

# 乡村本地招聘信息工具

---

## 一、项目背景

近些年随着城市化的发展，乡村面临严重的结构性人才失衡问题。由于缺乏各类专业经营人才，也缺乏合理的人力资本生态，乡村产业在现代化转型的过程中面临严重困难，乡村居民的生活也受到了很大影响。在这种情况下，对于招聘信息共享平台的需求应运而生。

## 二、需求和功能说明

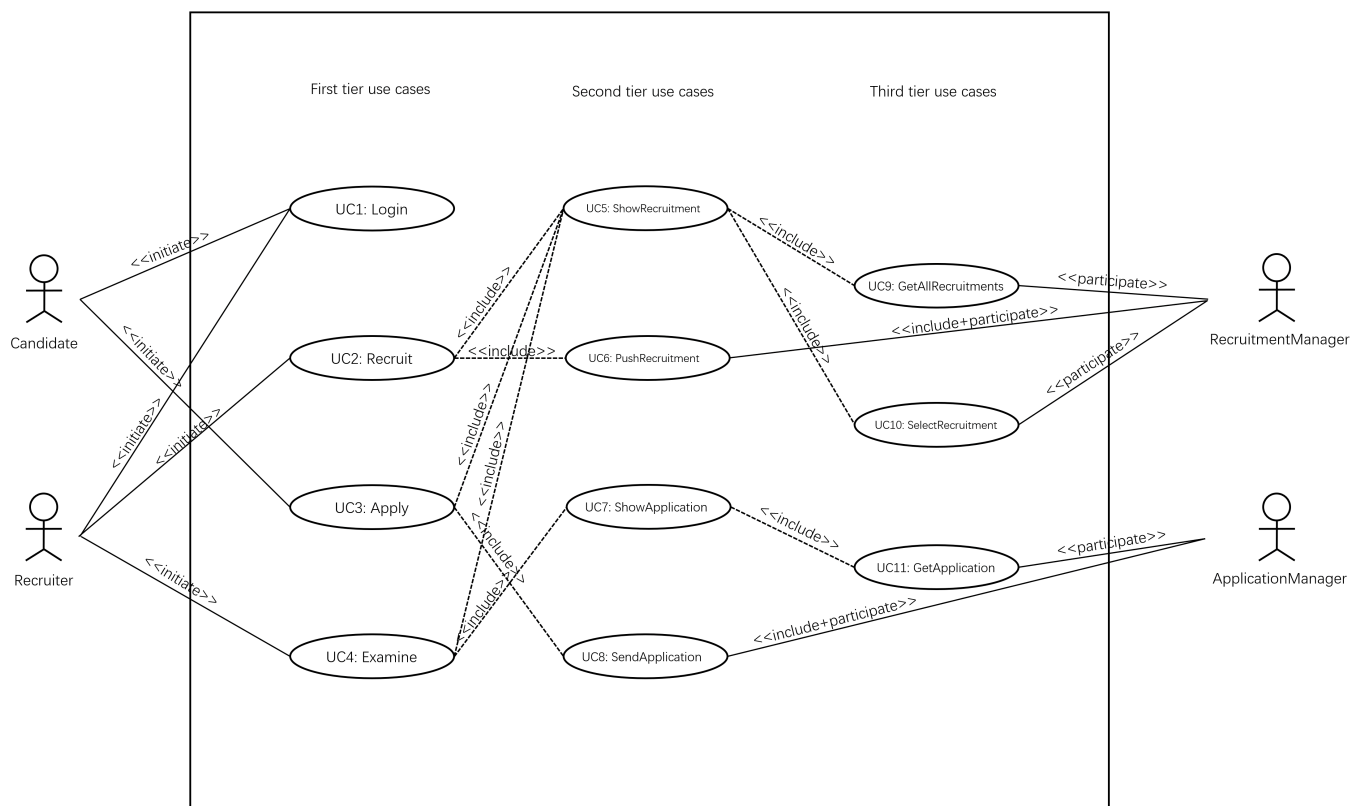
### 1. 应用场景

1. 由村委会、村党支部等发布招聘信息，从城市中招聘高级经营管理人才，支援乡村建设。
2. 地方政府发布创业扶持计划，吸引人才返乡创业。
3. 企业招聘本地人才，解决当地就业问题。
4. 个人和企业农忙等时期招聘短期工。

### 2. 实现功能

1. 支持帐号登录体系，帐号登录、登出等。（已实现）
2. 从数据库中读取招工信息并显示。（已实现）
3. 招工信息查询，可选择职位、薪酬待遇。（已实现）
4. 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。（已实现）
5. 求职信息发布，个人可编辑相关求职信息发布。（已实现）
6. 简历投递，可向心仪的招工企业或个人雇主投递简历。（已实现）
7. 企业或个人雇主可对收到的求职信息进行处理。
8. 企业或个人雇主可修改或删除已发布的招聘信息。

### 3. 用例图



## 4. 用例说明

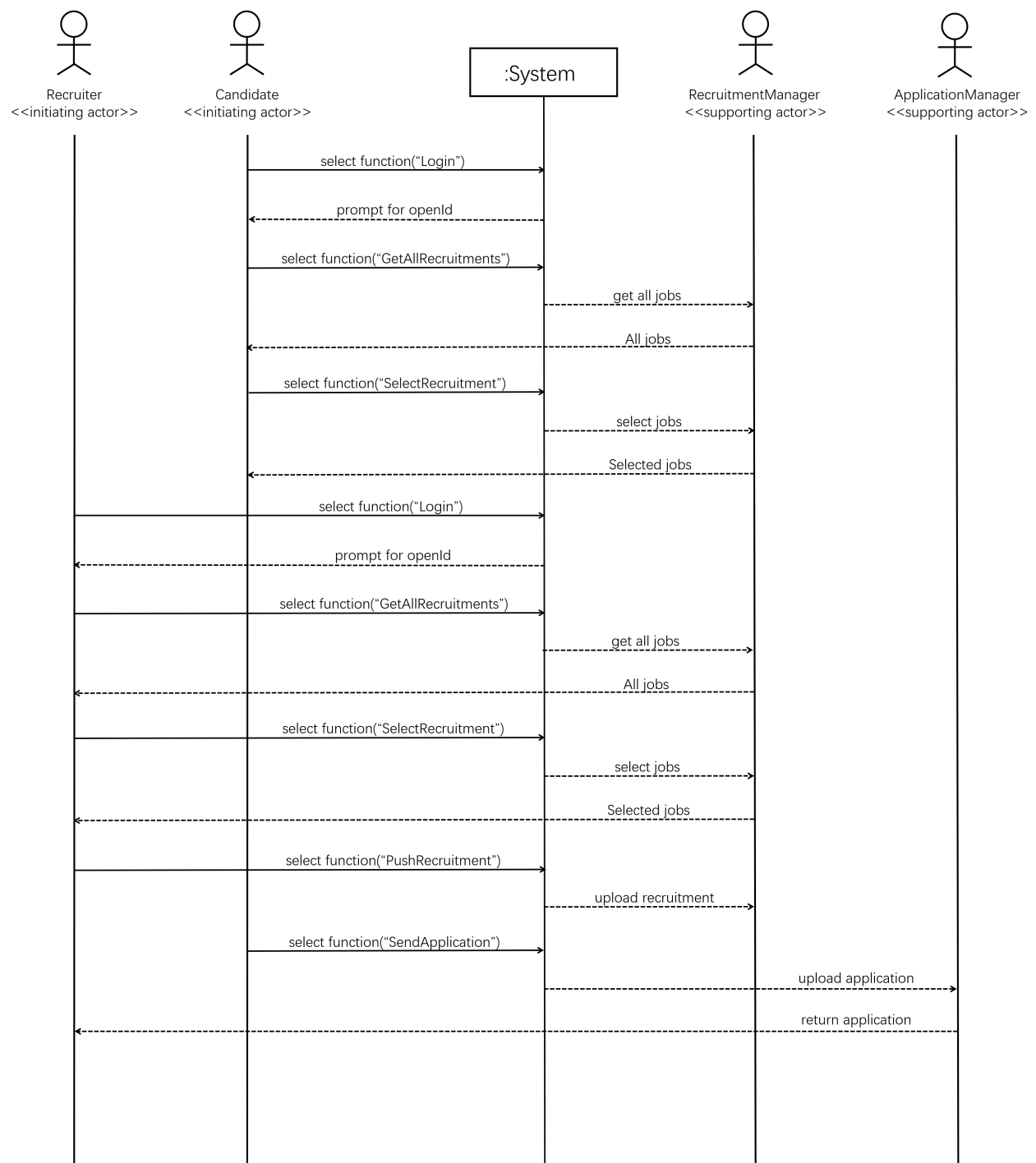
- 求职者在登录后，可以通过查找的方式获取招聘信息，并向对应的招聘者提交求职申请并上传简历。
- 招聘者在登录后，可以发布招聘信息，管理自己发布的招聘信息，以及管理求职者的求职申请。

## 三、技术实现说明

### 1. 技术栈

- 前端：微信小程序
- 后端：微信云开发

2. 时序图



3. 接口设计

1. getOpenId

- Request

```
wx.cloud.callFunction({  
  name: 'getOpenId',  
})
```

- Response

```
{  
  openid: wxContext.OPENID,  
  appid: wxContext.APPID,  
  unionid: wxContext.UNIONID,  
}
```

## 2. getAllJobs

- Request

```
wx.cloud.callFunction({  
  name: 'getAllJobs',  
})
```

- Response

```
{  
  data: [{  
    _id: "",  
    recruiterID: "",  
    isEnterprise: "",  
    jobtitle: "",  
    salary: "",  
    number: "",  
    location: "",  
    note: "",  
  }, {...}, ..., {...}]
```

```
}
```

### 3. getJobById

- Request

```
wx.cloud.callFunction({  
  name: 'getJobById',  
  data: {  
    jobid: "",  
  }  
})
```

- Response

```
{  
  data: [{  
    _id: jobid,  
    recruiterID: "",  
    isEnterprise: "",  
    jobtitle: "",  
    salary: "",  
    number: "",  
    location: "",  
    note: "",  
  }]  
}
```

### 4. pushRecruitment

- Request

```
wx.cloud.callFunction({  
  name: 'pushRecruitment',
```

```

data: {
  msg: {
    recruiterID: "",
    isEnterprise: "",
    jobtitle: "",
    salary: "",
    number: "",
    location: "",
    note: "",
  },
}
})

```

- Response

```

{
  success: true
}

```

## 5. selectJobs

- Request

```

wx.cloud.callFunction({
  name: 'selectJobs',
  data: {
    isEnterprise: "",
    jobtitle: "",
    minsalary: "",
  }
})

```

- Response

```
{
  data: [{
    _id: "",
    recruiterID: "",
    isEnterprise: "",
    jobtitle: "",
    salary: "",
    number: "",
    location: "",
    note: "",
  }, {...}, ..., {...}]
}
```

## 6. getResumeld

- Request

```
wx.cloud.callFunction({
  name: 'getResumeld',
  data: {
    msg: {
      jobID: "",
      recruiterID: "",
    },
  }
})
```

- Response

```
{
  id: "",
}
```

## 7. deleteResume

- Request

```
wx.cloud.callFunction({
  name: 'deleteResume',
  data: {
    _id: "",
  }
})
```

- Response

```
{
  true
}
```

## 8. getJobByRecruiter

- Request

```
wx.cloud.callFunction({
  name: 'getJobByRecruiter',
  data: {
    openID: "",
  }
})
```

- Response

```
{
  data: [{
    _id: "",
    recruiterID: "",
    isEnterprise: "",
    jobtitle: "",
  }
  ]
}
```



```

        salary: "",
        number: "",
        location: "",
        note: "",
        }, {...}, ..., {...}]
    }

```

## 9. submitApply

- Request

```

wx.cloud.callFunction({
  name: 'submitApply',
  data: {
    msg: {
      jobID: "",
      name: "",
      age: "",
      education: "",
      phone: "",
      email: "",
      resume: "",
    },
  }
})

```

- Response

```

{
  success: true
}

```

## 4. 数据库设计

### 1. job

- \_id: String
- recruiterID: String
- isEnterprise: Boolean
- jobtitle: String
- salary: Number
- number: Number
- location: String
- note: String

### 2. apply

- \_id: String
- jobID: String
- name: String
- age: Number
- education: String
- phone: String
- email: String
- resume: String

### 3. resume

- \_id: String
- jobID: String
- recruiterID: String

## 四、产品实现说明

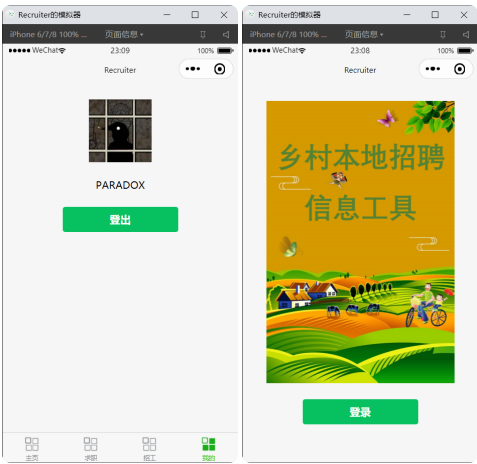
### 1. 界面功能说明

1. 登录



- 点击登录按钮，使用微信账号登录，允许获取用户个人信息，进入主页界面。

2. 登出



- 在我的界面点击登出按钮，登出账号。

3. 主页



- 在主页显示所有的招聘信息，点击“查看”进入工作详情界面。

## 4. 求职



- 在求职界面通过条件搜索招聘信息，点击“查看”进入工作详情界面。

## 5. 投递



- 在工作详情界面可以对当前工作进行投递，并且可以上传个人简历。
- 简历和求职信息暂时只存在数据库中，没有在界面中反映。招聘者对求职者投递的管理功能将在下周实现。

## 6. 招工



- 在招工界面查看本人发布的招聘信息，点击发布招聘信息按钮可以发布招聘信息。

## 2. 演示视频

<https://nankai.feishu.cn/docx/doxcn22FrtoFEcosbO15wIWTGkf>

## 3. 体验二维码

