

生物医药数字信息系统

一、项目介绍：

生物医药数字信息系包括中药材生长数据实时采集、存储，图谱比对，数据对比分析，溯源管理，线上教学、数据自动备份等基本功能，并能实现上传、下载数据及文件，线上视频教学等拓展功能；系统还能与手机APP、电脑终端共享；而且用户还能实现层级管理。

主要功能要求如下：

- 1.建立重庆市中药材分布网络地图，以方便科研人员、学生、科研机构在线查询品种分布地点。
- 2.具备与校内系统（采用SOAP协议）实现数据交换，以便校内师生获取中药材生长的相关数据。同时，应具备手机端的采集功能，可通过手机APP获取生长数据。
- 3.具备中药材试验课程存储，以方便教师、学生了解试验内容。
- 4.具备中药材试验课题研究存储，以方便教师、学生了解课题内容，参与课题研究及相关成果转化。
- 5.具备中药材培训素材的制作及存储，以方便教师、培训人员了解培训内容，对培训过程进行跟踪记录。
- 6.具备中药材的评价体系，对评价过程进行和结果进行跟踪记录，并且记录相关的数据为相关申报提供素材（如非遗申请等）。
- 7.实现工作业绩录入、审核、编辑、分级分类等功能，按照学校现有业绩分类认定办法进行分类分级统计与认定，并实现相关分级分类标准动态更新。

项目技术：

① Note

JavaEE：SpringMVC、Spring/SpringBoot、Mybatis、MyBatisPlus、前端框架；Linux。

开发语言：Java、TypeScript、Dart、Vue

框架及中间件：SpringBoot、MyBatisPlus、Electron、Flutter、Vue3、Vuetify3、OSS

配置管理系统：Git、

集成开发环境：Idea、Vscode、Navicat

工具：Visio、Figma、PowerPoint

数据库：MySQL

部署环境：Linux

项目初步计划：

1、项目计划仓库

在Github上建立两个仓库，一个前端，一个后端；

2、设计先行

力争在7月3日前完成项目的大部分内容，好在可以复用很多；
在此阶段小组成员学习新技术同时做已经熟练的项目；

3、小组分工：

小组成员	任务分工和详细计划说明
王银波（组长）（全栈）	规划整个项目，然后全方面设计辅助（不再单干而转化辅助）
张家豪（后端）（文档）	主力后端然后书写整理项目文档和计划
林文浩（前端）（安卓）	主力书写Vue同时学习Flutter
尚思宇（后端）（部署）	主力书写后端同时部署上线项目
陆岳（前端）（设计）	主力书写前端同时做设计（从头学习）

4、概要设计：

前后端解耦合：

 Note

模块一：

数据中心与可视化平台：

1、 中药材分布网络地图 (功能点1): 在地图上展示药材的地理分布。

设计数据库，核心：
接入高德地图API,实现药物的分布地图；
自定义设计界面设计比较好看的图标分布；

Web 端用 Vue+Vuetify+百度/高德地图渲染。Marker 分类图标和弹窗用 Material Design 风格，图片卡片支持轮播。保持UI资源和分类逻辑一致，实现跨端统一体验。

主要表结构举例

1. 药材主表 (herb)

字段名	类型	说明	约束
id	BIGINT	主键ID	主键, 自增
name	VARCHAR(64)	药材名称	唯一, 必填
scientific_name	VARCHAR(128)	学名	可空
description	TEXT	简介/药用价值描述	可空
created_at	TIMESTAMP	创建时间	默认当前时间
updated_at	TIMESTAMP	更新时间	自动更新

2. 药材地理分布表 (herb_location)

字段名	类型	说明	约束
id	BIGINT	主键ID	主键, 自增
herb_id	BIGINT	药材ID (外键)	关联herb(id)
longitude	DECIMAL(10,7)	经度	必填
latitude	DECIMAL(10,7)	纬度	必填
province	VARCHAR(32)	省份	可空
city	VARCHAR(32)	城市	可空
address	VARCHAR(255)	详细地址	可空
observation_year	INT	观测/采集年份	必填
created_at	TIMESTAMP	记录创建时间	默认当前时间

3. 药材图片表 (herb_image)

字段名	类型	说明	约束
id	BIGINT	主键ID	主键, 自增
herb_id	BIGINT	药材ID (外键)	关联herb(id)
location_id	BIGINT	观测点ID (可选外键)	关联herb_location(id) 可空
url	VARCHAR(255)	图片地址URL	必填
is_primary	TINYINT(1)	是否为主图 (0-否, 1-是)	默认0
description	VARCHAR(255)	图片描述	可空
uploaded_at	TIMESTAMP	上传时间	默认当前时间

4. 生长/统计数据表 (herb_growth_data)

字段名	类型	说明	约束
id	BIGINT	主键ID	主键, 自增
location_id	BIGINT	观测点ID (外键)	关联herb_location(id)
metric_name	VARCHAR(50)	指标名称	必填
metric_value	VARCHAR(100)	指标值	必填
metric_unit	VARCHAR(20)	指标单位	可空
recorded_at	TIMESTAMP	记录时间	默认当前时间

5. 数据历史记录表 (herb_growth_data_history)

字段名	类型	说明	约束
id	BIGINT	历史记录主键ID	主键, 自增
origin_id	BIGINT	原数据表主键ID	必填
location_id	BIGINT	观测点ID	必填
metric_name	VARCHAR(50)	指标名称	必填
old_value	VARCHAR(100)	变更前的值	可空
new_value	VARCHAR(100)	变更后的值	可空
action	VARCHAR(16)	操作类型	必填
changed_by	VARCHAR(64)	操作人	默认'system'
changed_at	TIMESTAMP	变更时间	默认当前时间
remark	VARCHAR(255)	变更备注	可空

示例插入数据表格 (部分)

herb_id	longitude	latitude	province	city	address	observation_year
1	126.63333	45.75000	黑龙江省	哈尔滨市	长白山脉周边	2022
2	106.27818	38.46814	宁夏回族自治区	银川市	中宁县枸杞种植基地	2023

2、数据对比分析 (功能点-简介): 对生长数据等进行统计和可视化。

Echarts图表展示, 各种图进行拿取组件

3、图谱比对:

在UI层面, 可以设计一个左右分栏的界面, 允许用户选择两张药材图片进行并排展示, 方便人工对比。如果想做得更深入, 后端可以利用简单的图像处理库 (如OpenCV) 提取颜色直方图等特征, 给出一个简单的“相似度”评分作为参考。

调用接口就行

4、溯源:

在数据库设计层面, 为关键数据表 (如药材生长数据表) 增加一个对应的“历史记录表”。每当主表数据发生变更时, 通过触发器或业务逻辑, 将变更前的数据、变更人、变更时间等信息存入历史记录表, 形成一条可追溯的链路。

 Note

模块二

多端数据采集与共享系统

一、核心思想与技术流程

本模块的核心是实现一个“前后端分离、云存储直传”的现代化数据流。

- **前端 (Flutter/Vue):** 负责数据采集和展示, 但不直接处理文件。
- **后端 (SpringBoot):** 作为“交通枢纽”, 负责处理业务逻辑、校验数据, 并**授权**前端进行文件上传。
- **云存储 (阿里云OSS):** 作为文件仓库, 直接接收前端上传的文件。
- 1. App请求上传凭证 -> 2. 后端向阿里云请求临时凭证
- 3. 阿里云返回凭证给后端 -> 4. 后端将凭证下发给App
- 5. App使用凭证直传图片到OSS -> 6. App将业务数据和图片URL提交给后端
- 7. 后端将数据写入数据库

二、后端API设计 (SpringBoot)

后端需要提供两类核心接口: **业务数据接口**和**文件上传授权接口**。

1. 文件上传授权API

这是实现OSS直传的关键。

- **Endpoint:** `GET /api/oss/sts-credential`
- **功能:** 生成一个临时的、有时间限制的、有特定上传权限的STS (Security Token Service) 凭证。
- **实现步骤:**
 1. 在SpringBoot项目中引入阿里云OSS和STS的SDK。
 2. 配置你的主AccessKey ID和Secret (这些保存在后端, 绝不外泄)。
 3. 编写一个Controller, 当收到此请求时, 调用STS SDK生成临时凭证, 其中包含:
 - `AccessKeyId` (临时的)

- `AccessKeySecret` (临时的)
- `SecurityToken`

4. 同时, 在后端定义好上传策略, 例如: 允许上传到哪个Bucket、哪个目录下 (如 `images/herbs/`)、文件大小限制等。将这些信息连同STS凭证一起返回给前端。

- 返回给前端的JSON示例:

```
{
  "accessKeyId": "STS.ABC...",
  "accessKeySecret": "xyz...",
  "securityToken": "CAIS...",
  "region": "oss-cn-hangzhou",
  "bucket": "your-herb-bucket-name",
  "uploadDir": "images/herbs/"
}
```

2. 核心业务API

这个接口用于接收Flutter App采集到的所有数据。

- **Endpoint:** `POST /api/locations`
- **功能:** 创建一条新的药材观测记录及其关联数据。
- **接收的JSON Body示例:**

```
{
  "herbId": 1, // 用户选择的药材ID
  "longitude": 127.98333,
  "latitude": 43.88333,
  "province": "吉林省",
  "city": "吉林市",
  "address": "集安市山区发现",
  "observationYear": 2024,
  "images": [ // 刚刚上传到OSS的图片URL列表
    {
      "url": "https://your-bucket.oss-cn-hangzhou.aliyuncs.com/images/herbs/uuid1.jpg",
      "isPrimary": true,
      "description": "植株整体形态"
    },
    {
      "url": "https://your-bucket.oss-cn-hangzhou.aliyuncs.com/images/herbs/uuid2.jpg",
      "isPrimary": false,
      "description": "叶片特写"
    }
  ],
  "growthData": [ // 伴随的生长统计数据
    {
      "metricName": "平均株高",
      "metricValue": "50",
      "metricUnit": "厘米"
    }
  ]
}
```

- **实现步骤:**

1. 创建一个DTO (Data Transfer Object) 来接收这个复杂的JSON结构。
2. 使用 `@Transactional` 注解保证数据库操作的原子性（要么全部成功，要么全部失败）。
3. 在一个Service方法中，按顺序执行以下数据库操作：
 - a. 向 herb_location 表插入一条记录，并获取返回的主键 location_id。
 - b. 遍历 images 数组，向 herb_image 表插入记录，并关联上一步得到的 location_id。
 - c. 遍历 growthData 数组，向 herb_growth_data 表插入记录，并关联 location_id。

三、移动端 (Flutter) 实现步骤

移动端的核心是**用户交互**和**与后端API的协同**。

1. **环境准备:**

- 在 `pubspec.yaml` 中添加必要的库: `http` (网络请求), `image_picker` (访问相册/相机), `geolocator` (获取GPS), 以及阿里云官方的 `aliyun_oss_flutter_sdk`。

2. **数据采集页面 (UI):**

- 使用Flutter的表单控件 (`TextField`, `DropDownButton` 等) 创建一个采集界面。
- 药材列表 (`herb` 表的数据) 应通过调用后端的 `GET /api/herbs` 接口动态获取，让用户选择。
- 放置“获取当前位置”、“拍照/选择图片”等按钮。

3. **核心功能实现 (Logic):**

- **获取GPS:** 调用 `geolocator` 库的 `getCurrentPosition()` 方法。
- **选择图片:** 调用 `image_picker` 库的 `pickImage()` 方法，让用户拍照或从相册选择。

4. **上传流程实现 (关键步骤):**

- 当用户点击“提交”按钮时，触发以下异步流程：
 - a. (并行) 图片上传:
 - i. 对于用户选择的每一张图片，先调用后端的 `GET /api/oss/sts-credential` 接口获取上传凭证。
 - ii. 使用 `aliyun_oss_flutter_sdk`，传入获取到的凭证和图片文件，执行上传。
 - iii. 上传成功后，OSS会返回文件的URL。将这些URL收集起来。
 - b. (等待图片上传完毕后) 提交业务数据:
 - i. 组装前面提到的 `POST /api/locations` 所需的JSON Body，其中 `images` 数组里填入上一步获取到的所有图片URL。
 - ii. 使用 `http` 库将这个JSON发送给后端。
 - c. 完成与反馈:
 - i. 根据后端的返回结果，给用户明确的提示（“上传成功”或“上传失败，请重试”）。在整个过程中，显示一个加载指示器 (`CircularProgressIndicator`) 以提升用户体验。

四、 电脑终端 (Vue/Electron) 功能

电脑终端是数据的消费者和管理者。

1. 数据展示:
- 调用后端的查询接口（如 `GET /api/locations?herbId=1`），获取数据并在地图上渲染 Marker 点，或在表格中展示详细信息。
2. 数据管理:
- 实现对 `herb_location`, `herb_growth_data` 等数据的增、删、改、查（CRUD）操作。

例如，管理员可以在Web端修正一个错误的地址信息，或为某个观测点补充新的统计数据。
3. 数据溯源查看:
- 提供一个界面，调用后端API查询 `herb_growth_data_history` 表，清晰地展示某条数据的每一次变更记录。

直接粘过来就行（不做最优先考虑）

 Note

模块三

在线管理教育平台

核心功能：

1. 教学资源管理 (功能点3, 4, 5): 试验课程、课题研究、培训素材的存储和管理。
2. 线上视频教学 (功能点-简介): 提供在线视频播放功能

设计要点：

这本质上是一个内容管理系统 (CMS)。你需要为课程、课题、培训素材分别设计数据表。

后台管理界面需要集成一个富文本编辑器（如WangEditor、TinyMCE），方便教师编辑图文并茂的课程内容。

视频教学可以直接利用OSS的文件链接，在前端使用成熟的视频播放器组件（如vue-video player）来实现

1. 用户表 (users)

字段名	类型	说明	约束
id	BIGINT	用户主键ID	主键，自增
username	VARCHAR(64)	用户名	唯一，必填
role	VARCHAR(32)	角色 (student, teacher, admin)	默认student
created_at	TIMESTAMP	创建时间	默认当前时间

2. 教学资源分类表 (edu_categories)

字段名	类型	说明	约束
id	INT	分类主键ID	主键, 自增
name	VARCHAR(50)	分类名称	必填
slug	VARCHAR(50)	分类别名 (URL用途)	唯一, 必填
description	VARCHAR(255)	分类描述	可空

3. 教学资源主表 (edu_resources)

字段名	类型	说明	约束
id	BIGINT	资源主键ID	主键, 自增
title	VARCHAR(255)	资源标题	必填
category_id	INT	关联分类ID (外键)	必填
author_id	BIGINT	作者的用户ID (外键)	必填
cover_image_url	VARCHAR(255)	封面图片URL	可空
content	LONGTEXT	资源主体内容 (富文本)	可空
status	ENUM	状态 (draft/published/archived)	默认draft
created_at	TIMESTAMP	创建时间	默认当前时间
updated_at	TIMESTAMP	更新时间	自动更新
published_at	TIMESTAMP	发布时间	可空

4. 教学视频库 (edu_videos)

字段名	类型	说明	约束
id	BIGINT	视频主键ID	主键, 自增
title	VARCHAR(255)	视频标题	必填
description	TEXT	视频简介	可空
video_url	VARCHAR(255)	视频文件URL (OSS)	必填
cover_url	VARCHAR(255)	视频封面URL	可空
duration	INT	视频时长 (秒)	默认0
uploader_id	BIGINT	上传者的用户ID (外键)	必填
created_at	TIMESTAMP	上传时间	默认当前时间

5. 资源与视频关联表 (edu_resource_video_link)

字段名	类型	说明	约束
id	BIGINT	关联主键ID	主键, 自增
resource_id	BIGINT	教学资源ID (外键)	唯一约束 (组合)
video_id	BIGINT	教学视频ID (外键)	唯一约束 (组合)
display_order	INT	显示顺序 (章节排序)	默认0

表之间主要外键关系说明:

- edu_resources.category_id → edu_categories.id
- edu_resources.author_id → users.id
- edu_videos.uploader_id → users.id
- edu_resource_video_link.resource_id → edu_resources.id
- edu_resource_video_link.video_id → edu_videos.id

如果需要插入数据的Markdown表格样例，可以补充说明。

 **Note**

模块五

日志和用户管理

1. 用户账户与角色表 (users)

字段名	类型	说明	约束
id	BIGINT	用户主键ID	主键, 自增
username	VARCHAR(64)	用户名, 用于登录	唯一, 必填
password_hash	VARCHAR(255)	加盐哈希后的密码	必填
role	VARCHAR(32)	角色标识 (admin/teacher/student)	默认student
status	TINYINT(1)	账户状态 (1-正常, 2-禁用)	默认1
created_at	TIMESTAMP	注册时间	默认当前时间
updated_at	TIMESTAMP	更新时间	自动更新

2. 用户信息表 (user_profiles)

字段名	类型	说明	约束
user_id	BIGINT	用户ID (外键)	主键, 关联users.id
nickname	VARCHAR(64)	用户昵称	可空
avatar_url	VARCHAR(255)	头像URL	可空
gender	ENUM	性别 (male/female/unknown)	默认unknown
bio	VARCHAR(255)	个人简介	可空