

软件需求规格说明(SRS)



小组名称：_____软工加油队_____

小组成员：____付翔宇、冯彬、孟维汉、翟江浩、杨雨泽_____

项目名称：_____人才招聘系统_____

年 级： ☐ 一年级 ☐ 二年级 ☒ 三年级

山东大学制

目录

软件需求规格说明(SRS)	1
成员分工	4
第一章 引言	5
1.1 定位与目标	5
1.2 对象	5
1.3 系统概述	5
1.4 文档概述	6
1.5 软件需求分析理论	6
1.6 软件需求分析目标	6
1.7 参考文献	7
第二章 需求	7
2.1 需求概述	7
2.1.1 运行环境	7
2.1.2 用户的特点	8
2.1.3 关键点	8
2.1.4 约束条件	10
2.1.5 项目背景	11
2.1.6 需求概述	11
2.1.7 人才招聘系统结构	11
2.1.8 人才招聘系统拓扑图	12
2.2 需求规格	13
2.2.1 软件系统总体功能/对象结构	13
2.2.2 软件子系统功能/对象结构	13
2.2.3 描述约定	15
2.3 系统功能需求	16
2.3.1 用户注册、登录与个人信息管理	16
2.3.2 发布招聘信息、搜索职位与申请职位	17
2.3.3 简历上传、管理与智能解析	17
2.3.4 招聘流程管理与面试安排	18
2.3.5 数据统计、报表生成与导出	18
2.4 接口需求	19
2.4.1 外部接口需求	19
2.4.2 内部接口需求	20
2.4.3 接口标识和接口图	22
2.5 数据需求	24
2.5.1 数据分类与结构	24
2.5.2 数据存储策略	24
2.5.3 数据管理机制	25
2.5.4 数据安全设计	25
2.5.5 数据字典	26
2.5.6 数据库描述	27
2.6 环境需求	27

2.6.1 硬件环境需求	27
2.6.2 软件环境需求	27
2.6.3 网络环境需求	28
2.7 资源需求	28
2.7.1 硬件资源需求	28
2.7.2 软件资源需求	29
2.8 非功能性需求	29
2.8.1 系统响应时间	29
2.8.2 数据处理能力	29
2.9 其他需求	30
2.9.1 设计和实现的约束	30
2.9.2 数据需求	31
2.9.3 操作需求	31
2.9.4 故障处理需求	32
2.9.5 包装需求	32
2.9.6 适应性需求	32
2.9.7 保密性需求	33
2.9.7 保密性和私密性需求	34
第三章 其他规格说明	35
3.1 合格性规定	35
3.2 需求可追踪性	36
3.3 尚未解决的问题	37
3.4 故障处理	38
3.5 算法说明	39
3.6 有关培训需求	40
3.7 有关后勤需求	41
附录	42
Petri 网建模	42

成员分工

组长：付翔宇

账号：FUxiangyu9

任务：① 负责 SRS 整体框架的搭建

② 负责组织组员进行项目需求分析与讨论、总结以及任务分工安排、监督核查落实到位程度

③ 进行需求概述、需求规格、系统功能性需求的详细撰写以及人才招聘系统拓扑图的绘制以及 Petri 网的分析与制作

④ 负责最终文档的排版审核以及定稿

组员：孟维汉

账号：Yourleler

任务：负责接口需求以及非功能性需求的详细撰写

组员：翟江浩

账号：Astesia-6

任务：负责环境需求以及资源需求的详细撰写

组员：冯彬

账号：feng1036

任务：负责接口需求以及数据需求的详细撰写

组员：杨雨泽

账号：noucanp

任务：负责其他需求以及其他规格说明的详细撰写

第一章 引言

1.1 定位与目标

计算机高度发达的今天，利用信息技术对大量复杂的信息进行有效的管理已成为一种普遍而实用的手段。一方面，这极大的减少了线下人力与时间的开销，另一方面，现代计算机强大的计算能力和网络的普遍部署，大大加快了信息的流动与普及。本人才招聘系统旨在为个人用户与企业提供高效、智能的人才招聘解决方案，通过整合先进的技术和人才管理理念，本系统致力于帮助企业与用户之间建立快速岗位匹配的渠道，提高招聘效率，降低用户和企业成本，实现人才与企业的良性互动。

1.2 对象

本《软件需求规格说明书》的预期读者是：

- 程序教学平台开发经理
- 技术部经理
- 项目组所有人员
- 测试组人员
- SQA 人员
- 开发公司授权调阅本文档的其他人员

1.3 系统概述

随着互联网的发展与招聘流程的数字化转型，传统的线下招聘方式已经难以满足企业对高效筛选人才、快速发布职位的需求，同时求职者也希望通过线上渠道更便捷地获取职位信息并进行申请。尤其是在高校中，学生在寻找实习、兼职或毕业就业岗位时，常常面临信息来源分散、职位匹配度低等问题。

本课程设计项目旨在开发一个面向学生、企业与管理层三方用户的人才招募系统，以实现职位发布、简历投递、申请管理等关键功能，帮助用户建立高效的人才供需对接平台。

本系统定位为一个中小型的 **Web** 应用平台，适合用于模拟真实的招聘流程，适用于以下场景：

企业用户发布岗位，筛选候选人；

大学生在线浏览职位、提交简历；

教师或管理员管理系统内容与用户权限；

用于展示软件工程课程中的分析设计、编程实现、功能测试等开发流程。

系统主要作为课程项目实现，功能适度，重点关注结构清晰、流程合理、代码规范与用户体验，既满足教学评估要求，也具备一定的实用性与扩展性。

该系统旨在实现以下几个主要目标：

信息发布与获取高效化：企业用户可在线发布招聘信息，求职者可按分类、关键词等条件筛选岗位；

招聘流程数字化：从职位发布、简历投递、筛选到反馈处理，支持完整的线上招聘闭环；

角色管理明确化：系统根据不同角色（求职者、招聘方、管理员）提供定制化功能，权限分工明确；

数据管理系统化：通过后台管理功能，管理员可对职位与用户数据进行监管，确保系统数据规范性和安全性；

界面交互简洁友好：注重用户体验设计，操作流程直观，利于演示与项目验收。

1.4 文档概述

本文档对实现软件的功能做全面的描述，帮助用户判断实现功能的正确性、一致性和完整性，促使用户在软件设计启动之前周密地、全面地思考软件需求。了解和描述软件实现所需的全部信息，为软件设计、确认和验证提供一个基准，为软件管理人员进行软件成本计价和编软件开发计划书提供依据。

需求分析的具体内容可以归纳为六个方面：软件的功能需求，软件与硬件或其他外部系统接口，软件的非功能性需求，软件的反向需求，软件设计和实现上的限制，阅读支持信息。

软件需求分析应尽量提供软件实现功能需求的全部信息，使得软件设计人员和软件测试人员不再需要需求方的接触。这要求软件需求分析内容应正确、完整、一致和可验证性。此外，为保证软件设计质量，便于软件功能的休整和验证，软件需求表达无歧义性，具有可追踪性和可修改性。

1.5 软件需求分析理论

软件需求分析（Software Requirement Analysis）是研究用户需求得到的东西，完全理解用户对软件需求的完整功能，确认用户软件功能需求，建立可确认的、可验证的一个基本依据。

软件需求分析是一个项目的开端，也是项目实施最重要的关键点。据有关的机构分析结果表明，设计的软件产品存在不完整性、不正确性等问题 80% 以上是需求分析错误所导致的，而且由于需求分析错误造成根本性的功能问题尤为突出。因此，一个项目的成功软件需求分析是关键的一步。

1.6 软件需求分析目标

软件需求分析的主要实现目标：

对实现软件的功能做全面的描述，帮助用户判断实现功能的正确性、一致性和完整性，促使用户在软件设计启动之前周密地、全面地思考软件需求；

了解和描述软件实现所需的全部信息，为软件设计、确认和验证提供一个基准；

为软件管理人员进行软件成本计价和编制软件开发计划书提供依据；

需求分析的具体内容可以归纳为六个方面：软件的功能需求，软件与硬件或其他外部系统接口，软件的非功能性需求，软件的反向需求，软件设计和实现上的限制，阅读支持信息。软件需求分析应尽量提供软件实现功能需求的全部信息，使得软件设计人员和软件测试人员不再需要需求方的接触。这就要求软件需求分析内容应正确、完整、一致和可验证。此外，为保证软件设计质量，便于软件功能的休整和验证，软件需求表达无歧义性，具有可追踪性

和可修改性。

1.7 参考文献

- 【1】莎丽·劳伦斯·弗里格, (加) 乔安妮·M. 阿特利. 软件工程（第四版）. 北京：人民邮电出版社，2024
- 【2】《软件工程基础》 赵一丁 北京邮电大学出版社
- 【3】《软件需求》 劳森 (作者), 刘晓晖 (译者) 电子工业出版社
- 【4】《软件需求工程：原理和方法》 金芝，刘璘，金英 科学出版社
- 【5】《实用软件工程》第三版 殷人昆 清华大学出版社
- 【6】《电子政务发展需求与效益分析》 朱建明 经济科学出版社
- 【7】《电子政务信息系统的规划与建设》 田景熙，洪琢 人民邮电出版社
- 【8】《电子政务信息公平研究》 唐思慧 世界图书出版公司
- 【9】《电子政务系统的需求分析》 甘明鑫，曹菁 机械工业出版社

第二章 需求

2.1 需求概述

2.1.1 运行环境

本系统的运行环境充分考虑了用户的多样性和设备的兼容性，具体如下：

Web 浏览器

系统支持主流的 Web 浏览器，包括 Edge、Chrome、Opera、Safari、Firefox 以及任何支持 HTML5 标准的浏览器。这些浏览器覆盖了市场上绝大多数用户群体，能够确保系统在不同的浏览环境中稳定运行。对于各主流浏览器，建议使用最新版本以获得最佳的用户体验和功能支持。例如，Chrome 和 Edge（基于 Chromium 内核）能够提供高效的 JavaScript 执行性能和良好的兼容性；Safari 则在苹果设备上具有出色的优化效果，确保系统在 macOS 和 iOS 设备上的稳定运行；Firefox 以其开源和高度可定制性，满足不同用户的个性化需求。支持 HTML5 标准意味着系统能够充分利用现代 Web 技术，实现丰富的交互效果和功能特性，如多媒体播放、本地存储、Canvas 绘图等，为用户带来更加生动和便捷的使用体验。

标准分辨率

系统适配多种标准分辨率，主要包括 1024x768、1920x1080 和 2K（2560x1440）。1024x768 分辨率适用于一些较旧的设备或较小屏幕的显示器，确保系统在这些设备上能够正常显示和操作，不会出现布局混乱或内容溢出的问题。1920*1080 是当前主流的全高清分辨率，广泛应用于各种台式机、笔记本电脑和显示器，系统在该分辨率下能够呈现清晰、细腻的界面，合理布局各个功能模块，使用户能够舒适地浏览和操作。2K 分辨率则为高分辨率设备提供

支持，如一些高端显示器和视网膜屏幕，在该分辨率下，系统界面的细节更加丰富，图像和文字显示更加锐利，提升用户的视觉体验。系统采用响应式设计技术，能够根据不同的屏幕分辨率自动调整界面布局 and 元素尺寸，确保在各种分辨率下都能保持良好的用户界面和操作逻辑，无论是在大屏幕的台式机上还是在小屏幕的笔记本电脑上，用户都能获得一致且优质的使用体验。

2.1.2 用户的特点

本产品的用户分为企业用户与个人用户，两类用户具有不同的需求和特点，具体如下：

企业用户

企业用户主要包括各类企事业单位、社会组织等，其核心需求是高效地完成员工招聘工作。企业用户存在相应的员工需求，希望在短时间内将岗位需求投送给大量求职者，以扩大招聘范围，吸引更多的潜在候选人。不同规模的企业需求有所不同，大型企业可能需要招聘多个岗位、大量的人员，对招聘流程的规范性和数据统计分析的需求较高；中小型企业则更注重招聘的效率和成本，希望能够快速找到合适的候选人，降低招聘成本。企业用户在使用系统时，通常会由人力资源部门的工作人员负责操作，他们需要系统提供便捷的招聘信息发布功能，能够详细描述岗位要求、职责、待遇等信息，并支持多种方式将招聘信息推送给求职者，如站内通知、邮件通知、短信通知等。同时，企业用户希望系统能够提供高效的简历筛选和管理功能，能够根据岗位需求对求职者的简历进行智能筛选，快速定位符合条件的候选人，并支持面试安排、录用管理等招聘流程管理功能，帮助企业顺利完成招聘工作。此外，企业用户还可能关注数据统计和报表生成功能，通过对招聘数据的分析，了解招聘效果，优化招聘策略。

个人用户

个人用户主要是有求职需求的各类人群，包括应届毕业生、在职跳槽者、自由职业者等。其核心需求是高效快速地锁定目标岗位，并能够在短时间内完成简历投递以及与企业取得联系。不同类型的个人用户需求也有所不同，应届毕业生可能更关注岗位的发展前景、培训机会等，对简历填写和求职指导的需求较高；在职跳槽者则更注重岗位的薪资待遇、工作地点、职业发展空间等，希望能够快速找到符合自己期望的岗位。个人用户在使用系统时，需要系统提供强大的职位搜索功能，支持多种筛选条件，如行业、职位类型、工作地点、薪资范围等，帮助用户快速定位到合适的岗位。同时，系统应提供便捷的简历上传和在线简历填写功能，支持多种简历格式，如 Word、PDF 等，并提供简历模板和填写指导，帮助用户制作出高质量的简历。此外，个人用户希望能够及时了解企业的反馈信息，如简历是否被查看、是否获得面试机会等，并支持与企业进行在线沟通，方便快捷地与企业取得联系，提高求职效率。

2.1.3 关键点

关键功能

用户注册、登录、个人信息管理

用户注册：支持手机号码、邮箱等多种注册方式，用户需填写基本信息，如用户名、密码、真实姓名、联系方式等。注册过程中，系统会对用户输入的信息进行格式验证和唯一性检查，确保用户信息的准确性和完整性。同时，为了保障用户账户的安全，注册时需设置强

度较高的密码，并支持验证码验证和短信验证等安全措施。

用户登录：用户可以通过注册时使用的手机号码、邮箱或用户名进行登录，支持密码登录和验证码登录等方式。为了提高用户的登录效率和便捷性，系统还支持第三方登录方式，如微信、QQ、支付宝等，用户可以通过授权第三方账号快速登录系统。登录过程中，系统会对用户的身份进行验证，确保只有合法用户才能访问系统资源。

个人信息管理：用户可以在个人中心对自己的个人信息进行管理，包括基本信息、联系方式、教育背景、工作经历、技能证书等。用户可以随时更新和修改自己的个人信息，确保信息的准确性和时效性。系统会对用户的个人信息进行加密存储，保障用户信息的安全。

发布招聘信息、搜索职位、申请职位

发布招聘信息：企业用户登录系统后，可以发布招聘信息，填写岗位名称、岗位类型、行业领域、工作地点、薪资范围、招聘人数、岗位职责、任职要求、福利待遇等详细信息。系统提供丰富的编辑功能，支持上传图片、附件等，帮助企业用户更好地展示岗位信息。发布招聘信息时，系统会对信息进行审核，确保信息的真实性和合法性，避免虚假招聘信息的发布。

搜索职位：个人用户可以通过系统提供的搜索功能查找合适的职位，支持关键词搜索、筛选条件搜索等方式。筛选条件包括行业、职位类型、工作地点、薪资范围、经验要求、学历要求等，用户可以根据自己的需求设置不同的筛选条件，快速缩小搜索范围，找到符合自己期望的职位。系统还支持智能推荐功能，根据用户的简历信息和求职偏好，为用户推荐相关的职位，提高职位搜索的效率和准确性。

申请职位：个人用户找到合适的职位后，可以直接在线申请职位，申请方式包括上传简历和在线填写申请表单。申请职位时，系统会自动检查用户的简历是否符合岗位要求，并提示用户补充相关信息。企业用户收到职位申请后，可以对申请进行处理，如查看简历、发送面试邀请、拒绝申请等，个人用户可以实时查看申请状态，了解自己的求职进展。

简历上传、在线简历填写、简历管理

简历上传：个人用户可以上传本地已有的简历文件，支持 Word、PDF 等常见格式。上传简历时，系统会自动解析简历内容，提取关键信息，如姓名、年龄、学历、工作经历、技能证书等，并自动填充到用户的个人信息中，减少用户的重复填写工作。同时，系统会对简历文件进行病毒扫描，确保文件的安全性。

在线简历填写：系统提供在线简历填写功能，用户可以根据系统提供的简历模板，逐步填写个人信息、教育背景、工作经历、技能证书、项目经验等内容。简历模板丰富多样，用户可以根据自己的求职方向选择合适的模板，填写过程中，系统会提供实时的格式提示和内容建议，帮助用户制作出专业、规范的简历。

简历管理：用户可以在个人中心对自己的简历进行管理，包括查看简历、编辑简历、删除简历、导出简历等操作。用户可以根据不同的求职岗位，制作多份不同版本的简历，针对不同的岗位需求进行优化和调整。系统还支持简历分享功能，用户可以将简历分享给企业用户或其他相关人员，方便求职沟通。

招聘流程管理、面试安排、录用管理等

招聘流程管理：企业用户可以在系统中对招聘流程进行全程管理，包括招聘计划制定、招聘信息发布、简历筛选、面试安排、录用决策等环节。系统提供可视化的招聘流程视图，企业用户可以清晰地了解每个招聘岗位的进展情况，及时发现和解决招聘过程中出现的问题。同时，系统支持招聘流程的自定义配置，企业用户可以根据自己的实际需求，设置不同的招聘流程和审批环节，提高招聘管理的灵活性和效率。

面试安排：企业用户在筛选出合适的候选人后，可以通过系统安排面试，选择面试时间、地点、面试官等信息，并自动发送面试邀请给候选人。候选人收到面试邀请后，可以在线确

认是否参加面试，企业用户可以实时查看候选人的反馈情况。系统还支持面试提醒功能，在面试前通过短信、邮件等方式提醒面试官和候选人，确保面试的顺利进行。面试结束后，面试官可以在系统中记录面试评价和意见，为录用决策提供参考。

录用管理：企业用户在确定录用候选人后，可以通过系统发送录用通知，通知候选人录用的岗位、薪资待遇、入职时间等信息。候选人收到录用通知后，可以在线确认是否接受录用，企业用户可以实时查看候选人的反馈情况。系统还支持入职流程管理，包括入职材料收集、入职培训安排等，帮助企业用户顺利完成新员工的入职手续。

2.1.4 约束条件

时间约束

项目截止时间为学期第 16 周，需要按时完成开发和测试。时间约束对项目开发提出了较高的要求，需要合理安排开发进度，确保每个阶段的任务都能按时完成。由于开发时间有限，可能会面临任务繁重、资源紧张等问题，需要制定详细的项目计划，明确各个阶段的时间节点和任务分工，合理分配人力和物力资源。在开发过程中，需要加强进度管理，定期召开项目会议，及时跟踪项目进展情况，发现问题及时解决，避免出现进度延误的情况。同时，需要合理安排测试时间，确保系统在上线前经过充分的测试，保证系统的稳定性和可靠性。

功能约束

该项目属于 WEB 软件而不是一般的桌面程序，无法实现桌面软件的一些特性。例如，无法直接访问本地硬件设备、无法在离线环境下完全运行等。这意味着在功能设计和实现过程中，需要充分考虑 WEB 软件的特点和限制，避免设计出依赖桌面软件特性的功能。对于需要与本地资源交互的功能，如文件上传下载，可以通过浏览器提供的接口和功能来实现，确保在 WEB 环境下能够正常运行。同时，需要优化系统的性能和用户体验，确保在网络环境下能够快速加载和响应，提供流畅的用户操作体验。

人力约束

开发小组仅 5 人，并且基本没有相关开发的流程经验。人力约束是项目开发过程中面临的重要挑战之一，小组人数较少，且缺乏开发经验，可能会导致开发效率低下、代码质量不高、项目管理困难等问题。为了应对人力约束，需要加强团队成员的培训和学习，提高团队成员的技术水平和开发能力。可以通过组织技术分享会、学习优秀的开源项目、参加在线课程等方式，让团队成员快速掌握相关的开发技术和流程。同时，需要合理分配任务，根据团队成员的技术特长和能力，分配适合的开发任务，充分发挥每个成员的优势。在项目管理方面，需要引入简单有效的开发流程和方法，如敏捷开发方法，提高项目的灵活性和适应性，及时调整开发计划和任务安排，确保项目能够顺利进行。

合作约束

开发小组成员间聚集时间较短，可能会导致沟通不畅、协作效率低下等问题。为了应对合作约束，需要加强团队成员之间的沟通和协作，利用在线协作工具，如项目管理平台、即时通讯工具、代码版本控制工具等，方便团队成员之间的信息共享和交流。定期召开线上或线下会议，讨论项目进展情况、遇到的问题和解决方案，确保团队成员之间保持良好的沟通和协作。同时，需要建立明确的分工和协作机制，明确每个成员的职责和任务，避免出现任务重叠或遗漏的情况，提高团队的协作效率。

2.1.5 项目背景

在计算机技术高度发达的当下，利用信息技术对复杂信息进行高效管理已成为主流手段，其不仅能大幅减少线下人力与时间成本，还能借助计算机强大的计算能力和网络普及性加速信息流动。本人才招聘系统的开发背景源于企业与个人用户在招聘领域的实际需求：企业需要快速匹配岗位与求职者、缩短招聘周期、降低成本，个人用户则期望高效锁定目标岗位、便捷投递简历并与企业建立联系。通过整合先进技术与人才管理理念，系统旨在搭建企业与用户间的快速匹配渠道，提升招聘效率，实现双方良性互动。

2.1.6 需求概述

运行环境：系统支持多类型 Web 浏览器，包括 Edge、Chrome、Opera、Safari、Firefox 及任何支持 HTML5 标准的浏览器；适配多种标准分辨率，如 1024×768、1920×1080、2K 等，确保不同设备上的界面展示效果。

用户特点：系统用户分为两类：企业用户有明确的员工招聘需求，希望快速发布岗位信息、触达大量求职者并完成招聘流程；个人用户以求职为核心诉求，需要高效搜索职位、管理简历并及时与企业沟通。

关键功能：涵盖用户注册 / 登录、个人信息管理、招聘信息发布、职位搜索与申请、简历上传与管理、招聘流程跟踪（面试安排、录用管理）、数据统计与报表生成、权限管理（系统管理员与普通用户分级控制）等。

关键技术：后端采用 Java、Python、Node.js 等技术栈，前端基于 React/Angular/Vue.js 框架开发，数据库支持 MySQL、MongoDB、PostgreSQL，前后端通过 RESTful API 交互。

约束条件：项目开发面临多重限制：时间上需在学期第 16 周前完成开发与测试；功能上受限于 Web 软件特性，无法实现部分桌面程序功能；人力方面开发小组仅 5 人且缺乏流程经验；团队成员聚集时间较短，合作效率需优化。

2.1.7 人才招聘系统结构

前端层：

负责用户交互界面展示，分为个人用户端、企业用户端和管理端。界面设计遵循简洁原则，个人用户端包含首页（岗位推荐）、搜索页（多条件筛选职位）、个人中心（简历管理、投递记录）；企业用户端包含岗位管理页（发布 / 编辑职位）、人才查询页（筛选求职者）、招聘流程页（面试安排、录用状态跟踪）；管理端包含用户审核页（会员资质校验）、数据统计页（招聘趋势分析）。

后端服务层：

通过模块化设计实现业务逻辑，核心模块包括：

认证模块：处理用户注册、登录及权限验证，使用 MD5 和 HMAC256 算法加密密码与 JWT 令牌。

用户模块：管理个人用户的简历信息、求职意向及企业用户的招聘信息、岗位详情。

流程模块：支持招聘全流程管理，如职位发布审核、应聘记录跟踪、面试通知发送（集成邮件 / 短信接口）。

数据模块：提供数据统计分析功能，生成岗位申请趋势、用户活跃度等报表，并支持

CSV/Excel 格式导出。

数据存储层采用混合存储方案：

关系型数据库（MySQL）：存储用户基础信息、岗位详情、应聘记录等结构化数据，通过数据字典定义字段类型（如用户名为 VARCHAR (20)，薪资为 INT）。

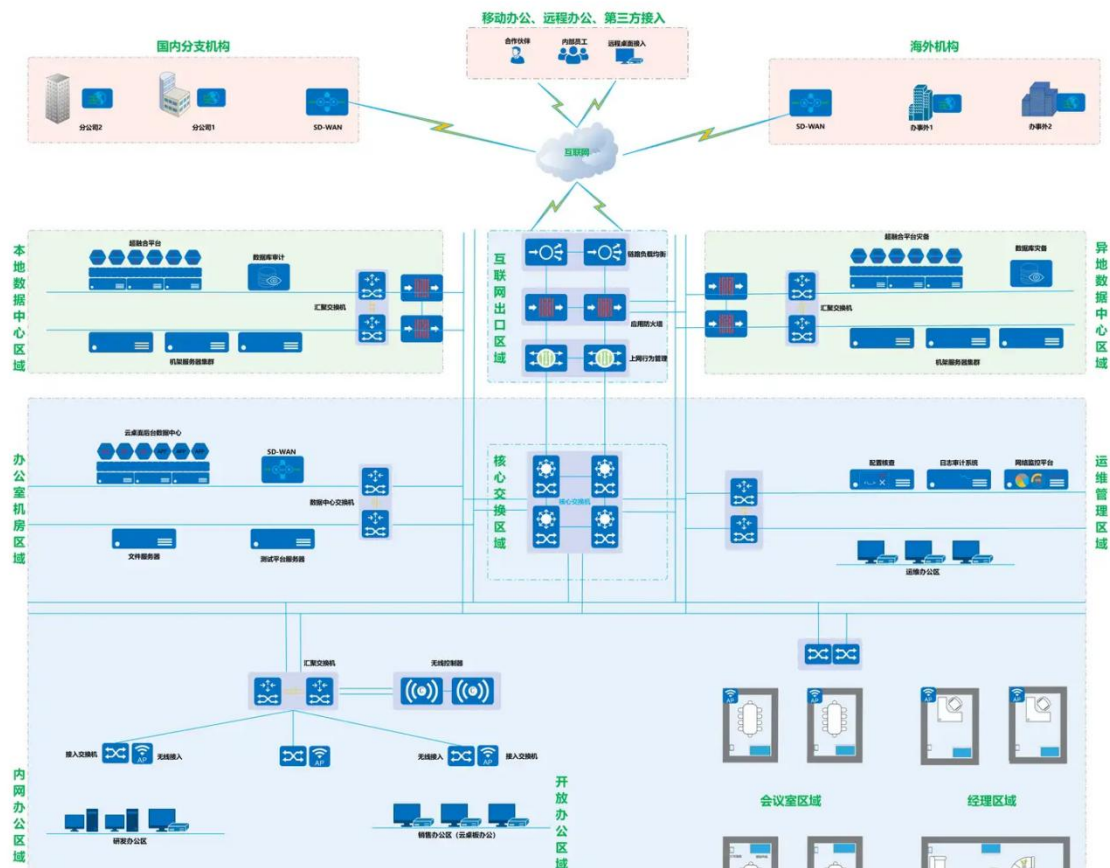
对象存储（OSS）：存储简历 PDF、企业营业执照等非结构化文件，数据库中仅记录文件路径与元数据。

缓存数据库（Redis）：存储用户登录态 JWT 令牌、高频查询结果（如热门岗位列表），提升系统响应速度。

接口交互

前后端通过 RESTful API 通信，例如：个人用户调用 GET /api/positions?keyword=Java 接口搜索职位；企业用户通过 POST /api/jobs 接口发布岗位信息，后端接收参数后校验格式、扫描敏感词并存储至 MySQL；管理员通过 GET /api/stats/recruitment-trends 接口获取招聘趋势数据，后端聚合数据库记录后返回 JSON 格式结果。

2.1.8 人才招聘系统拓扑图



2.2 需求规格

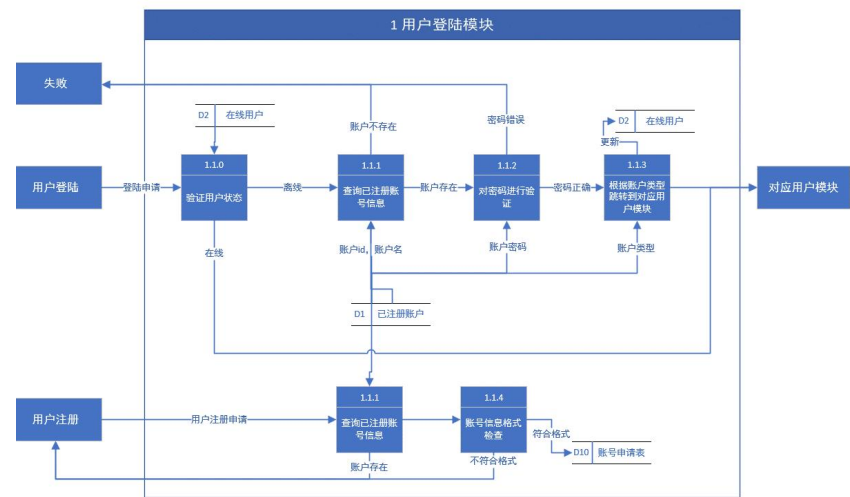
2.2.1 软件系统总体功能/对象结构

系统采用前后端分离架构，前端负责用户交互界面展示（如 Web 页面、移动端适配界面），后端通过 RESTful API 提供业务逻辑处理，分为登录模块、用户管理模块、招聘流程模块、简历解析模块、数据统计模块等核心服务，数据存储层使用 MySQL 数据库、Redis 缓存和阿里云 OSS 文件存储。用户层包含个人用户、企业用户和管理员，通过前端发送 HTTP 请求至后端，后端处理后调用数据存储层完成数据读写。

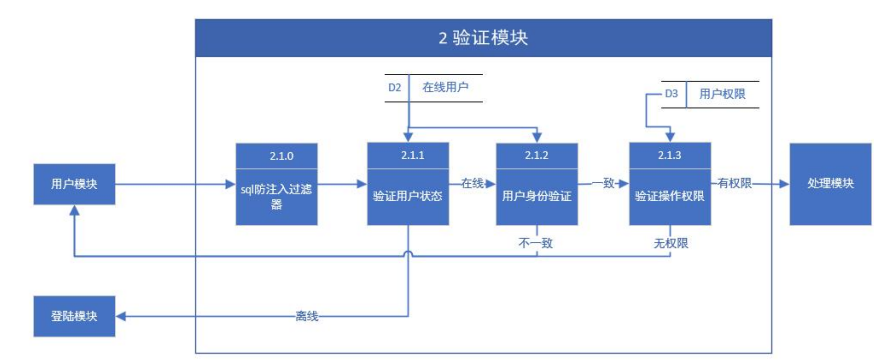
以用户登录流程为例：用户在前端输入用户名、密码及账户类型，前端验证格式后调用后端登录接口；后端查询 MySQL 验证账户存在性并对比密码哈希值，通过后生成 JWT 令牌存储于 Redis，同时更新用户在线状态至 MySQL；前端接收令牌后跳转至对应功能界面，如个人用户进入职位搜索页面，企业用户进入岗位管理页面。此外，系统扩展功能包括基于用户行为数据的推荐引擎模块（每小时更新推荐列表）和多渠道消息通知模块（集成邮件、短信、站内信，异步发送通知）。

2.2.2 软件子系统功能/对象结构

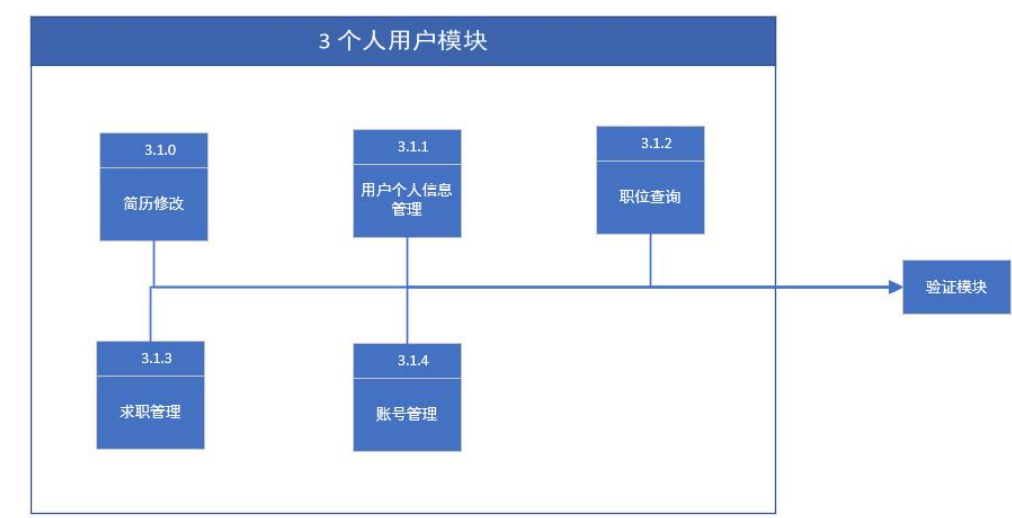
1、登录模块



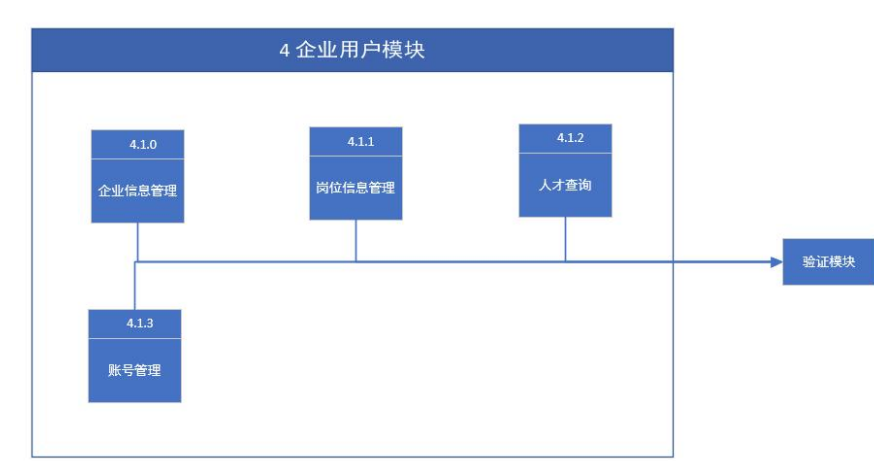
2、验证模块



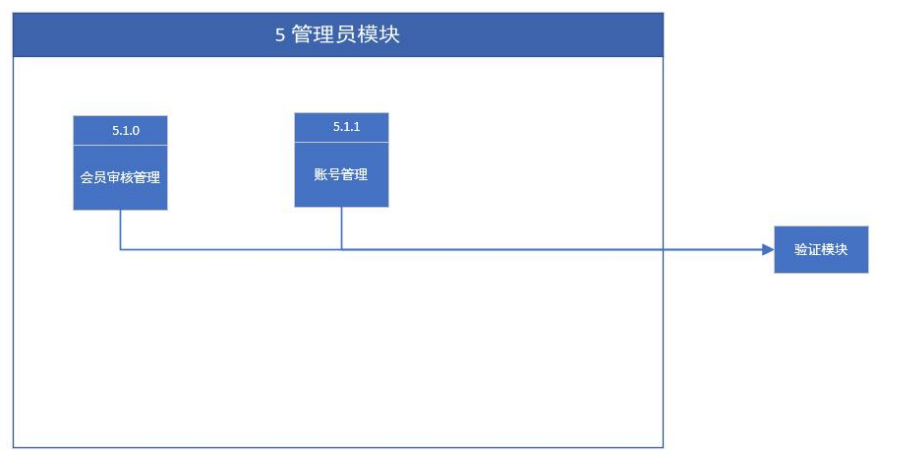
3、个人用户模块



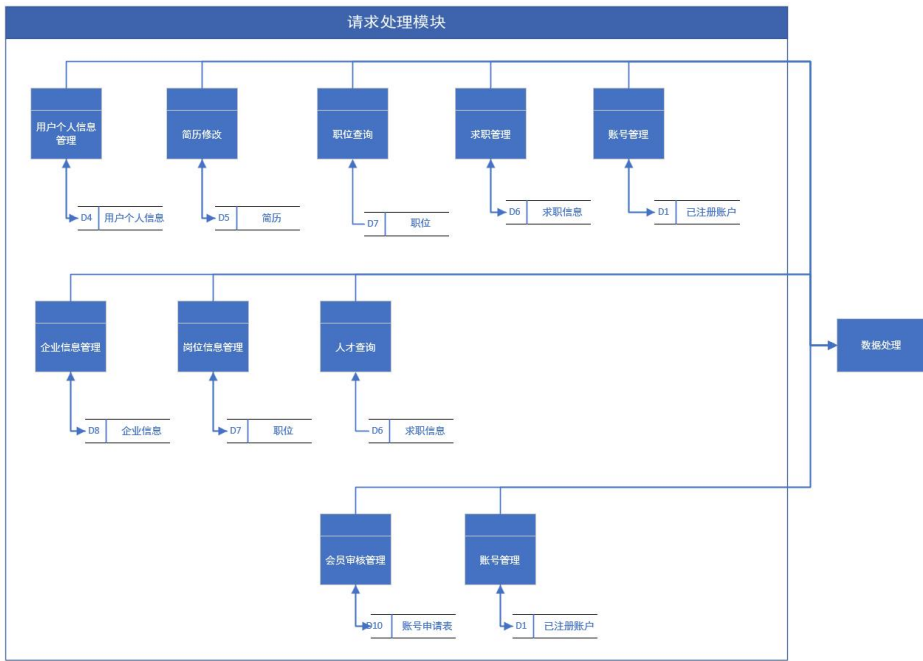
4、企业用户模块



5、管理员模块



6、请求处理模块



2.2.3 描述约定

为确保文档一致性，定义以下规范：

术语定义：明确关键技术术语，如 JWT（JSON Web Token，用于身份验证，包含用户信息、签名及过期时间）、RESTful API（基于 REST 架构的接口设计规范，通过 HTTP 方法操作资源）、OSS（阿里云对象存储服务，存储非结构化数据）、NLP（自然语言处理，用于简历解析和敏感词检测）、BCrypt（密码哈希算法，通过加盐增强安全性）。

符号与缩写规范：模块命名采用“功能域_模块名”格式（如 auth_login），接口命名遵循 RESTful 规范（如 GET /api/users/{user_id} 获取用户信息），数据类型标注明确长度及格式（如 VARCHAR (20) 表示最长 20 位字符串，DATETIME 格式为 YYYY-MM-DD HH:MM:SS）。

需求描述模板: 每条需求按“功能点 - 输入 - 处理 - 输出 - 约束”结构化描述,例如“用户登录” 功能: 输入为用户名、密码、账户类型; 处理包括账户验证、令牌生成、状态更新; 输出分成功(返回令牌及用户信息)与失败(错误码如“账户不存在”); 约束为密码错误尝试 ≤ 5 次, 超次锁定账户 30 分钟。

文档引用规范: 图表引用使用“图 X-X”“表 X-X”格式, 外部标准引用注明来源(如密码复杂度遵循 GB/T 22239-2019《信息安全技术 网络安全等级保护基本要求》)。

2.3 系统功能需求

2.3.1 用户注册、登录与个人信息管理

功能目标: 支持用户账号的创建、安全认证及个人信息维护, 确保数据准确性与隐私安全。

1. 输入

注册场景中, 用户通过前端页面(Web / 移动端)提交注册表单, 单次注册请求包含约 20 个字段(如用户名、密码、邮箱等)。其中用户名需为 6-20 位字符, 支持字母、数字及下划线, 不能包含特殊符号; 密码为 8-16 位, 需包含大小写字母、数字和至少 1 种特殊符号(如!@#\$%); 邮箱需符合 RFC 5322 标准的有效邮箱格式(如 user@domain.com)。登录场景中, 用户输入用户名 / 邮箱和密码, 并选择账户类型(个人 / 企业 / 管理员)。信息修改场景中, 用户在个人中心页面修改字段(如手机号、求职意向)或上传头像 / 简历文件。

2. 处理

注册流程包括: 前端实时验证输入格式(如邮箱有效性), 后端再次校验防止绕过; 查询数据库确认用户名 / 邮箱未被注册(响应时间 $\leq 300\text{ms}$); 使用 BCrypt 算法对密码进行加盐哈希处理(盐值长度 16 字节); 将用户基础信息存入“users”表, 个人 / 企业扩展信息存入对应子表。登录流程包括: 根据输入的用户名 / 邮箱查询账户, 若不存在则返回“账户未注册”; 将输入密码与数据库中哈希值对比, 允许最多 5 次错误尝试, 超过则锁定账户 30 分钟; 验证通过后使用 HMAC-SHA256 算法生成 JWT 令牌, 包含用户 ID、角色、过期时间(默认 2 小时)。信息修改时, 敏感字段(如密码)修改需二次验证(输入原密码或短信验证码), 简历文件上传时自动扫描病毒(调用 ClamAV 引擎), 支持 PDF/Word 格式, 大小限制 $\leq 20\text{MB}$ 。

3. 输出

注册成功后返回用户 ID、默认头像 URL, 并跳转至登录页; 登录成功后返回 JWT 令牌、用户类型及个人信息摘要(如用户名、头像); 信息修改后返回成功状态码(200)及修改后字段(如更新后的手机号)。异常输出包括注册失败时返回错误码“REG-001”(格式错误)、“REG-002”(用户名已存在), 登录失败时返回错误码“LOG-001”(账户不存在)、“LOG-002”(密码错误)、“LOG-003”(账户锁定)。

4. 性能指标

注册 / 登录请求处理时间 $\leq 1.5\text{s}$ (95% percentile), 密码哈希处理耗时 $\leq 800\text{ms}$ (使用硬件加速优化)。

2.3.2 发布招聘信息、搜索职位与申请职位

功能目标：支持企业高效发布岗位、用户精准搜索职位，并实现应聘流程自动化。

1. 输入

发布岗位时，企业用户通过后台管理系统填写岗位详情，单个岗位包含 30+ 字段（如职位名称、薪资、职责、要求等），其中薪资范围需满足最小值 \leq 最大值且符合行业标准（如实习生薪资 \geq 当地最低工资），有效期需为发布时间至截止时间 ≥ 7 天且最长不超过 90 天。搜索职位时，用户在搜索栏输入关键词（如“Java 开发”）或通过筛选条件（地区、薪资、经验）组合查询，单次搜索请求包含 5-10 个筛选参数。申请职位时，用户点击“申请职位”按钮，可选择上传最新简历或使用已有简历。

2. 处理

岗位发布流程包括：调用自然语言处理（NLP）接口扫描敏感词（如“996”“狼性文化”），人工审核不通过则返回修改建议；将岗位信息存入“positions”表，并更新企业用户的“在招岗位数”统计字段。职位搜索流程包括：使用 Elasticsearch 实现全文检索，支持模糊匹配（如“后端”匹配“后端开发”“Java 后端”）；根据相关性、发布时间、薪资高低等维度对结果排序，支持分页（每页显示 10 条）；基于用户求职意向、浏览历史，通过协同过滤算法生成个性化推荐列表（更新频率 ≤ 1 小时）。申请职位时，先检查用户是否已投递该岗位（避免重复申请）及岗位是否仍在有效期内，若通过则生成应聘记录（“applications”表），状态标记为“待筛选”，并向企业发送通知（邮件 / 系统消息）。

3. 输出

发布成功后返回岗位 ID、预览链接，岗位状态显示“发布中”；搜索结果返回岗位列表（含职位名称、企业名、薪资、距离等）、总条数及分页信息；申请成功后返回应聘记录 ID，岗位状态更新为“已申请”。异常输出包括发布失败时返回错误码“JOB-001”（敏感词违规）、“JOB-002”（有效期格式错误），搜索无结果时返回提示“未找到符合条件的岗位，建议放宽筛选条件”。

4. 性能指标

岗位发布审核平均耗时 ≤ 4 小时（自动审核 + 人工审核），搜索响应时间 $\leq 800\text{ms}$ （90% percentile），支持 1000+ 并发搜索请求。

2.3.3 简历上传、管理与智能解析

功能目标：支持用户高效管理简历，实现简历内容的结构化解析与精准匹配。

1. 输入

上传简历时，用户可上传 PDF/Word 文件或使用在线简历编辑器填写信息，单个文件大小 $\leq 20\text{MB}$ ，在线填写表单包含 50+ 字段（教育、工作、项目经验等）。简历管理场景中，用户可编辑简历字段、标记默认简历、删除旧版本简历。

2. 处理

文件解析流程包括：使用 Apache Tika 库将 PDF/Word 转为文本，识别准确率 $\geq 95\%$ ；通过正则表达式与 NLP 模型（如 spaCy）解析关键信息，包括姓名、手机号、邮箱（必填项）、教育背景（学校名称、专业、毕业时间）、工作经历（公司名称、职位、在职时间、职责描述）；自动纠正常见格式错误（如日期格式统一为 YYYY-MM-DD）。简历匹配时，计算简历与岗位的相似度得分（0-100 分），匹配维度包括关键词匹配（岗位要求 vs 简历内容）、经验匹配（工作年限、行业相关性）、教育匹配（学历、专业对口度），结果用于企

业筛选简历时的优先级排序。

3. 输出

上传成功后返回简历 ID、解析后的结构化数据摘要（如姓名、最近工作经历）；简历管理操作后返回更新后的简历列表，包含版本号、创建时间、匹配得分最高的岗位推荐。异常输出包括解析失败时返回错误码“RES-001”（文件损坏）、“RES-002”（不支持的格式），匹配失败时返回“暂无匹配岗位，建议完善简历信息”。

4. 性能指标

简历解析耗时 $\leq 20s$ / 份（文件大小 $\leq 10MB$ ），批量解析（一次处理 10 份简历）耗时 ≤ 3 分钟。

2.3.4 招聘流程管理与面试安排

功能目标：帮助企业高效管理应聘流程，实现面试全流程数字化。

1. 输入

筛选简历时，企业 HR 在后台查看简历列表，标记“通过”“拒绝”或“待沟通”。安排面试时，HR 选择应聘记录，填写面试时间、地点（支持线上 / 线下）、面试官信息。录用管理场景中，HR 发送录用通知，包含薪资 offer、入职时间、体检要求等。

2. 处理

招聘流程通过状态机管理，应聘记录状态流转为：待筛选 \rightarrow 已沟通 \rightarrow 面试中 \rightarrow 已录用 / 已拒绝，状态变更时自动触发通知（如“面试通过”时向用户发送短信，“录用拒绝”时发送邮件说明原因）。安排面试时，检查面试官日程（对接企业日历系统），避免时间冲突，支持自动生成面试邀请链接（如腾讯会议 ID），并添加至用户 / 面试官的日历。

3. 输出

筛选结果更新应聘记录状态，用户端显示“已查看”“待面试”等标签；面试通知通过邮件 / 短信发送，内容包含面试时间、链接、注意事项，系统内生成面试日程卡片；录用通知以电子 offer（PDF 格式）形式发送，用户需在 3 个工作日内确认接受或拒绝。异常输出包括流程阻塞时返回错误码“PROC-001”（面试官时间冲突）、“PROC-002”（用户未在有效期内确认 offer）。

4. 性能指标

批量处理 100 份简历筛选耗时 ≤ 5 分钟，面试通知发送延迟 ≤ 10 秒（通过消息队列异步处理）。

2.3.5 数据统计、报表生成与导出

功能目标：为企业和管理员提供多维度数据洞察，支持决策分析。

1. 输入

统计查询时，用户选择统计维度（如时间范围、行业、岗位类型），触发报表生成请求，支持查询近 1 年数据，单次请求涉及百万级记录聚合。导出时用户选择导出格式（CSV/Excel/PDF），设置字段过滤条件。

2. 处理

数据聚合使用 MySQL 窗口函数或 Spark SQL 实现分组统计，例如按月份统计企业招聘转化率（录用人数 / 申请人数）、按地区分析用户活跃度（注册量、简历投递量）。报表生成时动态生成可视化图表（柱状图、折线图、饼图），嵌入报表 PDF 中，支持自定义报

表模板（如企业品牌色、LOGO 展示）。数据导出采用分片下载（每片 10MB）避免内存溢出，敏感数据导出时隐去用户手机号中间四位、身份证号等信息。

3. 输出

统计报表以 HTML 格式返回，支持在线预览、缩放、打印；导出文件返回下载链接（有效期 1 小时），文件名称包含统计时间范围（如“2025 年 Q2 招聘报表.csv”）。异常输出包括导出失败时返回错误码“REPT-001”（数据量过大，建议缩小时间范围）、“REPT-002”（格式不支持）。

4. 性能指标

百万级数据统计耗时≤5 分钟，报表生成耗时≤2 分钟，大文件（≥100MB）导出成功率≥99%。

如需进一步调整格式（如字体、段落间距等），可在 Word 中通过“开始”选项卡的“样式”功能快速优化。

2.4 接口需求

2.4.1 外部接口需求

1. 用户接口

系统主要分为两个交互界面：客户端界面和后台管理界面。

客户端界面主要面向普通用户，包括求职者和企业招聘人员。该界面以简洁、清晰、易于操作为核心设计原则。首页主要展示系统的功能入口，如热门职位推荐、企业展示、注册/登录等内容。搜索页面提供关键词检索、行业分类、职位类型、地区等多维筛选功能，支持分页浏览职位信息。个人中心包含用户信息修改、简历上传与编辑、申请记录查看等模块，所有操作都应具有明确的提示和反馈机制，提升用户操作体验。整个客户端应在不同设备上（如电脑、平板、手机）具备良好的响应式兼容性，确保移动端也能顺利使用系统主要功能。

后台管理界面主要为管理员用户设计，用于对平台数据进行管理和监控。整体界面风格以“管理效率优先”为原则，采用左右分栏布局，左侧为功能模块导航栏，如“用户管理”、“职位审核”、“系统统计”等，右侧为数据操作区域，清晰展示列表、详情、操作按钮等。所有数据支持按条件筛选、批量操作，并提供必要的编辑、启用、禁用等管理功能。操作过程中应提供权限校验和操作日志记录，便于后期审计。

2. 硬件接口

为了保障系统的正常运行和稳定性能，服务器端应具备最低的硬件配置要求。系统需运行在具有不低于 512MB 内存、233MHz 主频 CPU 的设备上，且可用磁盘空间不少于 1.5GB，以满足数据库、日志文件、上传文件等基本存储需求。为了后期扩展和多人同时访问，建议部署在具备更高性能的硬件上，例如具备至少 2GB 内存、双核处理器、20GB 可用空间的云服务器或物理主机。客户端使用设备只需支持常见现代浏览器即可，无需额外硬件依赖。

3. 软件接口

本系统依赖于多种软件环境和第三方组件。服务器端操作系统建议使用 **Microsoft Windows 10** 或 **11**，也可选择兼容的 **Linux** 发行版。系统需安装并配置后端运行环境，例如使用 **Spring Boot** 的项目需要配置 **Java JDK 17** 及 **Maven**；若采用 **Node.js** 开发后端服务，则需安装 **Node.js 16** 及以上版本。

前端系统需要运行在支持 **ECMAScript 6** 标准的浏览器中，推荐使用 **Google Chrome**、**Microsoft Edge** 或 **Mozilla Firefox**。页面渲染与交互依赖现代浏览器支持的 **DOM**、**CSS3**、**ES6** 语法，部分页面可使用 **Vue.js**、**React** 等框架进行构建。

数据库方面，系统后端依赖 **MySQL 8.0** 或以上版本，需支持 **InnoDB** 引擎、事务操作、全文索引等特性。在服务器环境中，推荐安装 **Nginx** 或 **Apache** 作为反向代理服务器，用于托管静态资源和转发接口请求。

如系统需部署在校园服务器或公有云平台上，还应确保具备基本的运行权限、数据库连接权限与网络访问能力。

4. 通信接口

如果系统采用前后端分离架构，前端与后端之间将通过标准 **HTTP/HTTPS** 协议进行通信，接口遵循 **RESTful** 风格设计。所有接口的数据传输均采用 **JSON** 格式，接口地址具备统一前缀，并通过 **URL** 路径区分不同资源类型和操作。

接口访问需要基于身份认证机制，用户登录成功后获取的令牌（如 **JWT**）将用于后续请求的身份验证。系统应实现统一的异常处理机制，前端根据状态码与提示信息进行用户反馈。

举例来说，前端通过 **GET** 请求获取职位列表，**POST** 请求上传简历信息，**PUT** 请求修改用户资料，**DELETE** 请求撤回申请。对于管理端，还需实现数据分页、关键词搜索、状态筛选等接口功能。为支持接口开发与调试，系统使用 **Swagger/OpenAPI** 工具生成接口文档，便于开发团队理解和对接。

所有接口均应具备权限校验机制，不同用户角色仅能访问其对应范围的资源。例如，普通用户无法访问管理员接口，企业用户无法编辑其他企业发布的职位信息等。

2.4.2 内部接口需求

本节主要描述系统内部各模块之间的接口形式与交互机制，确保各功能模块协同工作、数据流通顺畅。系统采用模块化设计理念，将各功能划分为若干逻辑子模块，模块之间通过接口进行调用与数据交换。以下内容从模块边界、调用方式、接口约定等方面对内部接口需求进行详细说明。

1. 模块通信需求

系统采用逻辑分层与功能模块化的设计原则，不同模块之间通过明确的接口进行数据传输与功能调用。系统应定义清晰的接口边界，以保证：各模块之间松耦合、高内聚；任何一个模块的更改不会对其他模块造成影响；各模块具备独立测试和部署的能力。

2. 接口调用需求

1) 用户管理模块接口需求：

应为其他模块提供用户身份验证与用户信息查询服务；
接口需支持通过用户 ID 查询用户基本资料（如姓名、角色、联系方式等）；
对于登录后操作，应支持通过会话令牌或身份凭证校验用户身份有效性；
应支持根据角色类型（求职者、企业、管理员）返回权限信息，以供功能授权使用。

2) 职位管理模块接口需求：

应向外提供职位数据读取接口，支持根据职位 ID 查询详细信息；
支持提供职位列表查询接口，允许根据关键词、公司名、职位类别等条件进行筛选；
支持供企业用户更新职位信息，如发布新职位、编辑职位要求、设置截止时间等；
应提供职位状态接口，判断职位是否仍可被投递简历；
向申请处理模块提供职位有效性与可用性校验服务。

3) 简历管理模块接口需求：

应提供上传、编辑、删除简历的能力，并支持设置默认简历；
应支持供申请模块查询指定用户的简历列表及默认简历 ID；
应提供根据简历 ID 查询详细内容的接口；
接口需支持权限校验，仅简历所属用户或管理员可访问。

4) 申请处理模块接口需求：

应接收投递请求，处理求职者向职位的简历投递；
应支持查询某一用户的投递历史、某职位收到的投递记录；
应支持对申请状态进行更新，如“已查看”、“待面试”、“已拒绝”等；
应向通知模块提供事件回调，如“投递成功”、“申请状态变更”等。

5) 后台管理模块接口需求：

应具备统一查询所有用户、职位、申请记录等系统数据的接口；
应支持管理员对数据的修改权限，包括用户禁用、职位下架、数据清理等；
应支持系统参数的配置、查看日志记录、查询用户行为等管理功能；
应具备角色识别与权限校验接口，保证仅有管理员可访问敏感功能。

6) 消息通知模块接口需求：

应对接简历投递、职位状态变更、申请反馈等关键操作，触发相应消息；
应支持向指定用户发送系统站内消息，并存储历史消息记录；
可扩展支持电子邮件或短信等其他消息形式；
应提供用户消息列表查询及未读消息统计接口。

3. 接口交互需求说明

所有接口必须具备明确的输入和输出规范，数据字段结构需统一标准；
接口应支持分页查询、大数据量处理和多条件筛选，以满足前端用户的使用需求；
各模块调用接口时必须进行权限验证，防止未授权访问敏感信息；
各接口应具备错误处理能力，能够在发生异常时返回详细的错误信息和错误码；
模块间接口应具备稳定性和向后兼容性，避免因版本升级导致接口不可用；
高频调用接口应具备一定的调用限制（如限流）机制，以保障系统整体稳定性。

4. 接口安全性需求

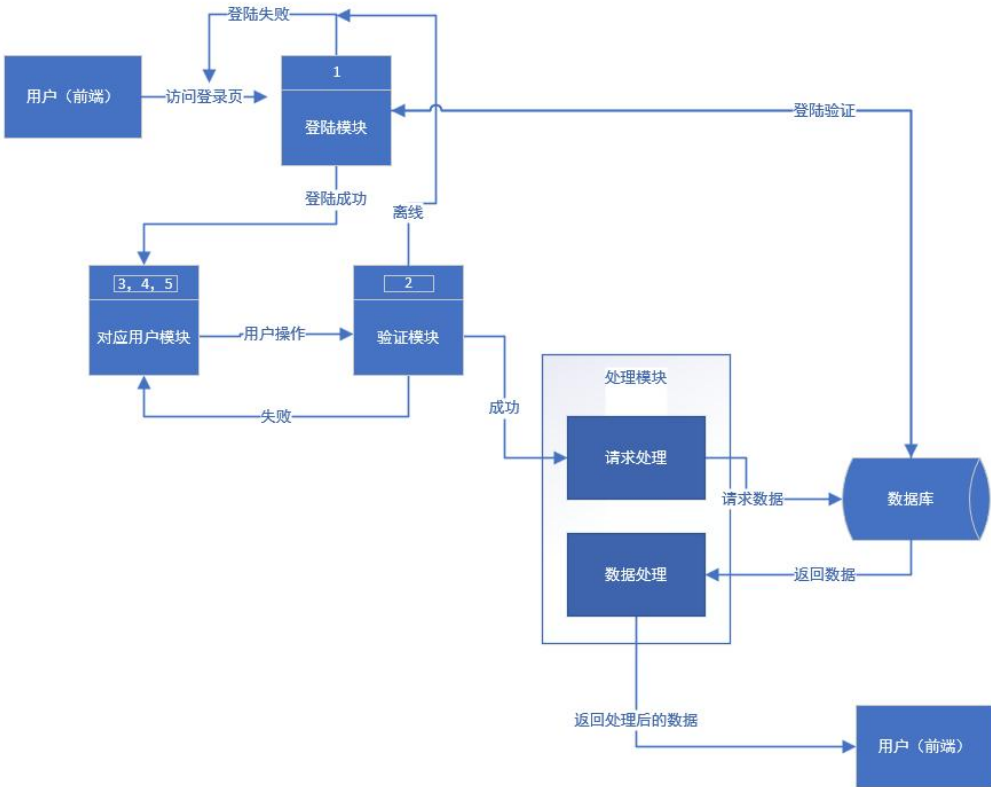
所有内部接口必须经过用户认证机制验证身份；
模块间接口调用应通过受控通道（如内部服务网络），避免接口暴露到公网；
接口传输的数据应在必要时进行加密，保护用户隐私与系统数据安全；
日志系统应记录模块间接口调用行为，以备审计与安全分析。

5. 接口性能要求

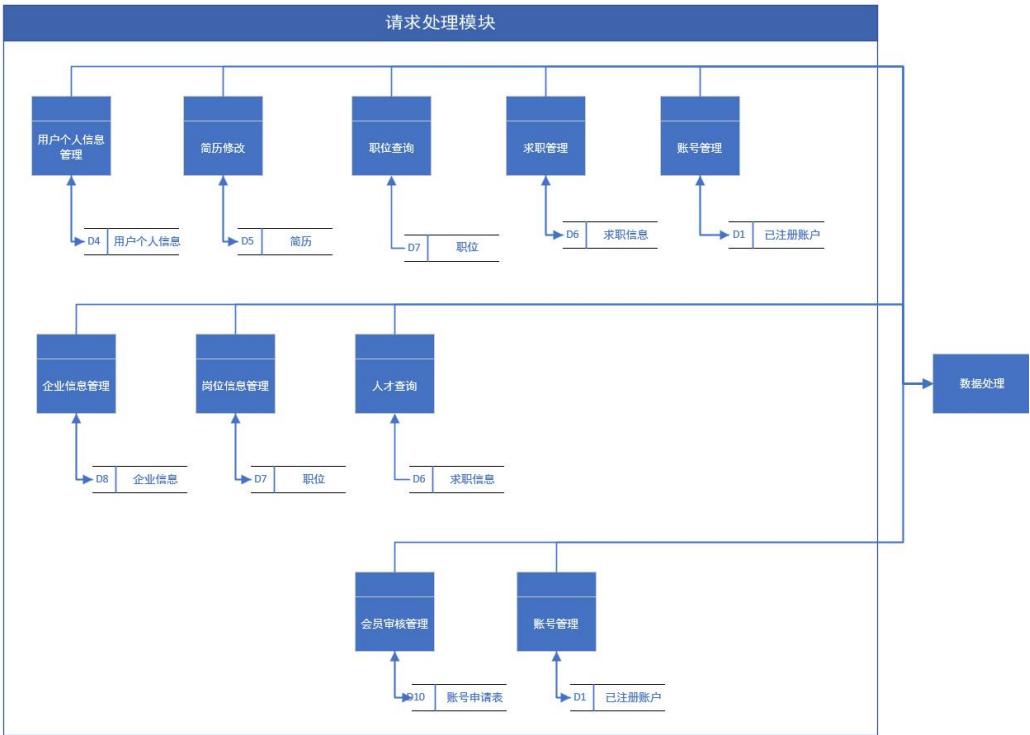
接口响应时间应控制在合理范围（如查询类接口在 200ms 内返回结果）；
数据访问类接口需支持缓存策略以提高响应效率；
高并发场景下的接口（如职位查询）应支持横向扩展；
所有接口应具备服务异常降级机制，保障核心功能可用性。

2.4.3 接口标识和接口图

- 1、接口优先级
- 2、本系统全局接口视图如下



3、 数据处理集中依赖于请求处理模块



2.5 数据需求

2.5.1 数据分类与结构

系统内部数据分为 静态数据、动态数据 和 元数据，各类数据的具体结构与用途如下：

1. 静态数据指系统运行过程中基本不变或极少变更的数据，用于支撑基础业务逻辑。

个人用户：包含用户名、账户 ID、密码（加密存储）、邮箱、手机号、头像、简介、简历文件路径、求职意向（行业、职位类型、薪资期望）、教育背景（学历、毕业院校、专业）、工作经历（公司名称、职位、在职时间）等。

企业用户：包含企业名、企业 ID、密码（加密存储）、邮箱、联系电话、简介、头像、营业执照文件路径、招聘负责人信息（姓名、职位、联系方式）、企业规模、所属行业等。

管理员：包含管理员 ID、用户名、密码（加密存储）、邮箱、手机号、权限等级（超级管理员 / 普通管理员）等。

岗位信息：包含岗位 ID、企业 ID、岗位名称、工作地点、薪资范围、招聘人数、岗位职责、任职要求、发布时间、有效期、申请人数统计等。

简历信息：包含简历 ID、用户 ID、简历名称、创建时间、更新时间、简历内容（文本摘要或 PDF 路径）、求职状态（已投递 / 已查看 / 已面试 / 已录用）等。

2. 动态数据指随用户操作或系统运行实时变化的数据，用于记录业务流程状态。

用户操作日志：包含操作时间、用户 ID、操作类型（登录 / 注册 / 修改信息 / 投递简历等）、操作对象（如岗位 ID、简历 ID）、IP 地址等。

招聘流程数据：包含应聘记录（用户 ID、岗位 ID、投递时间、简历版本）、面试安排（面试 ID、应聘记录 ID、面试官 ID、时间、地点、状态）、录用通知（录用 ID、应聘记录 ID、录用时间、薪资 offer）等。

系统交互数据：包含搜索关键词（用户搜索岗位的关键词）、推荐算法日志（用户对岗位的点击 / 收藏 / 申请行为数据）、数据统计临时结果（如岗位申请趋势、用户活跃度分析）等。

3. 元数据指描述数据属性的数据，用于数据管理与维护。

数据字典元数据：包含字段名称、数据类型、长度限制、默认值、是否必填、业务含义（如“薪资”字段的单位为“元 / 月”）等。权限元数据：包含角色（个人用户 / 企业用户 / 管理员）、操作权限（如企业用户可发布岗位，管理员可审核会员）、数据访问范围（如管理员可查看所有用户数据，企业用户仅查看自有岗位数据）等。

2.5.2 数据存储策略

系统采用 分层存储架构，根据数据类型与访问频率选择不同存储方案：

核心业务数据（高访问频率）存储介质：关系型数据库（MySQL 8.0），支持事务一致性和复杂查询。

表结构设计：用户表（users）：存储用户基础信息，含个人用户、企业用户、管理员的公共字段。

个人用户扩展表（personal_users）：存储个人用户特有的求职意向、教育背景等。

企业用户扩展表（enterprise_users）：存储企业规模、行业、营业执照等信息。

岗位表（positions）：与企业表通过外键关联，记录岗位详情。

应聘记录表（**applications**）：关联用户表与岗位表，记录投递行为与流程状态。

索引优化：对高频查询字段（如用户 ID、岗位 ID、投递时间）创建索引，提升查询效率。

非结构化数据（文件存储），存储介质：阿里云 OSS 对象存储，用于存储简历 PDF、营业执照图片、用户头像等。访问方式：通过 URL 直接访问，数据库中存储文件路径与元数据（如文件类型、大小、上传时间）。

缓存数据（热数据），存储介质：**Redis** 缓存数据库，用于存储用户登录态（JWT 令牌）、高频查询结果（如热门岗位列表）、推荐算法临时数据等。

过期策略：设置合理 TTL（生存时间），如登录令牌有效期为 2 小时，热门岗位缓存有效期为 10 分钟。

历史数据（低频访问），存储介质：分布式文件系统（如 HDFS）或归档数据库（如 PostgreSQL），定期将超过 6 个月的操作日志、已结束招聘流程数据归档。

归档策略：按年 / 月分区存储，支持快速检索与恢复。

2.5.3 数据管理机制

数据生命周期管理创建：用户注册 / 发布岗位时生成新数据，通过前端校验（如邮箱格式、密码复杂度）与后端验证（如唯一用户名检查）确保数据合法性。

更新：用户修改信息或管理员审核时触发数据更新，记录版本变更（如简历版本号递增），保留历史版本供追溯。

删除：用户注销账户或企业删除岗位时，执行逻辑删除（标记为“已删除”状态，保留数据供审计），物理删除需经管理员二次确认（如保留 7 天后自动清除）。

数据一致性保障事务控制：在涉及多表更新的场景（如用户投递岗位时更新应聘记录与岗位申请人数），使用数据库事务确保操作原子性。

分布式锁：对高并发场景下的共享数据（如岗位剩余招聘人数），通过 **Redis** 分布式锁避免超卖问题。

定时对账：每日凌晨运行数据对账脚本，校验数据库与 OSS 存储的文件元数据一致性，生成差异报告并自动修复。

数据查询与统计基础查询：支持按用户 ID、岗位 ID 等主键快速查询，响应时间需≤500ms。

2.5.4 数据安全设计

数据加密

传输层：使用 **HTTPS** 协议加密数据传输，防止中间人攻击。

存储层：

敏感字段（如密码、银行卡号）采用 **AES-256** 加密算法存储，加密密钥定期轮换。

简历 PDF 等文件在 OSS 存储中启用服务器端加密（SSE），支持客户主密钥（CMK）管理。访问控制角色权限：

个人用户：仅能访问 / 修改自身数据，可查看投递岗位的处理进度。

企业用户：可管理本企业的岗位与招聘流程，禁止访问其他企业数据。

管理员：按权限等级分配数据访问范围（如超级管理员可查看全量数据，普通管理员仅能审核会员）。

细粒度权限：通过 **RBAC**（角色 - Based Access Control）模型，对每个操作（如查看、修改、

删除）进行权限控制，例如：

企业招聘负责人可编辑本企业岗位，普通员工仅能查看。

管理员仅能删除违规用户数据，无法修改正常用户的核心信息（如账户 ID）。

2.5.5 数据字典

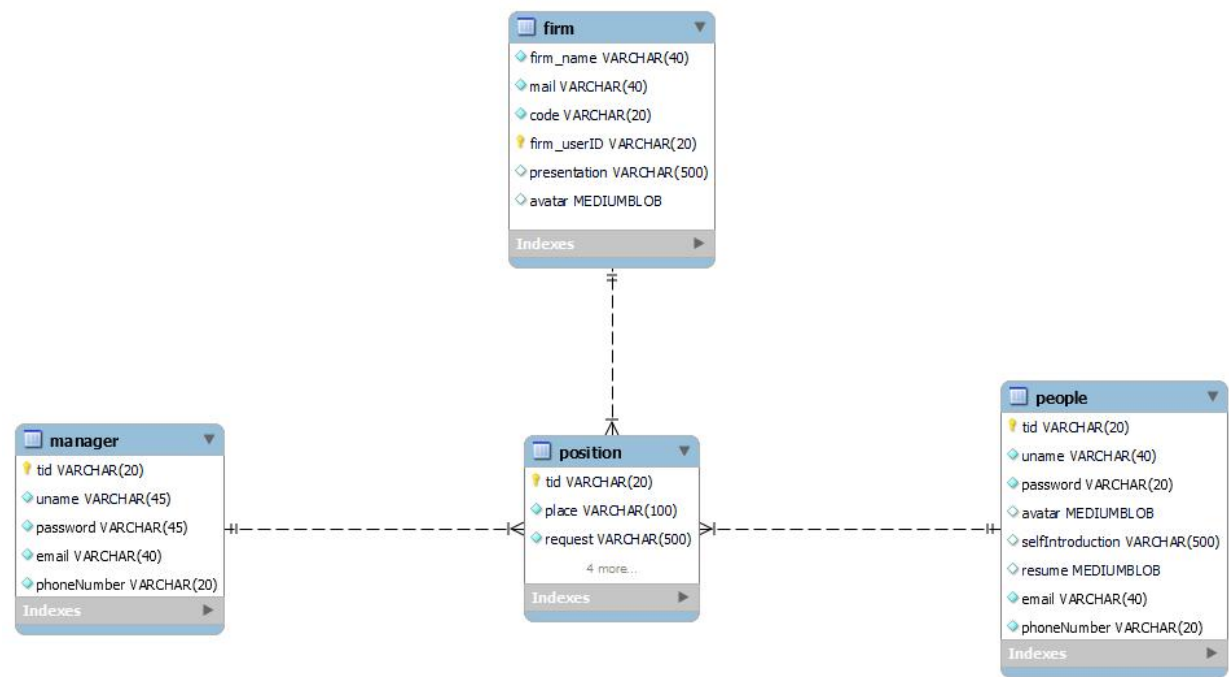
企业	
数据名	类型
企业名	Varchar（40）
邮箱	Varchar（40）
密码	Varchar（20）
账户 ID	Varchar（20）
简介	Varchar（500）
头像	Varchar（255）

个人用户	
数据名	类型
用户名	Varchar（20）
邮箱	Varchar（40）
密码	Varchar（20）
账户 ID	Varchar（20）
简介	Varchar（500）
手机号	Varchar（20）
简历	Varchar（255）
头像	Varchar（255）

岗位	
数据名	类型
岗位 ID	Varchar（20）
工作地	Varchar（100）
岗位要求	Varchar（500）
薪资	Int

管理员	
数据名	类型
管理员 ID	Varchar（20）
用户名	Varchar（45）
密码	Varchar（45）
邮箱	Varchar（40）
手机号	Varchar（20）

2.5.6 数据库描述



2.6 环境需求

本系统采用前后端分离架构，开发部署基于通用操作系统与主流编程语言框架。为保障系统的正常开发、测试、部署与运行，需满足以下硬件与软件环境要求。

2.6.1 硬件环境需求

本系统对运行所需硬件配置要求较低，适合在普通笔记本或台式机上开发和部署。主要硬件需求如下：

- CPU：主频不低于 2.0GHz，建议双核及以上处理器；
- 内存：最低要求 4GB，推荐 8GB 以上，以保证开发调试环境流畅运行；
- 硬盘空间：至少可用空间 10GB，用于安装开发工具、依赖库、数据库等；
- 显卡：无特殊要求，集成显卡可满足前端开发调试；
- 显示器：分辨率建议不低于 1366×768，推荐使用双屏以提升开发效率；
- 其他外设：标准键盘、鼠标及网络接口，建议具备 USB 接口以支持外部文件传输。

2.6.2 软件环境需求

本系统需要在以下软件平台和工具的支持下完成开发和运行：

1.操作系统要求：

支持系统运行的操作系统为：Microsoft Windows 10 或 Windows 11；

开发阶段可兼容 Windows 7 及以上版本；
若在 Linux 系统部署，需适配相关依赖。

2. 前端开发环境：

前端构建工具：Node.js

包管理器：npm 或 yarn

浏览器环境：支持 ECMAScript 6 标准的现代浏览器，如：Google Chrome，Microsoft Edge

3. 后端开发环境：开发语言：JavaScript（Node.js）、或 Java / Python（如需可拓展）；

运行时环境：Node.js

框架：Express.js；

数据库：MySQL 8.0；

接口调试工具：Postman 或类似工具；

2.6.3 网络环境需求

系统部署与运行需具备基本的内网或互联网连接条件；

若进行远程访问或异地协作开发，应配置本地 Web 服务器或使用云平台；

若进行测试部署，建议局域网中配置一台开发机作为服务端；

数据交互应保证在 HTTP/HTTPS 协议下进行，必要时可加装 SSL 证书保障传输安全。

2.7 资源需求

2.7.1 硬件资源需求

为确保本人才招聘系统在开发、测试与部署运行过程中具有良好的性能与稳定性，需配置满足最低标准的硬件资源。以下为本系统对客户端与服务端的主要硬件资源需求说明：

1. 客户端硬件资源需求

客户端主要用于访问系统前端页面、提交操作请求、浏览职位信息与管理个人账户等操作，硬件资源需求如下：

处理器：双核处理器及以上

内存：最低 4GB，推荐 8GB 及以上；

硬盘空间：至少剩余 5GB 可用空间（用于浏览器缓存、临时文件、下载简历等）；

显示设备：分辨率不低于 1366×768，推荐全高清；

网络接口：支持稳定的 Wi-Fi 或有线网络，推荐 100Mbps 以上带宽；

其他设备：标准 USB 接口、音频输入。

2. 服务器硬件资源需求

服务器承担系统核心业务逻辑处理、用户请求响应、数据库操作等任务，应具备较强的处理能力与并发支持，具体配置如下：

处理器：四核处理器及以上

内存：最低 8GB，推荐 16GB 或以上；

硬盘空间：

系统盘：不少于 50GB；

数据盘：根据职位、简历和日志数据量，推荐预留 100GB 可扩展空间；

网络带宽：至少 100Mbps 独享带宽，推荐部署在公网可访问的稳定环境下；

稳定性：建议配备 UPS 电源及定期自动备份设备；

可用性：若系统面向多人同时访问，推荐启用负载均衡与横向扩展能力。

2.7.2 软件资源需求

为保障本系统在开发、测试和正式运行阶段的各项功能能够稳定实现，需要配置相应的软件平台与工具。软件资源包括操作系统、运行支撑环境、开发工具、数据库系统、应用服务器、中间件及浏览器支持环境等。

1. 操作系统

1) 开发环境: Windows 10 / Windows 11

2. 编程语言与运行环境

1) 前端开发语言与框架: JavaScript Vue.js

2) 后端开发语言与框架: Node.js

3. 数据库系统

系统使用关系型数据库用于存储用户信息、职位信息、简历内容及投递记录等核心数据，采用 MySQL 8.0 存储信息。

2.8 非功能性需求

2.8.1 系统响应时间

页面加载时间：系统的页面加载时间应在 2 秒以内

操作响应时间：系统的操作响应时间应在 0.5 秒以内

2.8.2 数据处理能力

并发处理能力：系统应能够支持同时处多个用户请求

可靠性：

故障率：系统在正常运行条件下的故障率应低于每 1000 小时不超过 1 次

数据完整性：数据丢失或损坏的概率应低于每百万次操作不超过 1 次

可维护性

系统应符合可维护性指标，如 MTTR（Mean Time To Repair）、MTBF（Mean Time Between Failures）等

可用性：

系统的可用性应达到 99.9%，即系统每年不可用时间不超过 8.76 小时

可移植性：

跨平台支持：系统应能够在不同操作系统（如 Windows、Linux、MacOS 等）上运行，并保持稳定性和性能

数据迁移：系统应支持数据的快速迁移和导出，以便在不同环境中进行部署和使用

可复用性：

模块化设计：系统应采用模块化设计，以便系统的各个模块能够被复用在不同的项目中

接口标准化：系统应遵循标准化的接口规范，以便其他系统能够方便地集成和复用系统的功能模块

可测试性：

系统应具备自动化测试框架，以便进行持续集成和自动化测试，提高测试效率和可靠性

易用性：

用户可通过本系统快速完成整个在线预定流程，有比较良好的用户体验

2.9 其他需求

2.9.1 设计和实现的约束

特殊 CSCI 体系结构的使用或体系结构方面的需求：

需要使用特定的数据库和其他软件配置项，如 MySQL、MongoDB、PostgreSQL 等。

使用标准部件和现有的部件，如 Java、Python、Node.js 等作为后端开发技术。

需要使用 RESTful API 进行前后端交互。

特殊设计或实现标准的使用：

用户界面设计要求各界面清新简洁，配色合理，分首页、搜索、个人中心三个主页，并提供必要的修改功能。

硬件接口要求内存 512M 以上，磁盘空间 40G 以上，CPU 233Mhz 以上，硬盘空间 1.5G 以上。

软件接口要求操作系统为 Microsoft Windows 10/11，支持 ECMAScript6 的浏览器，服务器支持 node.js、MySQL 8.0 的环境。

灵活性和可扩展性需求：

系统需要支持企业用户和个人用户两类用户，以应对不同用户需求。

系统需要支持用户注册、登录、个人信息管理、发布招聘信息、搜索职位、简历上传、招聘流程管理等功能，以满足用户需求的多样性。

系统需要具备数据统计、报表生成、数据导出等功能，以支持系统管理员和普通用户的权限管理需求。

系统需要具备适应技术、风险或任务等方面预期的增长和变更的灵活性，以确保系统

的可扩展性和稳定性。

2.9.2 数据需求

1、输入数据

类型：文本、jpg、pdf、csv、json、xml

处理量：支持并发处理，可同时处理 10 个以上请求

数据量：能够处理 B~MB 大小的数据块

2、输出数据

类型：json 文件

格式：{

```
    "code":XX;
    "message":"XXX",
    "data":X
```

}

数据量：根据用户需求调整，可输出较小的简要反馈和较大的分析报告

3、数据管理能力

存储：以 SQL 数据库系统为支撑，能够存储和检索大量数据；媒体数据(图片 pdf 等)存储至专用服务器中(本软件使用的是阿里云 OSS 存储桶)

扩展性：随着数据量的增长，软件应能轻松扩展数据处理能力，如通过水平扩展增加数据库节点或使用分布式文件系统

4、安全性

数据加密：对敏感数据进行加密，确保数据在传输和存储过程中的安全性

访问限制：实施严格的用户权限管理，限制非授权访问

2.9.3 操作需求

说明本系统在常规操作、特殊操作以及初始化操作、恢复操作等方面的要求。

1、常规操作要求：

用户界面友好：系统应具有直观、易用的用户界面，用户能够快速上手并进行操作。

功能完备：系统应包含所有必要的功能模块，如数据录入、查询、分析、报告生成等。

响应速度：系统应具有快速的响应速度，用户操作不应出现明显的延迟。

数据一致性：系统应确保数据的一致性，避免数据冗余或数据不一致的情况发生。

2、特殊操作要求：

高级功能支持：系统应支持特殊操作，如高级数据分析、定制报告生成等功能。

异常处理：系统应能够处理特殊情况下的异常操作，如用户输入错误、系统故障等情况。

3、初始化操作要求：

系统配置：系统应具有简单的初始化配置过程，包括数据库连接设置、用户权限配置等。

数据导入：系统应支持数据导入功能，方便用户将现有数据导入系统进行管理和分析。

4、恢复操作要求：

数据备份：系统应具有数据备份功能，能够定期备份数据以防止数据丢失。

系统恢复：系统应支持系统故障后的快速恢复操作，包括数据恢复和系统配置恢复。

2.9.4 故障处理需求

1、 数据库未连接或连接失败

错误信息：“操作失败” 或

” SqlSession [org.apache.ibatis.session.defaults.DefaultSqlSession@16b0abfa] was not registered for synchronization because synchronization is not active”

补救措施：

检查是否安装了 mysql 数据库

检查 mysql 数据库是否正常运行

检查 application.yml 文件中与数据库相关的配置是否正确

检查 mysql 数据库中是否有与配置文件中相对应名称的数据库

与配置文件中相对应名称的数据库中的表项是否与说明书中的描述一致

2、 未连接到 Redis

错误信息：“Unable to connect to Redis”

补救措施：

检查是否安装了 Redis

检查 Redis 是否正常运行

检查 Redis 使用的端口是否正常

检查 application.yml 文件中与 Redis 相关的配置是否正确

3、 拒绝连接

错误信息:” connect ECONNREFUSED”

补救措施：

检查浏览器访问的网址，端口是否正确

检查 application.yml 文件中 server.port 是否与浏览器访问的端口一致

2.9.5 包装需求

1. 软件包装：将 CSCI 的软件部分进行打包，包括必要的安装程序、配置文件、文档等。软件包应当清晰标记版本号、发布日期等信息。

2. 标签要求：在软件包装上应附有标签，标明 CSCI 的名称、版本号、制造日期、适用操作系统、数据库管理系统版本要求等信息。

3. 交付文档：除了软件本身，还应提供用户手册、安装指南、系统需求规格说明等相关文档。

4. 可引用的标准：在进行软件包装和标记时，可以参考 ISO/IEC 19770-1 软件资产管理标准，以确保软件包装和标记符合行业标准和规范。

2.9.6 适应性需求

为保障人才招聘系统在实际运行过程中避免数据泄露、账户滥用和非法操作等安全隐患，

本系统对涉及用户隐私、操作权限、系统资源访问等方面提出了若干保密性与安全性需求，旨在防止潜在的操作风险，并降低可能对用户、数据或系统稳定性造成的威胁。

1. 数据依赖性

系统部分模块在运行过程中依赖于外部或预先加载的数据字典。这些数据不直接包含在系统安装包中，而需由运维人员在部署阶段手动导入或通过管理界面添加。例如，职位信息模块需要依赖标准化的职位分类数据，如技术岗、行政岗、销售岗等，以使用户准确检索与筛选；地区字段则依赖省、市、区三级行政区划结构，需加载完整的地域信息表；行业字段使用国家或平台自定义的行业代码，确保岗位归属分类的一致性。这些外部数据必须在首次部署时正确导入数据库，系统方可正常使用。

2. 运行参数的可配置性

系统包含一系列在运行过程中可动态调整的参数，以适应不同的使用需求。典型示例包括页面显示的记录条数（如分页默认 10 条），用户简历上传的最大文件大小限制（如不超过 5MB），招聘职位的有效期设定（如默认 30 天自动下架）等。这些参数均应通过配置文件或后台管理页面进行调整，而非硬编码固定在程序内部。通过参数配置机制，系统管理员可根据业务规模、使用频率或硬件能力灵活设置这些运行参数，从而提升系统弹性与易维护性。

3. 环境适配性

系统运行还需要适配不同的部署环境。后端服务器的监听端口、数据库连接地址、访问凭证等关键参数需要根据实际部署的服务器环境进行配置，一般通过 `.env` 配置文件完成。前后端分离部署架构要求系统具备动态设置跨域访问（CORS）的能力，以允许来自前端应用服务器的合法请求。此外，上传文件的本地保存路径必须支持配置，以适应不同操作系统的文件系统差异。同时，系统应根据开发、测试与生产环境的不同，自动切换日志级别与错误处理机制，以便问题排查与安全控制。

2.9.7 保密性需求

为保障人才招聘系统在实际运行过程中避免数据泄露、账户滥用和非法操作等安全隐患，本系统对涉及用户隐私、操作权限、系统资源访问等方面提出了若干保密性与安全性需求，旨在防止潜在的操作风险，并降低可能对用户、数据或系统稳定性造成的威胁。

1. 用户信息加密存储机制

系统中所有用户账号的密码信息采用不可逆的加密算法进行处理，当前使用 MD5 哈希算法对用户密码进行加密后再存储于数据库中。系统不允许任何明文密码在数据库中保存，也不在前端传输明文密码，从而防止中间人攻击与数据库泄露带来的信息安全风险。在未来扩展中，建议引入更强的加盐哈希算法（如 bcrypt、SHA-256+salt）以增强安全性。

2. 登录验证机制与会话控制

系统要求所有涉及用户数据访问、信息修改或资源提交等操作必须在登录状态下完成。未经授权的请求将被服务器拒绝。系统在登录成功后为用户分配会话令牌（Token 或 Session ID），并对每次请求进行身份校验，防止非法冒用。此外，系统应具备自动会话过期与重新登录机制，避免长时间空闲引发安全隐患。

3. 权限管理与功能隔离

平台根据用户类型（普通用户、企业用户、系统管理员）设定不同权限等级，并在系统后端进行角色权限校验。用户在客户端所能看到的操作界面将根据其权限动态渲染，避免因前端可见导致的越权操作。例如，普通用户无法访问后台管理页面，企业用户无法修改其他企业的岗位信息。所有权限控制逻辑应在服务端强制执行。

4. 敏感操作确认机制

对于简历删除、岗位下架、管理员封禁用户等敏感操作，系统需提供操作确认机制，如弹出确认框、二次输入密码、验证码验证等方式，防止用户误触或恶意脚本自动触发操作。同时，建议对这些操作写入安全日志，供后期追踪。

5. 防止非法请求与接口滥用

系统应对所有 API 请求做权限校验和参数合法性检查，防止通过构造非法请求绕过界面限制执行后台操作。此外，系统可在高频操作接口（如登录、注册、密码重置）中加入访问频率限制（限流）、验证码验证等防护机制，以防止暴力破解或接口滥用攻击。

6. 数据泄露应对与日志管理

系统应保留关键操作的安全日志记录，包括登录行为、修改操作、异常请求等，便于安全审计与问题溯源。同时，日志文件应具有只读权限，不得被用户直接访问或下载。敏感数据应避免输出至日志，防止日志泄露带来次生风险。

2.9.8 保密性和私密性需求

人才招聘系统涉及大量用户注册信息、个人简历、企业岗位详情等敏感数据。为了保障这些信息在存储与传输过程中的安全性，并防止系统功能被非法使用，系统从账户保护、数据访问、操作控制与隐私信息隔离等方面提出了明确的保密性与私密性需求。

1. 用户账户加密与身份验证

系统中所有用户的密码必须采用不可逆加密算法进行存储，目前选用 MD5 哈希处理，未来可升级为更安全的加盐算法（如 bcrypt 或 PBKDF2）。任何时候，系统均不允许明文密码在客户端或服务端以明文形式存储或传输。

登录功能必须通过用户名与加密后的密码匹配完成，且所有涉及用户身份的操作均需验证登录状态。系统通过 Session 或 Token 实现身份管理，采用 HTTP-only 方式存储认证令牌，有效防止会话被脚本读取，提升抗攻击能力。

2. 角色权限控制机制

系统应明确区分普通用户、企业用户和系统管理员三类角色，不同角色拥有不同的功能权限。权限验证应在服务端强制执行，任何绕过前端的请求均应被拒绝。

例如，普通用户只能访问职位浏览、简历编辑和职位投递等功能；企业用户则可管理本企业的岗位信息和接收到的简历数据；管理员则拥有用户管理、岗位审核、系统参数配置等

高级权限。前端应根据权限动态渲染界面，隐藏不具备权限的功能入口，防止误操作。

3. 隐私数据隔离与访问控制

系统对个人用户的简历信息、联系方式、教育经历等敏感数据必须严格限制访问权限。未获得授权的企业用户无法直接查看用户完整简历，仅在用户主动投递后才能查看对应信息。

系统后台应通过数据库字段隔离、数据加密存储、接口访问验证等方式实现隐私信息的隔离与受控访问，确保用户在平台上的数据仅在授权条件下被访问或处理

4. 敏感操作保护机制

所有涉及敏感数据变更或重要业务行为的操作，如简历删除、职位发布、岗位下架、管理员用户封禁等，系统应引入操作确认机制。可通过二次弹窗、图形验证码、密码复验证等方式，防止误触与自动化攻击。

对于高风险功能，建议采用双重验证机制，如短信验证码或管理员审批流程，以进一步强化操作安全。

5. 安全日志记录与操作审计

系统应对所有登录、身份切换、权限变更、数据修改、删除等关键行为进行日志记录。日志内容包括操作者身份、时间、操作类型与结果，用于系统安全审计与问题追踪。

日志文件应以只读权限存储，并限制管理员访问，防止日志被篡改或删除。用户行为记录不包含明文密码或私密字段，确保日志本身不构成新的泄密风险。

第三章 其他规格说明

3.1 合格性规定

合格性规定用于明确如何判断软件是否满足各项需求，为软件的验收和测试提供清晰的标准。

用户登录功能（REQ - 001）：采用演示的合格性方法。在测试环境中，模拟不同类型用户（企业用户、个人用户、管理员）进行登录操作。演示过程需涵盖正常登录场景，输入正确的用户名和密码，验证系统能否准确识别并顺利进入对应功能模块；同时要模拟异常登录情况，如输入错误密码、不存在的用户名等，检查系统是否给出恰当的错误提示，以此判断用户登录功能是否符合需求。

发布招聘信息（REQ - 002）：运用测试的合格性方法。通过编写专门的测试用例，针对企业用户发布招聘信息的功能进行全面测试。测试内容包括输入各种合法和非法的岗位信息，如不同格式的岗位名称、薪资范围、工作地点等，检查系统对输入数据的有效性验证机制是否正常工作；测试在并发操作情况下，系统能否准确保存和处理招聘信息，避免数据冲突或丢失，确保发布招聘信息功能达到预期要求。

数据加密要求（REQ - 003）：同样使用测试的合格性方法。利用专业的安全测试工具和技术，对系统在数据传输和存储过程中的加密情况进行检测。检查敏感数据（如用户密码、企业机密信息）在网络传输时是否被加密，通过抓包工具分析数据传输内容，确认数据以密文形式传输；对于存储在数据库中的敏感数据，查看数据库存储结构和加密算法应用情况，验证数据加密的完整性和有效性。

系统性能要求（REQ - 004）：采取审查的合格性方法。收集系统在不同负载情况下的性能数据，包括页面加载时间、操作响应时间、并发处理能力等指标。审查性能测试报告，分析数据是否满足需求规格说明书中规定的性能标准，如页面加载时间应在 2 秒以内、操作响应时间应在 0.5 秒以内等，评估系统性能是否达标。

用户界面设计（REQ - 005）：通过分析的方法进行判断。从界面的布局合理性、色彩搭配协调性、操作便捷性等方面进行分析。邀请专业的 UI 设计师和目标用户进行评估，收集他们对界面设计的反馈意见，对比需求中对用户界面的设计要求，判断用户界面设计是否符合预期。

数据备份需求（REQ - 006）：运用审查的合格性方法。检查系统的数据备份策略和实际执行情况，查看备份计划是否合理，是否按照规定的时间间隔进行数据备份；审查备份数据的完整性和可恢复性，通过实际的恢复操作测试，验证在系统故障或数据丢失情况下，能否利用备份数据成功恢复系统数据，确保数据备份需求得到满足。

安全审计功能（REQ - 007）：若系统具备安全审计功能，需制定特殊的合格性方法。例如，通过模拟各种安全事件，如非法登录尝试、数据篡改行为等，检查系统的审计日志是否准确记录相关事件，包括事件发生的时间、操作人、操作内容等信息；验证审计功能能否及时发现异常行为，并生成有效的审计报告，为安全管理提供有力支持。

可扩展性要求（REQ - 008）：采用综合评估的特殊合格性方法。从系统架构设计、代码结构、数据库设计等方面进行分析，评估系统是否具备良好的扩展性。检查系统在增加新功能模块、用户数量增长、数据量增大等情况下的适应能力，通过模拟未来可能的业务扩展场景，验证系统是否能够方便地进行功能扩展和性能优化，满足可扩展性要求。

3.2 需求可追踪性

需求可追踪性确保软件需求在整个开发过程中能够被准确识别、关联和管理，有助于提高开发效率、保证软件质量，便于在需求变更时进行影响分析和控制。

从 CSCI 需求到系统（或子系统）需求的可追踪性：在本软件中，CSCI - 001 代表用户管理子系统。该子系统的需求包括用户登录功能，这要求系统能够验证用户输入的用户名和密码，实现用户身份的准确识别，确保只有合法用户能够登录系统；同时包含用户权限管理需求，即系统要具备不同用户角色（管理员、普通用户）权限的设置和管理功能，不同权限用户能够访问和操作相应的功能模块。CSCI - 002 对应数据处理子系统，其需求涵盖数据加密要求，保证用户数据在传输和存储过程中的安全性，防止数据泄露；还包括数据备份需求，确保在系统出现故障或数据丢失时，能够利用备份数据快速恢复系统正常运行。

从系统（或子系统）需求到 CSCI 需求的可追踪性：以系统需求中的用户登录功能为例，其引用 CSCI 的需求体现在支持用户输入用户名和密码进行登录这一具体功能实现上。系统通过对用户输入信息的验证，与数据库中存储的用户信息进行比对，判断用户登录的合法性。对于用户权限管理这一系统需求，引用 CSCI 的需求表现为管理员可以设置用户权限级别，

系统根据管理员的设置，控制不同用户对系统功能的访问和操作权限。所有分配给 CSCI 的系统（或子系统）需求都应在文档中明确说明，以保证每个 CSCI 的需求都能追溯到其所涉及的系统需求，形成完整的需求链条。当追踪到 IRS（Interface Requirements Specification）中包含的 CSCI 需求时，通过引用 IRS 中的相应部分，确保追踪的完整性和准确性，使接口需求与 CSCI 需求紧密关联，便于在开发过程中进行统一管理和协调。

3.3 尚未解决的问题

安全性问题：在软件的整个生命周期中，安全性是至关重要的。在用户注册、登录过程中，虽然提及使用 MD5 算法对密码进行加密，但对于密码加密存储的具体机制和强度缺乏详细说明。此外，在防止 SQL 注入攻击方面，仅在验证模块提到进行 SQL 防注入过滤，但未明确具体的过滤方式和防护级别。随着网络安全威胁的不断变化，需要进一步研究和确定更完善的安全防护策略，如采用更先进的密码加密算法（如 bcrypt、argon2 等），结合参数化查询、输入验证等多种手段来有效防止 SQL 注入攻击，保障用户数据安全。

用户体验优化：尽管对用户界面设计提出了要求，但在具体的交互设计方面，如页面跳转的流畅性、操作按钮的位置和大小、提示信息的准确性和及时性等，都需要进一步细化。对于响应速度，虽然规定了页面加载时间和操作响应时间的上限，但在实际开发中，需要深入分析和优化系统架构、算法以及数据库查询等环节，以确保在各种网络环境和用户负载情况下都能达到理想的响应速度。同时，关于用户友好性，需要开展用户调研，收集用户的使用习惯和反馈意见，对系统的交互流程进行优化，提高用户满意度。

数据一致性与完整性：在多用户多数据交互的复杂环境下，数据一致性和完整性是保证系统正常运行的关键。目前尚未提及如何处理数据冲突，例如当多个用户同时对同一数据进行修改时，系统应采取何种策略来确保数据的准确性和一致性。对于重复数据的处理，也缺乏相应的机制，需要建立数据查重和去重功能，避免数据冗余对系统性能和数据质量的影响。此外，在数据备份与恢复方面，虽然提到了数据备份需求，但对于备份的频率、备份数据的存储位置、恢复数据的具体流程以及恢复过程中的数据一致性保障等问题，都需要进一步明确和规划。

性能优化：虽然列出了硬件和软件接口的要求，但在实际开发中，要达到良好的系统性能，还需要从多个方面进行优化。在硬件方面，需要根据系统的实际负载情况，合理配置服务器资源，如优化内存分配、提高磁盘 I/O 性能等；在软件方面，需要对系统的算法进行优化，减少不必要的计算和数据传输；对数据库进行索引优化、查询优化，提高数据检索效率。同时，要考虑如何处理大量数据时的性能问题，如采用分布式存储和计算技术，提高系统的吞吐量和处理能力。

错误处理与日志记录：在系统运行过程中，不可避免会出现各种异常情况。目前仅针对部分常见错误（如数据库连接失败、Redis 连接失败等）列出了错误信息和补救措施，但对于系统运行过程中可能出现的其他异常情况，如网络中断、服务器崩溃等，缺乏全面的错误处理机制。此外，在日志记录方面，需要建立完善的日志系统，详细记录系统运行过程中的关键事件、错误信息以及用户操作等，以便在系统出现故障时能够快速定位问题根源，进行有效的故障排查和系统优化。

3.4 故障处理

1. 软件系统问题分类

a. 数据库相关故障

数据库未连接或连接失败：数据库服务未启动、配置错误或网络异常导致无法访问数据库服务器。

数据一致性问题：多用户并发操作引发数据冲突（如简历重复投递），或分布式事务未正确提交导致数据丢失。

SQL 注入攻击：用户输入未过滤非法字符，导致数据库遭受攻击。

b. 接口与服务故障

接口响应超时：后端服务负载过高或网络延迟导致前端请求无法及时处理。

权限验证失败：登录令牌过期、伪造或角色权限配置错误，导致用户越权访问

c. 文件存储与解析故障

简历文件解析失败：上传文件格式不支持（非 PDF/Word）或文件损坏，解析引擎异常。

文件存储服务异常：阿里云 OSS 链接失效、存储空间不足，导致文件上传失败。

d. 系统性能故障

页面加载超时：前端资源加载失败或后端接口响应缓慢（如未优化数据库索引）。

并发请求处理失败：服务器资源不足（内存 / CPU 瓶颈），无法应对高并发场景。

2. 错误信息规范

以下为不同故障类型对应的错误信息及错误码

数据库连接失败："操作失败：无法连接到数据库，请检查服务器状态"（错误码：DB-001）

账号不存在："登录失败：该用户名或邮箱未注册"（错误码：LOG-001）

密码错误："登录失败：密码错误，请重新输入"（错误码：LOG-002）

接口响应超时："请求超时，请刷新页面后重试"（错误码：API-001）

文件解析失败："简历解析失败：不支持的文件格式，请上传 PDF 或 Word 文件"（错误码：RES-002）

权限不足："访问被拒绝：您没有权限执行该操作"（错误码：PER-001）

SQL 注入攻击检测："非法请求：检测到不安全的输入，请检查参数"（错误码：SEC-001）

数据冲突："操作失败：该职位已被申请，请刷新后重试"（错误码：CON-001）

3. 补救措施

a. 数据库故障处理

数据库连接失败：检查 MySQL 服务状态并重启，验证配置文件中的数据库连接参数，排查防火墙对数据库端口（默认 3306）的拦截。

数据一致性问题：启用数据库事务（如 InnoDB 引擎）保证操作原子性，高并发场景使用 Redis 分布式锁避免超卖，定期运行数据对账脚本修复差异。

b. 接口与服务故障处理

接口响应超时：优化后端代码并启用 Redis 缓存高频数据，扩展服务器资源或部署负载均衡。

权限验证失败：刷新登录令牌或重新登录，检查 RBAC 角色权限配置是否匹配用户操作。

c.文件存储与解析故障处理

简历解析失败：提示用户检查文件格式或重新上传，升级解析引擎（如 Apache Tika）或切换备用库。

OSS 存储异常：验证阿里云 OSS 访问密钥有效性，清理过期文件或扩容存储空间。

d.系统性能与安全故障处理

页面加载超时：压缩前端资源、启用浏览器缓存和 CDN 加速，优化数据库查询并添加索引。

SQL 注入攻击：使用参数化查询（如 MyBatis 的#占位符），部署 Web 应用防火墙（WAF）拦截恶意请求。

并发请求处理失败：引入消息队列（如 RabbitMQ）异步处理非核心任务，对高频接口启用限流（如 Redis 令牌桶算法）。

4.其他补充措施

日志记录：所有故障信息需写入系统日志（含时间、用户 ID、错误码、堆栈信息），便于追溯分析。

自动恢复：非关键故障（如缓存失效）自动重试或切换至备用服务（如数据库主从切换）。

用户引导：错误信息中提供清晰操作指引（如“点击此处检测网络连接”），降低使用门槛

3.5 算法说明

人才招聘系统的核心计算功能主要集中在岗位推荐匹配、搜索相关性排序、简历完整度评分等方面。通过合理设计这些算法，系统能够更高效地为用户提供精准的职位推送、快速搜索结果与简洁的数据呈现，增强用户体验。

1. 岗位与简历匹配算法

a.算法概况

岗位匹配是系统的重要功能之一，其目标是根据用户填写的简历信息（如专业、技能、学历、期望职位等），从数据库中选出匹配度较高的岗位。该算法基于关键词权重向量匹配和简单打分规则进行推荐。

b.算法公式

系统通过计算用户简历关键词与岗位需求关键词的匹配得分来评估匹配度。关键词通过预设权重加权，如果简历中包含岗位所需关键词，则该关键词得分为权重值，否则为零。最终匹配度得分为所有关键词得分的累加和。

$$\text{Score}(R, J) = \sum_{i=1}^n [w_i \times \delta(r_i, j_i)]$$

2. 搜索相关性排序算法

a.算法概况

系统提供了职位搜索功能，用户可以输入关键词进行职位检索。为了提升搜索体验，搜

索结果需按照岗位与搜索关键词的相关性排序，常用 TF-IDF 模型计算文本相关性。

b.算法公式

岗位搜索结果根据关键词与岗位描述的相关性排序。相关性由 TF-IDF 算法计算得到。关键词在岗位描述中的出现频率（TF）乘以其逆文档频率（IDF），IDF 反映关键词在所有岗位中的普遍程度。岗位与搜索关键词的相关性分数为所有关键词 TF-IDF 值的累加，得分越高表示相关性越强。

单个关键词的 TF-IDF 值：

$$TFIDF(q_i, D) = TF(q_i, D) \times \log(N / DF(q_i))$$

岗位相关性得分：

$$Score(Q, D) = \sum (q_i \in Q) [TFIDF(q_i, D)]$$

3.6 有关培训需求

本人才招聘系统涉及多类用户角色及开发场景，需针对不同对象提供差异化培训内容，确保系统功能的有效使用与开发维护。具体培训需求如下：

企业用户培训

涵盖岗位发布流程（如填写职位详情、设置薪资范围、审核机制）、简历筛选技巧（如关键词过滤、智能匹配结果解读）、招聘流程管理（面试安排、录用通知发送）、数据统计报表的解读与导出等。提供交互式在线教程，包含模拟发布岗位的操作演示；制作《企业用户操作手册》，图文并茂说明各功能模块；定期举办线上直播培训，解答企业用户常见问题（如简历解析失败处理、面试冲突调整等）。

个人用户培训

包括账号注册与登录流程、简历制作规范（如在线填写指南、文件上传格式要求）、职位搜索技巧（关键词组合、筛选条件应用）、申请状态查询与反馈机制等。开发“求职助手”引导模块，首次登录时弹出关键操作提示；提供简历模板下载及填写示例，降低用户操作门槛；通过站内信推送个性化培训内容（如针对应届生的简历优化建议）。

管理员培训

涉及用户权限管理（角色创建、权限分配）、数据审核流程（岗位真实性校验、违规用户封禁）、系统参数配置（如简历大小限制、职位有效期设置）、安全日志审计等。组织线下培训 workshops，结合实际案例演示后台管理操作；提供管理员专属测试环境，允许在模拟场景中练习数据维护与故障处理（如用户数据批量导出、异常操作回溯）。

开发与维护培训

技术团队培训

针对系统采用的技术栈（如 Node.js 后端、Vue.js 前端、MySQL 数据库）进行深度培训，包括 RESTful API 设计规范、前后端联调技巧、数据库索引优化策略等；讲解系统架构（分层设计、模块化开发），确保开发人员理解代码结构与扩展方式。安排内部技术分享会，由核心开发成员演示关键模块实现（如 JWT 认证流程、简历解析算法）；提供开源项目参考资料，引导团队学习类似招聘系统的最佳实践。

运维团队培训

涵盖系统部署流程（如 Docker 容器化、Nginx 反向代理配置）、服务器监控工具使用（如 Prometheus 监控 CPU / 内存占用）、数据备份与恢复策略（MySQL 定期备份、OSS 文件归档）、故障排查流程（如接口响应超时定位、数据库连接失败处理）等。通过模拟故障演练（如模拟 OSS 存储异常、Redis 缓存失效），提升运维团队的应急处理能力；提供《运维手册》，详细记录常见问题代码及补救措施（如错误码 DB-001 对应的数据库连接检查步骤）。

3.7 有关后勤需求

系统的稳定运行依赖于完善的后勤支持体系，需从维护、支持、部署、资源等方面确保软硬件环境的可靠性与可扩展性。

系统维护

建立定期巡检机制，每日检查服务器状态（CPU 利用率 $\leq 80\%$ 、内存剩余 $\geq 20\%$ ）、数据库连接池健康度、OSS 存储空间使用情况（阈值设为 80%，超阈值自动清理过期简历文件）。每周进行数据一致性校验，对比 MySQL 结构化数据与 OSS 非结构化文件的元数据（如简历 ID 与存储路径匹配性），生成校验报告并自动修复差异。

采用灰度发布策略，新功能先在测试环境验证（如模拟 100 个并发用户测试职位搜索响应时间），再逐步推送给部分用户（如 10% 企业用户），收集反馈后全量发布。

维护多套环境（开发、测试、生产），确保代码变更不影响线上服务；更新前备份数据库，支持一键回滚至历史版本。

软件支持

设立客服专线与在线工单系统，7×12 小时响应用户问题（如企业用户无法导出招聘报表、个人用户简历解析错误），平均响应时间 ≤ 30 分钟。

建立知识库系统，分类存储常见问题解决方案（如“LOG-003 账户锁定”需用户等待 30 分钟后重试），支持用户自助查询。

与云服务提供商（如阿里云）签订 SLA 协议，确保 OSS、Redis 等服务的可用性 $\geq 99.9\%$ ；定期参加供应商技术培训，获取最新产品动态（如 MySQL 8.0 新特性应用）。

引入安全审计服务，每年进行一次渗透测试与代码审计，防范 SQL 注入、XSS 等安全漏洞。

部署与运输

采用容器化部署，通过 Docker Compose 管理后端服务、数据库、缓存等组件，实现快速迁移与扩展；生产环境部署于阿里云 ECS 服务器，搭配负载均衡器应对高并发请求（如招聘旺季的职位搜索流量）。

前端静态资源（如 HTML、CSS、JS）存储于 OSS 并开启 CDN 加速，减少用户访问延迟。系统以软件形式交付，无需物理运输；若需本地化部署，提供离线安装包（含数据库初始化脚本、前端静态文件），支持通过安全介质（如加密 U 盘）传输。

资源供应

开发阶段使用个人笔记本（配置 $\geq 4\text{GB}$ 内存、50GB 硬盘），测试环境部署于校园云服务器（8GB 内存、100GB 数据盘）；生产环境根据用户规模动态扩展服务器资源（如每新增 1000 企业用户，增加 1 台 ECS 实例）。

备用设备方面，预留 20% 服务器资源作为故障切换节点，确保主服务器宕机时 5 分钟内完成 failover。

开发工具链包括 VS Code、Postman、Navicat 等，均使用正版授权；数据库采用 MySQL Community Edition（开源免费），Redis 为开源版本，降低 licensing 成本。

依赖的第三方库（如 Apache Tika 简历解析库、JWT 认证库）定期更新至稳定版本，避免已知漏洞风险。

对现有设施的影响

若部署于校园现有服务器，需评估硬件性能（如 CPU 主频 $\geq 2.4\text{GHz}$ 、网络带宽 $\geq 100\text{Mbps}$ ），必要时升级网卡或增加存储设备；若使用公有云，对校园网络无特殊要求，只需确保出口带宽满足用户访问需求。

系统兼容主流浏览器（如 Chrome、Edge），无需用户升级硬件或安装插件；移动端通过响应式设计适配屏幕尺寸，支持分辨率 $\geq 1366 \times 768$ 的设备。

灾难恢复

制定灾难恢复计划，定期将数据库备份至异地数据中心（如阿里云上海与北京机房互备），备份频率为每日全量备份 + 每小时增量备份；

模拟服务器宕机场景，测试从备份恢复系统的时间（目标 ≤ 4 小时），确保关键业务（如职位申请、面试安排）在灾难发生后快速恢复。

附录

Petri 网建模

确定系统元素

首先明确系统中的状态和事件。状态对应 Petri 网中的库所（Place），比如在生产流水线系统中，“原材料库存”“加工设备空闲”等可作为库所；事件对应变迁（Transition），像“原材料投入加工”“设备完成加工任务”等可设定为变迁。

定义元素间关系

确定库所和变迁之间的连接关系，通过有向弧来表示。若“原材料投入加工”变迁发生需消耗原材料，就从“原材料库存”库所向“原材料投入加工”变迁画一条有向弧；若该变迁完成会使加工后的产品进入库存，就从“原材料投入加工”变迁向“成品库存”库所画有向弧。

设定初始标识

标识（Marking）表示系统的初始状态，即库所中托肯（Token，可理解为资源或实体）的分布情况。如在初始时，“原材料库存”库所中有一定数量托肯代表原材料数量，“加工设备空闲”库所中有托肯表示设备可用。

模型验证与优化

构建好 Petri 网模型后，对其进行动态模拟，检查模型是否符合预期的系统行为，比如是否存在死锁（系统无法继续运行的状态）、活锁（系统持续执行无意义操作无法推进有效进程）等问题。根据模拟结果对模型进行调整优化，确保其能准确反映真实系统并用于分析和预测系统行为。

应用到本项目：

1. 确定库所（Place）

P1： 用户发起登录请求：代表用户在前端点击登录，输入账号密码并提交的初始状态。

P2： 验证模块接收请求：表示登录请求已传递到验证模块，等待验证处理。

P3： 账号不存在：当验证模块查询数据库，发现输入账号无匹配记录时进入该状态。

P4： 密码错误：若账号存在，但密码校验不通过，系统进入此状态。

P5： 登录成功：账号和密码均验证正确，用户成功登录，可进入相应功能模块。

P6： 系统异常：在登录验证过程中，若出现数据库连接失败、程序报错等问题，导致登录流程无法正常进行，进入该状态。

2. 确定变迁（Transition）

T1： 发起登录请求：用户在前端完成账号密码输入并点击登录按钮，触发此变迁，使系统从 P1 状态转移到 P2 状态。

T2： 验证账号密码：验证模块接收到登录请求后，开始查询数据库验证账号密码，该变迁连接 P2 与 P3、P4、P5 状态，根据验证结果决定后续走向。若账号不存在，激活 P3；账号存在但密码错误，激活 P4；两者都正确则激活 P5。

T3： 处理登录异常：在登录验证过程中，如果发生数据库连接异常、SQL 注入攻击等错误，触发此变迁，使系统从 P2 状态转移到 P6 状态。

3. 确定弧（Arc）

从 P1 到 T1：表示用户发起登录请求这一行为触发了“发起登录请求”变迁。

从 T1 到 P2：“发起登录请求”变迁发生后，登录请求传递到验证模块，即进入 P2 状态。

从 P2 到 T2：验证模块接收到请求后，开始进行账号密码验证，触发“验证账号密码”变迁。

从 T2 分别到 P3、P4、P5：根据“验证账号密码”的结果，决定进入不同状态。若账号不存在，T2 到 P3 的弧被激活；密码错误，T2 到 P4 的弧被激活；登录成功，T2 到 P5 的弧被激活。

从 P2 到 T3：若登录验证过程中出现异常，P2 到 T3 的弧被激活，触发“处理登录异常”变迁。

从 T3 到 P6：“处理登录异常”变迁发生后，系统进入 P6 系统异常状态状态

用户发起登录请求

