

## 可行性分析(研究)报告(FAR)



小组名称：\_\_\_\_\_软工加油队\_\_\_\_\_

小组成员：\_\_\_\_付翔宇、翟江浩、冯彬、孟维汉、杨雨泽\_\_\_\_

课程名：\_\_\_\_\_软件工程与实践\_\_\_\_\_

年 级： ☐ 一年级 ☐ 二年级 ☒ 三年级

山东大学制

## 目录

可行性分析(研究)报告(FAR)	1
1 引言	4
1.1 标识	4
1.2 背景	4
1.2.1 提出	4
1.2.2 项目名称	4
1.2.3 项目提出者	4
1.2.4 项目要求	4
1.2.5 限制条件	4
1.2.6 项目目标	5
1.3 项目概述	5
1.4 文档概述	5
2 引用文件	5
3 可行性分析的前提	6
3.1 项目的要求	6
3.2 项目的目标	6
3.3 项目的环境、条件、假定和限制	7
3.3.1 环境	7
3.3.2 条件	7
3.3.3 假设	8
3.3.4 限制:	8
3.4 进行可行性分析的方法	8
4 可选的方案	8
4.1 原有方案的优缺点、局限性及存在的问题	8
4.2 可重用的系统, 与要求之间的差距	9
4.3 可选择的系统方案(人员)	9
4.4 可选择的系统方案(模块)	10
4.5 选择最终方案的准则	10
5 所建议的系统	11
5.1 对所建议的系统的说明	11
5.2 数据流程和处理流程	11
5.3 与原系统的比较(若有原系统)	17
5.4 影响(或要求)	17
5.4.1 设备	17
5.4.2 软件	17
5.4.3 运行	17
5.4.4 开发	18
5.4.5 环境	18
5.5 局限性	19
6 经济可行性	19
6.1 投资分析	19
6.2 预期的经济效益	21

6.2.1 一次性收益 .....	21
6.2.2 非一次性收益 .....	21
6.2.3 不可定量的收益 .....	22
6.2.4 收益/投资比 .....	22
6.2.5 投资回收周期 .....	23
6.3 市场预测 .....	23
6.3.1 短期市场 .....	23
6.3.2 长期市场 .....	23
7 技术可行性(技术风险评价) .....	24
7.1 技术风险 .....	24
7.2 时间管理风险 .....	24
7.3 需求变更风险 .....	25
7.4 团队协作风险 .....	25
7.5 测试与质量风险 .....	25
7.6 文档与验收风险 .....	25
7.7 技术债务风险 .....	26
7.8 外部依赖风险 .....	26
7.9 未来功能扩展风险 .....	26
8 法律可行性 .....	26
9 用户使用可行性 .....	27
10 其他与项目有关的问题 .....	27
11 注解 .....	28
附录 .....	31
一、CASE 工具在本项目中的应用（实验二） .....	31
二、SCRUM 在本项目中的应用（实验三） .....	31
三、预期实现界面 .....	32

# 1 引言

## 1.1 标识

软件系统：人才招聘系统

缩略语：TRS

文档适用版本：Window10/11 系统 office2021 及以上版本

软件标识：MYSQL, Intelij IDEA, Spring Boot, Vue, Element Plus, Tomcat, Postman

## 1.2 背景

### 1.2.1 提出

随着中国教育高质量发展以及网络的兴起与应用，市场上涌现出越来越多的人才，为了优化招聘流程、提高招聘效率，并提升人才招聘的质量，现需要开发一套全面、易用、高效的人才招募系统。

### 1.2.2 项目名称

人才招聘系统（Talent Recruitment System）

### 1.2.3 项目提出者

软工加油队

### 1.2.4 项目要求

流程精简：通过系统化管理，减少招聘环节的复杂性，提升整体效率。

人才吸引：提供清晰、优质的招聘信息以及简便的投递方式，吸引更多优秀候选人。

效率提升：借助自动化工具和数据分析功能，降低人力成本，加快招聘进程。

质量保障：通过系统化管理和智能筛选机制，确保招聘到的人才符合高标准要求。

### 1.2.5 限制条件

功能需求：涵盖招聘信息发布、简历投递、简历筛选、面试安排及数据分析等核心功能。

安全保障：确保招聘信息和候选人个人数据的保密性与安全性，防止信息泄露。

## 1.2.6 项目目标

实现在 windows10/11 环境下可运行的人才招聘软件，该软件将有三部分组成：

- ① 个人用户模块：该模块包括简历修改，用户个人信息管理，职位查询，求职管理，密码设置与退出等。
- ② 企业用户模块，该模块主要包括企业信息管理，岗位信息管理，人才查询，密码设置与退出等。
- ③ 客服管理人员模块，该模块主要包括会员审核管理，留言管理，密码设置与退出等。

## 1.3 项目概述

人才招聘系统（Talent Recruitment System, TRS）是一款用于优化招聘流程的智能化平台，旨在高效连接招聘方与求职者，提升人才匹配的精准度和招聘效率。该系统不仅为求职者提供丰富的岗位信息和智能化的查询功能，还为招聘方搭建便捷的简历筛选与管理平台，助力企业快速找到符合需求的人才。为此，本项目旨在打造一个能够快速定位目标岗位和求职者的智能化平台，将零散的招聘信息进行系统化整合，使其更加直观、合理，从而提升招聘与求职双方的工作效率，实现精准匹配。

开发、运行、维护历史：暂无

投资方：山东大学计算机科学与技术学院

需方：山东大学计算机科学与技术学院

开发方：软工加油小组

其他有关文档：暂无

## 1.4 文档概述

本报告仅用于该项目的可行性分析以及醒目规划，项目参与者可随意阅读。

# 2 引用文件

《软件工程》（第4版·修订版）莎丽·劳伦斯·弗里格，乔安妮·M.阿特利 著 人民邮电出版社

## 3 可行性分析的前提

### 3.1 项目的要求

人才招聘系统主要用于人力资源管理，旨在利用计算机技术对求职与招聘信息进行高效管理，包括查询、修改、增加和删除等基本操作。要求主要分为以下几类：

系统功能：

- 招聘方信息管理。
- 求职者个人信息管理。
- 岗位信息查询及相关管理。
- 操作日志管理，确保系统可追溯性。

用户需求：

- 招聘方需求：便捷的职位发布、人才筛选和简历管理功能。
- 求职方需求：高效的职位查询、简历管理及求职申请流程。

数据需求：

- 数据统计分析。
- 自动生成报表。
- 数据导出功能，便于存档与分析。

### 3.2 项目的目标

功能需求：

- 提供 用户注册、登录及个人信息管理功能。
- 实现 职位发布、职位搜索和求职申请 功能。
- 支持 简历上传、在线编辑和管理。
- 提供 招聘流程管理、面试安排和录用管理 功能。
- 具备 数据统计、报表生成和数据导出 能力。
- 实施 系统管理员和普通用户权限管理 机制。

性能需求：

- 确保 系统响应速度快，界面友好，交互流畅。
- 支持 大规模用户并发访问。
- 采用 高效的数据库设计，优化查询性能。

安全需求：

- 加密 存储用户数据，确保信息安全。
- 严格 权限控制，防止未授权访问。
- 防范 SQL 注入、XSS 攻击 等常见安全威胁。

界面设计：

简洁直观，易于操作。  
采用响应式设计，适配不同设备和屏幕尺寸。

技术需求：

后端 采用 Java、Python 或 Node.js 作为技术栈。  
前端 选用 React、Angular 或 Vue.js 进行开发。  
数据库 采用 MySQL、MongoDB 或 PostgreSQL。  
采用 RESTful API 进行前后端交互。

可扩展性：

预留 业务扩展接口，保证未来系统可扩展。  
便于 添加新功能或模块，提升灵活性。

测试需求：

进行 单元测试、集成测试和系统测试，确保稳定性。

文档需求：

编写 用户手册和管理员手册，便于系统操作。  
提供 系统架构设计文档和数据库设计文档，支持维护和升级。

部署需求：

采用 云部署或本地部署 方案。  
编写 自动化部署脚本，简化安装和更新流程。

## 3.3 项目的环境、条件、假定和限制

### 3.3.1 环境

开发环境：代码开发使用 IntelliJ IDEA（jdk17 及以上，Springboot3，vue3）  
测试环境：接口测试使用 Postman，html 页面测试使用 IDE 集成的 Tomcat 和 chromium  
硬件环境：优先选择轻量级服务器进行部署，次选本地部署  
运行环境：Windows10/11

### 3.3.2 条件

该项目需要网络服务器和客户端机器。

### 3.3.3 假设

用户需求：假设用户需求已经明确并且团队已经对需求进行了充分的分析和讨论。

技术选型：假设团队已经确定了合适的技术栈和架构设计。

合作伙伴：假设与外部合作伙伴（如设计团队、产品团队）之间的沟通顺畅。

开发进度：假设开发小组内部沟通顺畅，能够按时完成每周的开发任务。

### 3.3.4 限制：

时间限制：项目截止时间为学期第 16 周，需要按时完成开发和测试且投入时间有限。

功能限制：该项目属于 WEB 软件而不是一般的桌面程序，无法实现桌面软件一些特性。

人力限制：开发小组仅 5 人，并且基本没有相关开发的流程经验。

合作限制：开发小组成员间聚集时间较短。

## 3.4 进行可行性分析的方法

市场可行性分析：

通过对现今人才招聘系统进行调研，分析该市场的需求与趋势；同时确定目标用户的规模、特点和需求，多方面确定系统开发方向。我们可以通过发布网络调研文件，对各方需求进行调研。

经济可行性分析：

进行成本估算，包括项目开发成本、运营成本、维护成本等。

进行收益预测，评估项目的盈利能力和投资回报率。

进行成本与收益的比较，确定项目的经济可行性。

技术可行性分析：

评估项目所需技术是否已经成熟，是否有相关的技术方案和解决方案可供选择。

考虑项目所需的技术资源、技术团队的能力和技术支持等因素。

## 4 可选的方案

### 4.1 原有方案的优缺点、局限性及存在的问题

优点：结构简单，实现较为容易。



缺点及存在的问题：功能过于简陋，使得用户的很多操作，诸如查询不方便。

存在问题：未加入登录验证等安全措施，存在安全隐患。

## 4.2 可重用的系统，与要求之间的差距

可重用的系统要求代码编写遵循严格的编码规范和最佳实践，以确保系统的模块化、可扩展性和可维护性。然而，由于本系统开发团队成员经验尚浅，在开发过程中可能存在编码不够规范、代码复用率较低、模块耦合度较高等问题，从而影响系统可重用性。

以下几个方面存在一定差距：

- ① 代码规范性不足：部分代码未严格遵循统一的编码标准，可能导致可读性和可维护性较差，从而降低了代码的可复用性。
- ② 模块化设计不完善：系统中的部分功能模块耦合度较高，缺乏清晰的接口设计，导致在不同项目或场景下的复用难度增加。
- ③ 文档和注释不完整：部分代码缺乏详细的注释和开发文档，使得后续团队在复用或优化代码时理解难度较大，影响开发效率。
- ④ 测试覆盖率不足：由于测试体系尚未完善，部分模块的单元测试覆盖率较低，可能导致代码在不同环境或新需求下的适应性较弱，影响其复用价值。
- ⑤ 依赖性较强：部分代码对特定框架或库的依赖性较高，使得在不同项目或技术栈中迁移和复用的难度增加。

## 4.3 可选择的系统方案（人员）

用户：

向“后端”发送“投递简历”和“查询岗位”请求

后端：

- ① 接收用户请求
- ② 处理请求并与“数据库”交互（提交请求和返回数据）
- ③ 接收“管理员”的账户增删改查请求
- ④ 接收“企业”的发布岗位和查询简历请求

管理员：

向“后端”发送账户的增删改查请求

企业：

向“后端”发送发布岗位和查询简历请求

数据库：

存储和返回数据给“后端”

## 4.4 可选择的系统方案（模块）

人才招聘系统包含以下模块：

- 登陆模块
- 验证模块
- 用户模块
- 企业模块
- 管理员模块

登陆模块包含以下功能：

- 用户登陆
- 用户注册

验证模块包含以下功能：

- SQL 防注入过滤器
- 登陆状态验证
- 操作权限验证
- 用户身份验证

个人用户模块包含以下功能：

- 简历修改
- 职位查询
- 求职管理
- 账号管理
- 用户个人信息管理

企业用户模块包含以下功能：

- 企业信息管理
- 岗位信息管理
- 人才查询
- 账号管理

管理员模块包含以下功能：

- 会员审核管理
- 账号管理

这些模块和功能共同构成了一个完整的人才招聘系统，涵盖了用户注册、登陆、信息管理、职位查询、求职管理、权限验证等多个方面。

## 4.5 选择最终方案的准则

各个子模块的划分清晰，便于开发；

功能更丰富，更贴合实际需求；  
添加了登陆模块和验证模块，安全性相对更高；  
增添了留言功能，便于项目运行后收集用户反馈进行升级。

## 5 所建议的系统

### 5.1 对所建议的系统的说明

该系统下辖五个模块，分别为登陆模块、验证模块、个人用户（求职者）模块、企业模块和管理员模块。

登陆模块：

实现用户登陆功能和用户登陆状态的更新，以及跳转至对应控制台的功能。

验证模块：

实现用户登陆状态的验证和无权限操作的拦截。

个人用户模块：

实现简历修改、用户个人信息管理、职位查询、求职管理、系统问题反馈、账号管理。

企业用户模块：

实现企业信息管理、岗位信息管理、人才查询、系统问题反馈、账号管理。

管理员模块：

实现会员审核管理、留言管理、用户反馈管理、账号管理。

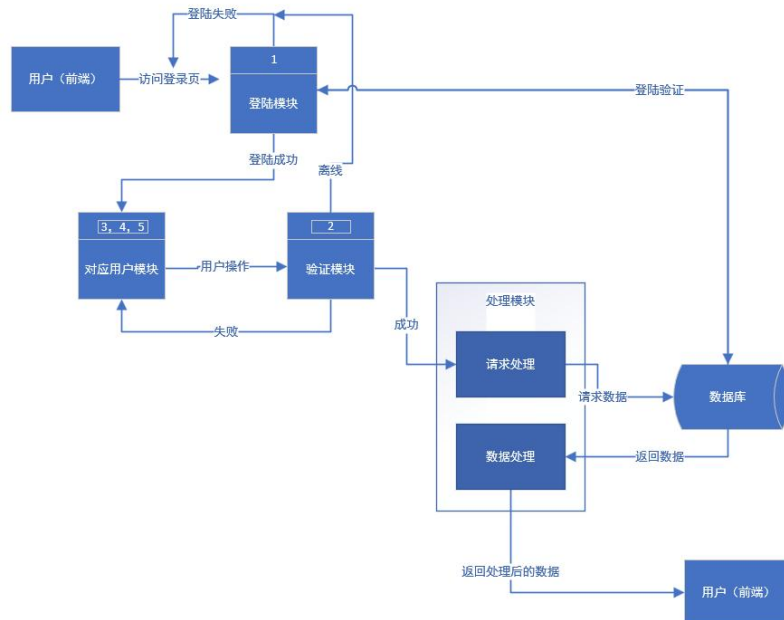
### 5.2 数据流程和处理流程

- ① 用户（前端）访问登录页  
用户进入登录模块：  
如果登录失败，返回用户（前端）。  
如果登录成功，进入验证模块。
- ② 在验证模块：  
用户进行操作。  
如果验证失败，返回对应用户模块。  
如果验证成功，进入处理模块。
- ③ 处理模块包括：  
请求处理：请求数据。  
数据处理：处理请求的数据。

数据库返回数据给处理模块。

处理模块将处理后的数据返回给用户（前端）。

用户可以选择离线操作，返回登录模块。



模块细化流程：

#### ① 用户登陆模块

用户登陆：

用户提交登陆申请。

验证用户状态：

在线用户（D2）：验证用户状态。

离线用户：继续下一步。

查询已注册账号信息：

账户不存在：返回失败。

账户存在：继续下一步。

对密码进行验证：

密码错误：返回失败。

密码正确：继续下一步。

根据账户类型跳转到对应用户模块：

在线用户更新（D2）。

根据账户类型（账户类型）跳转到对应用户模块。

用户注册：

用户提交注册申请。

查询已注册账号信息：

账户存在：返回失败。

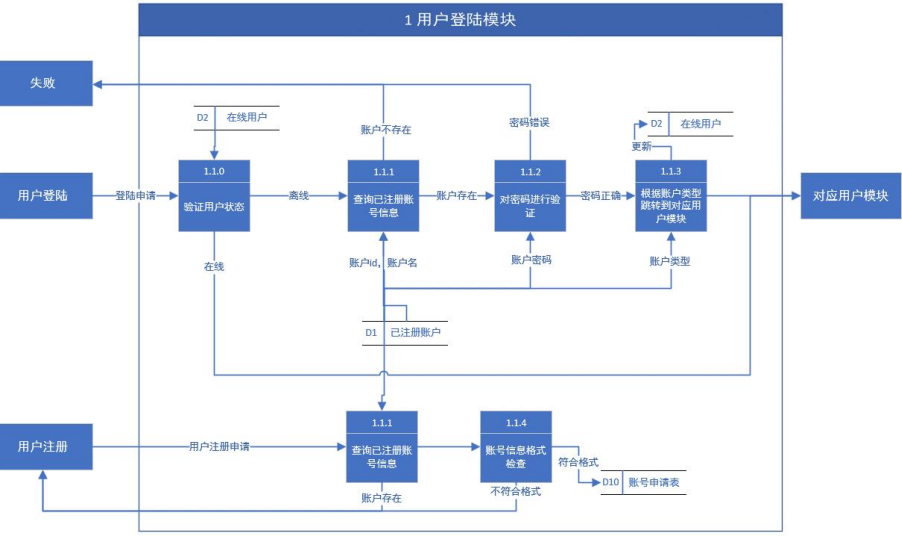
账户不存在：继续下一步。

账号信息格式检查：

符合格式：继续下一步。

不符合格式：返回失败。

账号申请表（D10）：提交账号申请表。



② 验证模块

用户模块：

用户模块是验证流程的起点，用户通过该模块发起登陆请求。

QL 防注入过滤器：

用户请求首先经过 SQL 注入过滤器，以防止 SQL 注入攻击，确保系统安全性。

验证用户状态：

经过过滤后，系统验证用户是否在线（D2）。

如果用户在线，继续进行下一步。

如果用户离线，返回到用户模块。

用户身份验证：

在线用户进行身份验证。

如果身份验证一致，继续进行下一步。

如果身份验证不一致，返回到用户模块。

验证操作权限：

验证用户的权限（D3）。

如果用户有权限，进入处理模块。

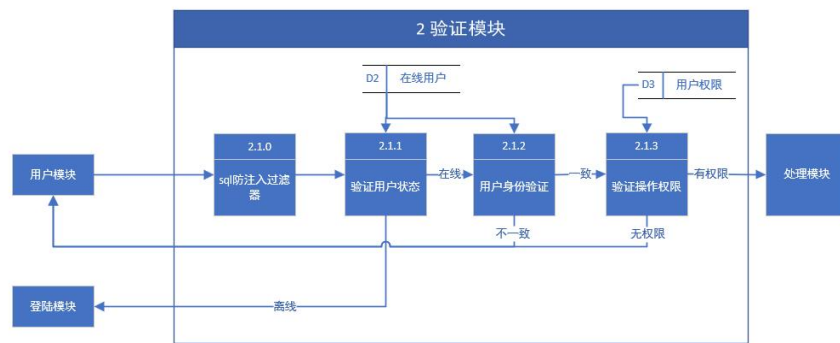
如果用户无权限，返回到用户模块。

处理模块：

用户通过验证后，进入处理模块进行后续操作。

该验证模块通过多个步骤确保用户的身份和权限验证，防止未经授权的访问。流程包括 SQL 防注入过滤、用户状态验证、身份验证和权限验证，确保只有合法

用户才能进入处理模块。



### ③ 个人用户模块

简历修改：

用户可以修改自己的简历信息。

用户个人信息管理：

用户可以管理自己的个人信息。

职位查询：

用户可以查询职位信息。

求职管理：

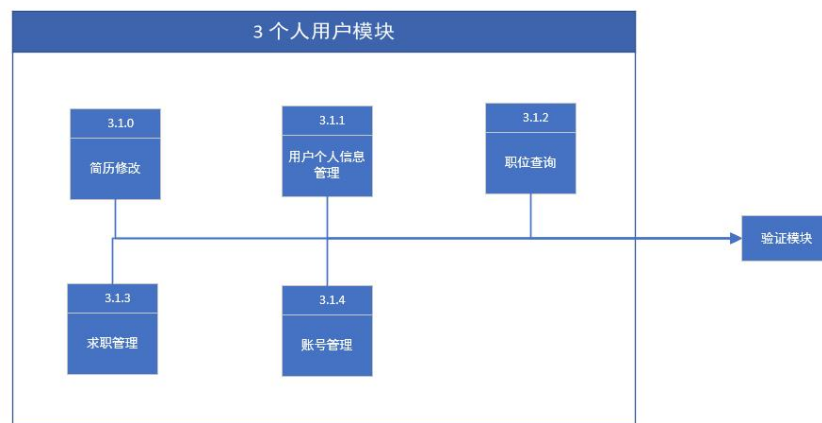
用户可以管理自己的求职申请。

账号管理：

用户可以管理自己的账号信息。

所有这些功能都与验证模块相连。

这意味着在用户执行这些操作之前，系统会通过验证模块进行身份验证，确保用户的身份和权限。



### ④ 企业用户模块

企业信息管理：

企业管理者在此功能中管理企业基本信息，如企业名称、地址、联系方式等。

岗位信息管理：

企业管理者在此功能中管理企业岗位信息，包括发布新岗位、修改现有岗位等。

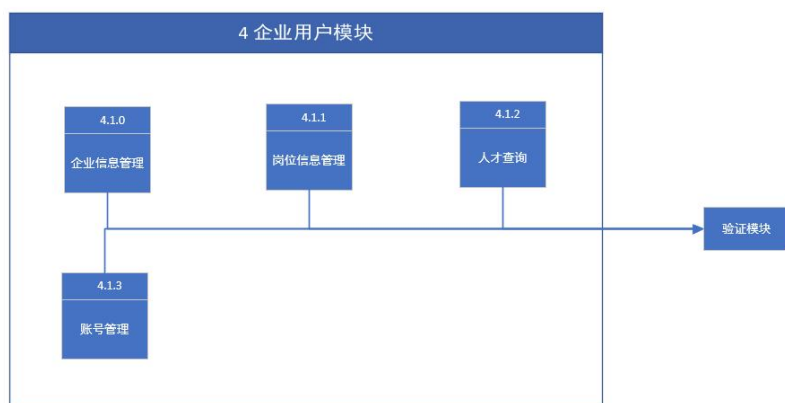
人才查询：

企业管理者在此功能中查询和筛选合适的人才，查看求职者的简历和信息。

账号管理：

企业管理者在此功能中管理企业账号信息，包括账号创建、修改和删除等操作。

该企业用户模块提供了企业管理者在系统中进行各种操作的功能，包括企业信息管理、岗位信息管理、人才查询和账号管理。所有这些操作都需要通过验证模块进行身份验证，以确保安全性和权限控制。



#### ⑤ 管理员模块

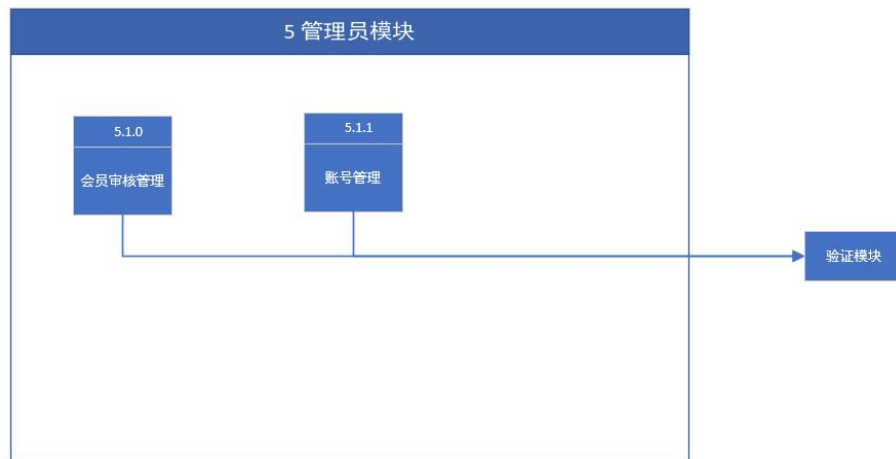
会员审核管理：

管理员在此功能中管理会员审核流程，包括审核会员的注册申请、更新会员状态等。

账号管理：

管理员可以在此功能中管理系统中的账号信息，包括创建、修改和删除账号等操作。

该管理员模块提供了系统管理员在系统中进行各种管理操作功能，包括会员审核管理和账号管理。所有操作都需要通过验证模块进行身份验证，以确保安全性和权限控制。



⑥ 请求处理模块：

用户个人信息管理：

负责管理用户的个人信息。

简历修改：

用户可以修改自己的简历。

职位查询：

用户可以查询职位信息。

求职管理：

用户可以管理自己的求职信息。

账号管理：

用户可以管理自己的账号信息，特别是已注册账户。

企业信息管理：

企业管理者可以管理企业信息。

岗位信息管理：

企业管理者可以管理岗位信息。

人才查询：

企业管理者可以查询人才信息。

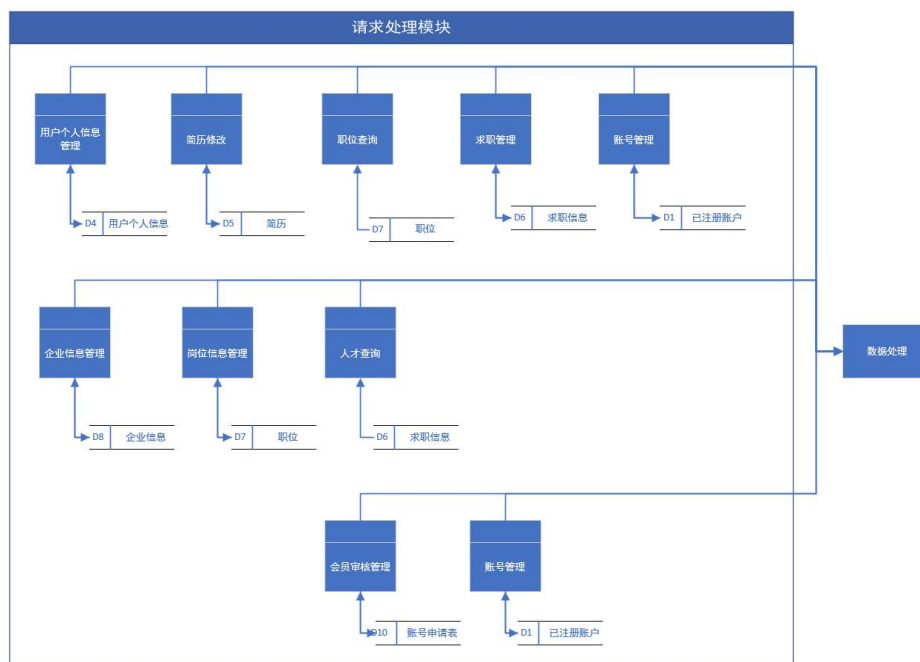
会员审核管理：

管理员可以管理会员审核流程。

账号管理：

管理员可以管理账号信息，特别是已注册账户。

该请求处理模块通过多个子功能模块，涵盖了用户个人信息管理、简历修改、职位查询、求职管理、账号管理、企业信息管理、岗位信息管理、人才查询、会员审核管理和账号管理等功能。所有功能模块都与数据处理模块相连，确保数据的准确性和安全性。





## 5.3 与原系统的比较(若有原系统)

暂无

## 5.4 影响(或要求)

### 5.4.1 设备

多台电脑。  
本地数据库。

### 5.4.2 软件

基于开发环境考量，项目软件运行时必须保证部分软件处于特定的版本范围：  
操作系统版本在 window10 及以上  
jdk 版本在 17 及以上  
MySqlSever 版本在 8.0 及以上  
SpringBoot 版本在 3 以上  
Vue 版本在 3 以上

### 5.4.3 运行

用户操作流程:

管理员：进行会员审核，账号管理，留言管理和用户反馈管理。

HR（企业）：管理企业信息，岗位信息，可查询已在网站注册的求职者

求职者：可修改简历和个人信息，可查询发布的职位和管理自己的求职投递。

源数据的处理：

相关账号数据存储于数据库的各张表中

用户数据要加密存储，确保用户信息安全。

系统要有完善的权限控制，防止未授权访问。

防止 SQL 注入、XSS 等常见安全漏洞。

对数据保存的要求：安全性、完整性、正确性

数据进入系统的过程：

将用户操作转化为对应的调用请求，经项目后端处理后与数据库进行交互。

软件（网站）的安全性：

用户在访问本软件提供的功能时，需进行权限检查，以保证各个用户账号的安全性实现对 SQL 注入攻击的拦截，保证数据库信息的安全。

对数据存储、恢复的处理：

数据存储在 MySQL 数据库中。使用 MySQL 相关功能实现数据的存储以及恢复。

系统失效的后果及恢复的处理办法：

系统失效会导致网站无法访问或出现严重错误（操作的结果与预期不符），严重可导致数据库中数据发生损坏。当系统发生失效时，应当重启软件服务并视情况对数据库进行回滚。

## 5.4.4 开发

使用 Java 语言和 SQL 语言和完成对软件系统后端的开发以及数据库管理，使 HTML，CSS，JS 等语言完成对系统前端页面的开发

开发工具选用：Intelij IDEA, Spring Boot3, Vue3, Element Plus

测试软件使用：Tomcat（网页测试），Postman（接口测试）

数据存储与管理选用：MySQL Sever 8.0

代码规范与质量控制：制定统一的代码规范，确保代码的可读性和可维护性。实施严格的质量控制措施，包括代码审查、单元测试、集成测试等，确保软件质量符合项目要求。

文档编写与培训：编写详细的开发文档，包括系统设计文档、接口文档、测试文档等。为团队成员提供必要的培训和支持，确保他们能够熟练掌握相关技术和工具。

## 5.4.5 环境

项目服务器应在适宜（干燥，适宜温度）的条件下运行；

软件工程在设计和开发软件时通常会对运行环境提出一些要求，以确保软件能够正常运行并达到预期的性能和稳定性。以下是一些常见的软件工程对于运行环境的要求：

操作系统环境要求：

操作系统 Windows

操作系统的版本要求：Windows10/11

硬件环境要求：

硬件配置：双核处理器及以上，2GB 内存及以上，30G 存储空间及以上等。

软件依赖要求：

jdk 版本 17 及以上

MySQLSever 版本在 8.0 及以上

网络要求:

带宽: 5Mbit/s (极小型规模) 及以上。

浏览器要求:

推荐 chromium 内核浏览器

## 5.5 局限性

程序在设计时没有考虑高负载和高并发情况下的稳定性和响应速度,导致在实际运行中可能出现性能瓶颈和响应延迟的问题。

# 6 经济可行性

## 6.1 投资分析

本项目的投资主要涉及开发环境、硬件设备、软件资源、培训、服务器托管等成本。由于团队成员均为在校大学生,资金来源有限,因此需尽可能利用现有资源、免费软件和开源工具,以最低成本完成项目开发。

基本投资分析:

**开发设备:** 团队成员使用各自已有的笔记本电脑或台式机进行开发,不需要额外购置硬件设备。对于性能要求较高的开发任务,我们计划利用学校计算机实验室的资源进行处理。虽然部分成员的电脑性能可能较低,但通过优化开发环境、合理分工,如后端成员使用更强性能的设备,可以有效降低对高端设备的需求。

**开发软件:** 为控制成本,我们将全面采用免费或开源软件进行开发工作。代码编写将使用 VS Code、Eclipse 或 PyCharm 社区版等免费 IDE,版本控制采用 Git 结合 GitHub/Gitee 的学生免费账户,前端设计将利用开源 UI 框架如 Bootstrap 和 Element UI,同时使用 Figma 免费版进行原型设计。开发环境将通过 Docker 进行本地容器化部署,确保团队成员间的环境一致性。所有软件选择均考虑了课程结束后的可持续使用性,避免因试用期到期导致的项目维护问题。

**数据库:** 项目将采用开源数据库解决方案,如 MySQL、PostgreSQL 或 MongoDB,根据具体业务需求选择最合适的数据存储方式。在开发阶段,团队成员将在本地安装数据库实例进行开发和测试,无需额外成本。若需进行在线演示,我们可以申请使用学校提供的数据库服务或利用云数据库服务商提供的免费额度。为防止数据丢失,我们将建立完善的数据备份策略,利用 Git 或其他版本控制工具定期备份重要数据。

**服务器：**开发阶段主要使用本地环境进行测试，不产生额外费用。在课程展示阶段，我们可以选择使用学校提供的内网服务器（如实验室服务器或学院开发平台）或申请云服务商提供的学生优惠套餐。若选择后者，如阿里云或腾讯云学生机，月均成本在 10-30 元之间，可由团队成员共同分担，人均成本极低。对于仅需短期展示的情况，我们还可以考虑使用 Heroku 免费额度或 Vercel 免费部署方案，进一步降低成本。

**API 费用：**对于可能涉及的人工智能简历解析、OCR 识别等功能，我们将优先使用提供免费额度的 API 服务。例如，可以利用开源 NLP 库（如 NLTK、spaCy）进行本地文本处理，或使用阿里云、腾讯云、百度 AI 开放平台提供的免费 API 额度。通过合理设计 API 调用策略，如实施缓存机制和批量处理请求，我们可以显著减少 API 调用次数，控制成本。同时，我们将设计功能降级方案，在 API 不可用或超出免费额度时，自动切换到本地简化算法处理，确保系统持续运行。

**学习资料：**团队将充分利用学校图书馆资源、电子期刊数据库和各类开源社区文档满足学习需求。我们将激励团队成员利用官方教程、MOOC 平台免费课程进行自学，并通过 StackOverflow、CSDN 等技术社区解决开发过程中遇到的问题。此外，团队内部将建立知识共享机制，定期组织技术分享会，避免重复学习成本，提高整体学习效率。技术文档、电子书和在线教程主要依靠开源社区、免费课程和学校提供的教学资源，如 Coursera、YouTube、CSDN 等，尽可能避免额外支出。

其他一次性和非一次性投资：

**技术管理费：**项目的代码托管将使用 GitHub 或 Gitee 的免费版进行版本控制，团队协作将采用 Notion、Trello 基础版、微信群等免费工具，文档管理则依托腾讯文档、石墨文档等免费协作平台。项目进度跟踪可通过 GitHub Projects 或简单的电子表格实现，这些工具都不需要额外付费，能够满足课程项目的基本管理需求，同时培养团队成员的项目管理能力。

**培训费：**团队成员将主要通过自学和互相指导的方式提升技能，充分利用学校提供的线上课程资源和技术讲座，积极参加校内技术社团活动获取经验和指导。针对项目中可能遇到的特定技术难点，我们将寻找相关开源项目源码进行学习和参考，这种学习方式不仅节约培训成本，还能培养团队成员的自主学习能力和解决问题的能力，为未来职业发展打下基础。

**管理费：**团队沟通将主要通过微信、QQ 群、腾讯会议等免费线上工具进行，定期线下讨论则利用学校空闲教室、图书馆或自习室，项目展示与答辩也将使用学校提供的场地和设备。由于不需要租赁办公场地和购买专业管理软件，项目的管理成本几乎为零，这符合大学生课程项目的实际情况，也锻炼了团队在有限资源下的管理能力。

**人员工资：**本项目的开发成员均为参与同一课程的学生，主要目标是完成课程任务和提升个人技能，团队成员的贡献将以获取学分和积累实践经验为主要回报。若项目参加比赛并获得奖金或申请到创新项目资助，可根据团队成员的贡献度进行适当分配，作为额外激励，但这并非项目的主要目标或预期收益。

**差旅费：**由于项目主要在校内进行，日常活动不会产生差旅费用。如需参加校外比赛或项目展示，我们将优先申请学校提供的创新创业基金或竞赛补助，尽可能减轻团队成员的经济负担。同时，我们也会尽量选择线上参会方式或就近展示机会，减少不必要的交通和住宿支出，确保项目在有限预算内顺利完成全部活动。

## 6.2 预期的经济效益

本项目的经济效益主要体现为一次性收益，如课程成绩、比赛奖金等、非一次性收益，如个人职业发展、项目商业化潜力等，以及难以量化但具有长期价值的附加收益。

### 6.2.1 一次性收益

**课程成绩：**完成高质量项目可为团队成员带来优秀的课程评分，有效提高 GPA，这对大学生而言是最直接的收益。优秀的项目成果可能获得课程设计一等奖或优秀作业等荣誉，增加学生档案中的亮点。在某些学校，表现突出的课程项目还可能被推荐参加校级或省级比赛，为团队成员赢得额外学分和荣誉。此外，本项目的开发经验也可作为毕业设计的前期准备，减少后续工作量，提高毕业设计质量。

**竞赛奖金：**团队可将项目进一步优化后提交至“挑战杯”、“互联网+”等创新创业比赛，这类比赛通常设有丰厚奖励。校级比赛奖金一般在 500-3000 元不等，若能晋级省级或国家级比赛，奖金可能达到 5000-20000 元。除直接奖金外，获奖项目还有机会获得学校创新基金支持，金额通常在 5000-10000 元，这些资金可用于项目进一步开发和团队成员激励，为项目可能的商业化探索提供初始资金。

**实践学分：**许多高校已建立创新创业学分认定机制，允许学生用创新项目成果置换部分选修课学分。团队成员可以申请将本项目认定为创新创业学分，计入综合评价体系。这不仅减轻了正常课业负担，还能提升综合测评成绩，增加奖学金获取机会，间接带来经济收益，同时为学生未来发展提供更多选择空间。

### 6.2.2 非一次性收益

**简历提升：**将实际项目经历写入简历，能够有力展示团队成员的实际开发能力和项目经验。相比纯理论学习，实践项目更受企业青睐，特别是在技术岗位招聘中。团队成员可以通过此项目展示自己的技术栈、解决问题能力和团队协作经验，大幅提升求职竞争力。这种简历加分不仅适用于毕业求职，也对在校期间申请实习岗位极为有利，为团队成员创造更多接触行业的机会。

**实习与就业机会：**优秀的课程项目可能引起企业关注，为团队成员提供实习或全职工作机会，尤其是那些关注校园人才的互联网公司 and 创业企业。通过项目展示，团队成员有机会接触到行业专家和企业招聘人员，建立初步的职业人脉。这些机会的价值难以

量化,但对大学生职业发展的长期影响不言而喻,可能带来数万甚至数十万元的职业收益差异。

**技术积累:**项目开发涉及后端开发、前端交互设计、数据库管理、API 集成等多个技术方向,团队成员将获得全面的技术锻炼。这些技能不仅适用于当前项目,也为后续更复杂的项目开发打下坚实基础。在快速发展的 IT 行业,持续的技术积累是保持竞争力的关键,本项目为团队成员提供了在实际开发环境中应用和提升技术能力的宝贵机会,其价值将在未来职业生涯中不断显现。

**人脉拓展:**团队成员通过项目开发过程、比赛参与、项目展示等活动,有机会接触同领域的学生、教师和行业专家,建立广泛的人脉网络。这些联系可能带来未来的合作机会、技术交流和职业推荐。在当今竞争激烈的就业市场,人脉资源的价值日益凸显,本项目为团队成员创造了建立专业人脉的平台,这种社会资本的积累将对未来职业发展产生长期积极影响。

### 6.2.3 不可定量的收益

**技术成长:**团队成员通过实际项目开发将理论知识转化为实践能力,提升代码编写、系统设计和问题解决能力。这种在真实项目中的学习体验远优于课堂教学,能够培养工程思维和技术直觉,为未来的技术生涯奠定基础。团队成员还能学习到项目管理、需求分析、质量控制等软技能,这些能力在职场中同样重要,但很难通过常规课程获得。

**创新思维培养:**开发创新性项目能够激发团队成员的创造力和批判性思维,培养发现问题、分析问题和解决问题的能力。这种思维方式一旦形成,将影响学生的整个职业生涯,使他们在未来工作中更具创新性和适应性。在当今快速变化的技术环境中,创新思维比特定技术知识更具持久价值,是学生最宝贵的收获之一。

**团队协作经验:**项目开发过程中,团队成员将学习如何有效沟通、分工协作、解决冲突和共同解决问题。这种团队协作经验对于未来进入企业工作或自主创业都有重要帮助,能够显著提高团队成员的沟通和管理能力。许多企业在招聘时特别看重应聘者的团队协作能力,本项目为团队成员提供了培养这一关键能力的实践平台。

### 6.2.4 收益/投资比

由于本项目投入成本极低(主要利用现有资源和免费工具),而收益体现在学习成果、课程成绩和职业发展等多个方面,因此投资回报率(ROI)极高。即使仅考虑课程学分和成绩这一直接收益,投入产出比也是显著正向的。如果将长期职业发展收益计入考量,本项目的价值更是难以估量。对于大学生而言,这种"低投入、高收益"的项目是最理想的学习和实践方式,能够在有限资源条件下最大化个人成长和能力提升。

## 6.2.5 投资回收周期

本项目的投资回收周期可分为短期、中期和长期三个阶段。短期回报主要体现在课程结束后获得的成绩评定、技术成长和可能的比赛奖金，这些收益通常在 3-6 个月内即可实现。中期回报包括实习机会、简历加分和人脉积累，通常在 6-18 个月内逐步显现。长期回报则体现在毕业后的就业优势、职业发展和创业可能性上，这种收益可能持续数年甚至更长时间。相比传统投资项目，本课程项目的回收周期较短，且风险极低，非常适合大学生团队实施。

## 6.3 市场预测

### 6.3.1 短期市场

在课程项目背景下，该人才招聘系统的主要"市场"是课程评审和校园展示环境。系统的主要用途是展示团队的技术能力和创新思维，目标受众是课程导师、评审组和可能参观项目展的企业 HR。在这一环境中，系统的评价标准主要是技术实现的完整性、创新性和实用性，而非商业价值。由于定位为学术项目，市场竞争压力相对较小，主要挑战来自团队内部的技术实现和项目管理。这种相对宽松的环境为团队提供了充分的学习和尝试空间，允许在保证基本功能的同时进行创新尝试。

### 6.3.2 长期市场

从长远角度看，随着团队成员技能的提升和系统功能的完善，该项目可能具有一定的实际应用价值。当前市场上，大型招聘平台如智联招聘、BOSS 直聘等主要服务于成熟企业，而中小企业和校园招聘市场仍存在诸多痛点，如招聘成本高、匹配效率低、简历筛选耗时等。本项目可以针对这些细分市场，特别是高校内部就业服务中心、学生社团招新、校园创业团队等场景，提供轻量级、智能化的人才匹配解决方案。

通过对项目进行适当调整和优化，团队可以探索多种发展路径：一是将系统作为校内就业服务的辅助工具，服务于学校就业指导中心；二是针对学生社团和校园活动的人才招募需求，提供定制化解决方案；三是关注特定行业或岗位的招聘痛点，开发针对性功能模块。这些方向虽然市场规模有限，但竞争相对较小，更适合学生团队起步和积累经验。

如果项目表现优异并获得一定认可，未来还可能吸引天使投资或创业孵化器的关注，为团队成员提供进一步发展的机会。即使不考虑商业化，项目中开发的部分模块和算法也可以作为开源项目贡献给社区，提升团队成员在技术社区的影响力和知名度。

## 7 技术可行性(技术风险评价)

### 7.1 技术风险

技术选型不当：

学生团队可能对某些技术栈（如 Spring Boot、Vue.js、MySQL 等）不够熟悉，导致开发效率低下或代码质量不高。团队成员可能对 Vue.js 的前端框架不熟悉，导致前端开发进度缓慢，或者对 Spring Boot 的后端框架理解不深，导致后端接口设计不合理。选择团队成员较为熟悉的技术栈，避免使用过于复杂或未接触过的技术。如果必须使用新技术，提前安排学习时间。团队成员可以通过在线课程、技术文档或实践项目来快速掌握新技术。

开发经验不足：

团队成员可能缺乏实际项目开发经验，导致在需求分析、系统设计、编码实现等环节出现问题。需求分析阶段可能遗漏某些关键功能，系统设计阶段可能设计出不符合实际需求的架构，编码实现阶段可能写出低效或不稳定的代码。在项目初期开展多次讨论与会议，明确需求分析等部分内容，在开发编码阶段与同学或者老师请教开发经验，并学习网上相关项目的编程过程积累开发经验。可以使用代码审查工具进行代码审查，确保每个提交的代码都符合规范。

系统性能问题：

由于实验项目通常不会进行大规模的性能测试，系统在高并发或复杂操作下可能出现性能瓶颈。当多个用户同时访问系统时，数据库查询速度变慢，或者前端页面加载时间过长，影响用户体验。在开发过程中进行基本的性能测试，确保系统能够满足实验要求。可以使用轻量级的测试工具（如 Postman）进行接口测试。制定统一的代码规范，定期进行代码审查，确保代码的可读性和可维护性。

### 7.2 时间管理风险

项目进度缓慢：

由于学生团队成员有其他课程和学习任务，可能导致项目开发时间不足，无法按时完成。团队成员可能因为期中考试或其他课程作业而无法全身心投入项目开发，导致项目进度滞后。可以将项目分为需求分析、系统设计、编码实现、测试和交付等阶段，并为每个阶段设定明确的时间节点。在项目初期制定详细的时间计划，明确每个阶段的任务和时间节点。可以使用项目管理工具来跟踪进度。每周或每两周进行一次进度检查，及时发现并解决问题，避免任务积压。

任务分配不合理：

任务分配不均或任务难度超出个别成员的能力范围，导致部分任务无法按时完成。根据团队成员的技术能力和兴趣合理分配任务，确保每个人都能胜任自己的工作。可以将前端开发任务分配给对 Vue.js 较为熟悉的成员，后端开发任务分配给对 Spring Boot 较为熟悉的成员。



## 7.3 需求变更风险

在实验项目中，需求可能不够明确或频繁变更，导致开发方向偏离原计划。指导老师可能在项目中期提出新的需求，导致团队需要重新调整开发计划。学生团队可能希望实现过多的功能，导致项目复杂度增加，无法按时完成。团队可能希望在系统中加入论坛功能、消息通知功能等，导致开发工作量大幅增加。在项目初期与指导老师充分沟通，明确项目的核心需求和优先级，避免功能过多或需求模糊。在开发过程中，尽量避免频繁变更需求。如果必须变更，需经过团队讨论并评估对项目进度的影响，并将不必要以及严重影响项目进度的需求进行冻结。可以在项目中期进行一次需求评审，确保需求变更不会对项目进度产生重大影响。

## 7.4 团队协作风险

团队成员之间沟通不足，可能导致任务分配不清或开发进度不一致。学生团队成员可能有不同的课程安排，导致团队聚集时间有限，影响项目进展。每周安排固定的团队会议，讨论项目进展、遇到的问题以及下一步计划。使用协作工具（如 GitHub、Slack、钉钉等）进行任务分配和沟通，确保团队成员能够随时同步信息。在项目初期明确每个成员的职责和任务，避免任务重叠或遗漏。

## 7.5 测试与质量风险

测试不充分：

由于时间和资源有限，可能无法进行充分的测试，导致系统存在未发现的 bug。在开发过程中编写单元测试，确保每个模块的功能正确性。在系统集成阶段进行基本的集成测试，确保各个模块能够协同工作。

质量不达标：

代码质量不高，可能导致系统运行不稳定或难以维护。代码中可能存在冗余、重复或低效的部分，导致系统性能下降。定期进行代码审查，确保代码符合规范并具有较高的质量。

## 7.6 文档与验收风险

文档不完整：

由于时间紧张，可能忽略文档的编写，导致项目交付时缺乏必要的文档支持。在项目初期制定文档编写计划，确保每个阶段都有相应的文档支持，如需求文档、设计文档、测试文档等。

验收项目不符合要求：

项目验收时可能不符合实验要求，导致评分较低。在项目交付前，定期检查交付物是否符合实验要求，确保最终交付物完整且符合标准。

## 7.7 技术债务风险

由于时间紧迫，可能采用一些临时解决方案，导致技术债务积累，影响系统的可维护性和扩展性。尽量避免采用临时解决方案，确保每个功能模块的设计和实现都具有较高的质量。如果必须采用临时方案，需记录技术债务，并在后续开发中进行修复。

## 7.8 外部依赖风险

项目中使用的第三方库或工具可能存在兼容性问题或未发现的 bug，影响系统开发。选择稳定且广泛使用的第三方库或工具，避免使用过于新颖或未经验证的版本。在使用第三方库或工具前，进行基本的测试，确保其能够满足项目需求。

## 7.9 未来功能扩展风险

系统设计时未考虑未来的功能扩展，导致后续添加新功能时遇到技术障碍。例如，系统可能设计为单体架构，导致后续添加新功能时遇到技术障碍。在系统设计时采用模块化架构，确保各个功能模块之间的独立性。这样在后续添加新功能时，可以减少对现有系统的影响。可以将系统设计为前后端分离的架构，确保前后端模块之间的独立性。在技术选型时，优先选择支持高扩展性的技术栈，确保系统能够方便地添加新功能或模块。可以选择支持高扩展性的 Spring Boot 和 Vue.js 技术栈，确保系统的可扩展性。

# 8 法律可行性

法律合规性：确保项目符合当地法律法规，包括数据保护、隐私保护等方面的合规性。例如：

### ① 劳动法：

人才招聘系统在招聘过程中必须遵循《劳动法》的相关规定。招聘广告应真实、合法，不得包含虚假或者引人误解的内容，确保招聘信息的准确性和透明度，避免误导求职者。同时，在招聘过程中，应确保招聘流程的公平、公正，避免因性别、年龄、民族等因素产生的歧视。

### ② 反就业歧视法

根据《反就业歧视法》，人才招聘系统应避免在招聘过程中出现任何歧视性行为或语言。系统设计应确保招聘标准和流程的公平性，不对特定群体造成不公平待遇。例如，在筛选简历和面试过程中，应依据岗位需求和能力进行评估，而非其他无关因素。

### ③ 个人信息保护法

人才招聘系统在处理求职者的个人信息时，必须严格遵守《个人信息保护法》。该法律要求企业在收集、存储、使用和共享个人信息时，必须获得用户的充分知情同意，并遵循最小必要原则。系统应确保求职者的信息仅用于招聘目的，并采取适当的安全措施防止信息泄露。

## 9 用户使用可行性

### 组织结构：

了解用户单位的组织结构，包括各部门之间的关系、决策层次等。这有助于确定系统实施后的管理流程和信息流动。

### 工作流程：

分析用户单位的工作流程，了解各个部门或岗位之间的协作方式和信息交流方式。新系统应该能够整合到现有的工作流程中，提高工作效率。

### 信息安全：

评估用户单位对信息安全的重视程度，包括数据保护、权限管理等方面。确保新系统能够满足用户单位的信息安全需求，防止数据泄露和信息丢失。

### 管理支持：

确定用户单位管理层对新系统实施的支持程度，包括预算支持、资源分配等。管理支持是新系统成功实施的关键因素之一。

### 员工素质：

评估使用人员的技能水平、学习能力和适应能力。不同员工可能具有不同的素质，需要根据实际情况设计培训计划。

### 培训需求：

确定使用人员对新系统的培训需求，包括系统功能操作、数据输入、故障处理等方面。制定针对性的培训计划，确保员工能够快速上手并熟练使用新系统。

### 培训方式：

选择适合的培训方式，可以是面对面培训、在线视频教程、培训手册等。根据使用人员的学习习惯和工作安排，设计多样化的培训方式。

## 10 其他与项目有关的问题

由于项目周期的限制，部分设想功能将根据未来开发进度选择性实现。

未来计划添加的功能：

- \* 问题反馈
- \* 留言功能
- \* 操作日志记录
- \* 企业分组权限：一个企业账户可分属于多个组，其账户权限为所有组的并集。
- \* 企业账户权限分级：顶级企业账户下有多个子账户，每个账户仅有顶级账户部分功能。

\* 求职者论坛：内置论坛功能，求职者可在其中发布问题贴和经验贴等。

## 11 注解

### MySQL

类型：关系型数据库管理系统（RDBMS）

特点：开源、免费（社区版）。支持标准的 SQL 语言。支持事务处理、触发器、存储过程等高级功能。适用于中小型应用，也可以处理大规模数据（通过优化和集群）。

用途：存储和管理结构化数据（如表、行、列）。常用于 Web 应用的后端数据存储，如用户信息、订单数据等。

典型场景：与 Spring Boot 结合，作为 Java 应用的后端数据库。通过 SQL 语句进行数据的增删改查（CRUD）。

### IntelliJ IDEA

类型：集成开发环境（IDE）

特点：由 JetBrains 开发，支持多种编程语言（如 Java、Kotlin、Python、JavaScript 等）。提供智能代码补全、代码分析、重构、调试、版本控制集成等功能。社区版免费，旗舰版支持更多高级功能（如 Spring Boot、数据库工具）。

用途：主要用于 Java 开发，尤其是 Spring Boot 项目。支持前端开发（如 Vue、React）和数据库管理。

典型场景：开发 Spring Boot 后端应用。调试和运行 Java 代码。

### Spring Boot

类型：Java 框架

特点：基于 Spring 框架，简化了 Spring 应用的开发。提供自动配置（Auto-configuration），减少手动配置。内嵌 Tomcat 服务器，无需单独部署。支持 RESTful API 开发、数据库访问（如 JPA、MyBatis）、安全控制（Spring Security）等。

用途：快速构建独立、生产级的 Java 应用。常用于开发 Web 应用的后端服务。

典型场景：开发 RESTful API，供前端（如 Vue）调用。与 MySQL 等数据库集成，进行数据管理。

### Vue

类型：JavaScript 框架

特点：轻量级、渐进式框架，易于上手。支持组件化开发，提高代码复用性。响应式数据绑定，自动更新 UI。支持单页应用（SPA）开发。

用途：构建现代 Web 应用的前端界面。与后端（如 Spring Boot）通过 API 交互。

典型场景：使用 Vue 开发用户界面，结合 Element Plus 组件库快速构建页面。通过 Axios 等工具与后端 API 通信。

### Element Plus

类型：UI 组件库

特点：基于 Vue 3 开发，提供丰富的 UI 组件（如表单、表格、按钮、对话框等）。设计风格简洁，易于定制。支持国际化、主题切换等功能。

用途：快速构建美观、一致的前端界面。减少开发者从零开始编写 UI 组件的工作量。

典型场景：在 Vue 项目中使用 Element Plus 组件库开发管理后台、表单页面等。

### Tomcat

类型：Web 服务器和 Servlet 容器

特点：开源、轻量级。支持 Servlet 和 JSP 规范。可以作为独立的 Web 服务器运行，也可以与其他服务器（如 Apache、Nginx）集成。

用途：部署和运行 Java Web 应用。处理 HTTP 请求，执行 Servlet 和 JSP。

典型场景：部署 Spring Boot 应用（Spring Boot 默认内嵌 Tomcat）。运行传统的 Java Web 项目（如基于 Servlet 的应用）。

### Postman

类型：API 开发工具

特点：支持发送 HTTP 请求（GET、POST、PUT、DELETE 等）。提供请求参数、头部、身份验证等配置。支持自动化测试和脚本编写。可以保存和分享 API 集合。

用途：测试和调试后端 API。模拟前端请求，验证接口功能。

典型场景：测试 Spring Boot 开发的 RESTful API。调试接口返回的数据格式和状态码。

### Tomcat

类型：Web 服务器和 Servlet 容器

特点：开源、轻量级。支持 Servlet 和 JSP 规范。可以作为独立的 Web 服务器运行，也可以与其他服务器（如 Apache、Nginx）集成。

用途：部署和运行 Java Web 应用。处理 HTTP 请求，执行 Servlet 和 JSP。

典型场景：部署 Spring Boot 应用（Spring Boot 默认内嵌 Tomcat）。运行传统的 Java Web 项目（如基于 Servlet 的应用）。

### Chromium

类型：开源 Web 浏览器项目

特点：由 Google 主导开发，是许多浏览器的基础（如 Google Chrome、Microsoft Edge）。支持最新的 Web 标准和技术（如 HTML5、CSS3、JavaScript）。提供高性能的渲染引擎（Blink）和 JavaScript 引擎（V8）。

用途：作为浏览器核心，用于浏览网页和运行 Web 应用。开发者可以使用 Chromium 进行 Web 开发和调试。

典型场景：开发者使用 Chromium 或基于 Chromium 的浏览器（如 Chrome）进行前端开发和调试。用户使用基于 Chromium 的浏览器访问 Web 应用。

### Postman

类型：API 开发工具

特点：支持发送 HTTP 请求（GET、POST、PUT、DELETE 等）。提供请求参数、头部、身份验证等配置。支持自动化测试和脚本编写。可以保存和分享 API 集合。

用途：测试和调试后端 API。模拟前端请求，验证接口功能。

典型场景：测试 Spring Boot 开发的 RESTful API。调试接口返回的数据格式和状态码。

### Notion

类型：笔记和项目管理工具

特点：支持多平台（Web、桌面、移动端）。提供笔记、任务管理、数据库、看板等多种功能。支持团队协作和内容共享。

用途：用于个人知识管理、团队项目管理、文档编写等。

典型场景：开发者使用 Notion 记录学习笔记、规划项目任务。团队使用 Notion 进行项目协作和文档管理。

### Trello

类型：项目管理工具

特点：基于看板（Kanban）的任务管理工具。支持任务卡片、列表、标签、截止日期等功能。提供团队协作和任务分配。

用途：用于个人任务管理和团队项目管理。

典型场景：开发团队使用 Trello 管理项目任务，跟踪任务进度。个人使用 Trello 规划日常任务和目标。

### Stack Overflow

类型：编程问答社区

特点：全球最大的编程问答平台。用户可以提问、回答问题、投票和评论。内容涵盖各种编程语言、框架和工具。

用途：帮助开发者解决编程问题，学习新技术。

典型场景：开发者在遇到编程问题时，在 Stack Overflow 上搜索解决方案或提问。通过阅读高质量答案提升编程技能。

### Apache JMeter

类型：性能测试工具

特点：开源、跨平台。支持多种协议（如 HTTP、FTP、JDBC 等）。可以模拟高并发用户负载，测试系统性能。提供图形化界面和脚本化测试。

用途：用于 Web 应用、数据库、API 等的性能测试和负载测试。

典型场景：开发团队使用 JMeter 测试 Web 应用在高并发情况下的性能表现，找出系统瓶颈。

### JIRA

类型：项目管理和问题跟踪工具

特点：由 Atlassian 开发，支持敏捷开发（如 Scrum、Kanban）。提供任务管理、问题跟踪、 workflow 定制、报告生成等功能。支持团队协作和集成其他开发工具（如 Confluence、Bitbucket）。

用途：用于软件开发项目的任务管理、问题跟踪和团队协作。

典型场景：开发团队使用 JIRA 管理开发任务、跟踪 Bug 和生成项目报告。

### StarUML

类型：UML 建模工具

特点：支持多种 UML 图（如类图、用例图、时序图等）。提供代码生成和反向工程功能。界面简洁，易于使用。

用途：用于软件设计和系统建模，帮助开发者可视化系统结构和行为。

典型场景：开发者在设计软件系统时，使用 StarUML 绘制 UML 图，帮助团队理解系统架构。

#### Visual Paradigm

类型：UML 建模和业务建模工具

特点：支持多种建模语言（如 UML、BPMN、ERD 等）。提供团队协作、代码生成、数据库设计等功能。功能强大，适合复杂项目。

用途：用于软件设计、业务流程建模、数据库设计等。

典型场景：开发团队使用 Visual Paradigm 进行系统设计、业务流程建模和数据库设计，支持从设计到代码的全流程开发。

## 附录

### 一、CASE 工具在本项目中的应用（实验二）

Visual Paradigm：用于数据库 ER 图、类图设计，提升系统建模能力。

StarUML：用于绘制系统流程图，帮助开发团队理解系统架构。

JIRA：用于任务拆解和项目进度管理，提高敏捷开发效率。

Apache JMeter：用于模拟高并发场景，优化系统性能。

注：上述均在项目中体现（具体各种工具的调研和分析用途已整理但未在此文档中展示）

### 二、SCRUM 在本项目中的应用（实验三）

采用项目管理工具（JIRA）进行任务拆解、进度管理。

设计 2 周 Sprint 周期，确保开发进度可控。

进行需求优先级排序，提高敏捷性。

需求分析与设计阶段：

采用传统开发模型（瀑布模型）为主，确保需求明确和文档完备，与指导老师充分沟通，明确项目的核心需求和优先级。制定详细的项目计划，明确每个阶段的任务和时间节点。

开发与测试阶段：

将项目分解为多个迭代，如每 2 周一个迭代，每个迭代交付一个可用的功能模块。使用 Scrum 框架，进行每日站会、迭代计划会、迭代评审会和迭代回顾会等形式，确保团队协作和进度跟踪。在每个迭代结束时，进行演示和反馈，确保功能符合需求。

文档与交付阶段：

在项目结束时，整理和完善文档，确保交付物符合课程要求。编写用户手册、管理员手册、系统设计文档和测试文档，便于后续维护和验收。

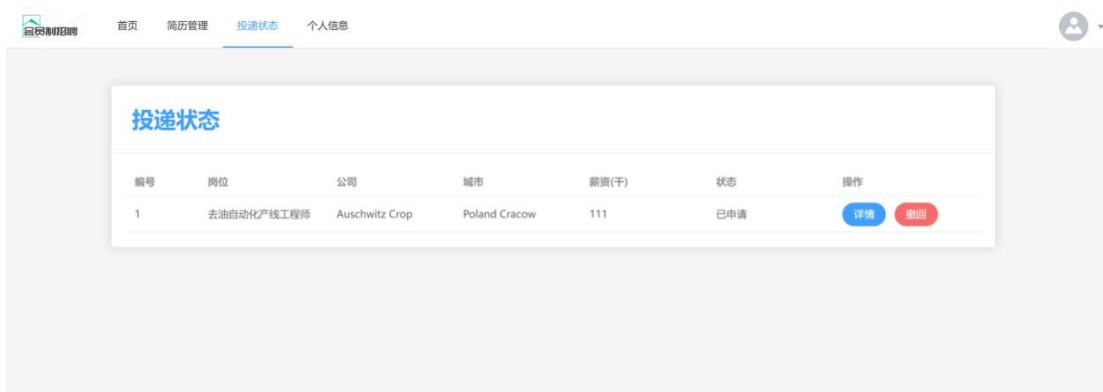
注：本项目还是以瀑布模型为主，吸取 SCRUM 敏捷模型的优点长处加以应用；细化风险管理见 [6 技术可行性](#)

### 三、预期实现界面

登入界面：

A login interface mockup. On the left is a blue-tinted background image of a laptop and keyboard. On the right is a white login form titled '登录'. The form includes a dropdown for '账户类型' (Account Type) with the placeholder '请选择账户类型', a text input for '请输入用户名' (Please enter username), a text input for '请输入密码' (Please enter password), a checkbox for '记住我' (Remember me), and a link for '忘记密码?' (Forgot password?). A blue '登录' (Login) button is at the bottom. Below the button is a link for '注册' (Register).

投递页面：

A mockup of a job application status page. At the top is a navigation bar with links: '会员招聘' (Member Recruitment), '首页' (Home), '简历管理' (Resume Management), '投递状态' (Application Status), and '个人信息' (Personal Information). The main content area is titled '投递状态' (Application Status) and contains a table with the following data:

编号	岗位	公司	城市	薪资(千)	状态	操作
1	去油自动化产线工程师	Auschwitz Crop	Poland Cracow	111	已申请	<a href="#">详情</a> <a href="#">撤回</a>

填写界面：



会员制招聘

首页简历管理投递状态个人信息

基本资料

登录名称

\* 用户昵称

性别

请选择

学历

毕业院校

专业

\* 用户邮箱

电话

自我介绍

提交修改

验证界面：

会员制招聘

首页简历管理投递状态个人信息

旧密码

请输入旧密码

\* 新密码

请输入新密码

再次输入新密码

请再次输入新密码

提交

清空