# 草悟暮语项目实现思路

## 项目概述

草悟暮语是一款智能应用，旨在帮助用户更好地了解植物。通过聊天机器人，用户可以通过文字或图片快速识别植物类型，了解其生长状况与养护方法。博客涵盖种植技巧等相关文章，并提供实时问答功能。图片识别功能能够快速获取植物的百科信息和花语。用户还可以聆听他人分享的植物培养故事。通过这些功能，草悟暮语帮助用户更深入地了解每种植物的生活方式，提升植物养护技能，增进对植物的热爱。草悟暮语可以处理90多种植物的问答。

## 技术架构

### 1. 前端

使用 Android（Java）作为应用的前端开发语言，通过 XML 文件定义界面布局，并在 Activity 以及 Fragment 中进行逻辑处理和 UI 更新。

### 后端

#### 普通问答

调用 OpenAI 的 API 实现与后端服务的通信，获取聊天机器人的回复。

#### 涉及植物的问答

##### 后端接口说明

###### POST /predict

此接口用于向腾讯云服务器发送 POST 请求，根据用户提供的问题和植物种类，使用部署在服务器上的模型进行文本分类和知识库查询，最终返回查询到的相关信息。

**请求参数：**

* question: 用户提问的问题，例如植物的生长习性、土壤要求、水分需求等。
* species: 用户提供的植物种类。

**请求示例：**

{"question": "植物的生长习性是什么？","species": "玫瑰"}

**响应示例：**

{"growth\_habits": "玫瑰的生长习性是...","soil\_requirements": "玫瑰的土壤要求是...","water\_needs": "玫瑰的水分需求是..." }

### 3. 通信

使用 OkHttpClient 库进行网络请求，包括向 OpenAI 的 API 发送请求以及处理植物识别服务的响应。

### 4. 数据库

通过 Room 库实现本地数据库的创建和管理，用于保存聊天记录等数据。SQLite 用于管理登录注册等后端功能。

### 5. 第三方服务

### PlantNet API 服务

草悟暮语应用使用 PlantNet API 进行植物识别。该 API 支持通过上传植物图片来识别植物种类，并提供详细的植物信息。

### API 详细描述

#### 请求参数

* project (string, path): 在哪个参照系中搜索植物（使用可用的项目之一或 "all" 获取最相关项目的结果）。
* 默认值：all
* include-related-images (boolean, query): 如果为 true，对于每个可能的物种，将返回最相似的图像。
* 默认值：false
* no-reject (boolean, query): 禁用 "no result" 在拒绝类匹配的情况下。
* 默认值：false
* lang (string, query): 国际化 (默认: en)。可用值：en, fr, es, pt, de, it, ar, cs, nl, sk, zh, ru, tr, pl, uk, he, el, fi, id, ms, ca, ja, hu, hr, da, ro, bg, nb, sl, sv, et, eu, ur, ml, cy, ku, gl, eo, sat, zh-tw, pt-br, hi, de-at, sr, zh-hant, bn, fa。
* 默认值：en
* type (string, query): 模型类型：使用 "kt" 用于新的 POWO / WGSRPD 基于植物区系和识别引擎 (2023+)，"legacy" 用于传统植物区系和识别引擎 (2022)。
* 可用值：kt, legacy
* api-key (string, query): 您的私人 API 密钥。
* authenix-access-token (string, query): Authenix 访问令牌 − Authenix 是一种替代身份验证方式。除非您了解此内容，否则使用 api-key。
* images (array[file], formData): 图像（所有图像必须代表同一植物）[最多 5 张图片]。
* organs (array[string], formData): 与图像相关的器官（必须包含一个：leaf, flower, fruit, bark, auto。可以包含：habit, other）[最多 5 个器官且 organs.length 必须等于 images.length]。
* 可用值：leaf, flower, fruit, bark, auto

#### 响应格式

* **Content-Type:** application/json

**成功响应示例 (200)**

{

"query": {

"project": "string",

"images": [

"string"

],

"organs": [

"string"

],

"includeRelatedImages": true,

"noReject": true

},

"language": "string",

"preferedReferential": "string",

"switchToProject": "string",

"bestMatch": "string",

"results": [

{

"score": 0,

"species": {

"scientificNameWithoutAuthor": "string",

"scientificNameAuthorship": "string",

"scientificName": "string",

"genus": {

"scientificNameWithoutAuthor": "string",

"scientificNameAuthorship": "string",

"scientificName": "string"

},

"family": {

"scientificNameWithoutAuthor": "string",

"scientificNameAuthorship": "string",

"scientificName": "string"

},

"commonNames": [

"string"

]

},

"images": [

{

"organ": "string",

"author": "string",

"license": "string",

"date": {

"timestamp": 0,

"string": "string"

},

"citation": "string",

"url": {

"o": "string",

"m": "string",

"s": "string"

}

}

],

"gbif": {

"id": 0

},

"powo": {

"id": "string"

},

"iucn": {

"id": "string",

"category": "string"

}

}

],

"remainingIdentificationRequests": 0,

"version": "string"

}

### 响应字段解析

* query: 包含了用户请求的详细信息。
  + - project: 搜索植物的参照系，例子中为 "all"。
    - images: 用户上传的图像，示例中为一个 base64 编码的图像字符串。
    - organs: 图像对应的植物器官，例子中为 "leaf"（叶子）。
    - includeRelatedImages: 是否返回相关图像，例子中为 true。
    - noReject: 是否禁用 "no result" 类匹配，例子中为 true。
    - language: 用户请求的语言，例子中为 "zh"（中文）。
    - preferedReferential: 优选的参照系，例子中为 "plants"。
    - switchToProject: 切换到的项目，例子中为 "none"。
    - bestMatch: 最佳匹配的植物种类，例子中为 "Rosa chinensis"（月季）。
    - results: 识别结果数组，包含每个可能匹配的详细信息。
    - score: 匹配度分数，范围从 0 到 1，例子中为 0.95。
    - species: 识别出的植物种类的详细信息。
      * scientificNameWithoutAuthor: 植物的学名（不包含作者），例子中为 "Rosa chinensis"。
      * scientificNameAuthorship: 学名的作者，例子中为 "Jacq."。
      * scientificName: 完整的学名，例子中为 "Rosa chinensis Jacq."。
      * genus: 属的详细信息。scientificNameWithoutAuthor: 属的学名，例子中为 "Rosa"。
        + scientificNameAuthorship: 属名的作者，例子中为 "L."。
        + scientificName: 完整的属名，例子中为 "Rosa L."。
      * family: 科的详细信息。scientificNameWithoutAuthor: 科的学名，例子中为 "Rosaceae"。
        + scientificNameAuthorship: 科名的作者，例子中为 "Juss."。
        + scientificName: 完整的科名，例子中为 "Rosaceae Juss."。
      * commonNames: 常见名称数组，例子中为 ["China Rose", "Chinese Rose"]。
    - images: 图像数组，包含与识别结果相关的图像信息。
      * organ: 图像对应的植物器官，例子中为 "leaf"。
      * author: 图像作者，例子中为 "John Doe"。
      * license: 图像使用许可，例子中为 "CC BY-SA 4.0"。
      * date: 图像日期。timestamp: Unix 时间戳，例子中为 1594857600。
        + string: 日期字符串，例子中为 "2020-07-16"。
      * citation: 引用信息，例子中为 "Doe, J. (2020). China Rose. CC BY-SA 4.0."。
      * url: 图像链接，包含三个不同大小的链接：o: 原始图像链接，例子中为 "https://example.com/image\_o.jpg"。
        + m: 中等大小图像链接，例子中为 "https://example.com/image\_m.jpg"。
        + s: 小尺寸图像链接，例子中为 "https://example.com/image\_s.jpg"。
    - gbif: GBIF（全球生物多样性信息机构）相关信息。
      * id: GBIF ID，例子中为 2891234。
    - powo: POWO（植物世界在线）相关信息。
      * id: POWO ID，例子中为 "urn:lsid.org:names:123456-1"。
    - iucn: IUCN（国际自然保护联盟）相关信息。
      * id: IUCN ID，例子中为 "12345"。
      * category: IUCN 保护级别，例子中为 "LC"（Least Concern, 无危）。
    - remainingIdentificationRequests: 剩余的识别请求次数，例子中为 99 次。
    - version: API 版本，例子中为 "1.0"。

## 功能实现

### 3.1 聊天机器人

● 识别与养护指南功能用户输入问题

● 模型搭建：训练针对植物信息和种类的问答模型，并部署到腾讯云轻量服务器

**模型搭建过程和实现细节**：该模型旨在为用户提供植物养护信息。通过用户输入的植物种类和问题，系统调用后端接口，返回相应的养护建议和措施。模型通过加载两个数据集（（见附件）Updated\_Info\_Dataset.csv**[知识库,** question为输入的问题匹配到数据集里最接近的问题和种类，然后来按照植物类型和匹配的类型输出information和measure里的内容**]** 和 （见附件）complex\_plant\_x\_questions.csv **[文本分类训练集]** ），解析并清理数据，将问题、信息和措施合并为一个综合文本特征，使用 TfidfVectorizer 进行向量化，并通过 MultiLabelBinarizer 处理标签。模型采用 OneVsRestClassifier 结合 LogisticRegression 进行多标签分类训练，可以根据输入问题预测其对应的多个养护分类。

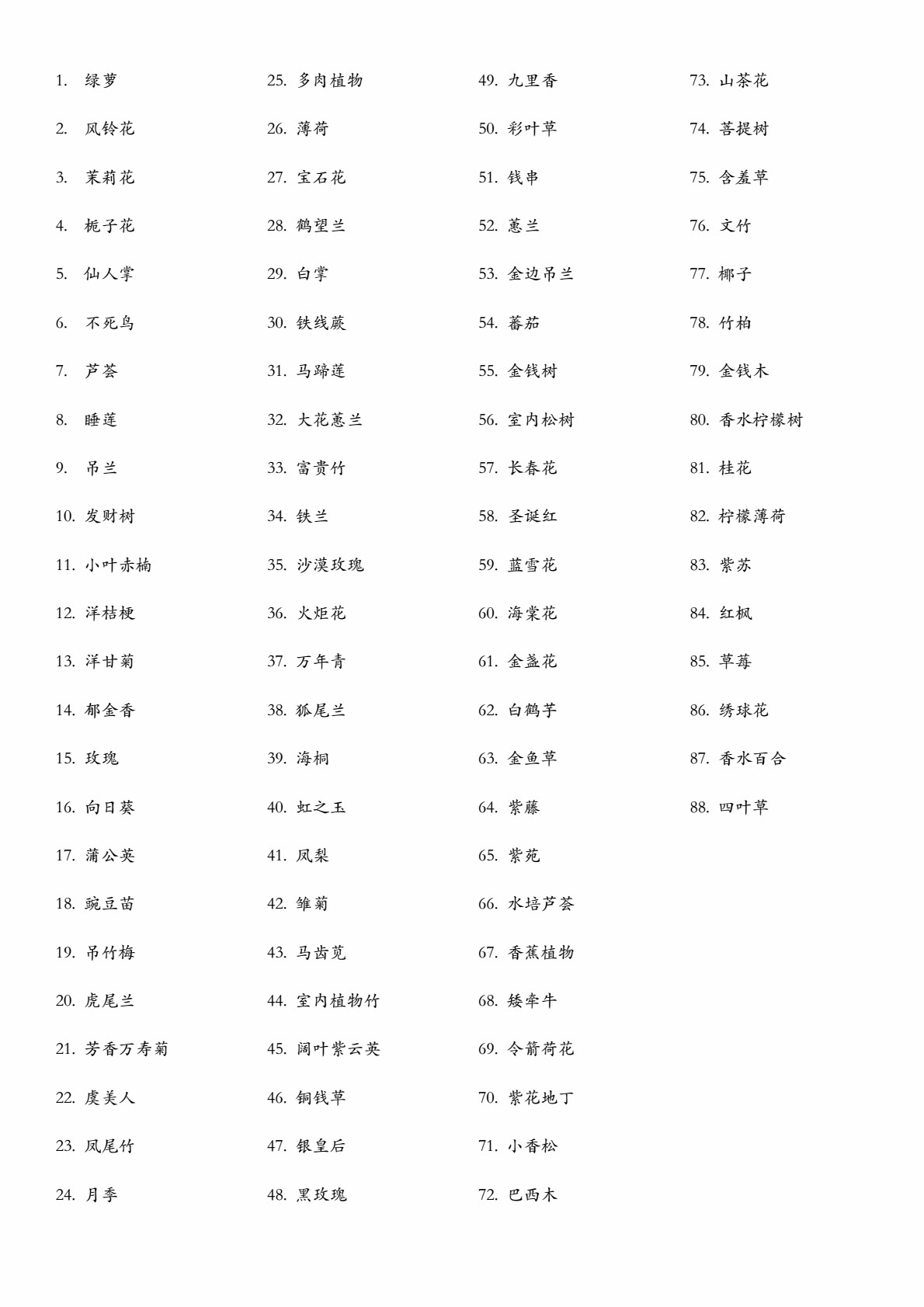
**应用中包含几个核心功能**：

1. 多标签分类：输入一个问题，模型预测出涉及的多个养护分类；
2. 最相似问题匹配：找到数据库中最相似的问题，并返回其对应的分类和物种；

3) 最相似物种匹配：找到数据库中最相似的物种名称；

1. 获取养护信息和措施：从数据库中提取相应的养护信息和措施。前端页面通过输入框让用户输入植物种类和问题，点击提交后调用后端接口 /predict，返回最相似的问题、分类、物种及对应的养护信息和措施。

**植物种类：**可以进行问答的植物



**部署：**Flask API 提供一个 RESTful 服务，用户可以通过 POST 请求提交问题和物种名称，系统返回相关信息。如果没有匹配结果，则返回错误信息。配置 Flask 应用并启动，同时考虑使用 Nginx 等工具作为反向代理服务器以提高安全性和性能。

**维护：**定期监控服务器资源使用情况，记录应用日志并进行数据备份，以防止数据丢失。定期更新系统和库，优化代码和数据库查询，并加强安全措施，以确保应用的稳定性、性能和安全性。

### 3.2 博客功能

博客功能主要包括展示种植技巧等相关文章，并提供实时问答功能。

1. **展示种植技巧等相关文章**：在应用中创建一个博客模块，用户可以浏览到各种植物的种植技巧、养护经验等相关文章。这些文章应该包含详细的步骤、注意事项和实用技巧，帮助用户更好地了解植物养护。
2. **提供实时问答功能**：在博客模块中添加实时问答功能，用户可以提出问题并得到即时回答。这可以是一个基于文本的交互界面，用户可以在其中输入问题并等待系统给出答案。答案可以是基于用户提供的问题和博客文章内容生成的。
3. **博客功能的实现细节**：
   * + 1. 创建一个博客文章数据库，存储各种植物的种植技巧和养护经验等相关文章。
       2. 实现一个界面，用于展示这些文章，并提供用户评论和反馈的功能。
       3. 为实时问答功能创建一个聊天机器人，它可以根据用户提供的问题和博客文章内容生成回答。

### 3.3 图片识别功能

**上传图片**：用户可以通过界面上传植物图片，用于识别植物种类和获取相关信息。

**调用 PlantNet API 获取植物种类**：上传的图片将被发送到 PlantNet API 进行植物识别，获取植物的种类信息。

**展示植物百科信息和花语**：根据识别结果，从数据库或网络获取该植物的百科信息和花语，并展示给用户。百科信息包括植物的分类、生长习性、养护要点等，花语则是该植物所代表的含义或象征。

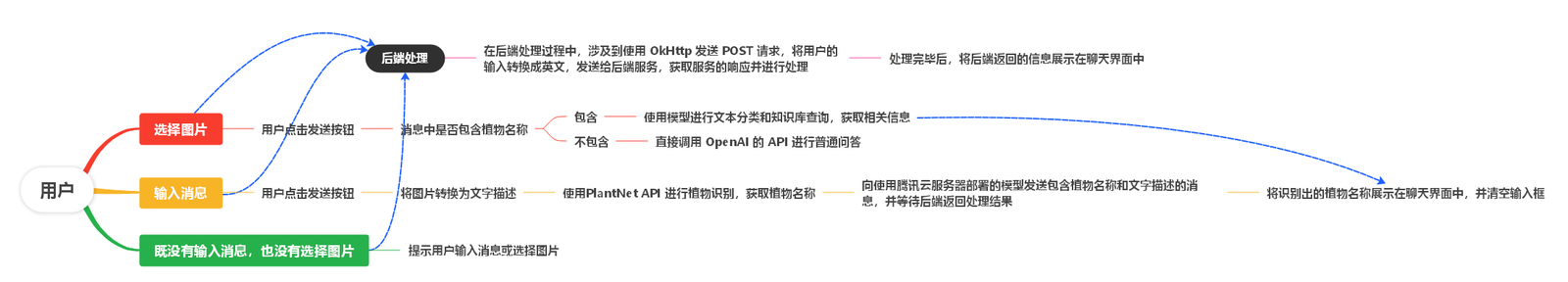
**图片识别功能的实现细节**：

* 创建一个界面，允许用户上传图片，并在上传后显示识别结果。
* 使用 Retrofit 或类似的库调用 PlantNet API，发送图片并获取识别结果。
* 根据识别结果，从数据库获取植物的百科信息和花语，并展示给用户。

## 5. 数据流程

* 用户输入消息或选择图片，并点击发送按钮。
* 如果用户既没有输入消息也没有选择图片，则提示用户输入消息或选择图片。
* 如果用户输入了文字消息，则进行以下处理：
  + - 检查消息中是否包含植物名称，如果包含，则使用模型进行文本分类和知识库查询，获取相关信息。
    - 如果没有植物名称，则直接调用 OpenAI 的 API 进行普通问答。
* 如果用户选择了图片，则进行以下处理：
  + - 将图片转换为文字描述。
    - 使用PlantNet API 进行植物识别，获取植物名称。
    - 向使用腾讯云服务器部署的模型发送包含植物名称和文字描述的消息，并等待后端返回处理结果。
    - 将识别出的植物名称展示在聊天界面中，并清空输入框。
* 在后端处理过程中，涉及到使用 OkHttp 发送 POST 请求，将用户的输入转换成英文，发送给后端服务，获取服务的响应并进行处理。

**流程图：**

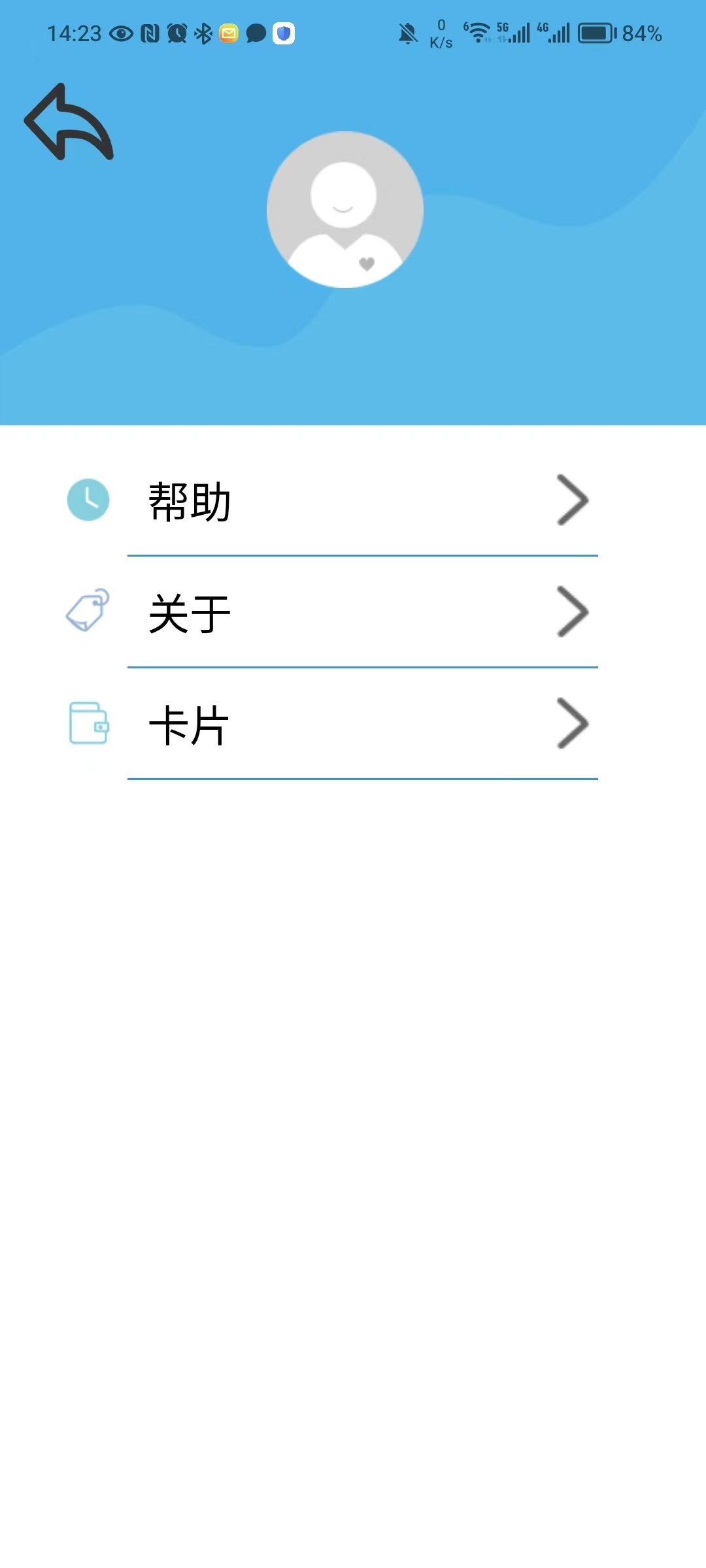
****

## 用户界面设计

### 主界面

* **背景虚化**：背景虚化处理，使logo和按钮文字更加突出，增强界面的视觉层次感。
* **紫色方框**：使用紫色的方框强调“开始聊天”界面的主要功能，居中放大，吸引用户注意力。
* **扫描界面入口**：将扫描界面入口放置在主界面的右上角，方便用户登录后快速了解植物信息和花语故事。
* **功能按钮**：在“开始聊天”功能下方分别放置“聊天记录”按钮、“博客”按钮和“我的”按钮，方便用户快速导航。
* **设计标语**：在界面底部添加设计标语，提升页面的整体观感。

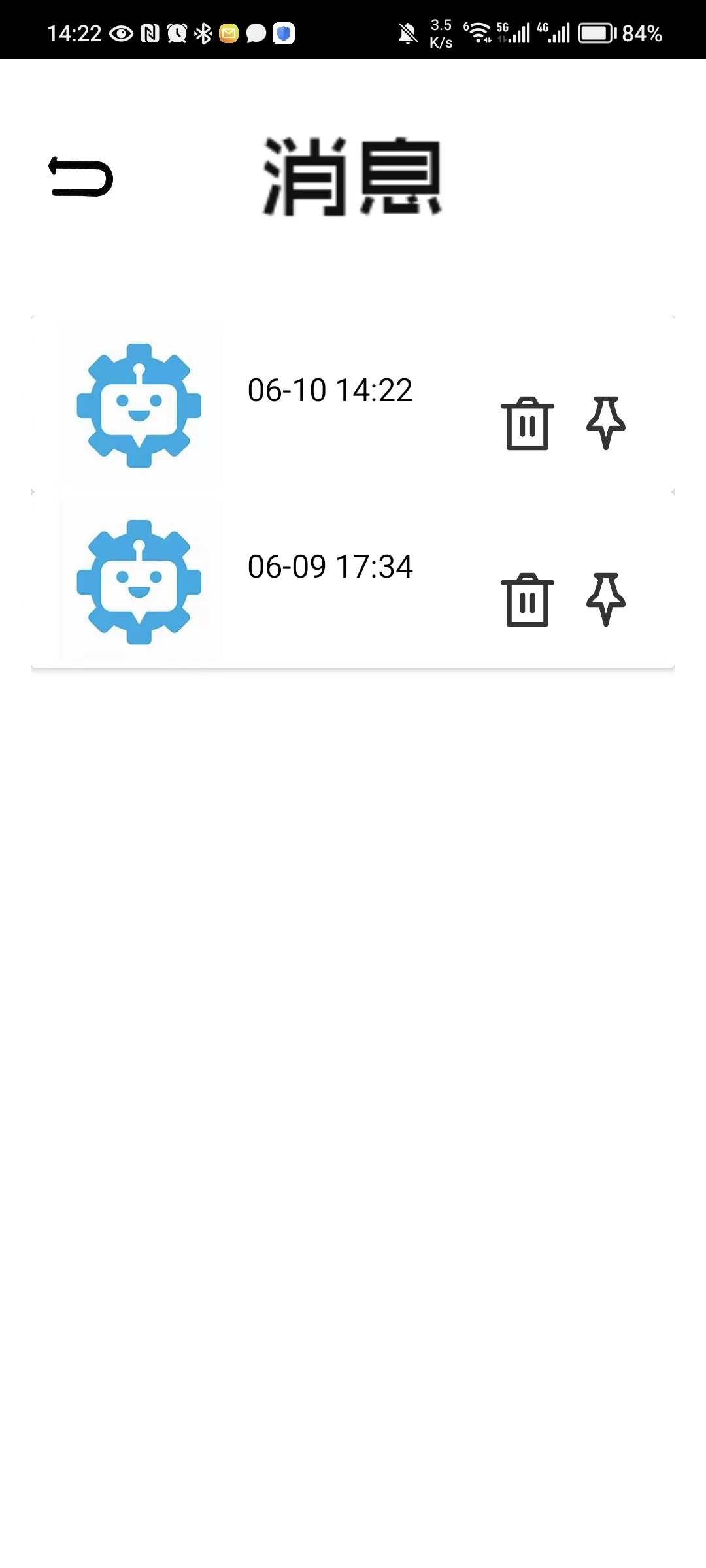
### 主界面 我的界面



### 2. 聊天界面

* **布局分区**：将聊天界面分为上中下三个部分，最上方放置返回键和设置按键，中间是聊天窗口，用来显示聊天内容，最下方是消息输入窗口，既可以输入文字又可以发送照片。
* **边框颜色**：窗口边框使用黄蓝色，清新活泼，与界面主体的颜色相呼应。上方窗口和其边框的颜色与下方窗口和其边框的颜色相互交换，使页面布局更加和谐和整体化。
* **Loading动图**：在最上方设置加载动画，方便用户跟踪系统运行状态。

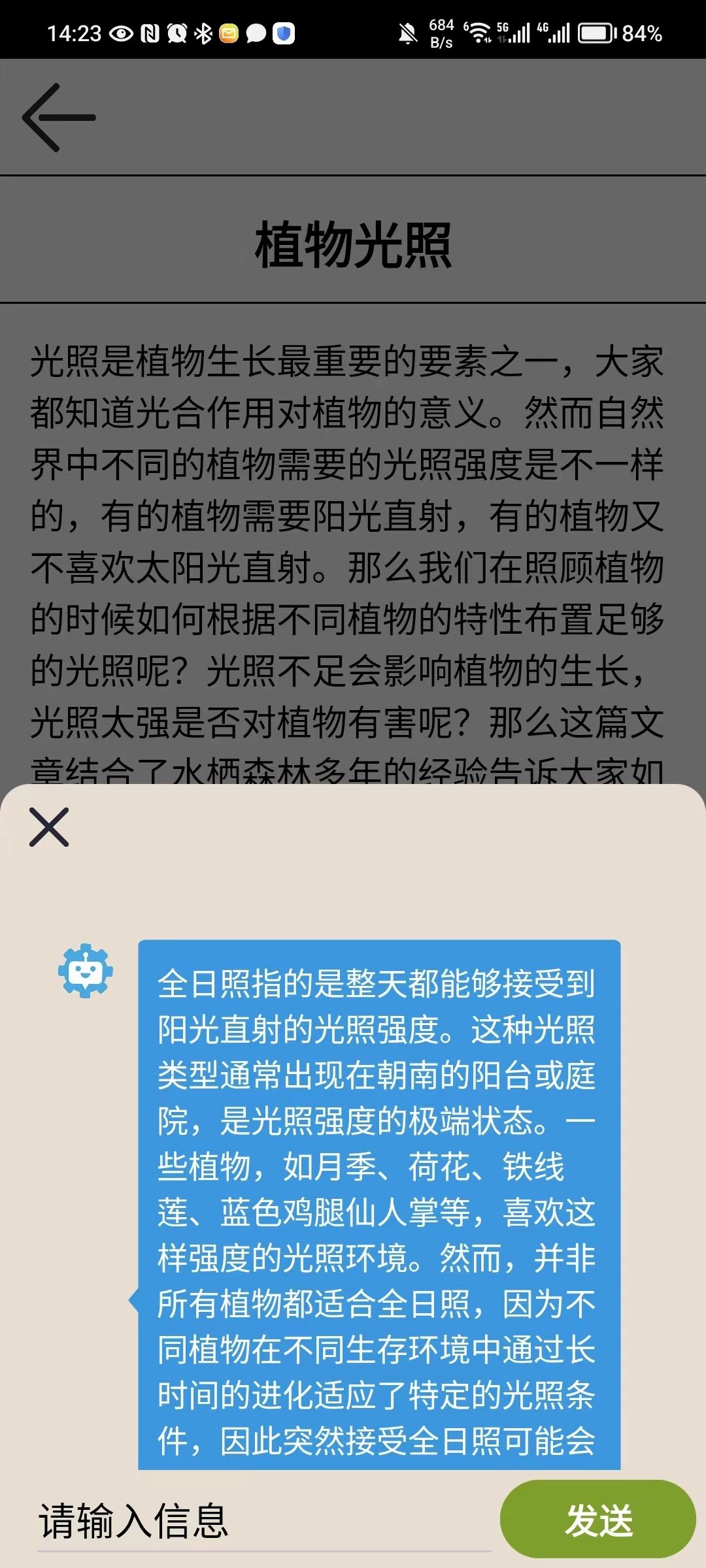
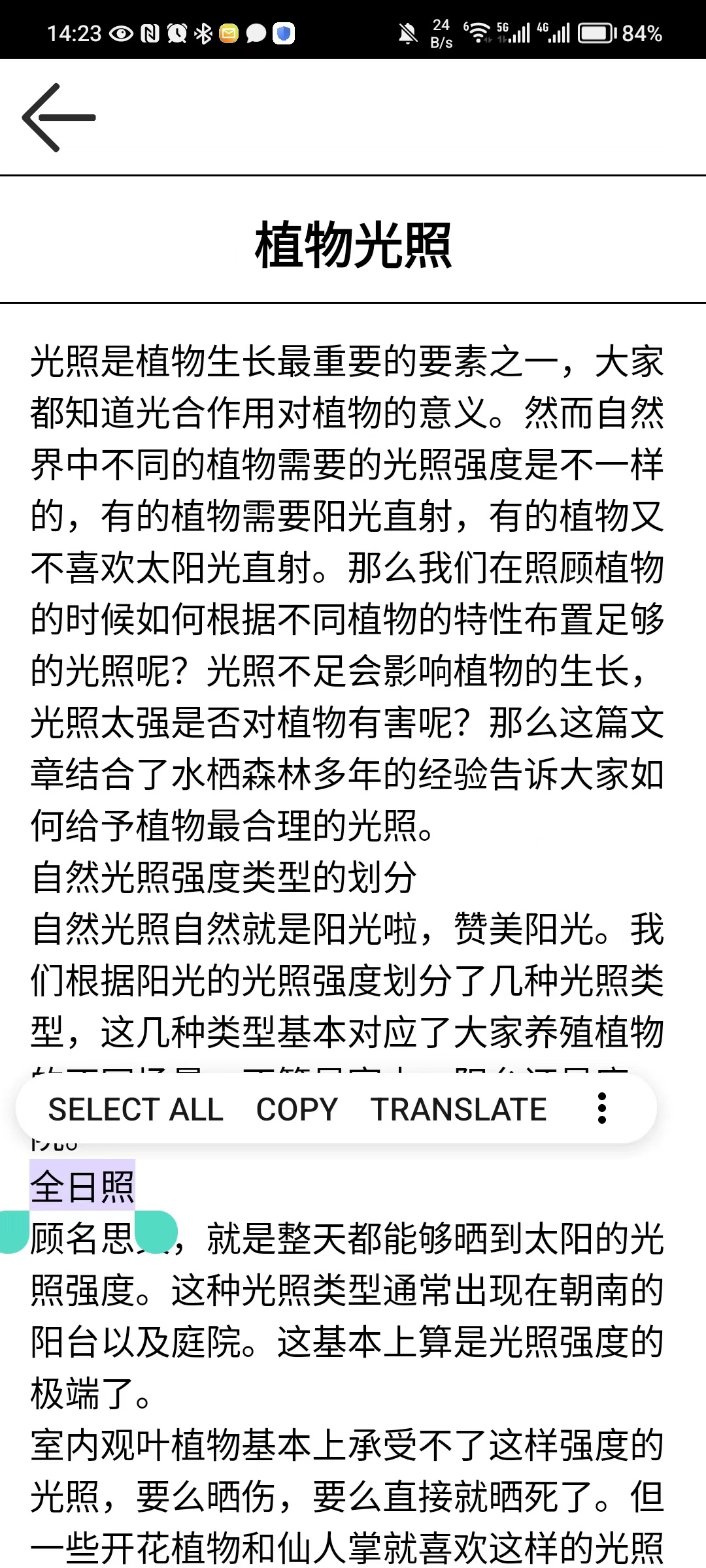
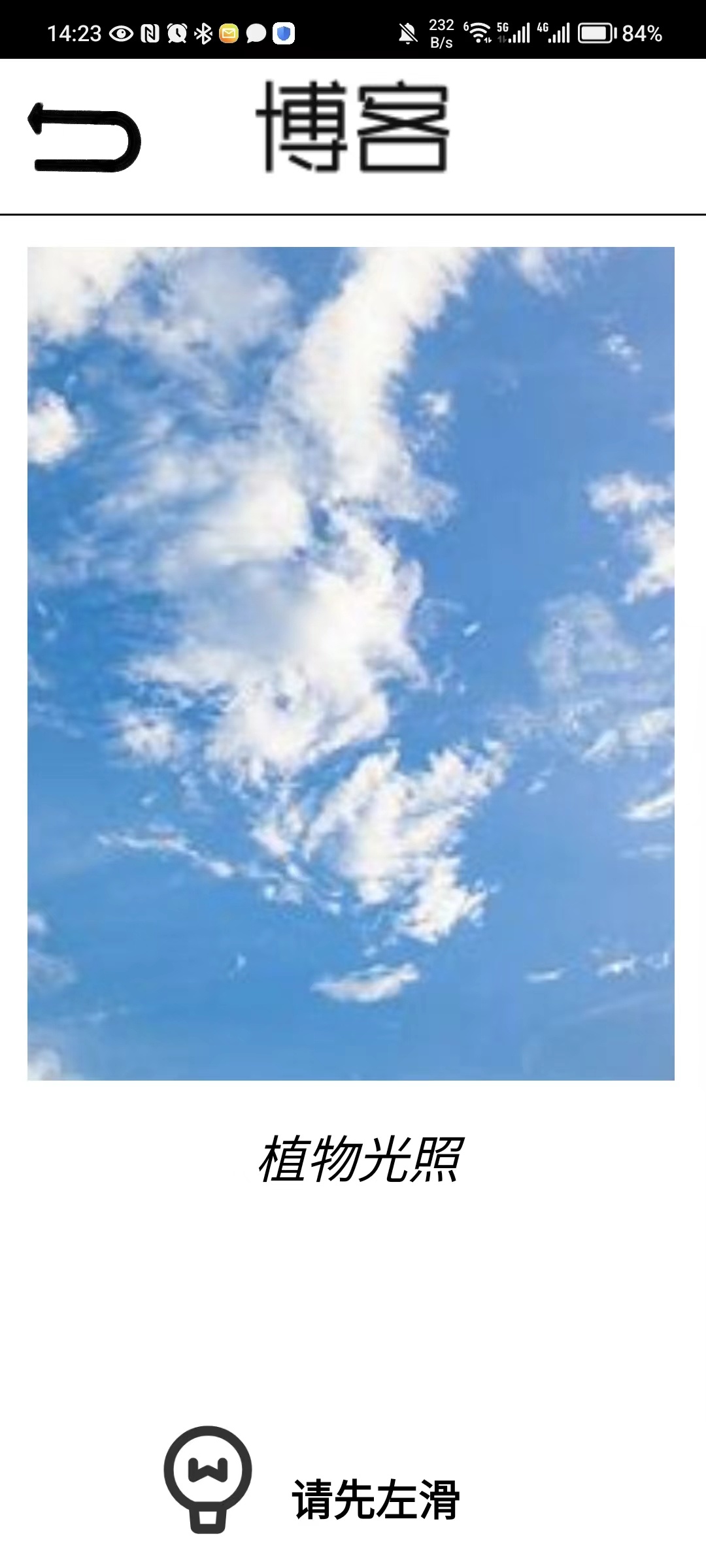
### 聊天机器人界面 聊天记录界面



### 3. 博客界面

* **目录设计**：博客界面由目录和目录内部的文字资料组成，单个目录界面由配图和文字说明组成。用户在目录界面左右滑动挑选种植模块后即可进入阅读界面。
* **文字互动**：用户可以在阅读界面的文字中选中需要深化理解的语句，并在出现的聊天框中求助聊天机器人翻译解释或直接输入问题进行查询。

### 截图：博客目录界面 博客文章界面 博客文章查询界面



### 4. 扫描界面

* **拍照界面**：用户可以选择照片或拍摄实时图像。
* **百科界面**：拍照后进入百科界面，显示植物的基本信息，如是否有毒、是否为花卉、是否为入侵植物、植物类型（室内或室外）、生长周期、种植时间和作用。此页面底部有返回键、页面翻转键和分享种植故事播放键。
* **卡片界面**：按动页面翻转键进入卡片界面，卡片上显示对应植物的花语。百科界面与卡片界面的前后配合作用，提供知识内容的双重体验。添加花语故事语音，增强沉浸式的种植体验。

**百科界面 卡片界面 扫描界面**

****

## 未来展望

**用户交互和知识论坛**

添加用户之间的交互功能和对博客资料的知识论坛，让用户可以分享种植经验和交流心得，增强用户参与感和粘性。

**详细手绘插图**

为每种植物添加详细手绘插图，增强界面的灵动性和视觉吸引力，提升用户对应用的使用兴趣。

**博客内容优化**

增强博客文字资料的容量、来源丰富度和内容质量，确保用户获取到的是有价值的种植知识和经验分享。

**每日打卡和消息通知**

添加每日打卡和任务功能，让用户记录植物养护情况并获得奖励，同时增加消息通知功能，提醒用户关注植物养护需求，增强用户使用的主动性和参与度。

## 8. 参考资料

PlantNet API. (n.d.). Retrieved from https://my-api.plantnet.org/?tags=my-api#/my-api/postV2IdentifyProject

Tencent Cloud. (n.d.). Plant Recognition. Retrieved from https://cloud.tencent.com/developer/article/1874691

Towards Data Science. (n.d.). How to Build Your Own Chatbot Using Deep Learning. Retrieved from https://towardsdatascience.com/how-to-build-your-own-chatbot-using-deep-learning-bb41f970e281