

# 技术文档

## 一、项目创新性和原创性

该项目的独特性在于提供了登陆注册、城市信息展示、天气查询、路径规划、帖子发布等功能的集成，能够帮助用户快速了解城市的风景名胜、饮食文化，以及查询实时天气和步行路径。用户中心页面展示了一个直观简洁的用户中心界面，结合了 HarmonyOS 的特性，界面元素布局合理，用户信息展示清晰。

该项目通过天气查询和路径规划功能，解决了用户在陌生城市中遇到的出行问题。用户能够实时获取天气信息，合理安排出行时间，避免天气影响；同时，路径查询功能帮助用户找到最佳的步行路线，提升了出行效率。这个解决方案有效改善了用户在城市生活中的便利性，尤其对游客和新居民来说，提供了很大的帮助。帖子发布功能使得用户可以分享自己的旅行见闻和当地美食，增强了用户之间的互动。

## 二、功能完整性

### 1、完成的主要功能：

登录注册界面：两个文本框（用户名和密码）和两个按钮（登录和注册）以及相关说明信息构成了页面的主要内容，采用 PersistentStorage 存储数据。

城市风景名胜、饮食文化展示、实时天气查询和路径规划功能。

实现发布分享帖、浏览分享帖的功能：

（1）点击 tabbar 分享，进入旅游分享页面；

（2）分享贴页面中展示多个分享贴，点击分享贴可以查看分享贴的详情页面，其中展示了分享贴的具体内容

(3) 在分享贴页面，点击新建贴按钮，输入帖子标题、所分享的旅游地点以及分享贴内容，之后再点击保存，就会返回分享贴页面。此时分享贴页面就会多出一条帖子，点开可以查看刚刚输入的分享贴具体内容

实现了用户中心界面，包括用户信息显示、功能列表动态加载以及账号切换和退出登录功能。

## 2、功能细节

### I、首页界面

(1) Views 组件：负责加载并显示风景名胜的列表。通过 `getViewsDataFromJSON` 从 JSON 文件中获取数据，并渲染每个风景名胜项为 View 组件。点击每个项目后，跳转到文章详情页面。

(2) View 组件：用于展示单个风景名胜的详细信息，包括图片、标题和简介。每个 View 项目有固定的布局，图片和文字信息都进行了样式调整，如文字溢出处理和圆角效果。

(3) FoodCultureView：从 JSON 文件 (`FoodCultureData.json`) 加载并显示饮食文化的文章列表。使用 List 显示每篇文章的标题和简短简介，点击某个项目会跳转到文章详情页面。

(4) FoodCulture：显示单篇文章的标题、简介和图片，采用卡片式布局。使用 Row 布局，左侧为文本部分，右侧为图片部分，设置了合适的间距、边距和对齐方式。

(5) 主要通过 `relationalStore` 提供的数据库管理工具和 SQL 语句来实现与数据库的交互。通过异步的 CRUD 操作，结合条件构造和回调机制，完成天气数据表的增、删、改、查操作。`RdbStore` 和 `RdbPredicates` 是实现这些操作的关键，`ValuesBucket` 用于构建插入和更新的数据，而 `ResultSet` 用于访问查询结果。

(6) 使用 HTTP 请求调用高德地图的步行路径规划 API，获取起点到终点的步行路线。通过 Promise 异步处理请求结果，解析返回的 JSON 数据，并根据结

果是否有效决定是否返回路径规划信息或抛出错误。

## II、分享界面

(1) 用自定义接口列表 `myPost[]` 来存储帖子的信息，通过 `list` 和 `foreach` 遍历展示所有帖子。

(2) 帖子存储在 `RdbTest.db` 数据库中的 `postsTable` 中。

(3) 用 `scroll` 展示可滚动文本（帖子详情部分），使用 `.gesture` 绑定手势，使用 `.transition` 添加转场效果。

## 三、代码质量

1、比较需要解释的部分都有简洁明了的注释。

2、代码结构清晰，逻辑分层合理。合理分到几个文件，建立自定义组件，逻辑清晰明了。

3、变量和函数命名简洁且能体现对应的语义，变量和函数命名采用下划线命名法。

4、在建立数据表，向数据表插入数据时使用 `try, catch` 捕捉异常，便于调试。

5、使用 `try-catch` 块进行异常处理。如在：查询路径规划功能中，异常处理通过 `try...catch` 语句实现。在 `try` 块中执行关键操作，如获取经纬度和路径规划，若发生错误，则跳转到 `catch` 块，显示错误信息。每个关键操作失败时，都会通过 `AlertDialog.show` 提示用户具体的错误或异常情况。

## 四、用户交互和界面设计

1、页面有明确的文字和图片描述，用户能清晰理解整个过程。

(2) 界面设计简洁且具有现代感，符合美学标准。

(3) 天气查询以及路线规划旁边有动态标识。

(4) 输入的文本框中有提示信息：“请输入 xx”。

(5) 使用按钮，超文本，弹窗等方式与用户进行交互。交互设计合理，从点击新建按钮、填入信息到访问帖子，整个过程清晰流畅。操作流程直观，用户可以轻松实现账号登录等功能。页面的滚动效果也为用户提供了流畅的交互体验。

(6) 界面布局采用了典型的用户中心设计，功能清晰、操作简便。按钮、图标和文本间的配色合理，确保了界面的易用性和可读性。

## 五、团队协作

### 1、会议记录

#### 12.12 第一次小组讨论

##### GitLab 群组及代码仓库创建

首先，四人讨论并创建了 GitLab 群组及代码仓库。每个人展示了自己的实现思路和代码。经过讨论，大家决定选择天气查询作为应用的基础功能，并决定在此基础上构建一个旅行咨询应用。

##### 应用初步构思

根据讨论的结果，团队决定开发一款旅行咨询 App，初步设计如下：

登陆注册功能实现：用户进入应用程序后，首先进行登陆。如果是新用户，则可以点击注册按钮进入注册页面，注册成功后返回登陆界面继续登陆。

应用主界面分为三个部分：

天气查询：提供用户当前所在位置或指定城市的天气情况。

旅行咨询推荐：用户可以进行发帖分享。

个人页面：显示用户的基本信息等个人内容。

当前计划：12.22 前搭建初步完整应用，并由四人分别负责不同页面的设计，最终整合成一个完整的应用。

分工：

孙：负责登陆和注册功能的实现。

罗：修改原有天气查询页面，并进行旅行咨询推荐页面的设计。

何：设计帖子创建页面。

叶：负责完成个人界面的实现。

## 12.22 第二次小组讨论

### 进展展示与代码更新

在第二次讨论中，四人轮流展示了各自的进展，并进行了代码更新。每个人分享了自己的实现细节以及遇到的问题，互相提供了改进建议。

孙：展示了登陆和注册功能的实现，代码已基本完成。

罗：对原有的天气查询页面进行了修改，增加了路径规划功能。

何：展示了帖子创建页面的设计，界面简洁明了。

叶：展示了个人页面的实现，初步完成了用户信息的展示功能。

### 下步计划

各自根据任务分配进行完善，进行 UI 设计的提升优化。

## 1.14 第三次小组讨论：线上腾讯会议

### 作业评分标准与分工细化

在第三次讨论中，团队成员依据作业评分标准进一步细化了各自的分工，并对现有功能进行了改进。

增加了更好的 UI 交互，提升了应用的用户体验。

进行了一些功能上的调整，以更好地匹配评分标准，确保每个部分都符合要求。

## 1.17 第四次小组讨论：线上腾讯会议

汇报人员与文档分配

在最后一次讨论中，团队确定了汇报任务，并进行了文档的分配。每个人将负责撰写部分内容，最终进行汇总和整理。

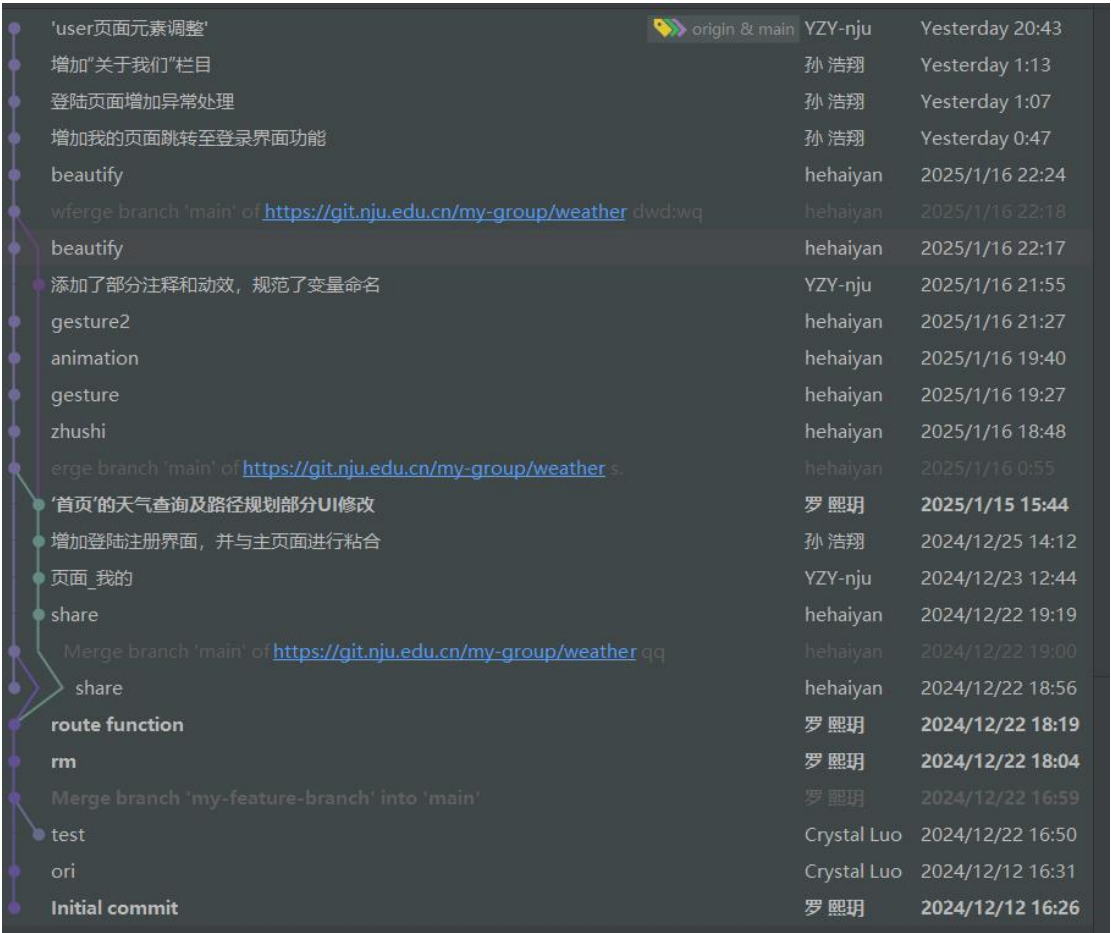
完成度检查

确定每个模块的最终功能并进行测试，确保功能的完整性和稳定性。

最终目标

统一文档格式，确保每个部分内容均衡，汇报时重点突出每个人负责的部分。

2、Git 记录



六、其他优化

- 1、界面布局、色彩搭配符合现代审美标准，色调和谐又不单一。
- 2、在跳转到帖子详情页面时有动画跳转效果。
- 3、在帖子详情页面和帖子页面都可以通过滑动的手势阅读。
- 4、离线缓存与数据同步

登录注册页面因用户名-密码对数据量不大，所以使用 PersistentStorage 保存数据，可以实现离线缓存的要求。