



广东工业大学

专项设计报告

基于 Django 的大学生就业信息共享平台

学 院 先进制造学院

专 业 人工智能

年级班级 22 级 3 班

学 号 3122009223

学生姓名 罗展鹏

指导教师 侯艳

2025 年 11 月



廣東工業大學

专业项目设计报告 基于 Django 的大学生就业信息共享平台

学 院 先进制造学院
专 业 人工智能
年级班级 22 级(3)班
学 号 3122009223
学生姓名 罗展鹏
指导教师 侯艳

2025 年 11 月

成绩评定栏（教师填写）				
平时+答辩（占 40%）	项目设计报告(60%)	总评百分制	总评等级	签名

摘要

针对当前大学生就业信息分散、企业招聘资质难核验、求职与招聘信息匹配效率低等问题，本文设计了一款基于 Django 框架的大学生就业信息资源共享平台，采用前后端分离架构，结合 Python 编程语言、数据库管理技术及软件工程方法论，构建高效、安全的就业信息交互体系。

平台围绕“学生-企业-系统管理”三类用户需求划分核心功能模块：学生模块支持用户登录注册、简历上传编辑、就业经验分享、招聘信息精准筛选及求职实时提醒，解决学生获取就业资源碎片化的痛点；企业模块聚焦资质审核与基础招聘功能，实现企业合规入驻与求职者简历定向查看，保障招聘信息真实性；系统管理模块通过用户账号与权限管控、招聘及发帖内容合规审核、公告发布管理，维护平台信息安全与运营秩序。

开发过程中，首先完成需求分析与整体方案设计，明确模块划分原则与数据交互逻辑；其次基于 Django REST Framework 构建后端 API 接口，设计 MySQL 数据库存储用户、简历、招聘等核心数据，结合 Vue.js 完成前端页面开发；最后通过功能模块测试、组装测试及需求验证，确保平台各功能稳定运行——测试结果表明，平台可实现就业信息高效共享与精准匹配，满足学生求职、企业招聘及平台管理的多元化需求，为高校就业服务数字化提供实践参考。

本文的创新点在于将“信息共享”与“合规审核”深度结合，在提升就业信息流通效率的同时，通过多层级审核机制保障信息质量；技术层面采用前后端分离架构，降低系统耦合度，便于后续功能扩展与维护。

关键词：Django 框架；就业信息平台；前后端分离；数据库设计；信息共享

Abstract

To address the current issues of fragmented job information for college students, difficulties in verifying corporate recruitment qualifications, and low efficiency in matching job seekers with available positions, this paper designs and implements a Django-based platform for sharing employment information resources. Adopting a front-end and back-end separation architecture, the platform integrates Python programming, database management technologies, and software engineering methodologies to establish an efficient and secure employment information exchange system.

The platform organizes its core modules around three user categories: Students, Enterprises, and System Management. The Student module provides essential features including account registration, resume editing, job experience sharing, targeted job search filters, and real-time alerts, effectively addressing students' fragmented access to employment resources. The Enterprise module focuses on qualification verification and basic recruitment functions, enabling compliant company registration and targeted resume screening for job seekers, while ensuring the authenticity of job postings. The System Management module safeguards platform security and operational order through account access control, content compliance review for recruitment posts, and official announcement management.

During the development process, we first conducted requirement analysis and overall solution design, establishing clear principles for module division and data interaction logic. Next, we built backend API interfaces using Django REST Framework, designed MySQL databases to store core data including users, resumes, and recruitment information, and integrated Vue.js for front-end page development. Finally, through functional module testing, integration testing, and requirement verification, we ensured stable operation of all platform features. Test results demonstrate that the platform enables efficient sharing and precise matching of employment information, meeting diverse needs for student job hunting, corporate recruitment, and platform management. This provides practical references for digitalizing university employment services.

The innovation of this paper lies in the deep combination of "information sharing" and "compliance audit", which not only improves the efficiency of employment information circulation, but also ensures the quality of information through multi-level audit mechanism. On the technical level, the front-end and back-end separation architecture is adopted to reduce the coupling degree of the system, which is convenient for subsequent function expansion and maintenance.

Keywords:Django framework; job information platform; front-end and back-end separation; database design; information sharing

目录

摘 要	3
1. 绪论	6
1.1. 立项背景	6
1.1. 立项原因概述	6
1.2. UML 概述	7
2. 项目背景	9
3. 需求分析	10
3.1. 功能需求	10
3.2. 性能需求	12
3.3. 系统安全性需求	13
3.4. 系统备份与恢复需求	13
3.5. 其他需求	13
3.6. 用例建模	14
4. 概要设计	20
4.1. 系统总体概述	20
4.2. 功能模块概述设计	20
4.3. 面向对象设计方法	23
4.4. 核心子系统结构设计	24
4.5. 动态行为建模	26
4.6. 交互建模	30
5. 结论及展望	32
5.1. 结论	32
5.2. 展望	32
6. 参考文献	34

1. 绪论

1.1. 立项背景

在数字化经济融入高校就业服务的当下，毕业生就业信息交互体系存在明显痛点，传统模式已制约就业资源配置效率，亟需通过项目破解。

学生端面临信息碎片化与经验壁垒：就业信息散于多渠道，需耗费大量时间筛选无效内容；学生间实习心得、面试技巧等隐性经验难共享，加剧求职信息差。企业端受资质失范与匹配低效困扰：部分企业缺乏统一资质审核，招聘信息存在岗位描述模糊、薪资不实等问题；简历筛选无精准工具，匹配率低、招聘周期长。高校端数字化能力不足：依赖线下服务，难整合信息、提供个性化指导；企业资质审核无系统化工具，就业风险预警滞后，难保障学生求职安全。

技术层面，现有平台多为前后端耦合架构，扩展性差，且信息共享与合规审核协同缺失，难平衡效率与安全。

在此背景下，立项开发大学生就业信息共享平台具备必要性与可行性：现有数字化技术可支撑信息整合、精准匹配等需求，项目能切实解决学生、企业、高校痛点，对推动就业市场数字化转型有实践价值。

1.1. 立项原因概述

在当前大学生就业市场中，学生、企业与高校就业服务端均面临突出痛点，具体问题如下：

1. 学生求职时，就业信息分散于多渠道，筛选成本高，且缺乏规模化的就业经验共享渠道，信息差明显。
2. 企业招聘时，资质审核无统一标准，招聘信息真实性难保障，同时缺乏精准简历筛选工具，招聘效率低、成本高。
3. 高校就业服务端，数字化工具缺失，难以整合就业信息、提供个性化指导，且对企业资质审核和风险预警能力不足。
4. 现有平台技术架构落后，多为前后端耦合设计，扩展性差，且未平衡信息共享效率与合规审核需求。

针对以上痛点，搭建基于 Django 的大学生就业信息共享平台具有显著价值，核

心机会点如下：

1. 整合多渠道就业信息，通过精准筛选功能减少学生无效信息筛选时间，提升求职效率。
2. 建立企业资质多层级审核机制，保障招聘信息真实性，同时为企业提供精准简历匹配工具，降低招聘成本。
3. 为高校提供数字化管理后台，支持就业信息整合、分析与个性化推送，强化就业指导与风险预警能力。
4. 采用前后端分离架构，降低系统耦合度，便于后续功能扩展，同时实现信息共享与合规审核协同运转。
5. 搭建学生就业经验共享社区，打破信息壁垒，形成互助生态，弥补求职过程中隐性知识传递不足的问题。

1.2. UML 概述

面向对象的系统分析与设计，包括 OOA（面向对象分析）与 OOD（面向对象设计）两个部分。其中 OOA 的主要任务是分析问题，找出问题解决方案，同时发现对象并分析对象内部构成和外部关系，建立软件系统的对象模型。OOD 的主要任务是根据已确立的系统对象模型，运用面向对象技术，设计对象与类，进而设计系统结构、人机界面、数据管理、任务管理等子系统。

UML（Unified Modeling Language）是第 3 代的面向对象建模语言，由 Rational 公司的面向对象技术研究专家 Grady Booch、Ivar Jacobson 和 Jim Rumbaugh 联合开发。它采纳和扩展了 Booch 标记法、OMT 的面向对象建模技术的方法，以及面向用例（Use-Case）的 OOSE 方法，并且吸取了其他学者和软件厂商的建议。UML 在提交给 OMG（Object Management Group）后，最终成为大众所接受的标准建模语言。UML 建模语言吸取和发展了多种方法的优点，融入了软件工程领域的新思想、新方法和新技术，提出如模板、扩展机制、活动图等新概念。它易于表达且功能强大，应用广泛，不但适用于面向对象的软件分析与设计，还支持从需求分析开始的软件开发的全过程。UML 定义了一系列图形工具，以对现实世界进行面向对象建模。

标准建模语言 UML 已成为面向对象技术的主流建模工具，支持系统分析、设计和实现等软件开发全过程。UML 定义了 5 类 10 种模型图，用于提供静态建

模机制和动态建模机制：

静态模型包括用例图、类图、对象图、包图、构件图、配置图。

动态模型包括行为图（活动图、状态图）和交互图（顺序图、协作图）。

各类模型图的核心作用如下：

- (1) 用例图：描述应用领域需求与功能，以及事务（用例）之间的逻辑关系。
- (2) 类图：显示一组类、接口、协作以及它们之间的关系。
- (3) 对象图：类的实例图，展示特定时刻对象的状态及关联。
- (4) 包图：提供类图的分组优化机制，简化复杂系统的建模。
- (5) 活动图：描述多用例和多对象的复杂行为及活动间的约束关系，或复杂行为中的活动及其执行顺序。
- (6) 状态图：描述单个对象的所有可能状态及转移条件。
- (7) 顺序图：强调消息的时间顺序的交互图，展示对象间交互的时序流程。
- (8) 协作图：强调发送和接收消息的对象之间的结构组织的交互图。
- (9) 构件图：描述软构件及相互间的依赖（及调用）关系，包括源文件、二进制文件、可执行文件。
- (10) 配置图：描述构件运行环境及系统软硬件体系结构。

2. 项目背景

在数字化经济深度融入高校就业服务的当下，高校毕业生就业作为连接教育成果与社会人才需求的关键环节，其效率与质量直接关系到学生职业发展、企业人才储备及高校就业服务水平。然而，当前大学生就业信息交互场景中，多主体需求与现有服务模式间的矛盾日益凸显，亟需针对性的信息整合与服务载体。

从学生群体来看，求职过程中面临信息获取分散化问题——就业信息散落在高校就业网、企业官网、第三方招聘平台等多个渠道，信息格式不统一、更新节奏不同步，导致学生需耗费大量时间筛选无效内容，难以快速定位匹配岗位；同时，学生间的就业经验（如面试技巧、实习心得）多依赖线下传递，覆盖范围有限，无法形成规模化的互助生态，加剧了求职信息差。

从企业端分析，招聘环节存在资质核验缺失与信息匹配低效的双重问题：部分企业尤其是中小微企业，缺乏统一的资质审核标准，发布的招聘信息可能存在岗位描述模糊、薪资信息不实等情况，既损害毕业生权益，也降低企业招聘公信力；此外，企业筛选简历时缺乏精准的信息匹配手段，面对海量简历常陷入“筛选成本高、匹配度低”的困境，延长招聘周期。

从高校就业服务层面观察，传统服务模式难以满足数字化需求：多数高校仍依赖线下宣讲会、纸质简历投递等方式，对就业信息的整合、分析与个性化推送能力不足，无法为学生提供精准的就业指导；同时，高校对企业招聘资质的审核缺乏系统化工具，难以全面把控招聘信息真实性，对就业市场风险的预警能力较弱，难以充分保障学生求职安全。

在此背景下，面向“学生 - 企业 - 高校就业服务”三类核心主体，构建一个聚焦就业信息共享与合规管理的平台成为必要。该平台需为学生提供集中化的求职信息获取与经验交流渠道，为企业提供规范化的招聘信息发布与简历筛选入口，为高校及平台运营方提供专业化的信息审核与服务能力，通过整合多方需求，弥补当前就业信息交互场景中的短板，助力就业市场资源高效配置。

3. 需求分析

3.1. 功能需求

A. 学生端功能需求

1. 账号管理：实现注册、登录、密码找回、个人信息编辑功能。
2. 简历管理：支持简历上传、在线编辑、多版本保存、简历预览与导出。
3. 招聘信息查询：支持按行业、岗位、薪资、地区、企业类型等条件精准筛选。
4. 信息互动：实现招聘信息收藏、投递简历、查看投递状态功能。
5. 经验共享：支持发布就业经验、面试心得、实习感悟，可点赞、评论、收藏他人内容。
6. 求职提醒：支持订阅岗位更新、投递反馈、面试邀约等实时提醒。
7. 消息中心：接收企业沟通消息、平台通知、提醒消息，支持已读 / 未读标记。

B. 企业端功能需求

1. 企业入驻：提交营业执照、招聘资质等材料，完成平台审核入驻。
2. 账号管理：实现登录、密码找回、企业信息编辑、操作员权限配置。
3. 招聘管理：支持发布、编辑、下架招聘信息，设置岗位要求与筛选条件。
4. 简历管理：查看精准匹配的简历、筛选简历、标记简历状态、下载简历。
5. 沟通管理：与投递学生在线沟通，发送面试邀约、录用通知等。
6. 数据统计：查看招聘信息曝光量、简历投递量、筛选通过量等数据报表。

C. 系统管理端功能需求

1. 用户管理：查看学生与企业用户列表，处理账号冻结、解封、注销申请。
2. 资质审核：审核企业入驻材料，给出通过或驳回意见，记录审核日志。
3. 内容审核：审核招聘信息、社区经验帖，处理违规内容（下架、屏蔽、警告）。
4. 公告管理：发布、编辑、下架平台公告，支持定向推送（全体 / 学生 / 企业）。
5. 权限管理：配置管理员账号权限，划分操作范围（如内容审核员、资质审核员）。
6. 数据管理：统计平台用户数、招聘信息数、简历数、投递量等核心数据，支持报表导出。

7. 违规处理：接收用户投诉举报，核查后处理违规账号或内容，反馈处理结果。

各部分功能描述

A. 学生端

1. 账号管理：学生通过手机号或学号注册，完成实名认证后登录；支持修改个人信息、更换绑定手机号、找回遗忘密码。
2. 简历管理：提供简历模板或自定义编辑功能，支持上传 PDF 格式简历，可保存多版本，编辑后可预览效果或导出打印。
3. 招聘信息查询：首页展示推荐岗位（基于简历关键词匹配），学生可通过多条件组合筛选，快速定位目标岗位，查看岗位详情（职责、要求、薪资、企业信息）。
4. 信息互动：对意向岗位进行收藏，方便后续查看；通过平台直接投递简历，实时查看投递状态（待查看、已查看、面试邀约、已拒绝）。
5. 经验共享：学生可发布图文结合的经验帖，分类标签（如面试技巧、简历优化、行业认知）；浏览他人经验帖，进行点赞、评论交流，收藏优质内容。
6. 求职提醒：自定义订阅提醒类型，通过平台消息、短信或邮件接收岗位更新、投递反馈等通知，避免错过关键信息。
7. 消息中心：集中展示所有消息，区分系统通知、企业沟通消息、面试邀约，支持标记已读、删除消息。

B. 企业端

1. 企业入驻：企业提交相关材料，填写企业基本信息（行业、规模、简介、联系方式），等待平台审核。
2. 账号管理：企业 HR 使用注册手机号登录，支持修改企业简介、联系方式等信息；可创建多个操作员账号，分配不同权限（如发布招聘、筛选简历）。
3. 招聘管理：发布招聘信息时填写岗位名称、职责、要求、薪资范围、工作地点等内容，支持上传企业宣传图片；可随时编辑已发布信息，或下架过期岗位。
4. 简历管理：平台根据岗位要求自动匹配简历，企业可按学历、专业、实习经历等条件二次筛选；对简历标记“合适”“不合适”“待沟通”等状态，

支持批量下载符合要求的简历。

5. 沟通管理：通过平台内置沟通工具，向学生发送消息，沟通求职疑问；可一键发送标准化面试邀约，明确面试时间、地点、方式。
6. 数据统计：查看每个招聘岗位的曝光量（被多少学生查看）、投递量、筛选通过量、面试量等数据，辅助优化招聘信息。

C. 系统管理端

1. 用户管理：管理员可按条件查询学生或企业用户，查看账号状态；对违规账号执行冻结、解封操作，处理用户提交的注销申请，保障账号安全。
2. 资质审核：接收企业入驻申请后，核查提交材料的真实性与完整性，符合要求则审核通过，开通企业端功能；不符合则驳回并说明原因，企业可补充材料重新申请。
3. 内容审核：采用“机器初审 + 人工复核”模式，审核招聘信息与社区内容，排查虚假招聘、违规宣传、敏感信息等，对违规内容执行下架、屏蔽操作，对账号发送警告通知。
4. 公告管理：管理员编辑公告内容（支持图文），选择发布范围与生效时间，发布后在对应用户端展示；可随时编辑已发布公告或下架过期公告。
5. 权限管理：按工作职能划分管理员权限，如“内容审核员”仅能处理内容审核任务，“超级管理员”拥有全部操作权限，避免权限滥用。
6. 违规处理：接收用户对违规账号、虚假招聘、不良内容的投诉举报，核查后按规则处理，处理结果反馈给投诉用户，同时记录处理日志备查。

3.2. 性能需求

1. 响应速度：在 100M 网络环境下，页面加载时间 ≤ 2 秒，查询操作响应时间 ≤ 1.5 秒，简历投递、信息提交等操作响应时间 ≤ 3 秒。
2. 并发能力：支持同时在线用户数 ≥ 1000 人，并发请求处理数 ≥ 200 次/秒，招聘高峰期（如秋招、春招）可稳定运行，无卡顿或崩溃现象。
3. 数据处理：支持百万级数据存储（用户、简历、招聘信息），单表数据量达 50 万条时，查询响应时间仍 ≤ 2 秒。
4. 稳定性：系统连续运行无故障时间 ≥ 720 小时/月，故障恢复时间 ≤ 30 分钟，保障求职与招聘活动不受影响。

5. 兼容性：学生端支持主流浏览器（Chrome、Edge、Firefox、Safari）及移动端适配，企业端与管理端支持主流浏览器，兼容不同分辨率屏幕。

3.3. 系统安全性需求

1. 身份认证：采用手机号 + 验证码、账号 + 密码双重认证方式，学生可绑定学号进行实名认证，企业需完成对公验证，防止虚假账号注册。
2. 权限控制：严格划分三类用户权限，学生仅能操作个人相关功能，企业无法查看非投递简历，管理员权限按职能拆分，无越权操作可能。
3. 数据安全：用户密码采用加密存储（如 MD5 加盐加密），简历、企业资质等敏感数据传输过程中采用 HTTPS 加密，防止数据泄露。
4. 防攻击能力：具备防 SQL 注入、XSS 跨站脚本、CSRF 跨站请求伪造等常见网络攻击的能力，配置防火墙与入侵检测机制。
5. 操作日志：记录关键操作（如账号登录、简历下载、资质审核、违规处理），日志保留时间≥6 个月，便于追溯问题。
6. 隐私保护：严格遵守个人信息保护相关规定，不泄露学生简历、企业商业信息等隐私内容，仅授权用户可查看相关数据。

3.4. 系统备份与恢复需求

1. 数据备份：采用“全量备份 + 增量备份”结合的方式，全量备份每周执行 1 次，增量备份每日执行 1 次；备份数据存储在异地服务器，防止本地存储故障导致数据丢失。
2. 备份验证：定期对备份数据进行完整性验证，确保备份有效，可正常恢复。
3. 数据恢复：支持按时间点恢复数据，恢复过程中不影响系统正常运行；确保业务连续性。

3.5. 其他需求

1. 数据库需求：系统采用 MySQL 数据库，管理员需掌握基础数据库操作与维护技能，定期进行数据库优化（如索引优化、碎片整理）；根据用户增长情况，可扩展数据库集群，提升存储与访问能力。
2. 系统维护需求：提供系统监控工具，实时监测服务器状态、数据库性能、接口响应情况；支持日志查询与异常告警，便于快速定位并解决问题。

3. 可扩展性需求：系统架构设计预留扩展接口，后续可新增功能模块（如视频面试、内推功能、就业指导课程），无需重构核心架构。
4. 易用性需求：界面设计简洁直观，操作流程清晰，减少学习成本；提供帮助文档或操作指引，解答常见问题，提升用户使用体验。

3.6. 用例建模

3.6.1. 简历上传与编辑

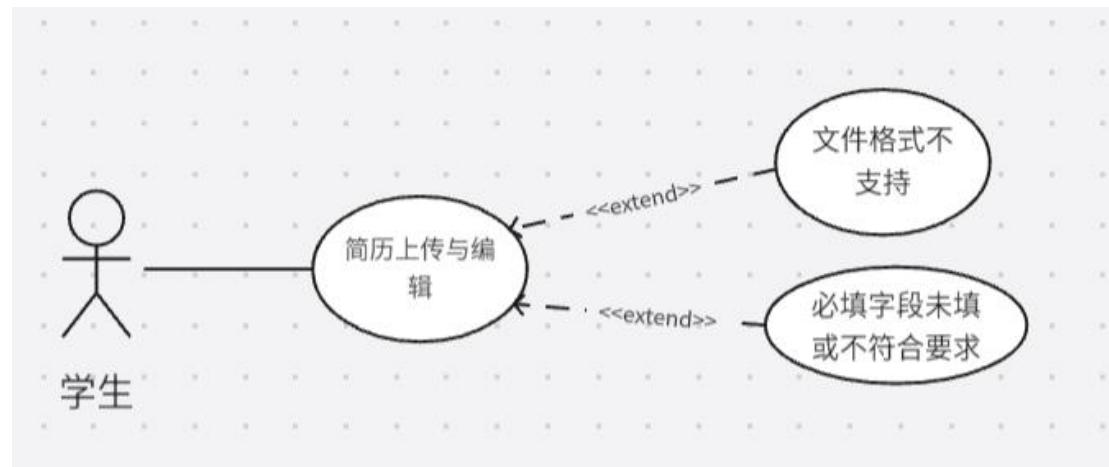


图 3.1 学生求职简历相关用例图

表 3.1 学生简历上传与编辑用例规约表

用例：简历上传与编辑
ID: 001
简单描述：学生可上传、编辑个人简历，保存多版本并导出
主要参与者：学生
副参与者：无
前置条件：学生已成功注册并登录平台
主事件流：1. 学生进入“简历管理”模块，选择“新建简历”或“编辑现有简历”。2. 选择简历模板或自定义编辑，填写个人信息、教育经历、实习经历等内容。3. 可选择上传 Word/PDF 格式简历文件，系统自动解析填充内

用例：简历上传与编辑

容。4. 编辑完成后，点击“保存”，可选择保存为新版本或覆盖原有版本。5. 支持预览简历效果，或点击“导出”生成 PDF 文件下载。

其他事件流：1. 若上传文件格式不支持，弹窗提示“仅支持 Word/PDF 格式，请重新上传”。2. 若必填字段未填写，弹窗提示“请完善必填信息后保存”。

后置条件：简历保存成功，可在简历列表中查看、再次编辑或投递

3.6.2. 学生端招聘信息筛选与投递

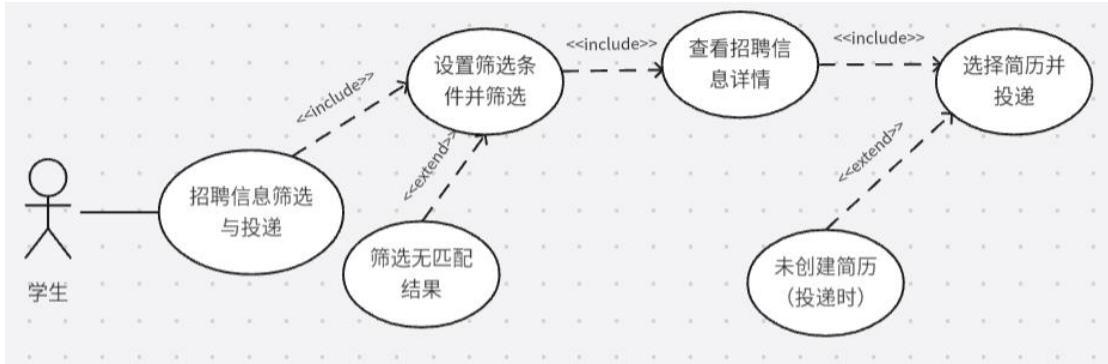


图 3.2 学生筛选岗位并投递用例图

表 3.2 招聘信息筛选与投递用例规约表

用例：招聘信息筛选与投递

ID: 002

简单描述：学生可按条件筛选招聘信息，查看详情并投递简历

主要参与者：学生

副参与者：无

前置条件：学生已成功注册并登录平台，且已创建至少一份简历

主事件流：1. 学生进入“招聘信息”模块，选择筛选条件（行业、岗位、薪资、地区等）。2. 系统展示符合条件的招聘信息列表，学生点击某条信息查看详情。3. 若有意向，点击“收藏”添加至收藏夹，或点击“投递简历”。4.

选择要投递的简历版本，点击“确认投递”。5. 系统提示“投递成功”，并跳转至投递记录页面。

其他事件流：1. 若筛选条件无匹配结果，显示“暂无符合条件的招聘信息，可调整筛选条件尝试”。2. 若未创建简历，点击“投递简历”时弹窗提示“请先创建简历后再投递”，并引导至简历管理模块。

后置条件：招聘信息收藏成功或简历投递成功，可在对应记录中查看状态

3.6.3. 企业入驻资质审核

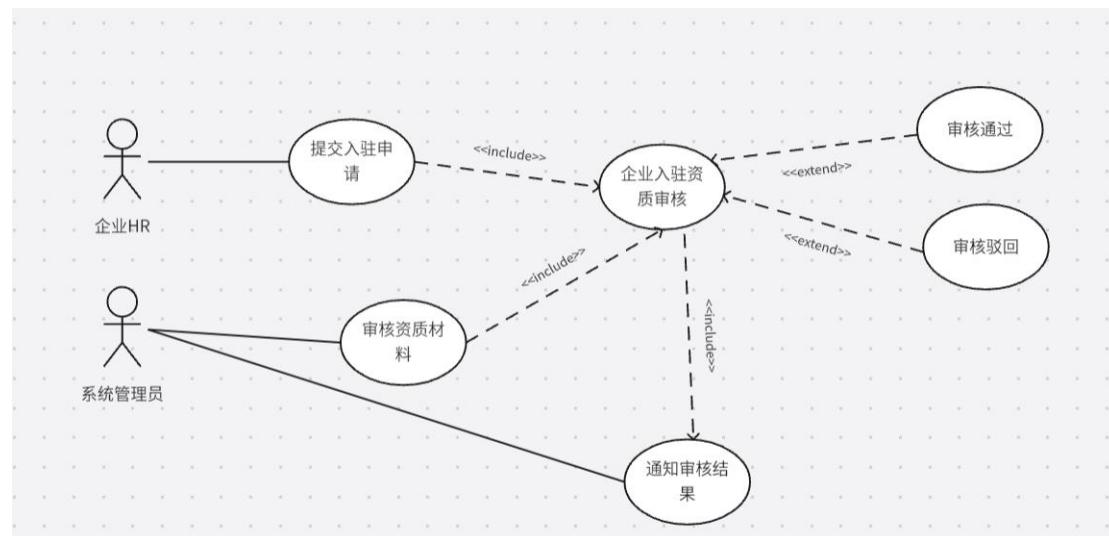


图 3.3 企业入驻资质审核用例图

表 3.3 企业入驻资质审核用例规约表

用例：企业入驻资质审核

ID：003

简单描述：企业提交入驻材料，管理员审核后开通企业端功能

主要参与者：企业 HR、系统管理员

副参与者：无

前置条件：企业未入驻平台，已准备营业执照、招聘资质等材料

主事件流：1. 企业 HR 进入平台注册页面，选择“企业注册”，填写企业基本信息（名称、行业、规模等）。2. 上传营业执照扫描件、法人身份证明、招聘

用例：企业入驻资质审核

授权书等资质材料。3. 提交入驻申请，系统提示“申请已提交，等待管理员审核”。4. 管理员收到审核通知，进入“资质审核”模块，查看企业提交的信息与材料。5. 若材料齐全且真实有效，审核通过，系统自动开通企业端功能，并向企业 HR 发送通知。6. 若材料不齐全或不符合要求，审核驳回，填写驳回原因，系统向企业 HR 发送通知。

其他事件流：1. 若企业名称已被注册，弹窗提示“该企业已入驻平台，可直接登录或找回密码”。2. 若上传材料模糊不清，管理员可驳回并要求“重新上传清晰的资质材料”。

后置条件：企业入驻审核通过，可登录企业端开展招聘工作；或审核驳回，可补充材料重新申请

3.6.4. 简历筛选与查看

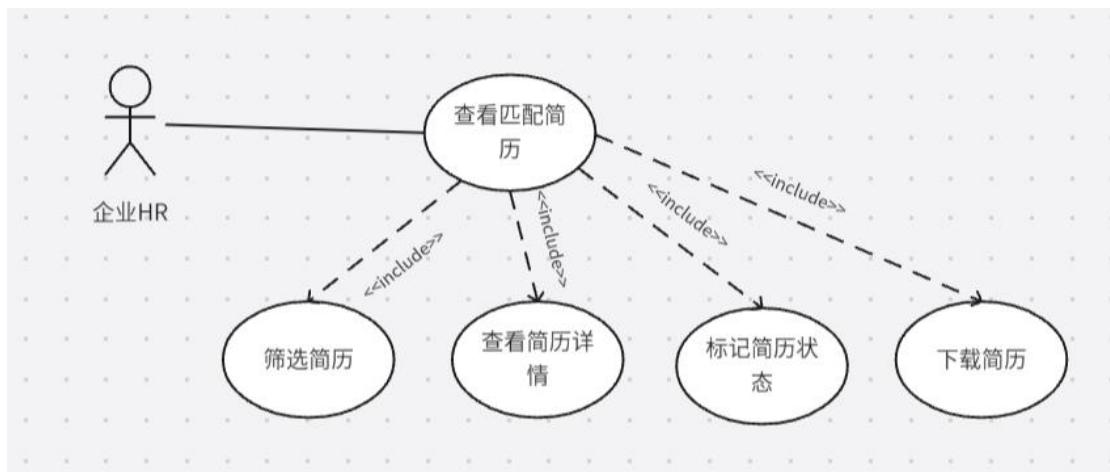


图 3.4 简历筛选与查看用例图

表 3.4 简历筛选与查看用例规约表

用例：简历筛选与查看

ID: 004

简单描述：企业 HR 可查看匹配简历，按条件筛选并查看详情

主要参与者：企业 HR

副参与者：无

用例：简历筛选与查看

前置条件：企业已入驻平台并成功登录，已发布至少一条招聘信息

主事件流：1. 企业 HR 进入“简历管理”模块，选择对应招聘岗位。2. 系统展示该岗位的匹配简历列表，支持按学历、专业、实习经历等条件筛选。3. 点击某份简历查看详情，包括个人信息、教育经历、实习经历、技能证书等。4. 可对简历标记状态（待沟通、合适、不合适），或点击“下载简历”保存文件。

其他事件流：1. 若该岗位暂无投递简历，显示“暂无简历投递，可优化招聘信息提升曝光”。2. 若筛选条件无匹配结果，显示“暂无符合条件的简历，可调整筛选条件尝试”。

后置条件：企业 HR 完成简历筛选与查看，可后续进行沟通或面试邀约

3.6.5. 招聘详细合规审核

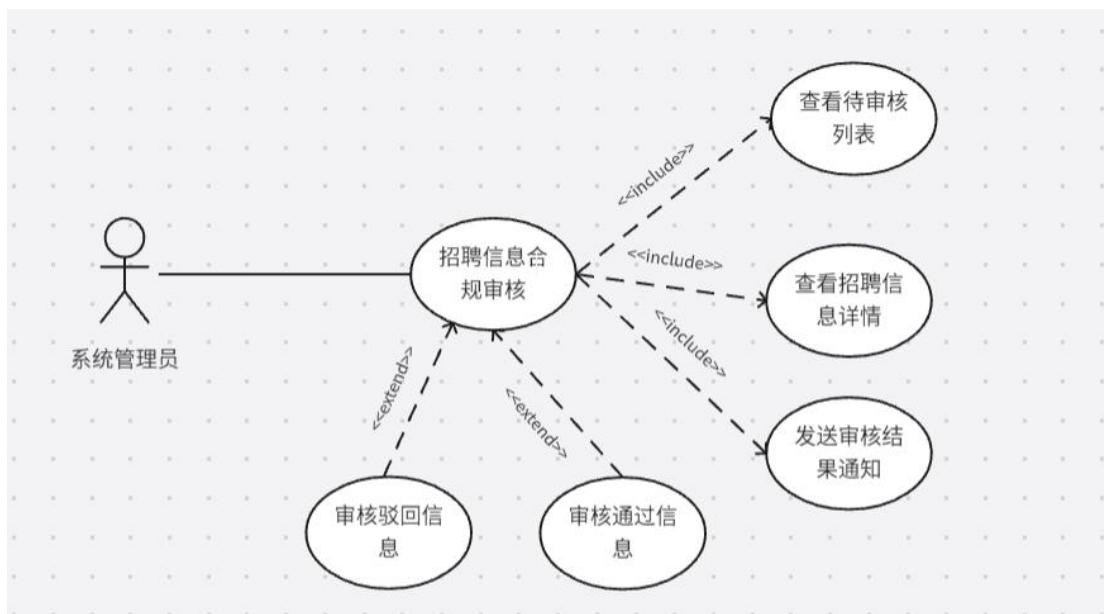


图 3.5 招聘信息合规审核用例图

表 3.5 招聘信息合规审核用例规约表

用例：招聘信息合规审核

ID: 005

简单描述：管理员审核企业发布的招聘信息，处理违规内容

用例：招聘信息合规审核

主要参与者：系统管理员

副参与者：无

前置条件：管理员已成功登录系统管理端，企业已发布待审核的招聘信息

主事件流：1. 管理员进入“内容审核”模块，选择“招聘信息审核”，查看待审核列表。2. 点击某条招聘信息，查看完整内容（岗位描述、薪资、企业信息等）。3. 若信息合规（无虚假宣传、敏感内容），点击“审核通过”，信息上线展示。4. 若信息违规，点击“审核驳回”，填写违规原因（如“岗位描述模糊”“薪资不实”），选择处理方式（下架、要求修改）。5. 系统向企业发送审核结果通知，违规信息按处理方式执行。

其他事件流：1. 若审核过程中发现信息存疑，可标记“待复核”，移交其他管理员二次审核。2. 若企业多次发布违规信息，可关联查看历史违规记录，按规则加重处理（如冻结发布权限）。

后置条件：招聘信息审核完成，合规信息上线，违规信息按要求处理

4. 概要设计

4.1. 系统总体概述

该系统围绕“学生 - 企业 - 系统管理”三类用户，采用前后端分离架构，分为三个核心操作端，各端功能相互协同，构建完整的就业信息共享生态：

1. 学生端（Web 端）：面向高校求职学生，核心功能包括账号管理、简历管理、招聘信息查询与投递、经验共享、消息接收等，满足学生求职全流程需求。
2. 企业端（Web 管理后台）：面向招聘企业 HR，核心功能包括企业入驻、账号管理、招聘信息发布与管理、简历筛选与查看、沟通邀约等，助力企业高效完成招聘工作。
3. 系统管理端（Web 管理后台）：面向平台管理员与高校就业服务人员，核心功能包括用户管理、资质审核、内容审核、公告管理、权限配置、数据统计等，保障平台规范运营。

系统技术架构分为三层：

- 表现层：采用 Vue.js 开发前端页面，实现跨端适配，通过 API 接口与后端交互，提供友好的用户界面。
- 业务逻辑层：基于 Django REST Framework 构建后端 API，处理核心业务逻辑（如用户认证、信息匹配、审核流程），实现各模块功能协同。
- 数据访问层：采用 MySQL 数据库存储核心数据，设计合理的数据表结构，保障数据存储安全与访问高效；同时可集成 Redis 缓存常用数据（如热门招聘信息、用户会话），提升系统响应速度。

4.2. 功能模块概述设计

A. 学生端功能模块图

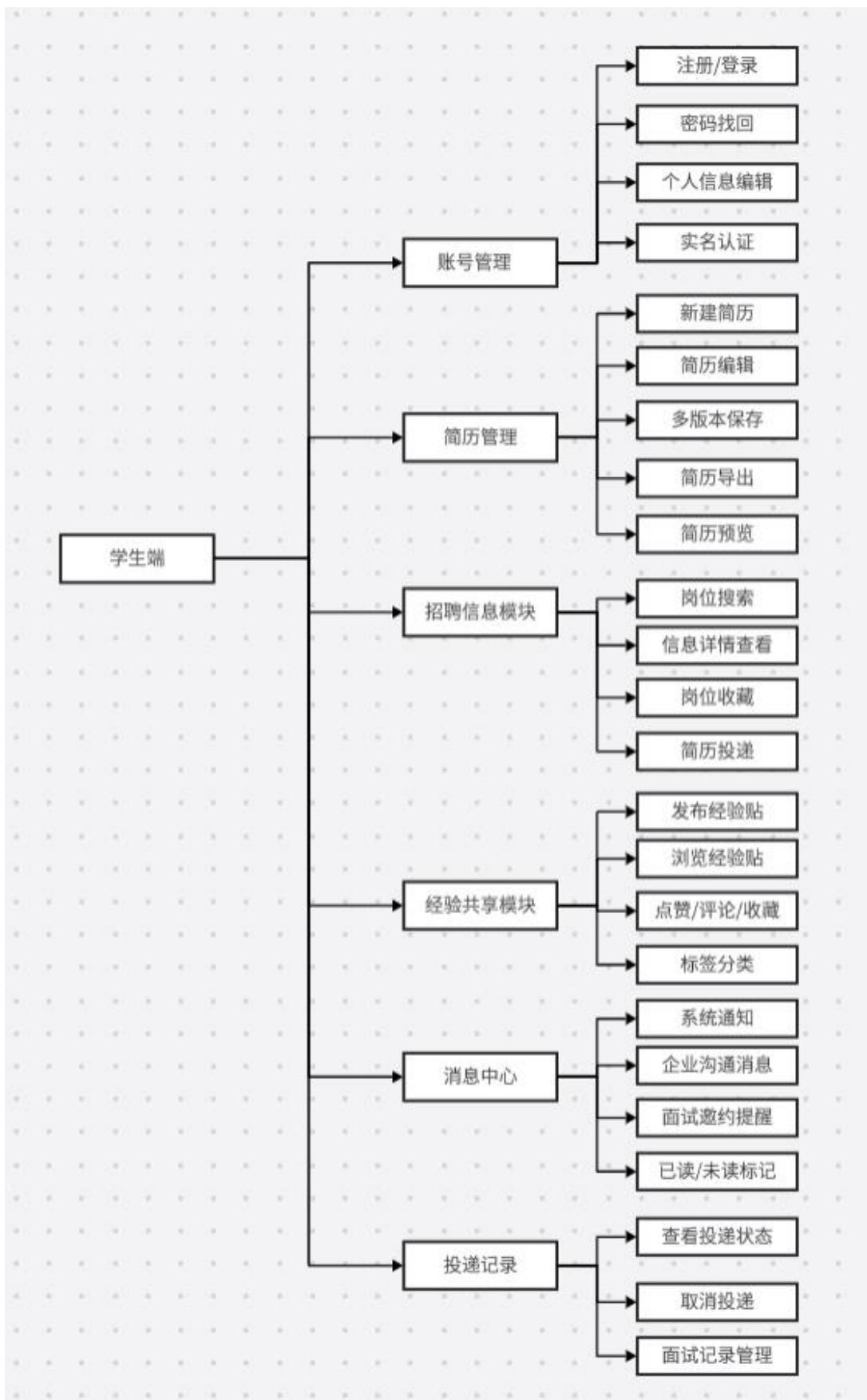


图 4.1 学生端功能模块图

B. 企业端功能模块图

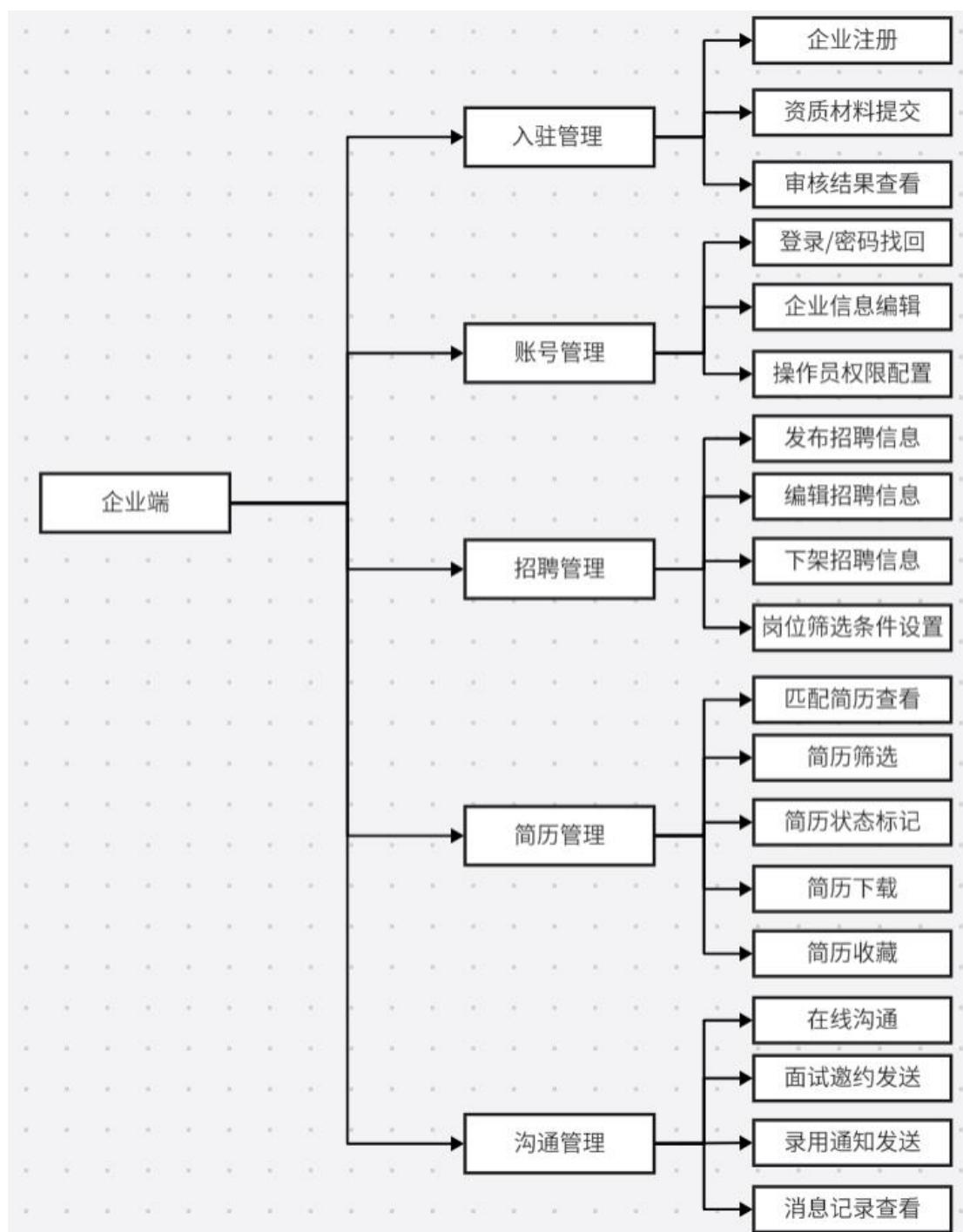


图 4.2 企业端功能模块图

C. 系统管理端功能模块图

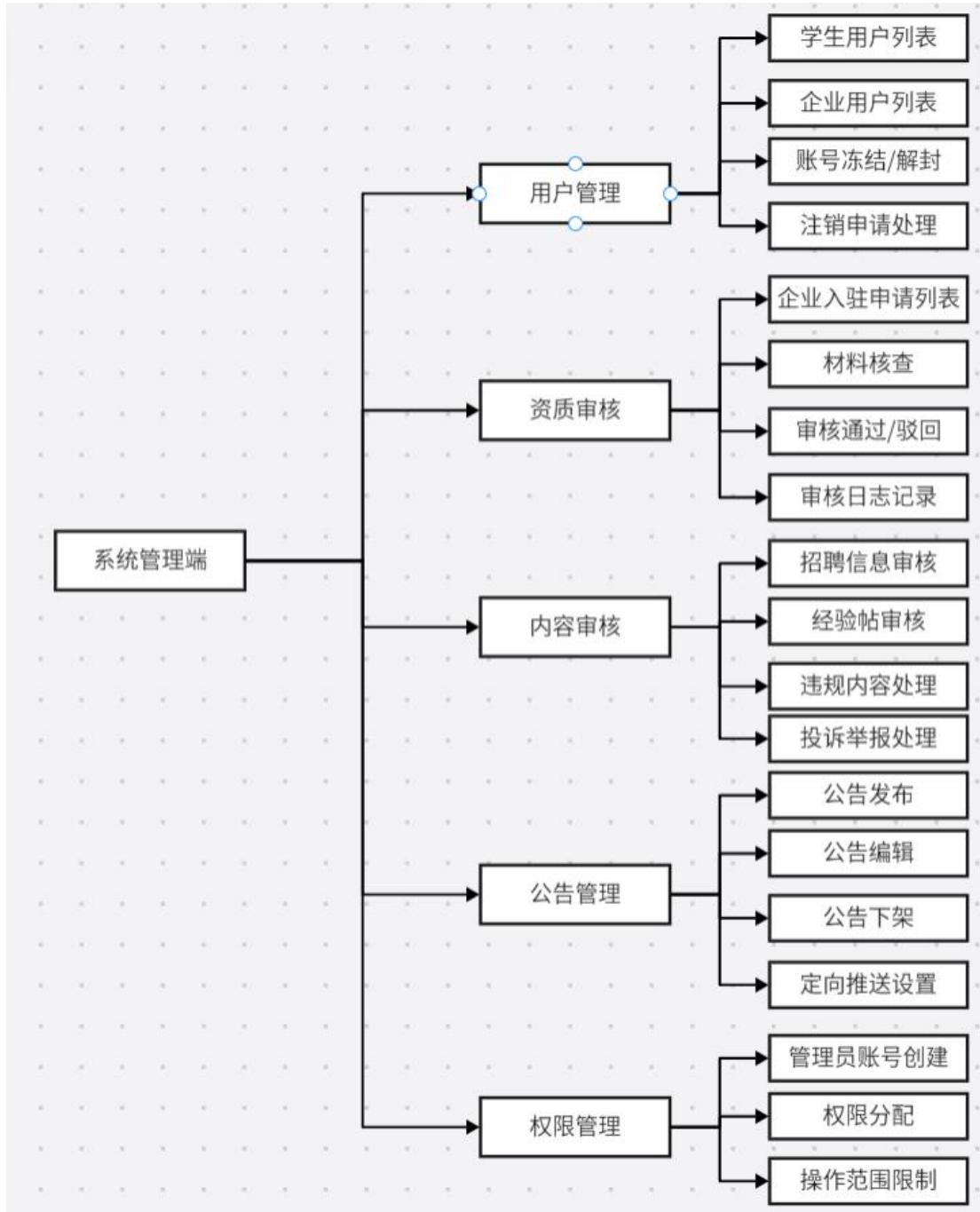


图 4.3 系统管理端功能模块图

4.3. 面向对象设计方法

本系统采用多层架构模式与面向对象设计方法结合，将整个系统划分为表现层、业务逻辑层、数据访问层，遵循“高内聚、低耦合”原则：

- 表现层：**负责与用户交互，接收用户操作请求，展示处理结果。采用 Vue.js 组件化开发，将页面拆分为多个独立组件（如简历编辑组件、招聘信息列表组件），提高代码复用性与维护性。

2. **业务逻辑层**: 核心业务处理层，接收表现层传递的请求，通过业务规则与逻辑处理后，调用数据访问层获取或存储数据。采用 Django 的视图类与序列化器，实现请求参数校验、业务逻辑处理、响应数据格式化。
3. **数据访问层**: 负责与数据库交互，封装数据查询、新增、修改、删除操作，屏蔽数据库底层细节。采用 Django ORM 框架，通过定义数据模型（Model）映射数据库表，简化数据库操作；同时设计数据访问接口，供业务逻辑层调用，降低层间耦合。

面向对象设计核心在于抽象系统中的实体与交互关系，定义核心类与类之间的关联：

- **核心实体类**: 学生类（Student/User）、企业类（Enterprise）、管理员类（Admin）、简历类（Resume）、招聘信息类（Recruitment）、经验帖类（Article）等。
- **类关系**: 学生类与简历类为“一对多”关系（一个学生可拥有多份简历）；企业类与招聘信息类为“一对多”关系（一个企业可发布多条招聘信息）；学生类与招聘信息类为“多对多”关系（一个学生可投递多个岗位，一个岗位可接收多个学生投递）。

4.4. 核心子系统结构设计

以“信息交互子系统”为例，该子系统负责学生、企业、管理员之间的信息流转与处理，核心结构如下：

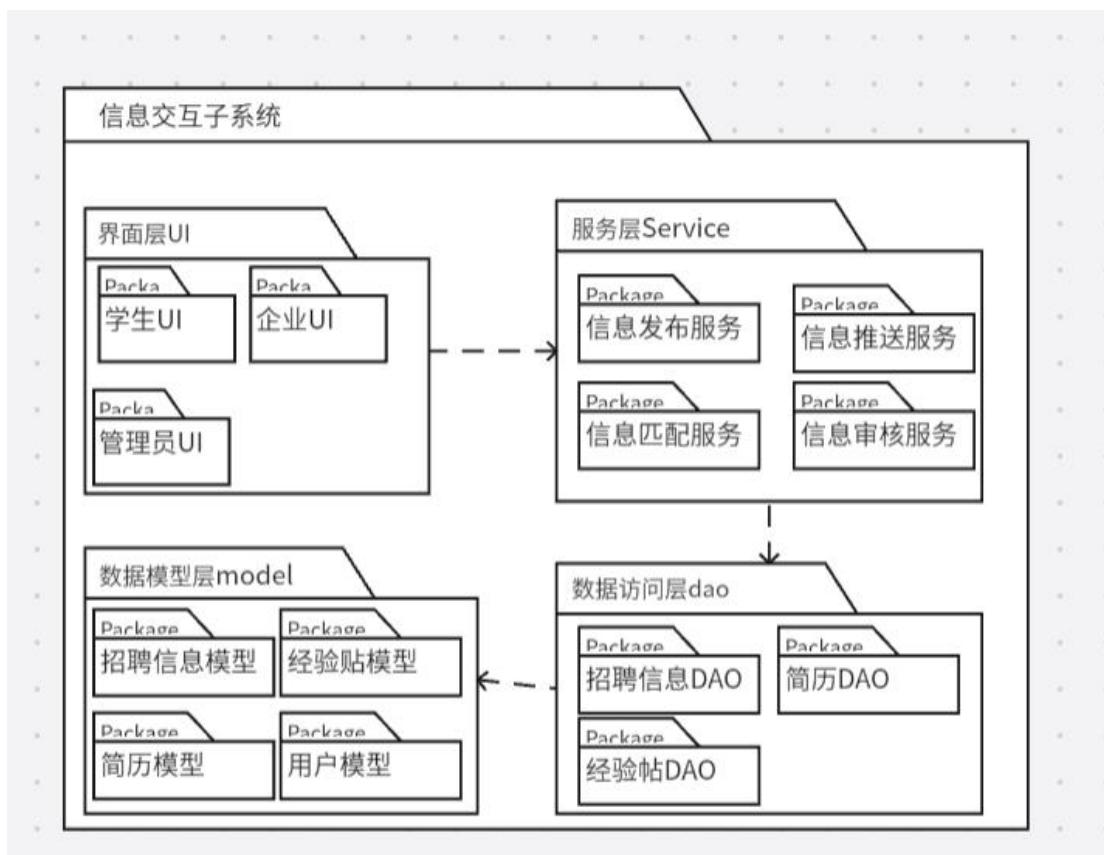


图 4.4 核心子系统结构图

各模块职责：

1. ui 层：提供用户操作界面，接收用户输入（如发布招聘信息、投递简历），展示信息处理结果（如审核通过、投递成功）。
2. service 层：实现核心业务逻辑，信息发布服务处理招聘信息、经验帖的发布流程；信息匹配服务根据岗位要求与简历关键词进行精准匹配；信息审核服务处理内容合规审核逻辑；信息推送服务将审核通过的信息、面试邀约等推送给目标用户。
3. dao 层：封装数据访问操作，为 service 层提供数据支持，如查询符合条件的简历、保存发布的招聘信息、更新信息审核状态。
4. model 层：定义数据模型，映射数据库表结构，存储信息的核心字段（如招聘信息包含岗位名称、薪资、要求等字段）。

4.5. 动态行为建模

4.5.1. 学生简历上传与编辑活动图

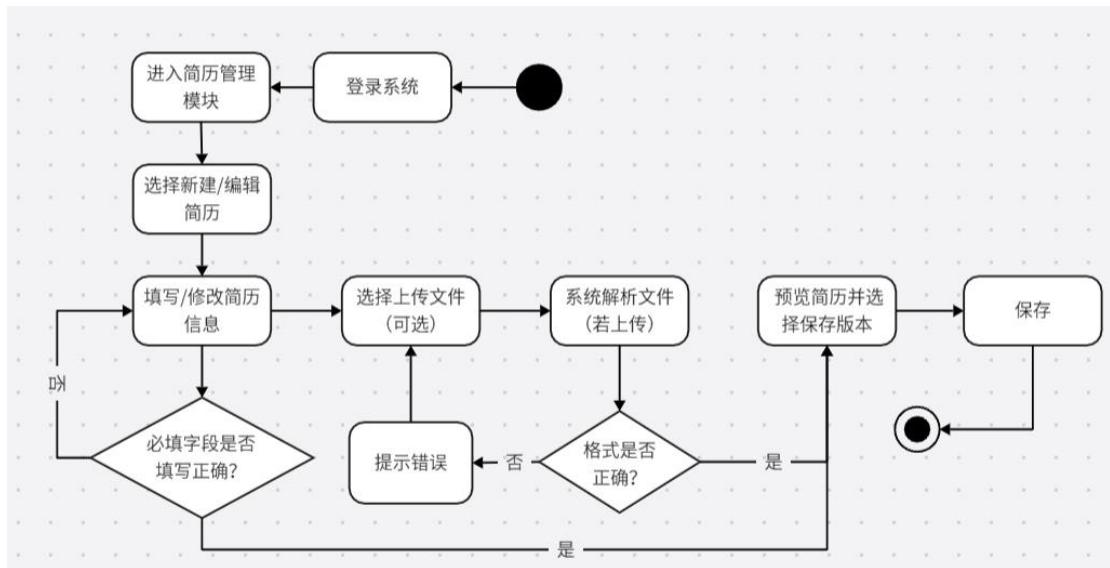


图 4.5 学生简历上传与编辑活动图

4.5.2. 企业资质审核流程活动图

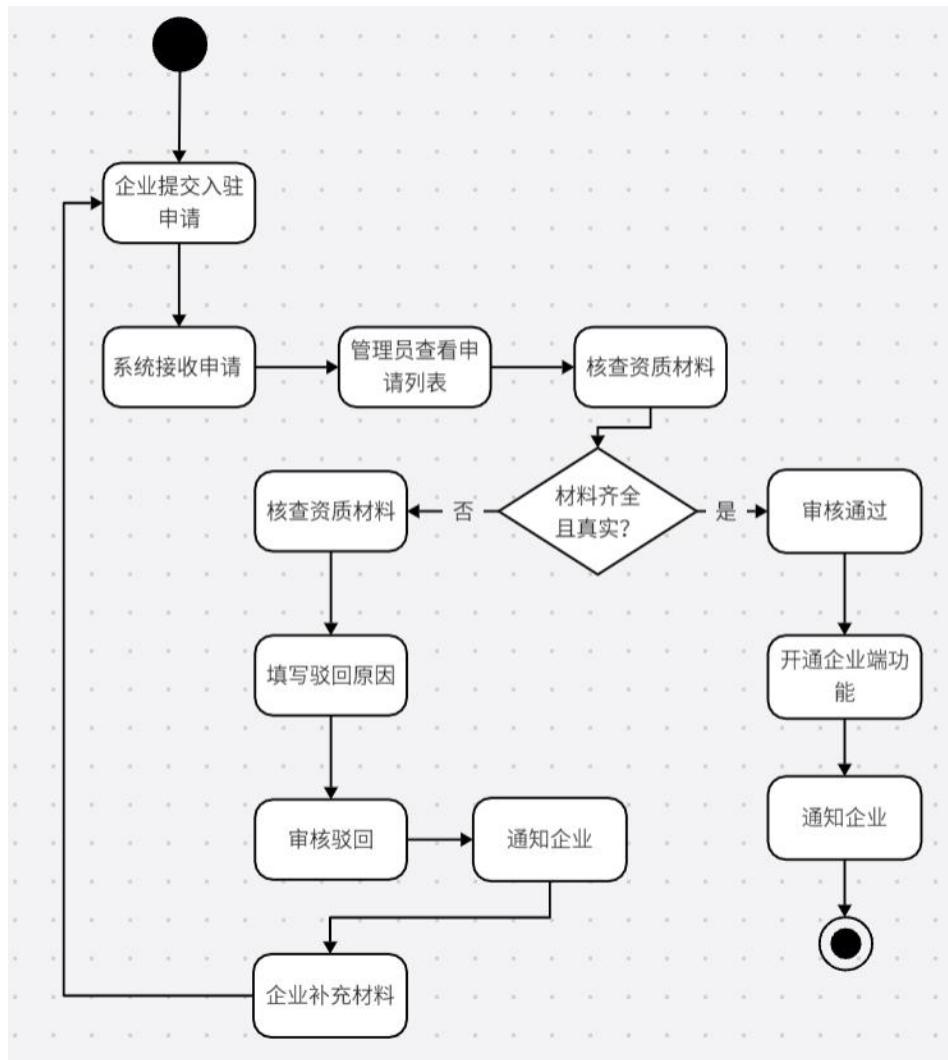


图 4.6 企业资质审核流程活动图

4.5.3. 招聘信息审核与发布活动图

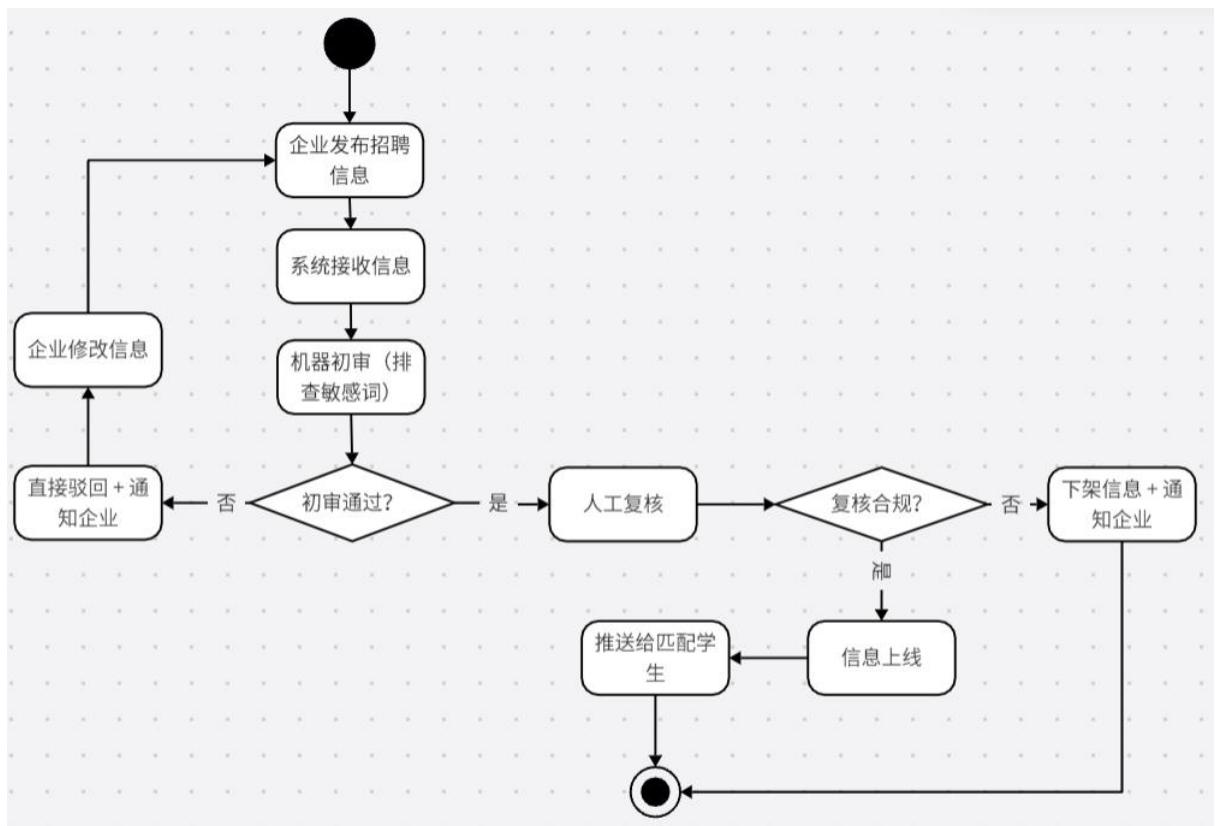


图 4.7 招聘信息审核与发布活动图

4.5.4. 学生发帖流程活动图

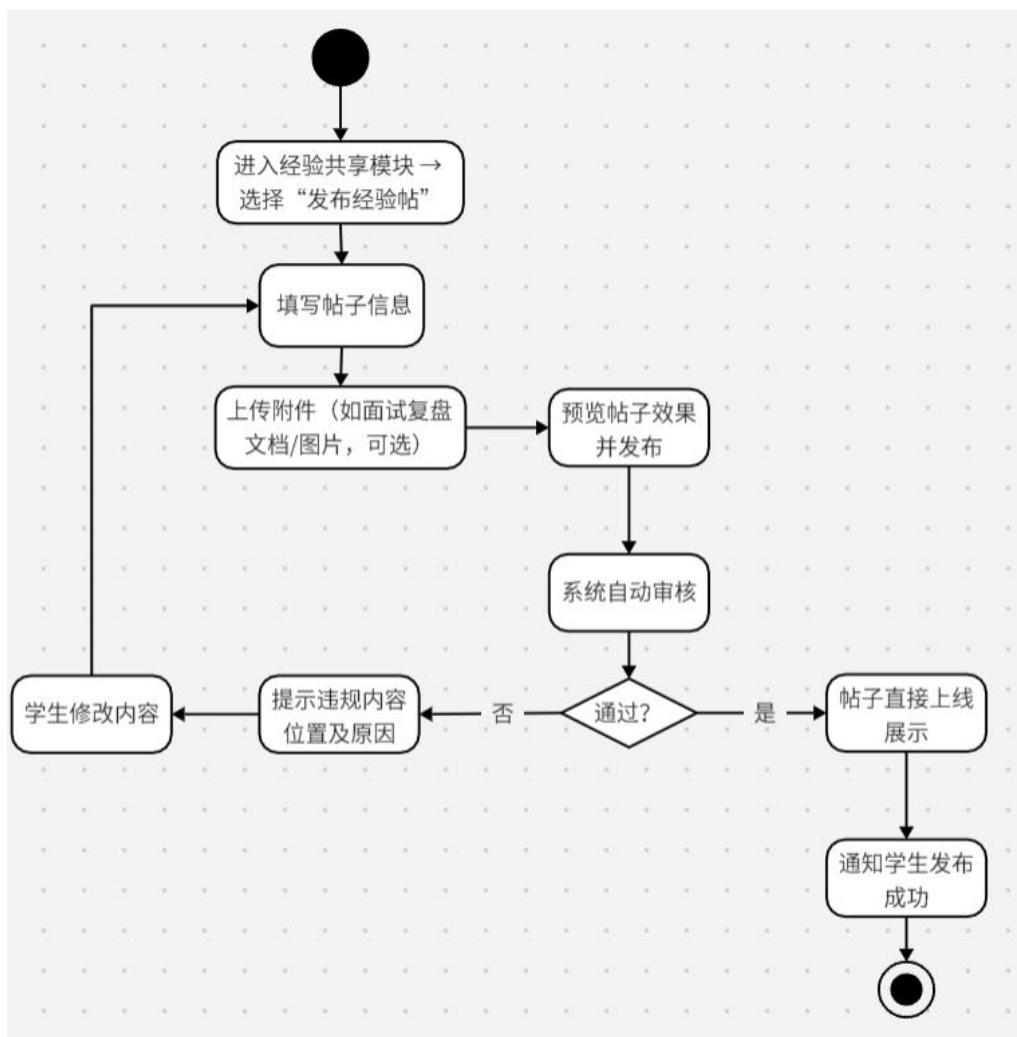


图 4.8 学生发帖流程活动图

4.6. 交互建模

4.6.1. 学生筛选招聘信息时序图

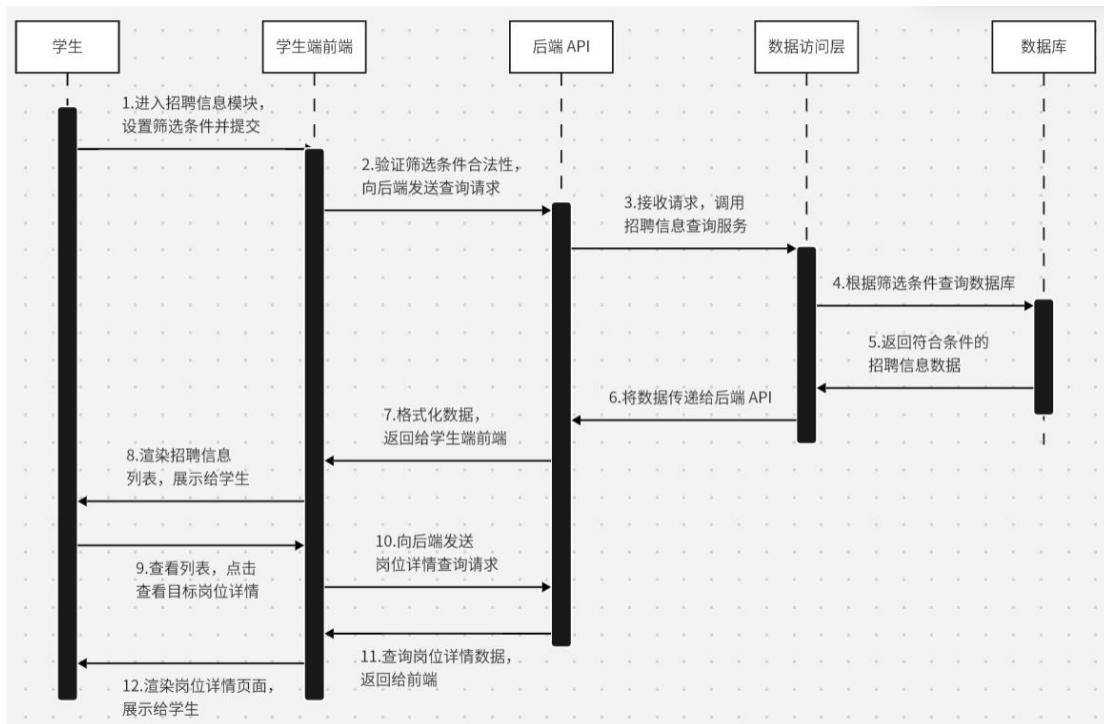


图 4.9 学生筛选招聘信息时序图

4.6.2. 企业查看定向简历时序图

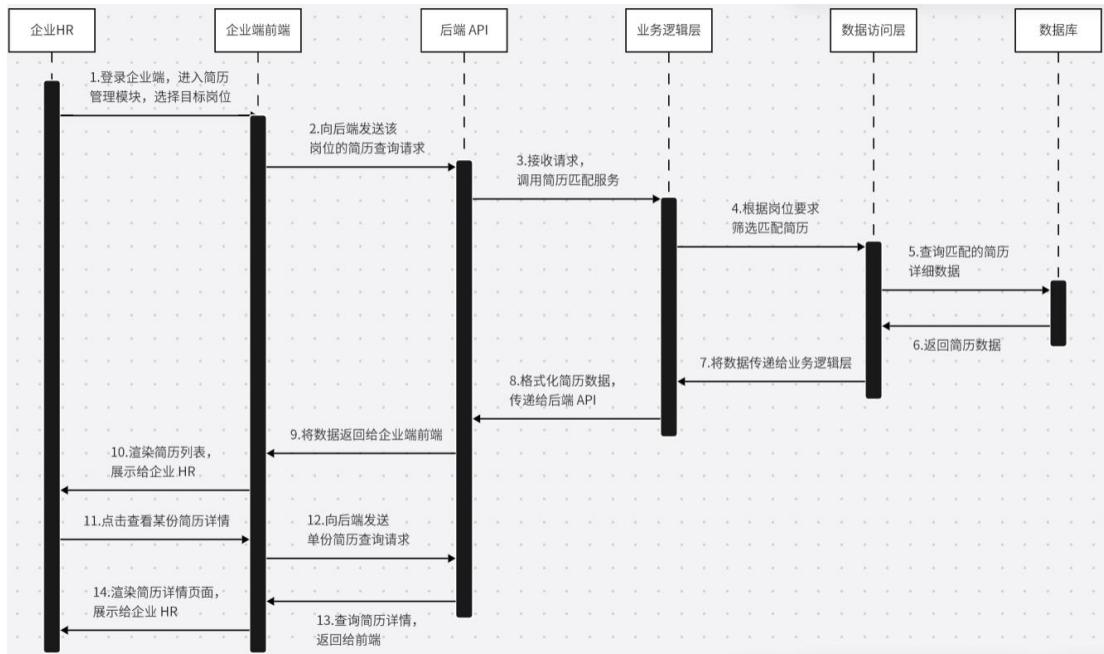
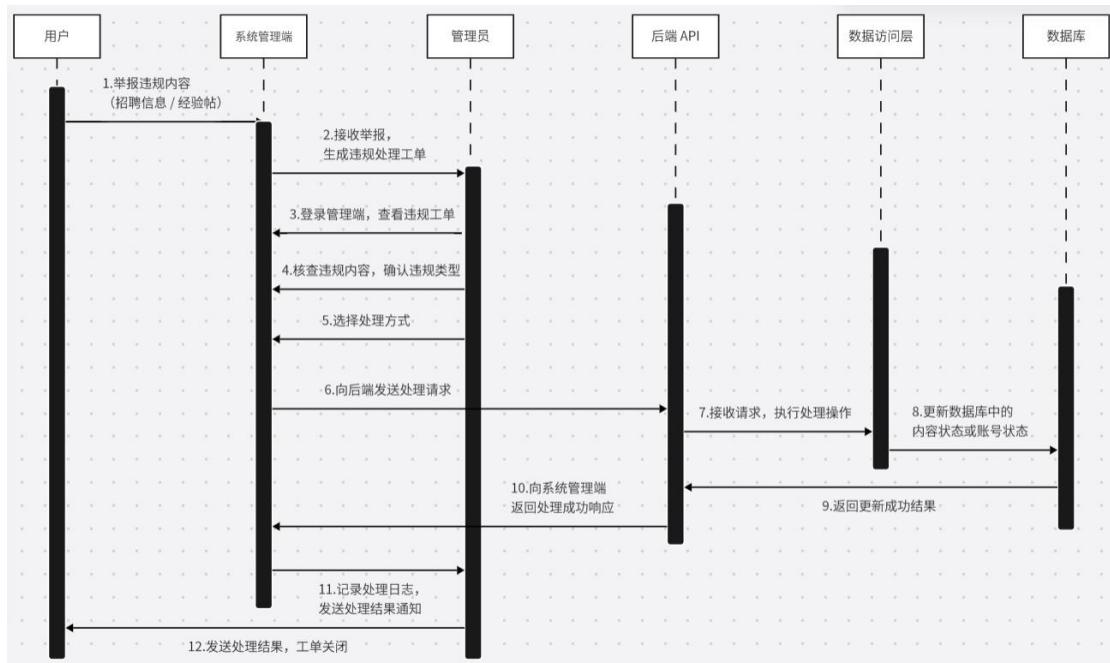


图 4.10 企业查看定向简历时序图

4.6.3. 管理员处理违规内容时序图



4.11 管理员处理违规内容时序图

5. 结论及展望

5.1. 结论

本项目聚焦大学生就业市场的痛点问题，设计了基于 Django 框架的大学生就业信息共享平台整体方案，围绕“学生 - 企业 - 系统管理”三类用户需求构建协同功能体系，采用前后端分离架构设计，结合 Python、MySQL、Vue.js 等技术栈规划技术路径，形成了就业信息高效共享与精准匹配的设计框架。

方案通过学生端简历管理、招聘信息筛选、经验共享等模块的设计，解决学生求职信息碎片化、经验传递壁垒的问题；企业端资质审核、简历匹配、招聘管理等模块的设计，提升企业招聘流程的规范性与效率；系统管理端多维度审核、权限管控、数据统计等模块的设计，保障平台运营的合规性与安全性。整体设计方案逻辑完整、模块划分清晰，兼顾了用户需求与技术可行性，为高校就业服务数字化平台的开发提供了系统化的设计参考。

本设计的创新点在于将“信息共享”与“合规审核”机制深度融合，通过多层级审核流程的设计保障信息质量，同时采用松耦合的前后端分离架构设计，为系统后续扩展预留空间。但受限于设计阶段的研究范围，方案仍存在可优化之处：一是简历与岗位的匹配逻辑设计可进一步深化，以提升匹配精准度；二是经验共享社区的互动功能设计较为基础，缺乏个性化推荐机制的细化设计；三是移动端适配仅完成整体框架规划，具体交互细节有待完善。

5.2. 展望

未来打算从以下方面对设计方案进行优化与拓展：

1. 算法逻辑设计深化：可引入机器学习匹配算法的设计思路，结合学生求职意向、企业招聘偏好等维度，细化简历与岗位的匹配模型设计，提升信息推送的精准性。
2. 功能模块设计扩展：新增视频面试、内推机制、就业指导课程等模块的设计，丰富平台服务场景；优化经验共享社区设计，增加个性化推荐、话题分类、导师答疑等交互功能的细节设计，提升社区活跃度。
3. 交互体验设计优化：深化移动端适配设计，完善响应式布局与操作流程设计，甚至规划独立 APP 的架构设计方案，提升移动场景下的用户体验；简化核心

操作流程的设计，减少冗余步骤，降低用户学习成本。

4. 生态架构设计完善：规划平台与高校就业系统、企业 HR 系统的数据对接方案，设计标准化接口实现数据互通；引入第三方服务集成设计，如背景调查、薪资查询等功能模块的对接架构，完善就业服务生态的设计体系。

6. 参考文献

- [1]林彬,杨彬彬,孙芳.基于 Django 框架的高校就业信息可视化平台的设计与实现[J].辽宁师范大学学报(自然科学版),2022,45(02):201-208.DOI:CNKI:SUN:LNSZ.0.2022-02-009.
- [2]刘洋.基于 WEB 结构学生就业信息平台的设计与实现[J].自动化与仪器仪表,2016,(06):146-148.DOI:10.14016/j.cnki.1001-9227.2016.06.146.
- [3]孙军,刘洋,吴枫,等.基于高校的信息共享平台功能设计与实现[J].项目管理技术,2019,17(09):102-107.DOI:CNKI:SUN:XMGJ.0.2019-09-018.
- [4]朱鹏威,曹烨帆,董天舒.“互联网+”视阈下高校信息化教学资源共享平台建设研究[J].情报科学,2016,34(12):133-136.DOI:10.13833/j.cnki.is.2016.12.025.
- [5]刘昶,李威,王德志.基于 Django 与 Vue 的煤矿企业员工心理健康平台的设计与实现[J].现代计算机,2024,30(21):206-210+216.DOI:CNKI:SUN:XDJS.0.2024-21-039.
- [6]陈峰,斯佳佳.基于 Celery 的国产服务器测试平台设计与实现[J].计算机测量与控制,2024,32(09):80-85.DOI:10.16526/j.cnki.11-4762/tp.2024.09.012.
- [7]胡从寅,杨文远,赵鑫,等.基于 Django+Vue.js 的设计作品交易平台的实现[J].软件,2023,44(11):42-46.DOI:CNKI:SUN:RJZZ.0.2023-11-045.
- [8]陈丽梅.基于微信小程序的高校就业信息平台的设计与实现[J].信息与电脑(理论版),2023,35(07):139-141.DOI:CNKI:SUN:XXDL.0.2023-07-039.