

# 上海大学

## 计算机工程与科学学院

### 计算机应用大型作业

### 实践报告



课题名称： 《募农帮》微信小程序

组长：学号 20121034 姓名 胡才郁 成绩：

组员：学号 20121036 姓名 黄逸弘 成绩：

学号 20121054 姓名 徐鼎力 成绩：

学号 20121706 姓名 张俊雄 成绩：

指导教师签名： \_\_\_\_\_

2023 年 8 月

<b>第一章 目的、要求和内容</b>	<b>5</b>
1.1 项目目的	5
1.2 项目要求	5
1.2.1 系统设计要求	5
1.2.2 系统开发要求	6
1.3 项目内容	6
<b>第二章 可行性报告</b>	<b>8</b>
2.1 选题背景	8
2.1.1 社会现状分析	8
2.1.2 市场分析	8
2.2 可行性分析	9
2.2.1 技术可行性	9
2.2.2 经济可行性	9
2.2.3 操作可行性	10
<b>第三章 需求分析</b>	<b>11</b>
3.1 工程概述	11
3.1.1 待开发软件的一般概述	11
3.1.2 待开发软件的功能	11
3.1.3 运行环境	12
3.1.4 条件与限制	12
3.2 系统综合要求	12
3.2.1 功能需求	12
3.2.2 外部接口需求	14
3.3 数据描述	14
3.3.1 静态数据	14
3.3.2 动态数据	15
3.3.3 数据库	15
3.3.4 数据流图	15
<b>第四章 概要设计</b>	<b>17</b>

4.1 前端概要设计 .....	17
4.2 云开发概要设计 .....	17
4.3 数据库设计 .....	17
4.4 用户界面设计 .....	20
<b>第五章 详细设计 .....</b>	<b>22</b>
5.1 前端详细设计 .....	22
5.1.1 小程序开发和组件库简介 .....	22
5.1.2 求职者相关功能 .....	22
5.1.3 招聘者相关功能 .....	24
5.2 云开发详细设计 .....	25
5.2.1 云开发简介 .....	25
5.2.3 云开发接口设计 .....	26
<b>第六章 程序模块 .....</b>	<b>29</b>
6.1 前端页面模块 .....	29
6.1.1 JS 文件 .....	29
6.1.2 JSON 文件 .....	35
6.1.3 WXML 文件 .....	36
6.1.4 WXSS 文件 .....	38
6.2 云开发模块 .....	43
<b>第七章 测试用例 .....</b>	<b>46</b>
7.1 提交/修改简历信息测试 .....	46
7.2 增加/修改岗位信息测试 .....	46
7.3 增加/修改公司信息测试 .....	47
7.4 聊天功能测试 .....	48
7.5 查看/筛选岗位信息测试 .....	49
7.6 投递简历状态测试 .....	50
7.7 地图定位测试 .....	50
7.9 浏览历史/收藏夹测试 .....	51
<b>第八章 用户手册 .....</b>	<b>53</b>

8.1 软件概述 .....	53
8.1.1 软件背景 .....	53
8.1.2 软件功能 .....	53
8.2 运行环境 .....	54
8.3 使用说明 .....	54
8.3.1 求职者模块 .....	54
8.3.2 招聘者模块 .....	57
8.3.3 帮助模块 .....	59
<b>第九章 个人体会 .....</b>	<b>60</b>
组长：胡才郁 .....	60
组员：黄逸弘 .....	60
组员：徐鼎力 .....	61
组员：张俊雄 .....	62
<b>参考文献 .....</b>	<b>63</b>

# 第一章 目的、要求和内容

## 1.1 项目目的

党的二十大进一步明确全面推进乡村振兴，是在党的十九大以来党中央关于乡村振兴一系列重大决策部署的基础上，关于农业农村现代化如何实现的宣示，党的二十大把乡村振兴战略作为“构建新发展格局，推动高质量发展”的一个重要方面进行了部署。

在乡村振兴任务中，主要需要解决主体“人”的就业问题。如今针对乡村就业的平台相对较少。本小组使用微信小程序为载体，致力于建立乡村振兴人才集聚平台，引导涉农高校学生、“城漂”、退伍军人、退休市民等各类群体回乡就业、创业，为农业农村发展注入新鲜的活力。同时我们瞄准了正蓬勃发展的电商领域，通过我们的平台，农民可以委托专业人员，对村落资源进行宣传、推销，为实现农村现代化提供途径。

“募农帮”小程序定位为招募人才的平台，应乡村振兴的号召而生，致力于帮助新农村建设，为农民朋友们在农忙时招募短期工或企业招募人才提供渠道。

## 1.2 项目要求

### 1.2.1 系统设计要求

本系统从功能上划分主要可以分为求职者和招聘者两种功能。当用户第一次登陆后，默认开通求职者功能。如果用户需要使用招聘者功能，则需要在系统中上传其企业的相关信息，便可开通招聘者功能。开通了招聘者功能后，用户可以在使用系统进行求职的同时，在平台上发布新的职位信息。

● 对于求职功能而言，其主要有以下的要求：

- (1) 登录系统；
- (2) 在系统中快速制作简历；
- (3) 根据地址、工作分类、薪资、工作时间等多种条件查找职位；
- (4) 直接与招聘者发起会话；
- (5) 向心仪职位投递简历；
- (6) 跟踪已投递职位的审查进度；

- (7) 查看最近浏览过的职位;
- (8) 收藏感兴趣的职位;
- (9) 获取劳动法律知识科普;
- 对于招聘功能而言,其主要有以下的要求:
  - (1) 登录系统;
  - (2) 填写公司基本信息,获取招聘资质;
  - (3) 发布新的职位信息;
  - (4) 修改已发布的职位的相关信息;
  - (5) 下架已经失效的职位信息;
  - (6) 查看求职者的简历信息;
  - (7) 直接与求职者发起会话;
  - (8) 对求职者进行快速筛选;
  - (9) 修改求职者的录用情况;
  - (10) 获取劳动法律知识科普;

### 1.2.2 系统开发要求

系统开发要遵循易操作性、健壮性、实用性、高效性和安全性的原则,实现系统设计对应的要求。本系统的开发中,前端主要是用了微信小程序中引入的 WXML、WXSS 和 JavaScript 语言来完成相应功能的实现。为了开发的便利以及界面的美观,还使用了 Vant Weapp 组件库来实现页面中的部分控件。

在后端方面,本程序使用了微信云开发的方式来实现。其提供了云数据库、云函数、云存储等一系列功能。具体来说,云数据库提供了一个 JSON 数据库,数据库中的每条记录都是一个 JSON 格式的对象,多条数据库的记录则组成了一个集合(即关系型数据库中的表),多个集合组成了整个数据库;云函数则是允许用户在微信开发者工具内编写完成后,直接上传到腾讯云。对应的代码会在云端运行,不需要额外部署新的服务器即可运行后端代码,简化了系统的部署。

## 1.3 项目内容

本项目主要使用微信小程序为载体,为求职者和招聘者提供了平台来发布求职信息和寻找对应的岗位。当用户第一次登陆系统,系统默认激活求职者功能。

当用户完成了招聘者的相关信息的填写之后，系统则会激活招聘者功能。用户可以同时具有两种状态，在求职的同时发布新的职位信息。

具体来说，求职者可以使用小程序制作自己的简历。小程序中提供了便利的模版，求职者只需要根据界面要求填写对应的个人信息就可以轻松的制作自己的简历。在制作了对应的简历之后，求职者可以根据分类或关键词来搜索其感兴趣的职位信息，并查看到对应职位的薪资、职位要求、工作地点等基本信息。对于感兴趣的职位，求职者可以直接与招聘者进行私聊、投递简历，并追踪简历筛选的进度。求职者还可以在个人页面查看其此前收藏的、此前浏览的相关职位。

在招聘者方面，在填写了公司的基本信息后，招聘者可以发布新的职位信息。当有新的求职者投递简历后，招聘者可以查看求职者的简历信息。招聘者可以直接在小程序中与求职者发送消息，商定面试时间等事宜。同时，小程序中也可以直接对求职者简历的状态以及录用状态进行修改。这一修改也会实时的反馈到求职者的后台中。

## 第二章 可行性报告

### 2.1 选题背景

#### 2.1.1 社会现状分析

此前，由于城市化进程的推进，使得中国农村人口流失严重，许多年轻人选择到城市工作，导致农村人口老龄化和劳动力短缺。与此同时，由于农村的总体受教育程度较低，人口密度相对较低等相关原因，此前的农村居民寻找工作的途径有限，往往只能通过招工广告或熟人介绍来寻找就业机会。这不仅会造成招聘信息传递的不及时、不准确，还会导致招聘双方的信息不透明。

近年来，随着网络基础设施的建设、网络资费的降低以及智能手机的普及，智能手机逐渐的走入到了千家万户，农村居民也能够接触到互联网。目前互联网上的招聘平台众多，这一定程度上增加了农村居民就业的机会，但是这些招聘平台大多相对复杂，往往不能够在手机端完成全部的招聘、应聘工作，对于农村居民相对不友好。

与此同时，如果农村居民想要在互联网上的招聘平台上发布招聘信息，往往需要提供一系列的证明材料证明其招聘资质，才能够有资格发布招聘信息。这样的方式一方面增加了农村居民发布招聘信息的工作量，另一方面其操作相对复杂，对于农村居民相对不友好。

因此，《募农帮》选择微信小程序作为平台的入口，其可以直接运行在移动设备上，具有更强的受众群体；使用了微信作为程序的载体，减少了用户的学习和适应成本。同时，《募农帮》针对农村招聘的需求而制作，简化了招聘和应聘的相关流程，使得农村居民可以轻松的平台应聘职位和发布招聘信息。

#### 2.1.2 市场分析

为了更清晰地分析搭建一个乡村振兴人才集聚平台的可行性，我们对贵州省遵义市正安县小马村进行了调研，该村于去年脱贫摘帽，但是村落仍是较为贫困。当地村官表示村庄主要产业为小麦、黄豆、玉米种植，人口男女比例失调，农忙时期压力较大，且缺乏高水平人才和招商渠道，同时村中每家每户都有2部以上智能手机，村民具备基本的文化素质，村里电力水资源充足，道路通畅，所以有



使用微信小程序进行人才招聘的能力。

目前绝大多数招工平台都是为了方便农民工在城市找工作,尚未发现有平台替农村招聘人才,就算有类似平台,也没有知名度,用户群体非常小。中国有六十九万个行政村,7.21 亿农村人口,这是一块非常大的空白市场。

对于人口方面,城乡发展差距导致乡村人才纷纷投入城市建设中,青壮年劳力、高素质人群流失严重,留守基层的主要有两类人,一类是基层干部,另一类是老人、妇女和儿童。《2017 年农民工监测调查报告》显示,农民工总量占到全国人口的 20%,占农村人口的 50%,留守农村的各类实用人才仅有 1690 多万人,占农村劳动力的 3.3%。务农人员普遍文化程度低,老龄化问题突出。第三次全国农业普查主要数据公报显示,农业从业人员小学学历与初中学历者占总人数的 85.4%,大专及以上学历的仅占 1.2%,高学历人才不足。此外,农业推广人才,农业生产型、经营型和技能服务型人才严重匮乏。

据此,我们判断我们的乡村本地招聘信息工具是有很高商业价值与公益价值的。

## 2.2 可行性分析

### 2.2.1 技术可行性

在前端方面,本系统使用微信开发者工具进行开发。前端开发中主要是用了 WXML、WXSS、JavaScript 作为开发的技术栈。其中前两者是微信小程序开发中的两种基本语言,其类似于 HTML 和 CSS,主要用于开发小程序前端的页面及其对应的样式;后者则是用于开发小程序的整体处理逻辑,连接数据库等操作。除此以外,前端还使用了 Vant Weapp 组件库来小程序的界面。其是一个轻量的、可靠的移动端组件库,专为微信小程序优化。

在后端方面,本系统使用了微信云开发来完成对应的数据库存储,函数调用等相关的操作。其无需自行搭建服务器或购买额外的服务器资源,可以直接使用微信提供的云数据库、云函数等一系列功能。用户的私聊功能则使用了 GoEasy 提供的 WebSocket 服务。

### 2.2.2 经济可行性

本系统的开发成本较低,主要的费用产生在微信云开发的租用费用以及用户

对话的 WebSocket 服务所产生的费用。由于总体的访问量较低、产生的数据相对较小,使用基本的微信云开发配额即可满足系统的运行,因此整体开发成本较低,具有经济可行性。

### 2.2.3 操作可行性

本系统使用微信小程序作为平台,使用手机端作为主要的运行平台,对于乡村地区的使用更加友好。系统整体界面简洁友好,操作方便,有多种的使用说明提示用户如何操作系统。用户只需要根据系统的说明输入相对应的信息,就可以轻松的在系统中完成简历投递、私聊招聘者或查看简历、筛选简历等操作,具有用户友好的特性。

## 第三章 需求分析

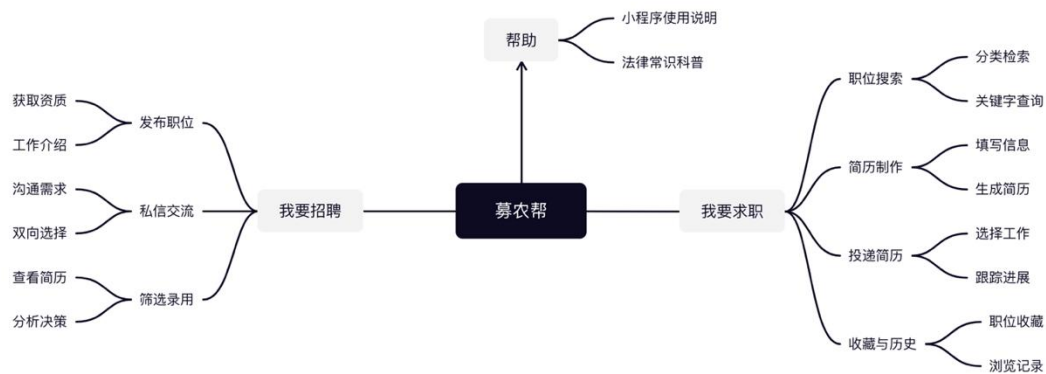
### 3.1 工程概述

#### 3.1.1 待开发软件的一般概述

本系统的主要定位为招募人才的平台，致力于帮助新农村建设，为农民朋友们在农忙时招募短期工或企业招募人才提供渠道。平台同时具备求职者和招聘者两种功能。用户默认激活招聘者功能，当用户完成了相关信息的填写，则会激活招聘者功能。

#### 3.1.2 待开发软件的功能

- (1) 求职者和招聘者（以下合称为用户）可以直接使用微信登录系统；
- (2) 求职者可以直接在系统中快速制作简历；
- (3) 求职者可以根据多种条件查找职位；
- (4) 用户可以直接在小程序中发起会话；
- (5) 求职者可以向心仪职位投递简历；
- (6) 求职者可以直接跟踪已投递职位的审查进度；
- (7) 求职者可以查看最近浏览过的职位、收藏的职位；
- (8) 招聘者可以填写公司基本信息，获取招聘资质；
- (9) 招聘者可以发布新的职位信息；
- (10) 招聘者可以修改、下架已经发布的职位信息；
- (11) 招聘者可以直接查看求职者投递的简历；
- (12) 招聘者可以对求职者进行筛选，并修改其简历审查进度；
- (13) 用户可以查看劳动法律知识科普；



### 3.1.3 运行环境

微信小程序。

### 3.1.4 条件与限制

本系统使用微信小程序作为运行平台，因此其运行的系统要求较低，只要手机能够正常运行微信即可使用本系统。本系统的整体操作难度较低，且具有明确的提示信息指引用户操作，用户的上手难度较低。

## 3.2 系统综合要求

### 3.2.1 功能需求

(1) 用户可以直接使用微信登录系统；

用户打开本系统后，点击下方导航栏处的“我的”页面，即可看到登陆/注册入口。用户点击对应的按钮后，系统会调用微信的授权登陆功能，引导用户给予小程序用户权限。当用户完成授权后，即可登陆本系统。

(2) 求职者可以直接在系统中快速制作简历；

用户在登陆系统的情况下，打开个人中心，点击“简历制作”按钮即可进入简历制作页面。如果用户此前已经创建了自己的简历，则先前的信息会被读入。用户可以在此页面修改自身的个人信息。如果用户部分信息没有填写完成就点击提交，系统会提示用户补充信息完成。当用户提交简历后，系统会自动打开简历供用户核对查看。用户可随时对自身的简历进行修改。

(3) 求职者可以根据多种条件查找职位；

在小程序首页会展示当前系统中的所有职位信息。用户可以直接使用搜索栏进行关键词的搜索，或通过职位的分类进行搜索。搜索到对应的职位后，用户可以根据薪资的多少、需求人数、招聘截止日等信息进行排序，从而选择最适合自己的职位信息。除此以外，系统还提供了距离最近的职位信息，以方便求职者查找职位。

(4) 用户可以直接在小程序中发起会话；

在职位详情页面点击“发消息”按钮后，会打开消息页面，此时求职者可直接与招聘者展开对话。对话同时具有回执功能，可在下方直观的看到对方是否已经阅读了消息。

(5) 求职者可以向心仪职位投递简历；

在职位详情页面，用户点击下方的投递简历后，即可完成简历的投递。简历投递完成后，下方的按钮会变为已投递来提示用户。

(6) 求职者可以直接跟踪已投递职位的审查进度；

用户投递了简历后，可以在个人中心中点击投递反馈按钮来查看用户投递的所有职位信息。在每一个条目的右上方可以看到当前简历的进度。如果用户想要查看更加详细的信息，可以点击条目，打开详细的信息。详细信息使用了时间线的方式，详细地列出了简历在投递后的状态变化，且具有时间的显示，给予求职者对当前求职状态的详细介绍。

(7) 求职者可以查看最近浏览过的职位、收藏的职位；

用户在个人中心页面可以看到“我的收藏”和“最近浏览”两个按钮。点击可以分别查看此前收藏的职位和之前浏览的相关职位信息。

(8) 招聘者可以填写公司基本信息，获取招聘资质；

用户在个人中心界面，点击招聘资质。如果用户在此前没有填写公司的基本信息，则会弹出对话框，提示用户没有填写信息。点击确认后，则会弹出信息编辑窗口。招聘者可以在此输入公司的相关信息，并在地图中选择准确的定位地址。填写完成后，点击保存即可获取招聘资质。

(9) 招聘者可以发布新的职位信息；

在获取了招聘资质后，用户在个人中心中点击发布职位即可发布新的职位信息。用户需要填写职位的数量、薪资、分类等相关信息，并选择工作地址，即可

发布新的职位信息。

(10)招聘者可以修改、下架已经发布的职位信息；

用户在获取了招聘资质后，点击“招聘资质”按钮就可以打开公司的主页。在此可以看到当前公司发布的所有的职位信息。用户可以直接点击对应条目旁的修改或下架按钮来修改对应职位的状态。

(11)招聘者可以直接查看求职者投递的简历，并进行筛选和审查进度；

用户获取了招聘资质后，点击投递详情可以查看所有发布的职位的简历投递情况。在此界面可以直接查看投递的简历人数。点击对应的职位，可以看到所有的求职者。在此界面可以查看简历、发起私聊、修改求职者的简历状态等操作。

(12)用户可以查看劳动法律知识科普；

用户点击帮助界面，即可查看对应的法律知识介绍。

### 3.2.2 外部接口需求

(1) 用户界面

本系统使用微信小程序作为开发的平台，同时使用了 Vant WeApp 组件库来构建用户界面，并且对手机等移动设备适配良好，其对用户友好，界面具有一致性。系统也对多种的错误进行了处理，使用用户友好的方式提示用户错误的类型。整体界面中有多处提示，使用语言和图形的方式提示用户正确填写相关信息。

(2) 硬件、软件接口

本系统使用微信小程序作为开发的平台，只需要能够访问互联网、运行微信的智能手机即可使用本系统。本系统使用的所有接口均是微信开放给开发者的相关接口。

## 3.3 数据描述

### 3.3.1 静态数据

(1) 用户 ID：用于标识唯一的用户信息

(2) 用户昵称：用户的微信昵称

(3) 公司 ID：用于标识唯一的公司信息

(4) 公司地址

(5) 公司分类

(6) 劳动法原始文本

3.3.2 动态数据

(1) 求职者制作简历；

用户输入简历制作页面中的相关信息，即可修改或制作用户的简历。如果用户没有填写完整，则会提示用户特定的项目未填写。

(2) 求职者投递简历；

求职者选择对应的职位投递后，会将其简历投递给招聘者，求职者处则会显示简历已投递。

(3) 新建对话；

用户可以直接与其他新建对话，对方发送的消息会显示在屏幕上，在对话框中输入的消息点击发送后也会发送给对方。

(4) 招聘者新建职位信息；

招聘者输入职位所需要的基本信息，即可建立新的职位。职位简历后，系统会自动打开对应的职位以供招聘者核对。

(5) 招聘者新建公司信息；

招聘者输入公司所需要的基本信息，即可建立新的公司，获得招聘资质。

3.3.3 数据库

本系统使用的数据库为微信云开发中提供的云数据库。

3.3.4 数据流图

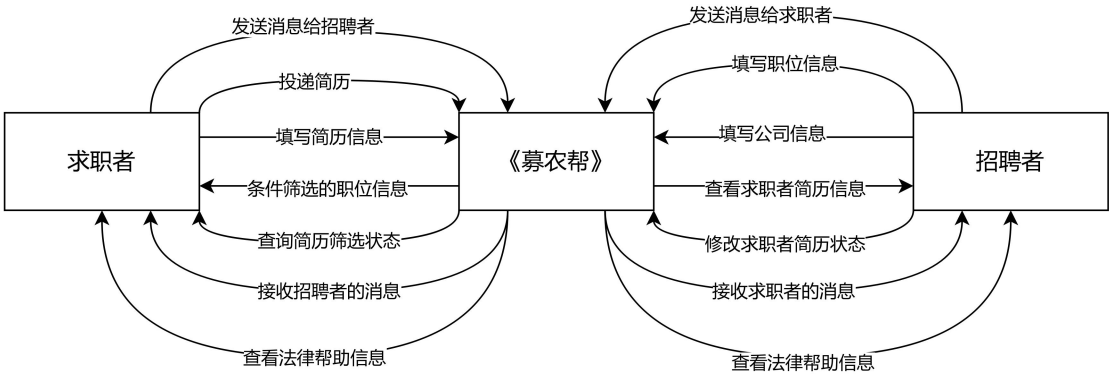


图 3.2 系统数据流顶层图

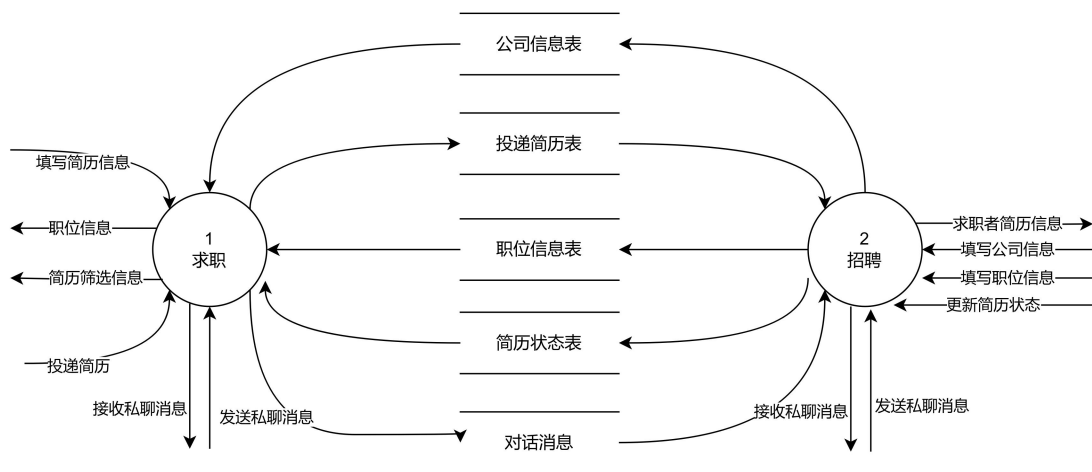


图 3.2 系统数据流 0 层图



## 第四章 概要设计

### 4.1 前端概要设计

本微信小程序使用微信开发者工具开发，从功能上区分主要可以分为应聘者 and 招聘者两个功能。

应聘者部分为用户提供了制作简历、投递简历、查看岗位信息、搜索岗位、根据条件筛选显示岗位、与招聘者对话、追踪投递情况、收藏夹和浏览历史等功能。此外用户还能够查看小程序的使用指南、功能介绍，在帮助部分可以查看劳动法的详细信息、劳动法、劳动者基本权利和应聘者关心的问题。

招聘者相比应聘者，还增加了创建招聘资质功能，能够添加自己的公司信息进行招募。通过支持公司信息修改、发布职位信息、管理应聘者简历状态等功能。

### 4.2 云开发概要设计

云开发部分主要运用了其中的云数据库和云存储功能，用于各种静态数据和动态数据的增删改查，同时使用云存储功能存储公司的图片，同时将对应的路径存储在云数据库中。在云数据库中，主要存储了用户的个人信息、公司的信息、岗位的信息、简历的信息、简历的状态、收藏夹、浏览历史和科普知识。同时使用云开发提供的函数与数据库进行通信，使用搜索、筛选、排序等功能。

### 4.3 数据库设计

表 4.1 用户信息表

数据列名	数据类型	备注
_id	string	表索引
_openid	string	用户唯一标识符
avatar	string	
nickName	string	

表 4.2 简历信息表

数据列名	数据类型	备注
_id	string	表索引
_openid	string	用户唯一标识符
age	number	
beginTime	string	
eduBg	string	学历背景
endTime	string	
gender	string	
phone	string	
pickCity	string	用户所在地
realName	string	
selfEval	string	自我评价
skill	string	
userId	string	用户表索引
workExp	string	工作经历

表 4.3 简历状态信息表

数据列名	数据类型	备注
_id	string	表索引
_openid	string	用户唯一标识符
postId	string	
status	number	简历状态
updateTime	date	

表 4.4 岗位收藏信息表

数据列名	数据类型	备注
_id	string	表索引
_openid	string	用户唯一标识符
	string	岗位表索引
status	string	
updateTime	string	收藏时间

表 4.5 岗位浏览历史表

数据列名	数据类型	备注
_id	string	表索引
_openid	string	用户唯一标识符
jobId	string	
viewTime	string	浏览时间

表 4.6 公司信息表

数据列名	数据类型	备注
_id	string	表索引
_openid	string	用户唯一标识符
address	string	
averageSalary	string	
category	string	公司类别
campanyName	string	
introduce	string	
userId	string	用户表索引

表 4.7 岗位信息表

数据列名	数据类型	备注
_id	string	表索引
_openid	string	用户唯一标识符
beginTime	date	
content	string	
category	string	公司类别
continueTime	string	
endTime	date	
jobAddress	string	用户表索引
jobName	string	
location	string	
locationGeo	geopoint	
needNum	number	
postNum	number	投递简历人数
remainNum	number	剩余岗位数量

salary	number	
status	number	收藏标签
tags	array	岗位标签
updateTime	date	
userId	string	用户表索引

表 4.8 科普信息表

数据列名	数据类型	备注
_id	string	表索引
chapter	number	
content	string	
entry	string	

## 4.4 用户界面设计

该小程序总共分为四个主要页面。

首页是微信小程序用户的入口和主要浏览页面。在首页的顶部，我们可以设计一个宣传轮播图，用于展示不同的宣传信息，以便用户了解更多相关信息。中间部分是岗位信息列表，展示不同岗位的信息，包括公司名称、职位名称、薪资待遇等。这些信息以列表形式展示，用户可以通过滑动屏幕来浏览不同的岗位信息。同时，为了方便用户找到符合自己需求的岗位，我们可以在页面顶部设计一个搜索框和筛选条件。用户可以输入关键词进行搜索，或者根据需要、薪资等条件进行筛选。

科普信息页面是用来向用户提供相关知识和法律信息的。页面顶部展示一个视频，表示为小程序操作指南。中间部分是科普信息列表，展示不同的科普内容。用户可以点击其中一篇内容，进入详情页面进行阅读或观看。这个页面可以用于教育用户、提供咨询和帮助用户更好地了解劳动者权利。

聊天功能页面是为用户提供与招聘者进行实时交流的功能。页面顶部显示一个标题，标识为聊天功能页面。中间部分是聊天界面，显示用户与其他人或系统的消息记录。用户可以发送文本消息，并查看对方的查看状态，与对方进行实时交流。这个功能可以用于用户之间的沟通和求职者与企业的交流。

个人中心页面是用户管理个人信息和查看个人相关功能的地方。顶部显示用

户头像和用户名，展示用户的身份。中间部分是个人信息展示，用户可以查看和编辑自己的基本信息，包括收藏夹、浏览历史以及个人简历。此外，个人中心页面还提供了多个功能模块，如简历状态跟踪。在简历部分用户可以填写个人信息以创建个人简历，并查看已投递的简历状态和反馈，通过个人中心页面，用户可以方便地管理自己的求职过程。而对于应聘者来说，也提供了公司的信息设置，发布不同岗位以及对应聘者的简历进行查看和筛选。

## 第五章 详细设计

### 5.1 前端详细设计

#### 5.1.1 小程序开发和组件库简介

微信小程序使用的是前端开发技术，主要使用的是 JavaScript 语言。开发者可以使用原生的 JavaScript 进行开发，也可以选择使用框架如 Vue.js 或 React 进行开发。微信提供了官方的开发者工具，开发者可以在其中进行小程序的开发、调试和发布。开发者工具支持代码编辑器、调试器、模拟器等功能，方便开发者进行开发和调试工作。为了方便开发者快速构建小程序界面，微信小程序开发中会使用各种组件库。其中，Vant Weapp 是一款基于微信小程序的组件库，提供了丰富的 UI 组件，能够快速满足开发需要。此外，微信小程序开放了一系列的接口 API，开发者可以通过这些 API 来实现小程序的各种功能，如获取用户信息、支付、调用云存储等。通过这些接口 API，可以实现与微信平台的交互和接入多种能力。

Vant Weapp 是基于微信小程序的组件库，提供丰富、易用的 UI 组件来帮助开发者快速构建小程序界面。其高度兼容 Vue.js 的语法和风格，使得开发者能够无缝迁移 Vue.js 项目到小程序开发中。其中，提供了一系列常用的基础和业务组件，例如按钮、表单、轮播、下拉刷新等，具有统一的样式和交互，能够快速满足大部分小程序开发场景的需求，也能进行一定自定义的设计和开发，其中的属性和事件设置来配置和交互组件，实现所需的功能。API 的设计借鉴了 Vue.js 的思想，使得 Vue.js 开发者能够更快地上手使用 Vant Weapp。此外，其内部做了大量性能优化工作，以确保组件在小程序上的渲染性能和响应速度。这些优化包括减少不必要的渲染和重绘、合理利用小程序的生命周期等。由于 Vant Weapp 是基于微信小程序开发的，所以它可以无缝集成和运行在微信小程序的所有端，包括小程序端、小游戏端和微信轻应用端等。

#### 5.1.2 求职者相关功能

##### (1) 岗位列表展示和查询

小程序首页会展示当前系统中的所有职位信息，包括顶部的搜索栏、轮播图、

筛选规则按钮以及工作简要信息列表。用户可以直接使用搜索栏，点击进入专用的搜索页面，进行关键词的搜索，或通过职位的分类进行搜索。此外，用户可以根据薪资的多少、需求人数、招聘截止日等信息进行排序，从而选择最适合自己的职位信息。除此以外，系统还提供了距离最近的职位信息，以方便求职者查找职位。点击岗位的条目后，根据岗位 id 转入相应的详细页面。

## (2) 岗位详细信息

展示当前岗位的详细信息，包括对应的公司的信息以及公司对应找招聘者的信息，同时提供一键投递简历的按钮。同时下方通过 `work_collect` 中收藏夹展示红心的颜色状态。下拉后，可以看到联系人信息，点击招聘者头像，点击后可以查看公司的详细信息，以及公司包含的所有岗位列表，与首页的岗位列表类似。此外，也可以点击招聘者右侧的发消息按钮，可以进行绘画会话功能界面。

## (3) 简历投递功能

在工作的详情界面，提供一键投递简历的按钮。当用于点击后会弹出确认框进行最终确认，确认提交后，下方的按钮产生变化，显示“已投递”。

## (4) 跟踪投递岗位信息

用户投递了简历后，可以在个人中心中点击投递反馈按钮来查看用户投递的所有的职位信息。在每一个条目的右上方可以看到当前简历的进度。如果用户想要查看更加详细的信息，可以点击条目，打开详细的信息。详细信息使用了时间线的方式，详细地列出了简历在投递后的状态变化，且具有时间的显示，给予求职者对当前求职状态的详细介绍。在数据库中存储在 `work_cv_status` 表中，不同状态同时存储了创建时间信息，使得用户能够得到状态的完整时间信息。

## (5) 快速制作简历

用户在登陆系统的情况下，在个人中心，点击“简历制作”按钮即可进入简历制作页面。如果用户此前已经创建了自己的简历，在 `work_cv` 中以及存储该用户的简历信息，先前的信息会被读入。用户可以在此页面修改自身的个人信息。如果用户部分信息没有填写完成就点击提交，会调用相应的函数进行完整性和正确性的检验，系统会提示用户补充信息完成。当用户提交简历后，系统会自动打开简历供用户核对查看。用户可随时对自身的简历进行修改。在个人中心的“我的简历”按钮中，可以重新点开简历进行预览。

## (6) 个人中心

在个人中心用户，可看到登陆/注册入口。用户点击对应的按钮后，系统会调用微信的授权登陆功能，引导用户给予小程序用户权限。当用户完成授权后，即可登陆本系统。登录后展示个人信息，进行账户的退出和切换，提供各种功能的列表，包括简历制作、查看简历、收藏夹和最近浏览，以及获取帮助定位到相应的界面。

## (7) 会话功能

在职位详情页面点击“发消息”按钮后，会打开消息页面，此时求职者可直接与招聘者展开对话，使用 goEasy 进行实现，在 util.js 中定义实现 goEasy 的初始化和连接，传递用户的 openid、昵称以及头像信息，同时定义了发送消息和方法方便调用。此外，对话同时具有回执功能，可在下方直观的看到对方是否已经阅读了消息。在底部的导航栏可以查看所有的聊天信息，进入后查看详细的聊天信息。

## (8) 法律信息科普功能

在帮助界面，能够看到该小程序宣传视频，为用户提供使用教程和介绍，下方是不同的法律信息科普，其中劳动法部分的数据存储于 work\_labor\_law 表中，进行章节筛选后展示在不同路径的页面上，用户可以点击不同章节进行查看。

## (9) 岗位收藏夹和历史浏览列表

用户在个人中心页面可以看到“我的收藏”和“最近浏览”两个按钮。点击可以分别查看此前收藏的职位和之前浏览的相关职位信息，其中根据用户的 userId 以及时间点进行筛选以及排序，分别存储于 work\_collect 和 work\_record\_job 表中。

### 5.1.3 招聘者相关功能

#### (1) 获取招聘资质

用户在个人中心界面，点击招聘资质。如果用户在此前没有填写公司的基本信息，则会弹出对话框，提示用户没有填写信息。点击确认后，则会弹出信息编辑窗口。招聘者可以在此输入公司的相关信息，并在地图中选择准确的定位地址，地图提供经纬度搜索和地址名称定位两种搜索方式，也可以点击地图进行位置的定位，选择后下方会显示推荐的地址用于确认地址的正确性。填写完成后，点击保存即可获取招聘资质。



## (2) 公司详情信息和岗位信息修改

用户在获取了招聘资质后，点击“招聘资质”按钮就可以打开公司的主页。在此可以看到当前公司发布的所有的职位信息。用户可以直接点击对应条目旁的修改或下架按钮来修改对应职位的状态。当岗位被下架之后，其他用户不会看到该岗位的信息，重新上架后，用户可以看到。此外，也可以通过筛选的条件（按需求的大小、按截止日期、按日薪）进行升序或者降序的排列，方便管理。

## (3) 发布岗位

在获取了招聘资质后，用户在个人中心中点击发布职位即可发布新的职位信息。用户需要填写职位的数量、薪资、分类等相关信息，并选择工作地址，即可发布新的职位信息。在填写后，对应函数会对填写的表单进行验证，如果通过验证，将该岗位的信息增加一条新的条目。

## (4) 查看投递简历及简历状态修改

招聘者在获取了招聘资质后，上架岗位之后，当应聘者提交简历之后，可以点击投递详情可以查看所有发布的职位的简历投递情况。在此界面可以直接查看投递的简历人数。点击对应的职位，可以看到所有的求职者，并对不同求职者的录用状态进行改变，同时可以进入和不同的求职者进行对话聊天。

其他功能与求职者功能相同，不进行赘述。

# 5.2 云开发详细设计

## 5.2.1 云开发简介

微信小程序云开发是一项由微信官方提供的后端云服务，它使开发者能够以高效和低成本的方式构建和部署小程序的后台服务。使用微信小程序云开发，开发者无需投资和维护独立的服务器，而是利用微信提供的云端资源进行开发，从而节省了时间和资源。

云开发的核心概念是云函数。开发者可以编写自己的业务逻辑代码，并通过云函数发布和调用它们。云函数能够直接调用小程序端的请求、数据库和其他微信开放能力，实现了前后端的无缝交互。这种基于云函数的开发模式简化了开发流程，提高了开发效率。

另一个重要组成部分是实时数据库。微信小程序云开发提供了一个实时数据库，它是一种 NoSQL 数据库，具备实时同步功能以及强大的查询和索引功能。这种数据库可以方便地存储和查询数据，大大简化了数据管理的过程。

此外，其还具备强大的安全性。它基于微信自身的安全机制，开发者可以轻松进行用户身份验证和数据安全管理。此外，微信小程序云开发集成了微信支付、微信登录和微信推送等功能，帮助开发者快速实现各种丰富的功能和服务。

### 5.2.3 云开发接口设计

在本项目中，主要使用了云开发中云存储的功能，设置好云开发的环境 id 后，使用对应函数进行增删改查。

#### (1) 用户制作个人简历

接口获取数据：真实姓名、年龄、性别、教育经历、工作城市、手机、空闲时间、工作经历、擅长经历、自我评价、用户 id、用户 openid。

接口功能：用户填写表单信息，检验数据信息符合要求，同时检验当前用户是否已经登录，若满足以上条件，调用添加数据接口，提供表单中的简历相关信息，保存在云数据库的 work\_cv 表中。

#### (2) 展示用户简历信息

接口获取数据：用户 openid

接口功能：在用户修改简历或查看简历时，根据用户 openid 查询该用户的简历信息，并展示在前端。

#### (3) 用户修改个人简历

接口获取数据：真实姓名、年龄、性别、教育经历、工作城市、手机、空闲时间、工作经历、擅长经历、自我评价、用户 id、用户 openid。

接口功能：用户填写表单信息，检验数据信息符合要求，同时检验当前用户是否已经登录，若满足以上条件，查询当前用户 openid 对应的简历是否存在，若存在，调用更新数据接口（update），提供表单中的简历相关信息，保存在云数据库的 work\_cv 表中。

#### (4) 查看工作历史记录

接口获取数据：用户 userId。

接口功能：通过用户的 userId 获取 work\_record\_job 表中的数据，保存了工作的

基础信息，并制定该工作记录的浏览用户的信息，产生对应关系。获取数据之后，排序之后，展示在前端界面。

#### (5) 查看工作信息列表

接口获取数据：工作状态（上架/下架）

接口功能：在 `work_job` 中获取所有上架工作的数据。

#### (6) 查看工作信息详情

接口获取数据：工作 id、公司 id、用户 id、公司对应用户 id

接口功能：在工作详情界面，通过接口获取 `work_user`、`work_company`、`work_job` 中的数据，展示当前工作以及对应的用户和公司的信息，并查询当前用户的收藏信息，展示红心的效果。

#### (7) 招聘资质查看公司详情

接口获取数据：用户 id；排序依据（需求大小、截止日期、日薪）。

接口功能：在 `work_company` 表中获取当前用户的公司信息，包括公司本身的基本信息，同时在 `work_job` 表中获取当前公司的所有工作信息，以列表进行展示，同时可以通过按钮和下拉框改变排序的依据，以重新调用接口，传入排序依据，获取排序过后的数据。

#### (8) 创建/更新公司信息

接口获取数据：公司 id、公司图片信息、公司名称、平均薪资、公司地址、公司介绍、公司类别

接口功能：根据当前用户相关的公司信息，若公司 id 已经存在，获取已经创建的公司信息，通过表单展示，若公司不存在，则创建新的公司。在表单填写或修改信息之后，更新 `work_company` 中的数据。

#### (9) 获取/更新收藏夹工作信息

接口获取数据：用户 id、收藏状态。

接口功能：在工作详情页面中，若用户点击了爱心按钮，为当前用户创建收藏信息，保存在 `work_collect_job` 表中，并将收藏状态记录为 1。若用户取消收藏了，记录中的收藏状态改变为 0。当用户进入收藏夹后，获取当前用户所有收藏状态为 1 的记录，并展示在前端。

#### (10) 投递简历

接口获取数据：简历 id、用户 id、其他简历信息。

接口功能：获取用户的简历的信息，并将其与该岗位的信息进行连接，并保存在 work\_post 表中。

#### (11)更改简历状态

接口获取数据：简历 id、用户 id

接口功能：在当前用户的公司中的每个岗位中，可以查看当前投递简历的用户，在进行筛选的按钮上，改变 work\_cv\_status\_record 表中该记录的 status 信息，表示录用的状态，分别为已投递、待初筛、初筛通过、待面试、录用成功和录用失败。当每次改变筛选状态时，会产生一条新记录，并记录产生的时间，反馈用户。

#### (12)获取科普信息内容

接口功能：在 work\_labor\_law 表中获取劳动法相关的内容并展示。

## 第六章 程序模块

### 6.1 前端页面模块

本系统基于微信小程序进行开发，因此本系统的模块结构符合小程序的项目结构，即系统的前端页面位于小程序目录下的/pages 目录中。在该目录中，每一个文件夹对应的是一个页面，其中包含了 4 种文件，分别为 JS 文件、JSON 文件、WXML 文件和 WXSS 文件。下面以首页为例，详细的介绍一下这四种文件对于小程序页面中系统的作用。

#### 6.1.1 JS 文件

小程序页面中的 JS 文件主要承担了逻辑处理、数据处理和交互操作等功能，其中包括了页面绑定数据（Data 属性）、页面绑定方法以及后端数据请求函数等部分。这些部分主要是为了实现小程序前端的交互逻辑，前端与云数据库、云函数的数据交互等。

在这里以首页为例，在文件的最上方首先设置了小程序中使用的云数据库及其登陆的鉴权信息；随后引入了 dayjs 依赖，其实一个基于 JavaScript 的包，在这里主要用于对时间的快速处理。

在引入了相关依赖和完成了设置后，在 Data 区域主要存储了页面的有关的初始数据，例如初始化了职位的列表数组、将初始的页面设置为 0 等操作。这些数据在后续的程序或 WXML 的前端展示中都有使用。

随后，在此定义了一系列的函数。其中，toLoginPage 函数是用于当用户进入了需要登陆的情况下，将用户跳转至登陆页面；getJobList 函数是用于根据用户选择筛选条件，从数据库中检索所有的职位信息，并根据用户的筛选条件进行排序，在这里默认是根据工资进行降序排序；toSearchPage 函数则是当用户点击了上方的搜索栏后，将用户跳转到搜索页面。此外，下方还有 onLoad、onReady、onShow 等函数，其是微信小程序自带的生命周期回调函数，他们会在对应的事件触发后，执行函数中的代码。上方举例的三个生命后期回调函数分别会在加载开始、页面初次渲染完成、页面显示事件触发后，执行函数中的代码。

表 6.1 首页 JS 文件代码

```
// pages/home/home/home.js

const db = wx.cloud.database({
  env: '*****'
})

const dayjs =
require('../../miniprogram_npm/miniprogram_npm/dayjs/index')

import { storeBindingsBehavior } from
'mobx-miniprogram-bindings'

import { store } from '../../store/store'

Page({
  options: {
    styleIsolation: 'shared'
  },
  behaviors: [storeBindingsBehavior],
  storeBindings: {
    store,
    fields: {
      active: 'activeTabBarIndex'
    },
    actions: {
      updateActive: 'updateActiveTabBarIndex'
    },
  },

  /**
   * 页面的初始数据
   */
})
```

```

data: {
  keyword: '募农帮，助力乡村振兴！',
  activeTab: 0,
  jobList: [],
  curPage: 0,
  loading: true,
  showOverlay: true
},

toLoginPage() {
  wx.switchTab({
    url: '/pages/center/center',
  })
  this.setData({showOverlay: false})
},

onClickShowOverlay() {
  this.setData({ showOverlay: true })
},

onClickHideOverlay() {
  this.setData({ showOverlay: false })
},

// 推荐：按日工资排序
async getJobList(orderBy = 'salary', order = 'desc', limit
= 10) {
  let res = await
db.collection('work_job').skip(this.data.curPage *

```

```

limit).limit(limit)

        .where({
            status: 1
        }).orderBy(orderBy, order).get()
let jobList = res.data
for (let i = 0; i < jobList.length; i++) {
    const job = jobList[i]
    jobList[i].endTime =
dayjs(job.endTime).format("YYYY-MM-DD")

    const res = await
db.collection('work_user').doc(job.userId).get()

    jobList[i].user = res.data
}
this.setData({
    jobList: this.data.jobList.concat(jobList),
    loading: false
})
console.log(jobList)
},

toSearchPage() {
    wx.navigateTo({
        url: '/pages/search/search'
    })
},

/**
 * 生命周期函数--监听页面加载
 */

```



```

onLoad: function (options) {
    this.getJobList()
},

/**
 * 生命周期函数--监听页面初次渲染完成
 */
onReady: function () {

},

/**
 * 生命周期函数--监听页面显示
 */
onShow: function () {
    if (typeof this.getTabBar === 'function' &&
this.getTabBar()) {
        this.updateActive(0)
    }
},

/**
 * 生命周期函数--监听页面隐藏
 */
onHide: function () {

},

/**

```

```

    * 生命周期函数--监听页面卸载

    */

    onUnload: function () {

    },

    /**

    * 页面相关事件处理函数--监听用户下拉动作

    */

    async onPullDownRefresh() {

        wx.showNavigationBarLoading()

        this.setData({

            jobList: [],

            curPage: 0,

            loading: true

        })

        await this.getJobList()

        wx.hideNavigationBarLoading()

        // 数据成功后，停止下拉刷新

        wx.stopPullDownRefresh()

    },

    /**

    * 页面上拉触底事件的处理函数

    */

    onReachBottom: function () {

        wx.showLoading({

            title: '加载中..',

        })

    }

```

```

        this.setData({
            curPage: this.data.curPage + 1
        })
        this.getJobList().then(() => {
            // 数据成功后，停止下拉刷新
            wx.stopPullDownRefresh()
            wx.hideLoading()
        })
    },

    /**
     * 用户点击右上角分享
     */
    onShareAppMessage: function () {

    }
})

```

### 6.1.2 JSON 文件

JSON 文件为当前页面配置的文件，其中包含页面所用到的外置组件库、外置组件名称等页面信息，以及导航栏颜色、页面字体样式、页面使用组件等外观信息。其对程序的核心逻辑影响不大，主要是整体的 UI 设置。

**表 6.2 首页 JSON 文件代码**

```

// pages/home/home/home.json
{
    "usingComponents": {},
    "navigationBarBackgroundColor": "#00a896",
    "navigationBarTextStyle": "black",
    "backgroundTextStyle": "dark",

```

```
    "enablePullDownRefresh": true,  
    "navigationBarTitleText": "募农帮"  
  }
```

### 6.1.3 WXML 文件

小程序中的 WXML 文件类似于 HTML 的语法, 不过在其中引入了部分微信小程序中特有的语法。在此我们使用了 Vant Weapp 提供的部分组件, 使得整体的用户界面更加简洁美观。例如, 在首页中, 使用了 Vant 中提供的粘性布局 (Sticky), 使得无论用户如何滑动页面, 搜索栏始终在页面的顶端; 此外, 在这里使用了两个大括号, 引用了此前在 JS 文件中的 Data 部分声明的相关变量; 还使用了 bindtap, 将用户点击的触发事件连接到 JS 文件中定义的函数, 打开搜索页面。事实上, 在页面中, 多次使用了 Vant Weapp 中提供的组件。

表 6.3 首页 WXML 文件代码

```
<!-- pages/home/home/home.wxml -->  
  
<van-sticky>  
  <van-search placeholder="{{ keyword }}" shape="round"  
background='#00a896'  
  bindtap="toSearchPage" use-action-slot>  
    <view slot="action">搜索</view>  
  </van-search>  
</van-sticky>  
  
<home_swiper></home_swiper>  
  
<van-notice-bar left-icon="volume-o" color="#02c39a"  
background="#dcf7f3" speed='140'  
  text="工作不好找, 简历不会写? 来募农帮! 募农帮——专注于乡村招聘的微信小程序!" />  
  
<van-grid column-num="3" border="{{ false }}" clickable='true'>  
  <van-grid-item  
icon="https://shu-silence.oss-cn-shanghai.aliyuncs.com/img/2023/n
```

```

ew%20(3).svg" text="最新发布"

url='/pages/showJobList/showJobList?to=1'/>

    <van-grid-item
icon="https://shu-silence.oss-cn-shanghai.aliyuncs.com/img/2023/f
ire.svg" text="热门职位"

        url='/pages/showJobList/showJobList?to=2'/>

    <van-grid-item
icon="https://shu-silence.oss-cn-shanghai.aliyuncs.com/img/2023/m
ap%20(1).svg" text="距离最近"

        url='/pages/showJobList/showJobList?to=3'/>

    <!-- <van-grid-item
icon="http://cdn.itbear666.top/avatar/2.jpg" text="持续最长"
url='/pages/showJobList/showJobList?to=4'/> -->

</van-grid>

<jobList jobList="{{jobList}}"/>

<van-skeleton title avatar row="4" loading="{{ loading }}"
wx:for="{{2}}" wx:key="index"></van-skeleton>

<van-skeleton title avatar row="2" loading="{{ loading }}">

    <view class="end-line">

        到底了~

    </view>

</van-skeleton>

<!-- <van-overlay show="{{ showOverlay }}" z-index='10000'>

<view class="wrapper">

    <view class="block">

        <view class="cell" hover-class="hover_cell"
bindtap="toLoginPage">

            <van-image

src="http://cdn.itbear666.top/avatar/2.jpg" fit="cover"

```

```

radius='12' />

        <view class="explanation">我要求职</view>
    </view>

    <view class="cell" hover-class="hover_cell"
bindtap="toLoginPage">
        <van-image
src="http://cdn.itbear666.top/avatar/2.jpg" fit="cover" radius='12'
/>

        <view class="explanation">我要招聘</view>
    </view>

    <van-icon name="cross" custom-style='background-color:
#c4c6c9; border-radius: 100rpx; margin: 40rpx;'
bindtap="onClickHideOverlay"/>
</view>
</view>
</van-overlay> -->

```

#### 6.1.4 WXSS 文件

小程序中的 WXSS 文件类似于 CSS 的语法，其主要是定义了 WXSS 中各个元素的样式信息。下面展示的是首页中使用的 WXSS 代码。在这里，可以对 WXML 中的部分元素的部分样式进行修改。例如，在这里对 `page` 的背景颜色进行了设置。

**表 6.4 首页 WXSS 文件代码**

```

/* pages/home/home/home.wxss */
page {
    background-color: #f7f8fa;
}

.field-class {

```

```
        height: 1000rpx;
    }

    .divide-line {
        height: 20rpx;
        background-color: #f4f4f5;
    }

    .head-line {
        display: flex;
        justify-content: space-between;
        margin: 10rpx 28rpx;
    }

    .title {
        font-size: 36rpx;
        font-weight: bold;
    }

    .money {
        font-size: 32rpx;
        color: #02c39a;
    }

    .content {
        margin: 10rpx 24rpx;
        color: #909399;
        font-size: 28rpx;
        display: flex;
```

```
        justify-content: space-between;
    }

    .tags {
        margin: 16rpx 16rpx;
    }

    .user-style {
        display: flex;
    }

    .bottom-line {
        margin: 10rpx 28rpx;
        display: flex;
        justify-content: space-between;
    }

    .city-name {
        font-size: 24rpx;
        margin-top: 16rpx;
        color: #909399;
    }

    .user-name {
        font-size: 28rpx;
        margin-top: 16rpx;
        margin-left: 10rpx;
        color: #303133;
    }
}
```



```
.end-line {  
    height: 220rpx;  
    background-color: #f7f8fa;  
    color: #909399;  
    display: flex;  
    font-size: 28rpx;  
    justify-content: center;  
}
```

```
.wrapper {  
    display: flex;  
    align-items: center;  
    justify-content: center;  
    height: 100%;  
}
```

```
.block {  
    width: 66%;  
    height: 33%;  
    background-color: #fff;  
    border-radius: 28rpx;  
    display: flex;  
    flex-direction: column;  
}
```

```
.overlay-title {  
    text-align: center;  
    color: #606266;
```

```

}

.cell {
    display: flex;
    align-items: center;
    margin: 20rpx;
    background-color: #dcfffb;
    border-radius: 28rpx;
    height: 42%;
}

.explanation {
    margin-left: 30rpx;
    font-size: 44rpx;
    font-weight: bold;
    color: #02c39a;
}

.hover_cell {
    opacity: 0.9;
    background: #e4e8ee;
}

.van-image {
    display: inline-block;
    position: relative;
    width: 160rpx;
    height: 160rpx
}

```

## 6.2 云开发模块

在本系统中，部分与云数据库操作有关的操作，以及涉及获取定位的有关操作，都使用了微信云开发工具中的云数据库及云函数等功能。使用微信云开发可以将代码直接上传到微信提供的腾讯云空间中，而不需要部署在其他的服务器上。这样的开发模式极大的简化了开发者部署和维护的成本。

下面展示的是本系统中获取定位的云函数。当用户修改/建立简历、修改/新建职位信息、修改/新建公司信息的时候，都需要用户输入工作或公司地址。为了实现程序能够根据用户距离工作地点距离进行排序，在数据库中需要存储工作地点的精确地理坐标，在这里需要实现地址与经纬度坐标的互相转化。

由于在本系统中多次使用了这一功能，因此为了减少代码的重复，在这里将其定义为了云函数。当程序在调用这一云函数时，其会使用腾讯地图所提供的相应接口，将用户传入的地理坐标或详细地址进行相互转换，同时也包含了部分信息，如该地址所在的行政区等相关信息，返回给前台显示。这是实现根据地理位置对职位进行排序的重要依赖项目。除此以外，云数据库在本项目中也有重复的使用，其查询大多写在了小程序的 JS 文件中，在此就不再赘述。

**表 6.5 获取定位云函数代码**

```
async call(obj) {  
    return new Promise(async (resolve, reject) => {  
        try {  
            const cloud = await this.cloud()  
            if (obj.load !== false) { // 如果 load 不为 false  
                wx.showLoading({ // 展示加载 load  
                    title: obj.tips || '加载中', // 如果没有 tips,  
默认加载中  
                    mask: true  
                })  
            }  
            const res = await cloud.callFunction({ // 发起云
```

函数请求

```
        name: 'lbs_server', // 应用统一服务函数
        data: {
            type: obj.name, // 服务类型
            data: obj.data // 服务数据
        }
    })
    console.log('【云函数调用成功】', res)
    wx.hideLoading() // 取消加载
    if (res.errMsg === "cloud.callFunction:ok") { //
如果返回 code 为 0，则业务正常
        resolve(res.result)
    } else { // 如果返回 code 不为 0，则证明有点问题
        let tips = res.result.msg || '服务调用出现未知
问题，请重新尝试！' // 提示文字如果没回传，默认提示
        if (tips === '查询无结果') { // 如果提示文字为这
个，不说人话
            tips = '没有找到地址，请在地址中填写城市名，并尽
可能详细' // 改为说人话
        }
        wx.showModal({ // 弹出提示框
            content: tips + ', 请阅读项目目录 readme.md
文件，根据步骤设置',
            showCancel: false
        })
    }
} catch (e) { // 网络问题出现
    let flag = e.toString()
    flag = flag.indexOf('FunctionName') === -1 ?
```

flag : '请在 cloudfunctions 文件夹中 lbs\_server 上右键，创建部署云端安装依赖，然后再次体验'

```
        console.error('【云函数调用失败】', flag)

        wx.hideLoading()

        wx.showModal({
            content: flag, // 此提示可以在正式时改为 "网络服务
异常，请确认网络重新尝试！"

            showCancel: false

        })
    }

    })

}
```

## 第七章 测试用例

该部分对将测试小程序中大部分功能以及其中有特点的内容。

### 7.1 提交/修改简历信息测试

如图 7.1，在简历信息对应的表单中，当用户点击提交简历的按钮时，栏目左上角的红色星号表示必填项，在检验过程中，填空不能为空，其次手机号码必须满足手机号码的正则表达式，填写的年龄必须大于 18 岁等。若通过检验，显示正在保存的提示，并重定位到简历查看的界面。

The figure displays two screenshots of a resume creation interface. The left screenshot shows the form with red asterisks indicating required fields: \*真实姓名 (Real Name), \*年龄 (Age), \*性别 (Gender), \*教育经历 (Education), \*工作城市 (Work City), \*手机+86 (Mobile Number), and \*空闲时间 (Vacant Time). The right screenshot shows the form after submission, with a '正在保存...' (Saving...) overlay and a green '保存简历' (Save Resume) button at the bottom.

图 7.1 提交/修改简历信息测试

### 7.2 增加/修改岗位信息测试

如图 7.2，当招聘者获得招聘资质之后，可以在个人中心发布新的岗位，或者在招聘资质中对已有岗位信息进行修改。在发布岗位中，表单中的每个字段也进行正确性验证，正确填入后，可以点击最下方的保存并发布，操作成功后直接跳转到岗位详情界面。



图 7.2 增加/修改岗位信息测试

### 7.3 增加/修改公司信息测试

如图 7.3，若招聘者需要进行招聘，需要在个人中心获取招聘资质，当用户填写表单中的信息满足对应的函数的校验，即可点击保存，在 `work_company` 中添加新的记录，或者根据原有的 `job_id` 找到对应的记录进行修改。在完成保存之后，会跳转公司详情。

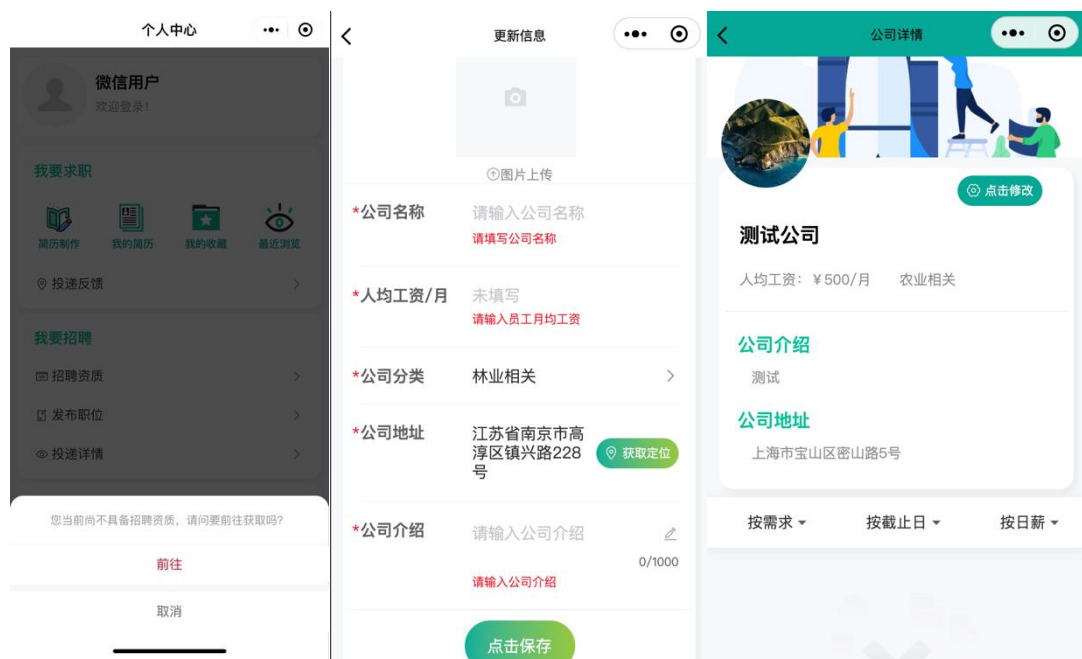


图 7.3 增加/修改岗位信息测试

## 7.4 聊天功能测试

如图 7.4，求职者在浏览岗位时，点击岗位详情最下方的“发消息”，可以和该岗位对应的招聘者进行聊天，其中可以使用文本和表情符号，同时自己的信息被对方读取后，会在消息下方显示“已读”标记。当招聘者修改投递过的简历状态为录用成功之后，可以收到来自应聘者的录用成功通知。

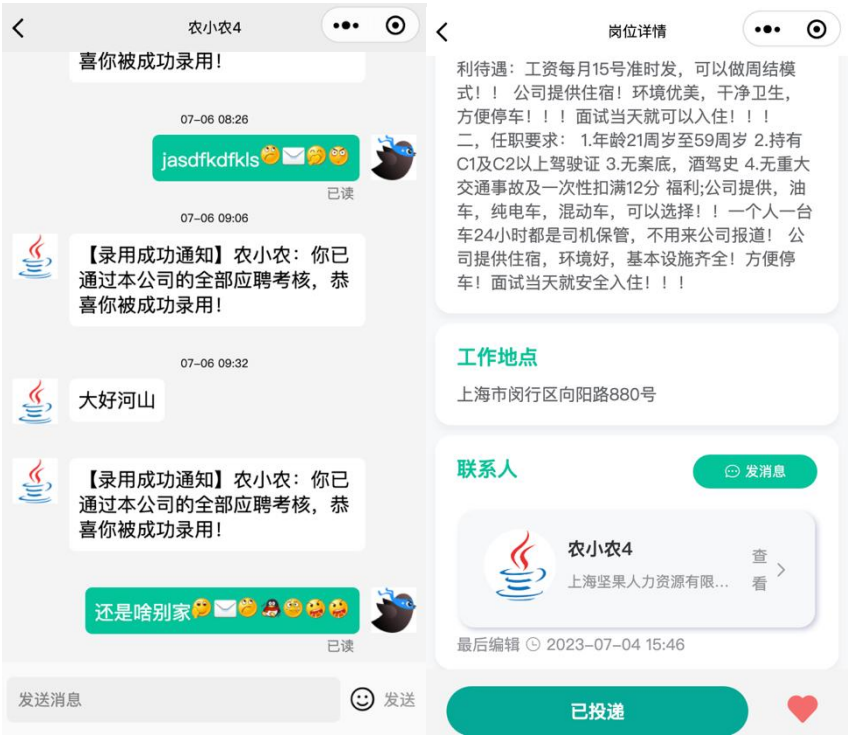


图 7.4 求职者聊天功能测试

如图 7.5，应聘者在投递详情中可以查看自己的公司所有岗位的列表，点击某一个岗位可以进入查看具体的信息，三个按钮可以分别进入发送消息、查看该求职者简历和进行筛选三个功能。其中通过点击“进行筛选”按钮后，从下方弹出不同阶段的选项，用于筛选求职者以及记录求职者目前所处的阶段，每次改变状态后，在 work\_cv\_status\_record 表中增加新记录，用于给用户求职状态追踪。当点击“录用成功”选项后，会调用聊天中对应的方法发送指定的内容。



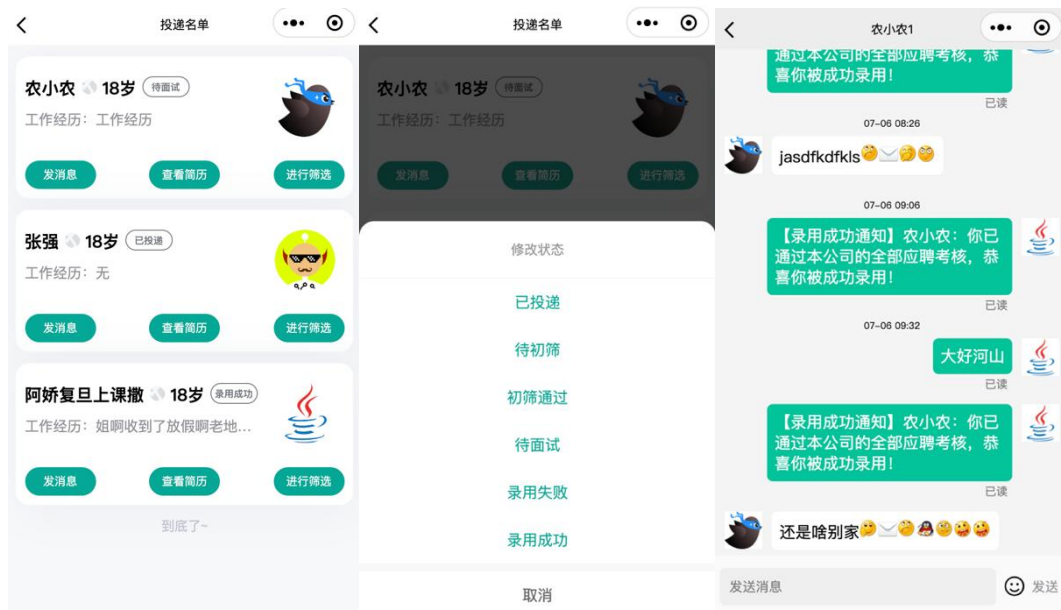


图 7.5 应聘者聊天功能测试

## 7.5 查看/筛选岗位信息测试

在首页可以看到各种岗位的信息列表, 首页中部的三个标记表示不同的筛选方式, 点击可进入按时间发布排序、按热门程度排序、按岗位距离排序的岗位信息。点击上方的搜索框可以看到不同的职业分类以及搜索记录, 选择分类后, 根据 work\_job 表中的字段进行筛选, 显示筛选过的内容。



图 7.6 查看/筛选岗位信息测试

## 7.6 投递简历状态测试

当求职者提交简历后，招聘者可以在投递详情中看到不同岗位中的列表，进入后可以看到不同投递者投递信息，在点击某一个岗位之后，可对筛选状态进行修改，而求职者可在投递反馈中看到自己所有投递的岗位，点击进入后，可看到投递的状态。



图 7.7 投递简历状态测试

## 7.7 地图定位测试

招聘者可对公司的定位进行设置和修改，进入定位页面后，可以使用经纬度或地址进行查询，同时可在地图上点击进行更精确的定位，下方显示经纬度坐标和推荐地址。在发布岗位中，同样适用。

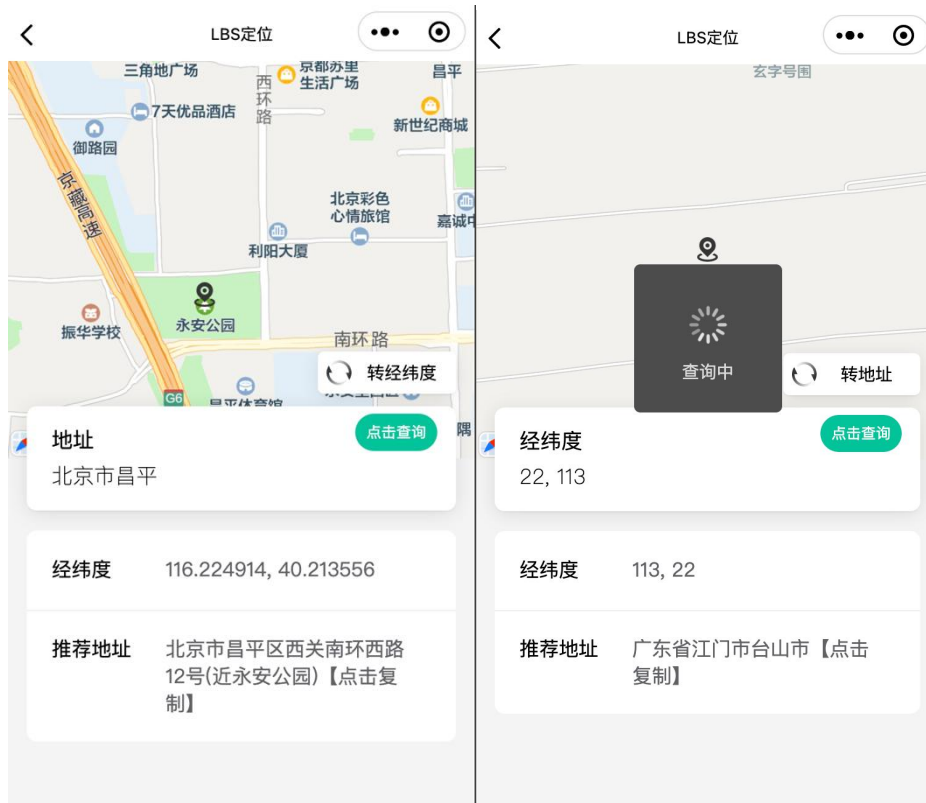


图 7.8 地图定位测试

## 7.9 浏览历史/收藏夹测试

求职者点击进入岗位详情后，记录为该用户的历史浏览记录，并可在个人中心右侧“最近浏览”中查看浏览过的岗位，在收藏后，work\_collective 表中加入一条新的记录，并保留当前岗位信息，当招聘者更新岗位信息后，收藏夹中保留旧版信息，方便求职者。当求职在岗位详情下方点击“爱心”后，样式改变为“红色爱心”，并可在个人中心右侧“我的收藏”中查看收藏过的岗位。

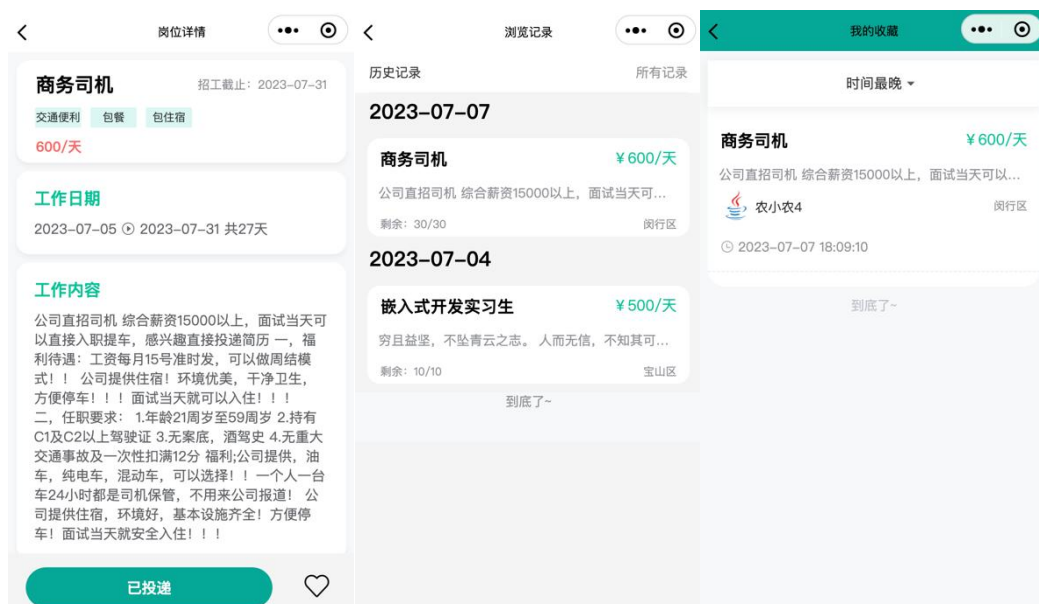


图 7.9 浏览历史/收藏夹测试

## 第八章 用户手册

### 8.1 软件概述

#### 8.1.1 软件背景

由于城市化进程的推进，使得中国农村人口流失严重，许多年轻人选择到城市工作，导致农村人口老龄化和劳动力短缺。由于信息相对闭塞，此前的农村居民寻找工作的途径有限，这不仅会造成招聘信息传递的不及时、不准确，还会导致招聘双方的信息不透明。

近年来，智能手机已经逐渐普及，农村居民也能够接触到互联网。尽管互联网上的招聘平台众多，但是这些招聘平台大多相对复杂，往往不能够在手机端完成全部的招聘、应聘工作，对于农村居民相对不友好。与此同时，用户想要发布招聘信息需要大量的资质审查，对于农村居民难度相对较大。

因此，本系统计划制作一个易于使用的平台，方便农村居民通过移动小程序，直接寻找心仪的工作岗位；同时，可以直接在小程序上发布招工消息，而不需要使用电脑或提交复杂的材料来获得资质。

#### 8.1.2 软件功能

- (1) 用户可以直接使用微信登录系统；
- (2) 求职者可以直接在系统中快速制作简历；
- (3) 求职者可以根据多种条件查找职位；
- (4) 用户可以直接在小程序中发起会话；
- (5) 求职者可以向心仪职位投递简历；
- (6) 求职者可以直接跟踪已投递职位的审查进度；
- (7) 求职者可以查看最近浏览过的职位、收藏的职位；
- (8) 招聘者可以填写公司基本信息，获取招聘资质；
- (9) 招聘者可以发布新的职位信息；
- (10) 招聘者可以修改、下架已经发布的职位信息；
- (11) 招聘者可以直接查看求职者投递的简历；
- (12) 招聘者可以对求职者进行筛选，并修改其简历审查进度；

(13)用户可以查看劳动法律知识科普；

## 8.2 运行环境

本系统对运行环境没有严格的要求，只需要移动设备能够运行微信客户端，并且能够连接到互联网，即可运行本系统。因此最新的 iOS 和 Android 设备均可运行本系统，具有良好的兼容性。

## 8.3 使用说明

### 8.3.1 求职者模块

#### (1) 职位搜索—分类检索、关键字查询

用户进入系统，点击上方的搜索栏即可进入职位搜索页面。求职者可以直接在上方的搜索栏中输入关键词进行职位的搜索，也可以在下方的职业分类中点击感兴趣的职业类型进行搜索。如果用户还想在上面的基础上对职位的地区进行限制，可以点击搜索栏左侧的“全国”按钮选择地区。打开后，系统会自动根据当前的定位，提示当前所在的区域；用户也可以在下方的选择栏中自行选择想要限制的范围。

搜索完成后，会显示所有符合条件的职位信息。用户可以自行根据职位需求量、招聘截止日以及薪资的多少自行排序查看。用户也可以在首页根据工作地点距离当前位置的远近来对所有职位进行排序。



图 8.1 岗位搜索

### (2) 简历制作—填写信息、生成简历

用户进入个人中心后，就可以选择简历制作来制作简历。本系统提供了简历模板来收集求职者的真实个人信息，用户只需要根据提示填写相关信息就可以完成简历的制作。同时，为了方便用户制作简历，部分信息如性别、教育水平、工作城市等都提供了选项，用户只需要选择对应的选项，就可以快捷地完成简历制作。用户保存简历后，系统会自动打开填写的简历供用户进行核对。

The figure displays two mobile application interfaces for resume creation. The left interface, titled '简历查看' (Resume View), shows a completed resume for a user named '小农女' (Xiao Nong Nu), age 25, female, with a phone number 13612345672. It includes sections for '基础信息' (Basic Information), '教育经历' (Education Experience) as a Master's degree, '工作城市' (Work City) as Yumeng District, '空闲时间' (Free Time) from 2022-6-1 to 2022-6-1, '自我评价' (Self-Evaluation), '技能特长' (Skills and Hobbies) in electronics and circuit design, and '工作经历' (Work Experience) at a power plant. The right interface, titled '简历制作' (Resume Creation), shows the same form with input fields and a green '保存简历' (Save Resume) button at the bottom.

图 8.2 简历制作

### (3) 投递简历—选择工作、跟踪进展

当用户检索到了其感兴趣的职位，并想要投递简历，可以直接进入到职位详情页面，求职者可以点击下方的投递简历按钮向心仪的公司投递简历。此外，求职者还可以使用发消息功能，直接给招聘者发送私聊消息，具体的内容在下文中还会具体介绍。投递完成后，用户可以从个人中心中的“投递反馈”功能来查看该用户已经投递的所有岗位以及简历筛选的状态。如果用户想要进一步的查看某一职位的具体录取状态，以及简历在何时状态被招聘者修改，可以点击对应的岗位。打开后，系统会使用时间线的方式直观的跟踪简历实时的进展，如已投递、面试成功、录用成功等状态。如果应聘者最终被录用，还会使用下文介绍的私信功能自动的给应聘者发送一条消息，提示其已经被录用。



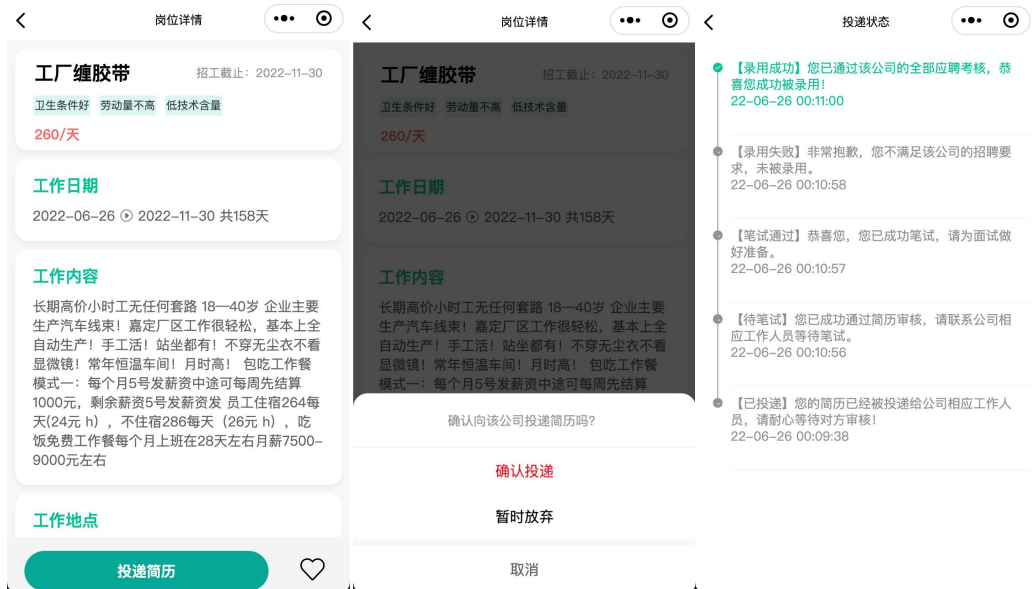


图 8.3 投递简历

#### (4) 收藏与历史—职位收藏、浏览记录

用户在浏览职位的时候，还可以对心仪的职位进行收藏。在职位详情页面，点击右下角心形图标的方式就可以收藏对应的职位。收藏的职位此后可以在“我的”页面中的“我的收藏”中查看收藏的职位以及收藏的时间。此外，当用户在浏览所有的职位的过程中，系统会自动记录其浏览历史。用户可以在“我的”页面中的“最近浏览”中按日期查看所有此前浏览的岗位信息。

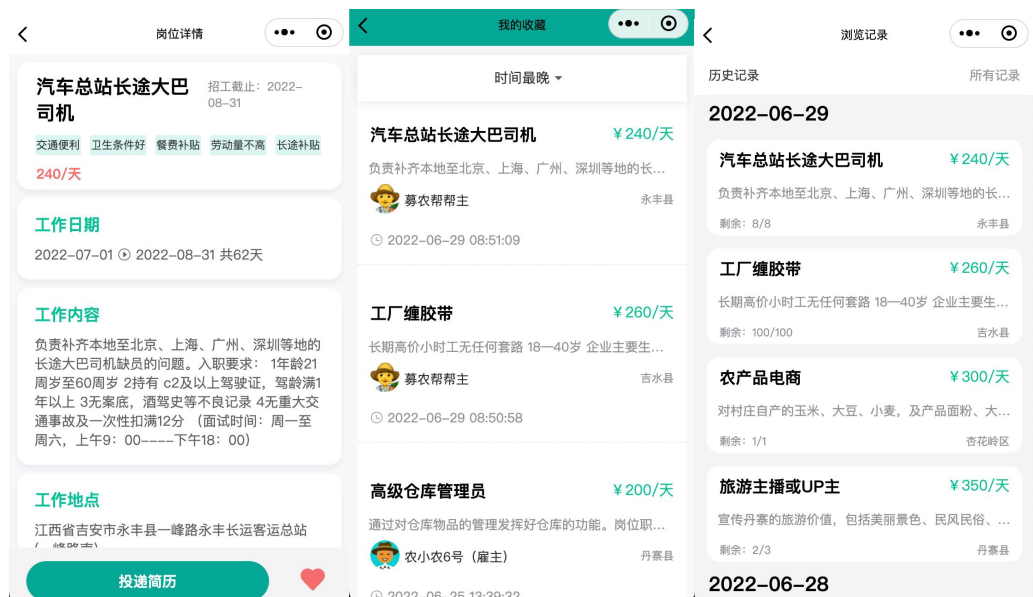


图 8.4 收藏与历史



### 8.3.2 招聘者模块

#### (1) 发布职位—获取资质、工作介绍

用户如果想要通过平台发布招聘信息，需要填写公司、集体或个体的相关基本信息，方可发布职位信息。招聘方可以通过平台创建新的职位要求，用户只需要在平台中填写职位的名称、薪资、招募人数、职业分类、工作地址、起止时间和工作内容后，即可一键发布职位。此处填写的工作地点会被用于计算用户距离工作单位的距离，从而完成先前提到的根据距离进行排序。

发布完成后，招聘方可以查看其发布的所有的职位信息。用户可以点击岗位旁的编辑或下架按钮，从而分别实现对岗位描述信息的修改，和对特定岗位的下架处理。被下架的岗位将无法被其他用户检索到。



图 8.5 发布职位

#### (2) 私信交流—沟通需求、双向选择

平台提供了私信交流功能，使得求职者可以主动与招聘者进行私信交流；同时，招聘者也可以主动与投递了简历的求职者进行私信交流。具体来说，求职者可通过“岗位详情”页面的“发消息”按键向职位发布者发起会话；招聘者可通过“投递详情”页面中的“进行筛选”按键，向应聘者发起会话。平台的私聊功能允许用户发送表情，同时还提供了已读回执功能，双方都可以看到对方是否已经阅读了发送的消息，提高了交流效率。

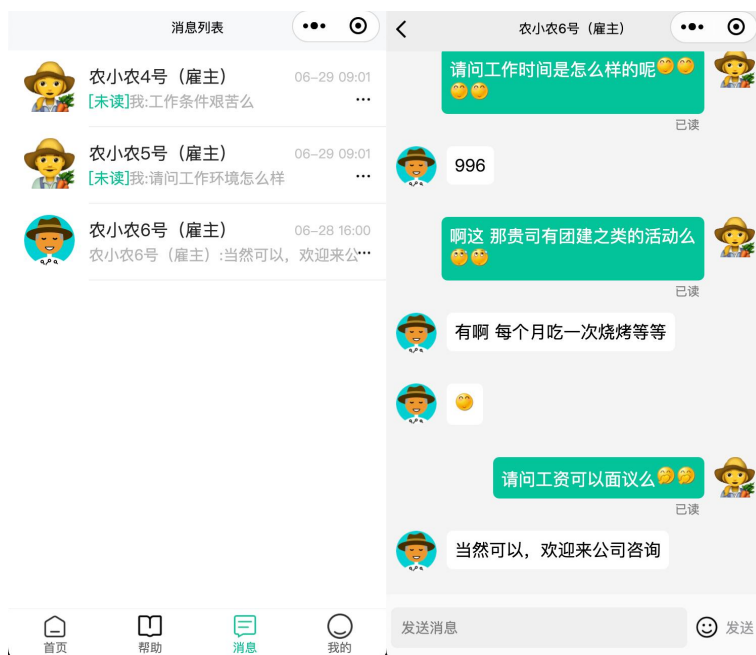


图 8.6 私信交流

### (3) 筛选录用—查看简历、分析决策

招聘者可以直接在平台上进行简历的筛选和录用工作。招聘者进入到“投递详情”功能后，选择对应的岗位，就可以直接在列表中看到所有投递简历的求职者的基本信息，包括姓名、年龄、性别、工作经历及其当前的简历投递状态。招聘者可直接点击“查看简历”按钮求职者的相关简历；可以使用“进行筛选”按钮修改求职者简历的录用状态。这一状态会被追加到数据库中，求职者可以在“投递反馈”功能中看到简历筛选的进度以及状态变化的时间。

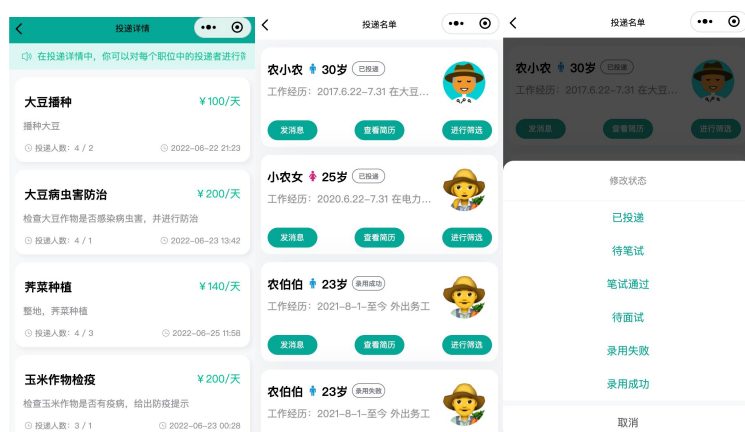


图 8.7 筛选录用

8.3.3 帮助模块

(1) 法律帮助—劳动法科普

平台提供了部分法律帮助功能，提供了劳动法以供用户查阅。求职者与招聘者可以在法律科普栏目了解劳动法，保护自己的合法权益。此外平台还整理了一些共性的问题，在平台上做了统一的解答。



图 8.8 法律帮助

## 第九章 个人体会

**组长：胡才郁**

在此次微信小程序的开发过程中，我不仅提升了自己的开发技能，更深入地领略了开发的全过程。做小程序开发，尤其是用户界面设计时，考虑到用户体验和交互设计等因素，既让我感受到挑战，也锻炼了我的权衡取舍能力。

一个在日常中看似简单的功能，实际在实现与维护过程中却是十分地精细。在开发中，各项功能之间如何有效地互相关联，使结构清晰、方便后续增加和维护，这让我实践了系统性和创新性的思维。我认识到，小程序的开发不仅是技术的学习和应用，更涉及到团队合作的能力。

开发微信小程序是一个不断学习、不断刷新知识的过程。随着开发技术的不断更新，我在开发过程中需要保持紧密的跟进状态，时时关注新的技术，并将其运用到实际项目中，这也锻炼了我解决问题的能力。

小程序开发中的云数据库和云函数，是这次项目中的最大技术难点。云数据库，则让我有机会体验到无服务器开发模型的便利。它自动处理扩展问题，使我们在处理大量数据时无需担心性能和存储问题，大大节省了我们的时间。并且，云数据库能够有效地实现数据的实时同步，极大地提高了用户体验。云函数，为微信小程序提供了在云端运行代码的能力，使得我无需关心服务器等硬件设施，只需专注于业务逻辑的实现。我发现，使用云函数不仅可以降低开发的复杂性，构建和维护也变得相对容易，极大地提高了开发的效率。

总的来说，此次微信小程序的开发让我感受到了开发的全过程和魅力，而关键的，是它让我更深刻地认识到团队协作和有效沟通的重要性。在遇到问题时，我们可以集思广益，找到问题的根源，这无疑能大大提升开发的效率。尤其是在需要频繁测试的过程中，有效的协作与沟通显得至关重要。

**组员：黄逸弘**

在「募农帮」微信小程序项目中，我主要承担了小程序中职位搜索功能的实现，以及按照多种筛选方式对职位进行排序的任务。尽管此前没有接触过小程序

的项目开发，但是通过这个项目，我学到了很多有关微信小程序开发和前端技术方面的知识和技能，以及后端相关的微信云开发、云函数、云数据库等功能。

在实现职位搜索功能的过程中，由于其需要从数据库中获取所有的职位信息，这让我更深入的了解云数据库用法。相比于传统的数据库，云数据库极大的简化了小程序的上线部署工作；在使用数据时，只需要在小程序的代码中，通过几行简单的代码，就可以调用到云数据库中的相关数据，简化了开发的流程。这一过程避免了传统数据库中需要写 SQL 语句的检索方式，降低了开发成本。

同时，在项目中还需要与其他团队成员紧密合作，沟通项目的进度以及共同解决遇到的问题。庞大的项目是无法通过一个人完成的，其必然需要团队合作，因此团队成员之间的沟通就显得格外的重要。通过有效的沟通，可以提高项目进行的效率。

## 组员：徐鼎力

这次作为参与微信小程序开发的一员，能够参与其中的每一个开发阶段，对自己的开发能力有一定的提升。微信小程序开发具有一定的挑战，尤其是在设计用户界面时，需考虑用户体验、交互设计和界面美观等方面，在需求和技术之间做出权衡。比如在小程序中的主要功能，简历的制作、投递、查看以及面试录用的多个环节，这些功能是相互关联的，要在实现功能的同时，使得结构较为清晰，方便后续功能的增加以及维护，这需要我们具备系统性的思维、创新思维和团队合作的能力。

同时，微信小程序开发是一个持续学习和不断迭代的过程，小程序的开发技术也在不断地更新，包括本次小程序中使用的云函数、云存储和云数据库功能。在开发过程中，我们需要及时跟进最新的技术动态，在发现问题时，查看文档，更新自己的开发知识，并应用到实际项目中。

最后，微信小程序开发让我深刻认识到团队协作和沟通的重要性。在项目中，与其他同学的紧密合作是必不可少的。通过有效的沟通和协作，我们能够更好地理解要求和问题，比如在碰到一些 bug 时，共同协助找到问题所在，提升了开发的效率，比如在聊天功能中，需要频繁测试，在写协作中可以更好地完成开发与测试。

## 组员：张俊雄

在参与开发旨在帮助农民工朋友找工作的“募农帮”微信小程序的过程中，我深刻感受到了技术的社会影响和创新力量。作为招聘简历投递、查看详情和公司简历审核等关键环节的主要负责人，我不仅是代码的编写者，更是社会责任的践行者。“募农帮”小程序为农民工朋友提供了一个便捷的求职平台，这让我深感振奋和自豪。每一个成功的匹配背后，都是我和团队不懈努力的结果，也是对于技术价值的验证。招聘、查看、审核等功能的实现，不仅展现了技术的实用性，更体现了技术的人文关怀，这是我深深被感动的地方。

同时，微信小程序开发和微信云开发为我打开了通向创新世界的大门。通过这些工具，我不仅可以将想法迅速实现，还能够实现复杂功能，提升用户体验。更重要的是，这个过程教会了我团队协作的重要性，锻炼了解决问题的能力，使我在技术和团队合作方面都得到了巨大的成长。这次经历让我更加坚定了在技术领域不断学习和创新的信念。我期待继续投身于技术创新，为社会贡献更多有益的解决方案。通过技术，我愿意持续为改善人们生活贡献力量，成为社会进步的推动者，创造更加美好的未来。

## 参考文献

[1] 微信开放文档[EB/OL]. [2023-07-06].

<https://developers.weixin.qq.com/miniprogram/dev/framework/>.

[2] 介绍 - Vant Weapp[EB/OL]. [2023-07-06].

<https://youzan.github.io/vant-weapp/#/home>.

[3] Webservice API | 腾讯位置服务[EB/OL]. [2023-07-06].

<https://lbs.qq.com/service/webService/webServiceGuide/webServiceOverview>.