**项目介绍文档**

**“呼叫邻里”农村招聘小程序v1.0**

小程序AppID: wx874904c38445429a

**开发人员**: 林德福2014580

刘志昊2014581

**文档编写**: 林德福2014580

刘志昊2014581

**编写日期**: 2022.05.24

目录

[一、项目简介 1](#_Toc30077)

[1.1 开发背景 1](#_Toc21658)

[1.2 应用场景 1](#_Toc22153)

[二、系统需求分析 1](#_Toc2835)

[2.1 用户角度需求分析 1](#_Toc13792)

[2.2 功能角度需求分析 2](#_Toc13801)

[三、系统设计 2](#_Toc26478)

[3.1 总体功能设计 2](#_Toc25903)

[3.2 技术选型 3](#_Toc32622)

[四、系统具体实现 3](#_Toc22238)

[4.1 开发环境 3](#_Toc31732)

[4.2 整体架构 3](#_Toc28853)

[4.3 前端技术 4](#_Toc29653)

[4.4 后端技术 5](#_Toc24062)

[4.5 数据结构和数据库 5](#_Toc15829)

[4.6 分模块具体实现 6](#_Toc6278)

[五、总结 7](#_Toc23108)

[5.1 系统主要优势和不足 7](#_Toc7753)

[5.2 使用到的开源代码列表 8](#_Toc31338)

## 一、项目简介

### 1.1 开发背景

长期来说，在乡村建设过程中，优质人才资源外流、人才结构失衡等问题日益凸显，乡村新业态的蓬勃发展、乡村集体产业的转型增值，迫切需要具有现代管理理念的经营型人才的加入，也需要乡村治理者具有更高的经营能力，职业经理人、各类专业经营人才的缺乏。希望设计些小程序，及时对接家乡人才需求，传递给在外的学子，或者想返乡的创业的人，以及当地想找工作的人。

短期来说，县域的农产品加工厂大多受季节影响，用短期工需求较多，农民朋友们也会有农忙与农闲的时候，双方都需要一个工具帮助对接起来，既解决农业企业的短期用工问题，又能帮助当地村民增收。

### 1.2 应用场景

针对上面提出的开发背景，我们结合农村招聘中需要兼顾地域与当地农耕需要的特点，开发了一款针对于短期用工需求进行招聘的微信小程序：“呼叫邻里”。它可以帮助需要招收短期工的雇主与需要兼顾农耕的乡亲们对接需求，为农闲时的农民创造更多的创收机会，为需要在农村开展业务的雇主快速找到雇员。

## 二、系统需求分析

### 2.1 用户角度需求分析

农村招聘程序的核心是对不同用户的核心需求进行精确分析。

1. 作为应聘者：我们希望其能够正常登录小程序，并且实时的编辑发布自己的简历，查看相关岗位信息并投递简历，可以及时的跟招聘人员进行交流，最好可以有一些技能培训和简历制作指导。
2. 作为招聘者：我们希望可以正常登录小程序，根据用工需求发布职位，实时查看收到的简历情况，对简历进行筛选，联系应聘者进行交流。

### 2.2 功能角度需求分析

**主要功能：**

1. 支持帐号登录体系，帐号登录、登出等；
2. 招工信息查询，可选择职位、薪酬待遇、行业类别、工作地点等；
3. 招工需求发布，可区分个人或企业身份，并能编辑相关用工信息；
4. 求职信息发布，个人可编辑相关求职信息发布；
5. 简历投递，可向心仪的招工企业或个人雇主投递简历；
6. 简单管理端，企业或个人雇主可对收到的求职信息进行处理，如简历查看，对信息打标记等。

**选做功能：**

1. 简历辅导，提供相关辅导资源，如视频、图文教程等；
2. 职业培训，提供相关培训资源，如视频、图文教程等；
3. 法律咨询，简单的劳动法知识宣传，如视频、图文等；
4. 简单客服咨询，可与招工企业或个人雇主进行客服咨询，支持文本、图片、文件传输（可适当使用官方或外部组件实现该能力）；
5. 支持订阅消息提醒。

（注：由于时间原因，可能无法将所有的功能都实现，因而优先实现主要功能）

## 三、系统设计

### 3.1 总体功能设计

1. 注册登录；
2. 身份选择（选择招聘者或求职者）和切换身份；
3. 招工需求/求职需求发布；
4. 已发布需求管理；
5. 招工信息/应聘信息列表；
6. 个人信息查看与修改；

### 3.2 技术选型

1. 使用Node.js技术编写后端代码，结合微信小程序云托管提供的部署服务，快速实现功能。
2. 数据库使用MySQL，可以与云托管提供的云存储适配。
3. 应用微信小程序开发工具，与云托管提供的前端API，编写前端代码。

## 四、系统具体实现

### 4.1 开发环境

·后端：本地调试使用Node.js版本为v18.2.0；云端Node版本未知，云端MySQL 5.7

·前端：微信小程序开发工具，调试基础库版本2.24.5。

·操作系统：Windows 11 家庭版

使用非LTS（长期支持版本）的Node.js的原因是，Node.js v18更新了原生的fetch方法，开发时方便在Node环境中调试网络接口。除此之外本程序的最终版本没有使用到最新Node的特性，因此后端服务使用低一些的Node版本应该也可以运行。

微信小程序运行时的基础库版本取决于客户端的微信版本，因此过于老旧的微信版本可能无法运行本程序。请确保运行时的基础库版本大于2.21.2。

### 4.2 整体架构



图4.1 系统整体架构图

本程序前端使用微信小程序技术实现，后端采用云托管技术托管了一个Node.js服务，属于传统的前后端分离架构。前端主要负责界面的渲染与呈现、界面与用户交互的逻辑。后端负责处理前端发送的网络请求，根据请求内容进行相应的运算和数据库操作。

### 4.3 前端技术

本程序的前端采用了微信小程序开发技术，主要使用wxml、wxss和JavaScript语言编写而成。另外，微信小程序还提供了一些可以影响程序编译的JSON配置文件，所以也用到了JSON语法。

**4.3.1 依赖管理**

微信小程序的依赖安装与传统使用Node构建的前端项目或基于HTML的前端项目有较大不同。本程序使用了两种方式管理依赖，一是直接导入，通过wxss文件的import语法和wxml的component语法，就可以直接将样式或组件依赖引入。本程序使用这种方法引入了colorUI组件库。这种引入方法的缺点是比较麻烦，需要手动添加引入语句。

此外，程序还使用了构建npm的方式引入依赖。方法是先使用npm init生成package.json，再使用npm安装依赖，最后使用微信小程序开发者工具自带的构建npm功能将安装的node\_modules编译成可以被小程序引入的形式。其默认文件夹名为miniprogram\_npm。本程序使用这种方法引入了weUI组件库。这种方法的缺点是，只有部分npm包提供了对微信小程序编译的支持，因此能够引入的包数量相对有限。

**4.3.2 微信小程序开发框架**

相比于传统前端开发框架，如Vue、React，微信小程序同样提供了组件化、模板语法、条件渲染、列表渲染、动态数据绑定等一系列功能，因此，有前端开发经验的开发者学习微信小程序开发会比较快。本程序在页面和组件中大量使用了这些功能。

此外，本程序还用到了微信小程序开发框架提供的一些特色功能，比如tabbar的配置、自带的UI组件、自带的网络请求封装（CallContainer）、自带的前端路由、与用户设备直接交互的API（如调起用户电话呼叫）等等。

### 4.4 后端技术

由于本小组为两人开发，因此采用前后端分离的云托管模式构建后端相比直接编写云函数后端更易分工，所以我们采用了云托管+Node.js服务的方式创建了小程序的后端。

在服务部署上，小程序使用的是微信提供的云托管服务。云托管服务可以自动关联开发人员的Github代码仓库，在本地编写完成相关代码之后，使用git命令将代码push到仓库中，可以直接触发流水线，进行后端服务的更新。经过研究，微信云托管应该是采用Docker技术，预先编写Dockerfile来实现这一自动化流程的。云托管赋予了项目CI/CD的能力，大幅提高了开发效率。同时，它还可以在一定范围内自动扩缩容Docker环境，来应对高负载场景，实现按量收费。

在后端代码编写上，本程序使用了Koa框架，原因其一是云托管有现成的可供修改的Koa后端模板，其二是Koa框架支持ES6 async await语法，可以提供一种更清晰的编程风格，便于后端代码的组织。另外，Koa框架的生态也比较完备，例如，本项目就使用了Koa-router依赖进行后端路由管理。

### 4.5 数据结构和数据库

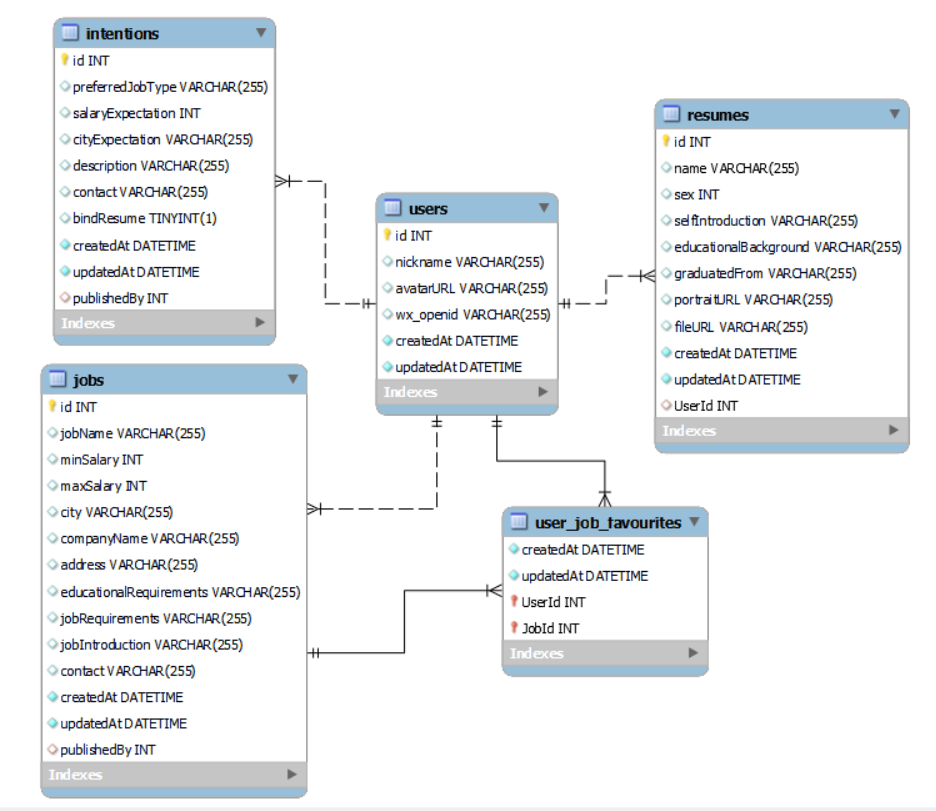


图4.2 数据库表关系图

由于云托管默认只支持云端MySQL作为数据库，因此本程序使用它存储数据。通过使用Sequelize ORM（Object Relational Mapping，对象关系映射），将关系型数据库的数据表抽象成模型（类），数据表中的记录抽象成实例（类的实例），大幅简化了后端代码中的数据库操作，避免了SQL语句的书写。

本程序的数据模型中，使用Sequelize 的关联功能定义了多个关联关系，其中包括一对一关系、一对多关系和多对多关系。Sequelize 通过数据库中的外键记录实现一对一关系和一对多关系的保存。而对于多对多关系，Sequelize 则是使用额外一张表存储数据之间的关联，并使用每一行记录的值作联合主键，最大化节约了空间。

### 4.6 分模块具体实现

**4.6.1 登录和注册**

为了响应微信保护用户隐私的号召，本小程序第一次登录时不会询问用户请求获得微信头像和昵称，而是以灰色头像和默认昵称代替。在用户第一次打开小程序时，前端代码会自动使用微信自带的封装方法CallContainer请求login接口。这个CallContainer方法会给网络请求附加一些额外信息，例如用户的wx\_openid。而每个用户的wx\_openid是唯一的，通过这个封装方法调用任何接口，后端都可以判断用户的身份。请求login接口后，如果后端发现是新用户，就会创建一个User模型的对象实例，将用户的wx\_openid录入，并设置默认的灰色头像和“微信用户”昵称，而后存入数据库。

**4.6.2 身份切换功能**

为了实现系统招聘者/求职者身份切换的功能，小程序将判断用户身份的字段在前端存储了两份。一份是使用wx.setStorageSync API将数据存储到本机缓存中。使用同步而非异步的API原因是该字段占用空间很小，使用同步API可以避免思考代码执行顺序带来的麻烦。这一份主要是为了保证在用户第二次及以后使用小程序的时候，可以记住用户之前的选择。另一份挂载在小程序实例对象上，通过在小程序入口文件中给小程序的构造函数传参时添加一个globalData对象，就可以实现全局状态管理，让globalData对象内的所有属性对所有页面和组件可见，使用getApp()方法就可以很轻松地访问。这一份主要是为了实现当状态改变时及时触发其他页面或组件的状态更新，实现类似vuex中的mutation功能。

**4.6.3 招工需求/求职需求发布**

招工需求/求职需求发布页面使用一个表单来实现。表单的前端使用了WeUI的Cell组件来实现简单的样式。在前端代码中，每个表单与微信小程序中ViewModel层的一个对象进行单向绑定，当表单发生变化时，对象的值就会发生相应的变化。当点击提交表单的按钮时，触发按钮的Click事件，JS会在事件的回调函数中以JSON格式携带ViewModel层表单对象的数据发送HTTP POST网络请求，将数据传递到后端。后端将前端的数据进行反序列化之后得到JS对象，在对对象的值进行验证后，返回失败或成功的HTTP Response，并使用Sequelize的相关API进行异步的数据库操作。

**4.6.4 已发布需求管理/招工信息、应聘信息列表**

标题所述功能前端使用wx:for语法列表渲染实现。在相关页面加载时，触发onShow()生命周期函数，函数会请求后端列表信息接口，从而获得对象列表。前端列表渲染的每个卡片对应一个JS对象，每个对象又对应后端数据表的一个实例，而每个实例又对应数据库中的一行记录。

**4.6.5 个人信息查看与修改功能**

个人信息查看和修改功能目前实现了头像与昵称部分。头像的上传功能是通过微信button组件open-type=avatar来实现的，该类型button可以触发头像上传事件，并在回调函数中提供头像图片的URL。通过对默认button组件的样式重写，就得到了点击头像即可修改头像的效果。个人信息页还加入了自定义gif动画效果，以提高美观度。

## 五、总结

### 5.1 系统主要优势和不足

本系统的主要优势在于，相比一般的招聘用工软件，作为求职者用户，可以主动发布附带自己个人信息的求职意向，系统会自动将求职意向推送给招工者。这个功能多数的招聘用工软件都是没有的。在农村招聘市场小、用工灵活性强的背景下，这个功能可以大大提高求职用户被成功招聘的可能，解决供需双方市场不对等的问题。

### 5.2 使用到的开源代码列表

微信小程序官方文档示例

WeUI（微信官方UI库）

ColorUI

微信云托管Koa框架模板

Node.js

Koa.js

Sequelize

koa-router koa-cors koa-bodyparser axios