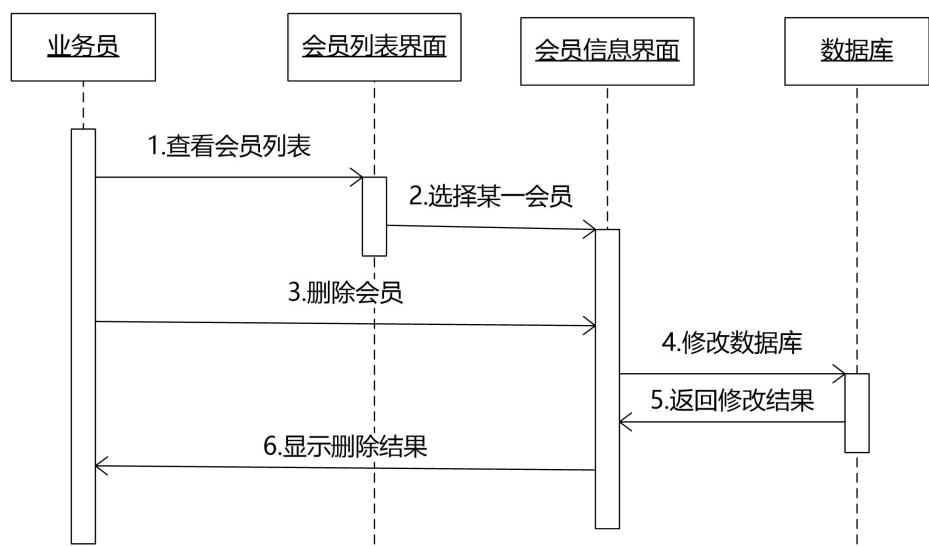


图 4.11 删 除会员资格时序图

#### 4.6.4 审核会员

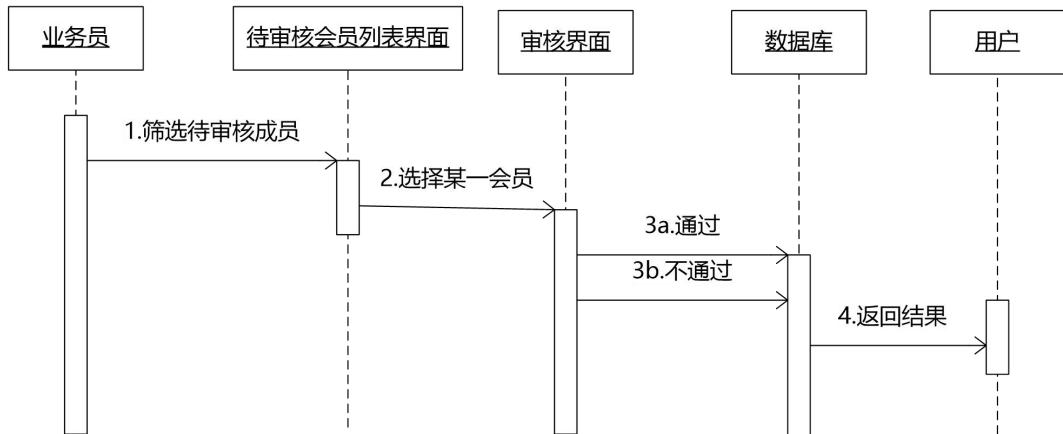


图 4.12 审核会员时序图

## 4.7 系统运行时界面

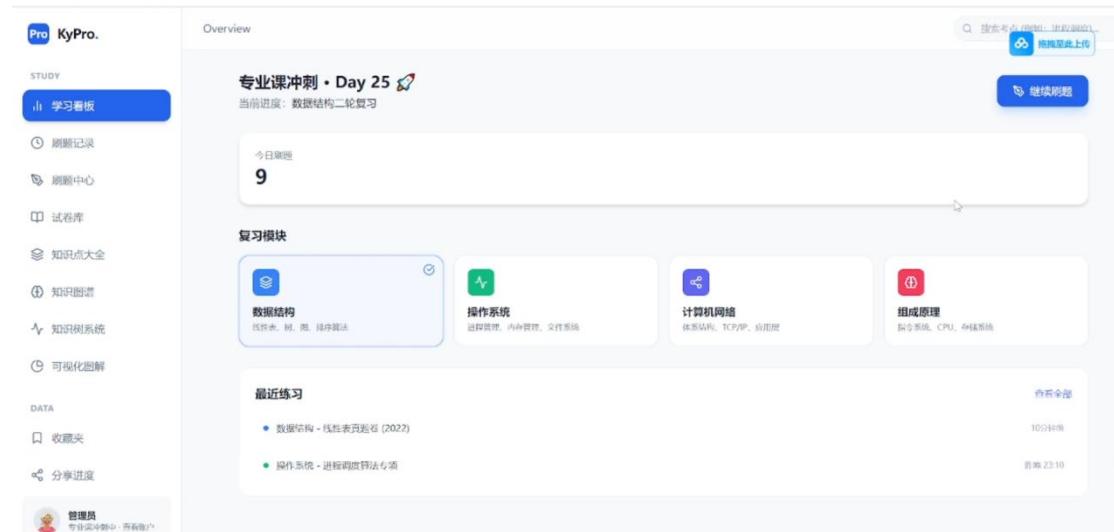


图 4.13 用户登录界面

The screenshot shows the KyPro solving center. At the top is a navigation bar with '刷题中心' (Solving Center), '全部难度' (All Levels), 'easy', 'medium', 'hard', '按标签' (By Tag), '单项选择题' (Single Choice Question), '408真题' (408 Actual Questions), '2009年' (2009), '树的遍历' (Tree Traversal), and '综合应用题' (Comprehensive Application Problems). Below this is a large question card for '2009年全国硕士研究生招生考试408计算机学科专业基础综合 - 单项选择题 - 第1题' (2009 National Graduate Entrance Examination for Computer Science Professional Basis Comprehensive - Single Choice Question - Question 1). The question asks: '为解决计算机主机与打印机之间速度不匹配问题，通常设置一个打印数据缓冲区，主机将要输出的数据依次写入该缓冲区，而打印机则依次从该缓冲区中取出数据。该缓冲区的逻辑结构应该是' (To solve the speed mismatch problem between the computer host and the printer, a printing data buffer is usually set up, where the host writes the output data sequentially into the buffer, and the printer takes data out sequentially from the buffer. What should be the logical structure of this buffer?). The options are: A. 栈 (Stack), B. 队列 (Queue), C. 树 (Tree), and D. 图 (Graph). Below the question are buttons for '#单项选择题' (Single Choice Question), '#408真题' (408 Actual Questions), and '#2009年' (2009). At the bottom of the card are buttons for '熟悉度' (Familiarity) with levels 1 to 5, '练习次数' (Practice Times) with a value of 11, a '+' button, a '查看解析' (View Solution) button, and a '0' knowledge notes button. Below this is another question card for '2009年全国硕士研究生招生考试408计算机学科专业基础综合 - 单项选择题 - 第2题' (2009 National Graduate Entrance Examination for Computer Science Professional Basis Comprehensive - Single Choice Question - Question 2). The question asks: '设栈S和队列Q的初始状态均为空，元素a, b, c, d, e, f, g依次进入栈S。若每个元素出栈后立即进入队列Q，且7个元素出队的顺序是b, d, c, f, e, a, g，则栈S的容量至少是' (Assume stack S and queue Q are initially empty. Elements a, b, c, d, e, f, g enter stack S sequentially. If each element exits the stack and immediately enters queue Q, and the sequence of 7 elements exiting the queue is b, d, c, f, e, a, g, then what is the minimum capacity of stack S?).

图 4.14 用户刷题界面

图 4.15 用户查看题解界面



图 4.16 用户查看思维导图

## 结论及展望

本文概述了一个基于 Web 的考研刷题平台系统的部分实现，重点完成了系统中用户管理与学习管理相关功能的需求分析与概要设计工作。通过对考研刷题平台系统的研究与设计，本文在一定程度上验证了将信息技术应用于在线学习与考试

备考领域的可行性与实用性，具有一定的理论意义和实践价值。通过本次系统设计与实现研究，主要得出以下几点结论：

- (1) 由于本人在系统设计与开发经验方面仍然有限，所设计的考研刷题平台系统在功能与细节上仍存在一定不足。例如，系统目前尚未引入更加智能化的学习推荐机制、学习路径规划等功能，与实际大规模在线学习平台相比仍有一定差距。因此，该系统在后续工作中仍需要不断进行功能扩展与性能优化，以逐步提升其实用性与完整性。
- (2) 通过本次毕业设计，对软件系统开发的整体流程有了更加清晰和全面的认识。软件系统的开发不仅仅是代码实现，更重要的是贯穿需求分析、系统设计、模型建立、实现与测试等多个阶段，其中各类设计文档和模型在整个软件生命周期中同样具有重要作用。合理的系统建模与结构设计是保证软件系统质量与可维护性的关键环节。

展望未来，基于 Web 的考研刷题平台系统在学习管理与数据分析方面具有较大的应用潜力。该系统能够充分利用计算机系统运行速度快、数据处理准确可靠的特点，为考研考生提供更加高效、系统的学习支持，同时也为平台管理人员提供多样化的数据统计与管理手段。然而，当前系统仍有许多方面有待进一步完善。在今后的学习与研究过程中，作者将继续深入相关技术领域，逐步解决系统中存在的问题与不足，不断优化系统功能与性能，以期将该平台建设成为更加完善、实用的在线学习系统。

## 参 考 文 献

- [1] 郑宇. 探析计算机软件工程项目管理[J].信息工程, 2017, (09) : 38.
- [2] 王伟敏. 软件工程化管理方法实践[J].电子质量, 2017, (10) : 58-62.
- [3] 王瑶. 系统软件开发过程中的软件工程技术应用策略[J].科技展望, 2017,27 (30) : 1-2+15.
- [4] 吴彦博. 软件工程中数据挖掘技术的运用探索[J].数字通信世界, 2017,(09): 187.
- [5] 王龙, 魏丽娟, 马艳娥, 武尧. 基于 CDIO“软件工程”课程教学模式改革与实践[J].计算机时代, 2017, (10) : 56-58.
- [6] 彭东海, 成运, 谢四连. 软件工程专业转型发展的探索与实践--以湖南人文科技学院为例[J].湖南人文科技学院学报, 2017,34 (04) : 83-86.
- [7] 白伟华. 基于协同的软件工程专业实践教学改革研究[J].科教文汇 (中旬刊), 2017, (08) : 47-48.
- [8] 卜晓波. 软件工程金融信息化[J].电子技术与软件工程, 2017, (20) : 45.
- [9] 王依楠. 数据库基于计算机软件工程的编程技术[J].电子技术与软件工程, 2017, (20) : 165.
- [10] 沈海波, 周如旗, 朱雄泳. 工程教育认证背景下的软件工程专业建设和实践[J].软件, 2017,38 (09) : 87-90.
- [11] 张博, 郝杰. 项目驱动与微课混合教学模式用于软件工程实践的教学研究[J].卫生职业教育, 2017,35 (21) : 73-74.
- [12] Fielding, Roy Thomas. Architectural Styles and the Design of Network-based Software Architectures (Doctoral dissertation). University of California, Irvine, 2000.

[13] Express. Express - Node.js web application framework (Official documentation/website).