**志愿者管理软件需求文档**

**1. 引言**

本文档描述了一款名为“VolunteerPro”的志愿者管理软件的需求。该软件旨在帮助组织有效地招募、管理和协调志愿者活动。

**2. 用户需求**

**2.1 注册和登录**

* 用户可以通过电子邮件进行注册和登录。
* 注册时，用户需提供电子邮件地址和密码，并验证电子邮件地址的有效性。
* 登录时，用户输入正确的电子邮件和密码进行验证。
* 将验证码存储到redis中，并且保存三十分钟

**2.2 个人资料管理**

* 用户登录后可以访问个人资料页面，编辑和更新个人信息，包括姓名、联系方式、技能和兴趣等。
* 用户可以上传个人照片，用于在活动中展示身份。

**2.3 活动浏览和报名**

* 用户可以在首页浏览发布的志愿者活动列表，包括活动描述、时间和地点等信息。
* 用户可以根据兴趣和可用时间筛选和搜索活动。
* 用户可以通过点击活动详情页了解更多信息，并选择报名参加活动。

**2.4 活动管理**

* 组织者可以通过管理端发布新的志愿者活动，包括活动名称、描述、时间、地点和所需志愿者数量等信息。
* 组织者可以编辑和取消已发布的活动，并通知已报名志愿者。
* 组织者可以查看已报名志愿者的列表，并导出报名数据以便进一步管理。

**2.5 通知和消息**

* 用户可以收到系统通知，如活动确认、变更或取消等。
* 用户可以收发个人消息，与其他志愿者或组织者进行沟通。

**2.6 统计和报告**

* 系统可以生成志愿者活动的统计报告，包括参与人数、活动类型等。
* 管理端可以查看活动数据和用户报名情况，并根据需求生成定制报告。

**3. 技术要求**

**3.1 后端技术栈**

* **Spring Boot**: 作为后端框架，提供RESTful API服务。
* **Spring Security**: 实现用户认证和授权，结合JWT实现无状态的身份验证。

也可以采用sa-token的方式进行认证和授权。

* **数据库**: 使用MySQL或其他关系型数据库存储用户信息、活动信息等数据。

**3.2 前端技术栈**

* **用户端（UniApp）**: 使用Vue.js作为主要框架，结合UniApp框架进行跨平台开发。
* **管理端(Vue)：**使用Vite作为主要框架elementPlus作为样式库，来对管理端开发
* **组织端（UniApp）：**也采用Uniapp作为主要框架来开发

**3.3 其他技术要求**

* **JWT**: 使用JWT（JSON Web Token）作为身份验证的机制，将用户信息加密到Token中，以实现无状态的身份验证。
* **电子邮件验证**: 实现用户注册时的电子邮件验证机制，确保用户提供的电子邮件地址的有效性。

**4. 其他需求**

* **用户界面设计**: 设计响应式的用户界面，确保在不同设备上都能够良好展现。
* **多语言支持**: 支持多语言界面，方便不同地区的用户使用。
* **性能优化**: 对前后端进行性能优化，提升系统响应速度和用户体验。

**5、数据库设计**

**基础表 x 18**

* **用户信息表：包含用户的角色，以及基础个人信息**
* **活动信息表：包含活动的信息**
* **活动参加状态表：进行多表关联活动&用户**
* **用户聊天：用户关联用户聊天**
* **动态发布信息表：用于个人动态的发布**
* **通知表：用于存储用户的系统、用户通知**
* **图片上传信息表：用于存储用户上传的图片信息**
* **活动类型表：关联活动发布类型**
* **动态评论表：动态关联动态评论表**
* **组织表：志愿者隶属于什么组织**
* **组织关联表：关联用户和组织**
* **活动类型关联表：活动类型关联活动表**
* **建议表：成员向组织，组织向总部管理端提建议**
* **成员审核信息表：成员进入组织需要审核**
* **组织审核信息表：组织需要申请才能被公开**
* **组织信息表：组织的相关信息**
* **志愿者活动参加完的心得信息：心得信息**
* **用户组织信息表：关联用户组织，以及成员在组织的职称**

**6、角色**

|  |  |
| --- | --- |
| 角色 | 权限 |
| User | 志愿者普通成员 |
| Manager | 管理者，也称为总部 |
| GroupManager | 组织管理者 |
| - | 游客 |