

软件方案设计说明书

软件方案设计说明书

Green Cloud 系统

版本： 1.0

编订：刘丽、火忻

团队：Just do IT!

日期： 2018-6-19

1.引言

1.1 编写目的

本文档的目的是说明Green Cloud系统各个层次中的每个模块或子程序和数据库系统的设计考虑，详细说明系统的设计方案，开发一个可以直接转化成程序语言的软件表示，使得管理员和软件设计人员对软件的各模块以及功能有更深入的了解。本文档的编写为程序员的编码提供依据，为软件开发组制定在设计时应该共同遵守的标准，以便协调组内各成员的工作，提供软件的表示，为软件的质量评价提供依据。

在前一阶段《系统详细设计说明书》中，已经说明了该系统各模块的设计。包括软件系统的整体结构的说明、模块的外部设计、接口的实现以及多错误的处理等。在以下的系统方案设计说明书中将对在本阶段中对系统所做的所有详细设计进行说明。

在以下的各个阶段中，《用户操作手册》将与本阶段的工作紧密结合，努力作到让用户易懂易学。《测试报告》和《维护报告》也将参考本说明书，检验本系统的各项性能指标，及时发现纰漏及时修补，一定要把功能强大、稳定可靠、便于维护的网上报名管理系统交到用户手中。

本文档的预期读者有客户（植物养殖者），项目经理，开发人员以及跟该项目相关的其他竞争人员和无关人员。

1.2 背景

- (1) 软件系统的名称：Green Cloud
- (2) 开发者：张琪、张永琪、付恩丽、火忻、刘丽、刘琼
- (3) 开发环境：JS、IDEA 、Android Studio
- (4) 用户群体：植物养殖者、物联网开发者
- (5) 系统模式：客户端/服务器模式
- (6) 数据库管理系统：mysql 5.7

(7) 项目目标：利用软硬件结合的技术，实现自动化养殖和实时监测的功能，为广大植物爱好者提供便利。开发WEB服务器，解决物联网开发者自己开发服务器耗时耗力、资金不够、风险大的问题。

1.3 参考资料

- (1) 项目开发计划 本项目开发小组
- (2) 可行性研究报告 本项目开发小组
- (3) 软件需求说明书 本项目开发小组
- (4) 概要设计说明书 本项目开发小组
- (5) 软件工程导论 张海潘、倪宁 人民邮电出版社

1.4 术语定义及说明

- (1) MySQL：系统服务器所使用的数据库管理系统(DBMS)。
- (2) SQL：Structured Query Language（结构化查询语言）
- (3) 事务流：数据进入模块后可能有多种路径进行处理。
- (4) 主键：数据库表中的关键域。值互不相同。
- (5) 外部主键：数据库表中与其他表主键关联的域。
- (6) FOLLBACK：数据库的错误恢复机制。
- (7) ATM：Asynchronous Transfer Mod（异步传输模式）

2.总体设计

2.1 程序系统组织结构

根据Green Cloud系统的特点，开发的Green Cloud系统主要分为4大模块：登录模块、植物监控模块、设备管理和植物推荐4个主要模块。Green Cloud系统的功能结构图如图2.1所示。

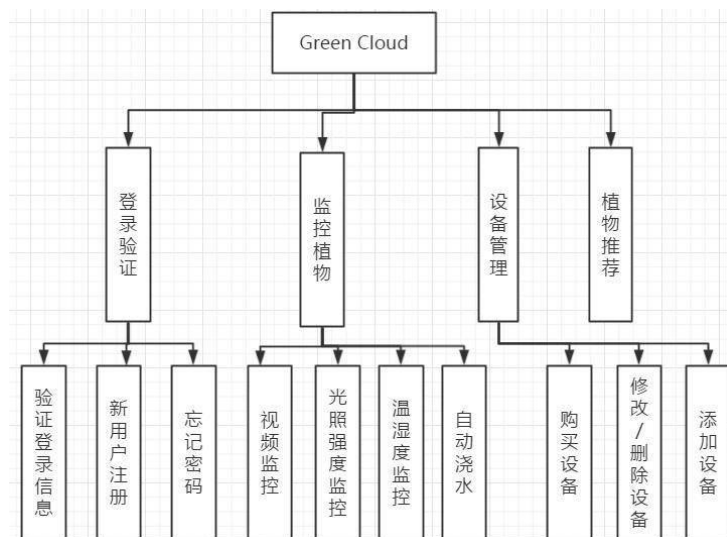


图2.1 Green Cloud 系统的功能结构图

2.2 总体目标

某用户要实时监控动植物生长状态，物联网开发者要实时获取所需数据，数据之间的传送和服务质量必须保证一定的稳定性及通用性。

2.3 设计原则

本项目的设计以需求为导向，尽量收集用户的要求，采用构件的设计思想，便于系统功能的重组和扩充。系统采用“总体规划、分步实施”的策略，第一步完成通用控制页面的设计，接

下来完成常用的后台服务程序的开发，最终达到实现质量体系的目标。为保障项目的各项性能，本项目在设计和实施过程中应遵循如下的原则：

先进性：思想领先性和功能领先性。既要保证系统在满足当前的功能需求，也为将来的功能需求提供条件，与此同时保持系统功能的先进性。

实用性：充分考虑到质量管理和流程控制的实际需要，设计方案留有一定的余地，即在设计中构造明晰的体系结构，便于更改；完全根据企业的生产实践进行软件设计，达到实用的目的。

可扩充性：采用面向对象的设计思想和开发技术以及构件设计思想，保证在用户的需求发生变化和增加时，使系统的修改量减到最小。

开放性：由于本系统涉及许多硬件设备和外部应用软件，为了更好地和各种硬件设备和应用软件接口，在设计时，应提供主要的接口标准。

标准化：在系统总体规划设计中，所有软、硬件产品的选择必须选择符合开放性和国际标准化的产品和技术；在应用软件开发中，必须遵循总体组制定的各项规范和要求。

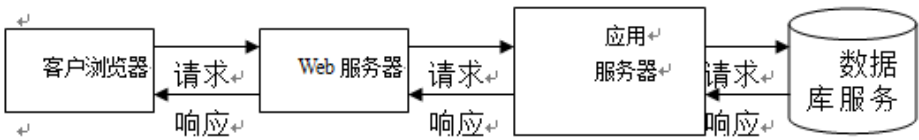
安全与保密：系统设计时应充分考虑系统的安全，提供备份的功能，保证当系统出现硬件或软件的故障时，能进行恢复；另外，应对用户进行权限管理，保证信息的安全。

以数据为中心的设计原则：本项目采用以主题数据库为中心的设计思想，首先把本项目的主题数据库设计出来，再在此基础上开发各种应用。由于主题数据库是项目的基本信息资源，它具有很好的不变性和稳定性，相对独立与具体的应用，当用户的需求发生变化时，使修改量减到最小。

2.4 总体方案思想建立

GREEN CLOUD植物管理推荐系统的主要组成从软件实现上可分为四部分，即植物数据信息收集、用户喜好植物推荐、植物状态实时监控、平台养殖经验交流；从系统架构上可分为三部分，即应用程序客户端、网络应用程序和数据后台。

软件设计采用三层结构的设计方法，即通过服务器层实现应用前端与后台数据库的数据共享，通过JAVA语言和相关的小程序以及数据库处理实现GREEN CLOUD系统。



- Web发布部分

Web服务器仅仅是把要显示的内容从站点上以文件的形式读取，然后以静态的HTML格式送到客户端的浏览器；也可以Applet增强表现能力，但它仅仅是利用ActiveX或JavaBeans通

过页面或组件，并没有通过任何事务数据服务器,目前项目展现成果是以HTML格式送至客户端的浏览器的。

- 数据处理部分

Web数据处理增强了标准Web站点存取数据的能力，包括许多数据类型。我们可根据数据的存取容量把数据分成两大组类：标准的在线事务处理(OLTP)程序将花费大量时间去检索和操作核心在线数据，这种数据需要连续读取和回写，如植物温湿度数据信息的实时提取。而另一种辅助数据是只读的，如帮助文件、用户信息和物联网开发者使用手册等等。Web数据处理主要集中在辅助数据，而Web OLTP主要集中在核心在线数据。

- 客户端

客户端是表现逻辑层

Web: Web是一个成功的平台，用户可以在断开连接后继续工作，这就意味着远程象本地一样可存取事务和逻辑数据。这样不仅要分发应用程序，而且还要分发数据。

APP: APP是直接为用户进行对接的部分，普通用户通过使用APP对自己的设备直接进行管理，站在用户层面，为用户的使用方便来考虑这部分的设计显得十分必要。

3.登录验证模块设计说明

3.1 程序描述

该模块用于用户根据个人帐号，密码登录该系统，获得使用该软件的权限；同时系统也会对错误信息进行反馈报错。

3.2 功能

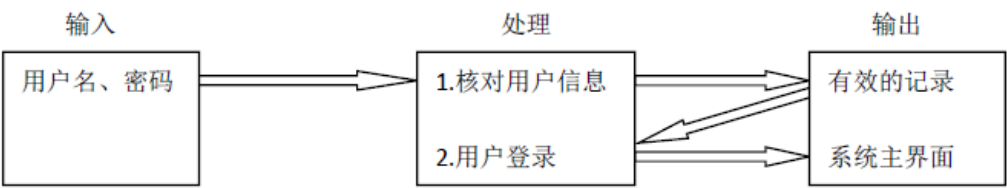


图 3.1 登录模块 IPO 图

3.3 性能

(1) 精度要求

根据用户的需求，本系统在各项输入、输出项给出了明确的精度要求。

(2) 时间需求

在软件方面，响应时间，更新处理时间都比较快且迅速，完全满足用户要求。

(3) 灵活性

当用户需求，如操作方式，运行环境，结果精度，数据结构于其他软件接口等发生变化时，设计的软件要做适当调整,灵活性非常大。

(4) 故障处理

内部故障处理：在开发阶段可以随即修改数据库里的相应内容。

外部故障处理：对编辑的程序进行重装载时，第一次装载认为错，修改。第二次运行，在需求调用时出错，有错误提示，重试。

(5) 安全与保密

用户间数据相对隔离，设置多重权限控制，用户密码等非还原较验数据采用 MD5 加密，防止泄密。

3.4 输入项

名称	数据类型	数据长度	输入方式	标识
用户名	varchar	64	鼠标或键盘键入	username
密码	varchar	64	鼠标或键盘键入	password
验证码	varchar	32	鼠标或键盘键入	Api_key

3.5 输出项

Green Cloud 系统主界面。

3.6 算法

- (1) 用户输入完用户名、密码点击登录按钮，判断用户名、密码是否为空，为空则弹出提示对话框。
- (2) 用户名、密码不为空时，与数据库中的记录核对。
- (3) 有效的用户记录，该用户根据用户类型不同进入对应的主界面，否则弹出错误对话框。

3.7 流程逻辑

根据算法画出本模块的流程图如图 3.2 所示

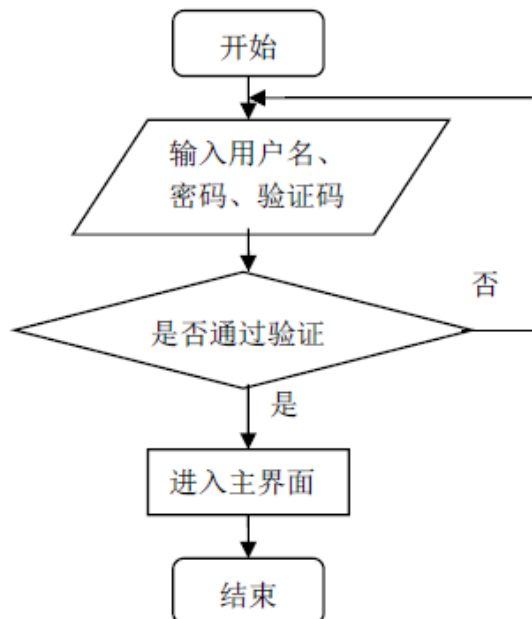


图 3.2 登录模块流程图

3.8 接口

(1) 外部接口

Android 应用软件用户界面的规范来设计，使用以对话框、按钮为主的用户界面，便于用户使用。

(2) 内部接口

此系统模块间采用数据耦合方式，通过参数表传送数据，交换信息。数据视图是基于数据结构来建立的，如果信息的结构使用相同的格式定义的话，多种信息可以共享相同的数据视图组件显示信息的内容。

数据结构是以描述方式定义的，与数据视图的实现平台没有关系，因此在构件环境下使用可以很方便。数据行为基本不关心数据视图是如何实现的，数据行为改变数据之后并不直接修改数据视图，而是通过让数据视图刷新来得到数据的变化，数据行为与数据视图的关联在参数传递之后就结束了。

数据视图的复用本身就意味着数据行为可以被复用，如果数据视图调用数据行为的参数是一致的话，相同数据行为还可以被多种数据视图所复用。由于数据视图和数据行为都可以通过上述方式被复用，因此在框架中数据视图和数据行为可以像积木那样被组装起来，只要他们在关联的地方具有相同的描述方式。当原先使用的数据视图或数据行为已经不符合新的需求时，可以将新的数据视图或数据行为构件代替原来的构件与其他部分重新组合就可以完成升级，不影响其他信息对象仍然使用原来的构件，也不需要维护相同构件的多个版本。

事实上，数据从头到尾都是隐藏在数据视图和数据行为的背后。只要数据结构定义的描述方式没有改变，即使数据结构本身改变了数据视图和数据行为仍旧是可用的。

(3) 用户接口

Green Cloud 系统与用户之间通过系统管理来进行信息交换，主要包括的外部接口为用户的账号输入，密码输入。只有当用户输入的账号和密码正确时，才能登录此系统，并利用此系统来对植物进行各种管理。用户与系统交互界面：

根据该系统的工作特点，将用户的界面设计分成以下几个部分，每一部分完成一项独立功能，在主窗口提供进入各个功能的入口，具体划分如下：

主控窗口：提供菜单，功能按钮，状态栏等操作部件，用户可选择进入具体的操作。

数据维护窗口：提供植物添加、修改、删除，植物查询即植物监控，包括植物的温湿度、光照强度以及实时画面，提供设备添加、修改、删除，植物推荐功能。

3.9 存储分配

所有数据存储于数据库服务器，采用 MySQL 数据库系统。

3.10 注释设计

(1) 用“//”进行单行的代码注释

(2) 用“/**xxxxxxxxxxxxx

xxxxxxxxxxxxxxxxx

*/”进行方法注释

(3) 用“/*xxxxxx

xxxxxxxxx

*/”进行多行注释

3.11 限制条件

(1) 技术约束

本项目的设计是在 java 程序设计语言的条件下进行的，技术设计采用软硬一体化的设计方法。

(2) 环境约束

操作系统：Windows98/NT/2000 以上

浏览器：IE5 以上

开发工具：MySQL NetBeans7.0

(3) 标准约束

该软件的开发完全按照企业标准开发，包括硬件、软件和文档规格。

(4) 硬件限制

- CPU：1GHz 以上
- RAM：256M 以上
- 存储容量：剩余存储容量大于 100M
- PC 工作站
- CPU：500MHz 以上
- RAM：128M 以上
- 显示设备：支持 1024X768 显示分辨率

3.12 测试计划

用例 ID	1111-11-11	用例名称	用户登录使用	
测试用例 ID	场景	测试步骤	预期结果	备注
TC1	正确有效的用户	进入登录页面,输入账号密码	登陆成功	无效输入和错误输入统一为错误信息报错
TC2	错误无效的用户		报错返回登录界面	
TC3	正确操作	进入选择操作界面并选择操作	进入相应管理模块	
TC4	无效操作		返回选择操作界面	

3.13 尚未解决的问题

用户间数据相对隔离，设置多重权限控制，用户密码数据为实现采用 MD5 加密，来防止泄密。

4.植物监控模块设计说明

4.1 程序描述

植物监控模块包括植物添加、植物删除、植物查询 3 项内容，植物查询包括植物的温湿度、光照强度、实时画面信息、自动化浇水四方面。

4.2 功能

植物监控模块的 IPO 图如图 4.1 所示。

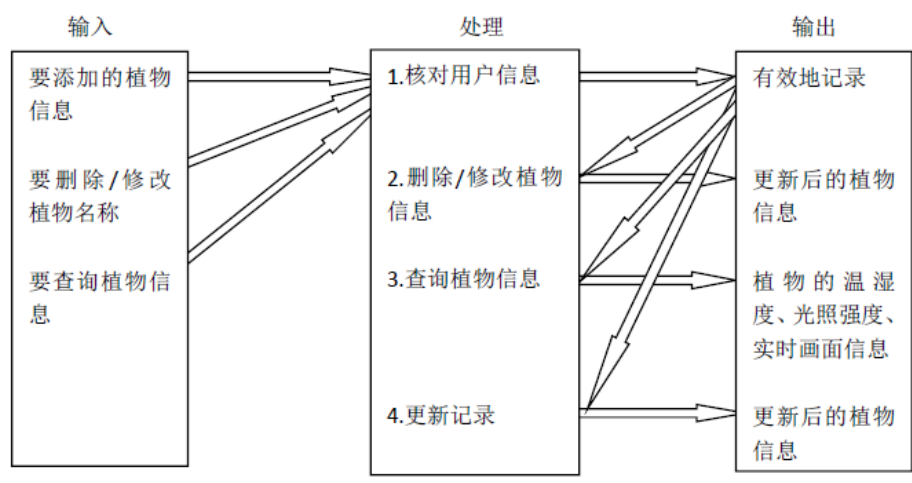


图 4.1 植物监控模块 IPO 图

4.3 性能

本软件中，文件的大小的计量单位最小为 KB（兆比特），其余依次为 MB（兆比特）、GB（千兆比特）以及 TB（千吉比特）。

响应时间：一般页面的相应时间不超过 5 秒，正常传输文件的时间与文件大小相对应。

对大多数操作系统保持良好的兼容性，其他无要求。

本软件设定支持中文以及英文的输入输出。

4.4 输入项

名称	数据类型	数据长度	输入方式	标识
用户名	varchar	64	鼠标或键盘键入	username
密码	varchar	64	鼠标或键盘键入	password
验证码	varchar	32	鼠标或键盘键入	Api_key
植物名称	varchar	64	鼠标或键盘键入	Plant_name

4.5 输出项

名称	数据类型	数据长度	输入方式	标识
植物名称	varchar	64	文本格式	Plant_name
温度	int	32	数字方式	Sensor_one
湿度	int	32	数字方式	Sensor_two
光照强度	int	32	数字方式	Sensor_three

4.6 算法

设备植物:

(1) 输入植物的各项信息，按提交按钮，判断用户名是否为空。如果为空则弹出警告对话框。

(2) 用户名不为空时，将该条记录插入数据库。

植物修改/删除:

- (1) 在植物名称处，直接从数据库中调用已添加的全部植物名称，用下拉列表显示。
- (2) 选好要修改/删除的植物名称后，点击确定，下面会显示该植物的部分信息。
- (3) 要修改植物信息时，在本已显示个原信息的基础上对其修改，点击修改按钮，更新数据库中的信息。
- (4) 要删除植物信息时，直接按删除按钮，删除数据库中该植物的全部信息。

植物查询：

可根据设备名称对其进行查询操作，显示该植物目前的温湿度、光照强度、实时画面以及自动化浇水开关。

4.7 流程逻辑

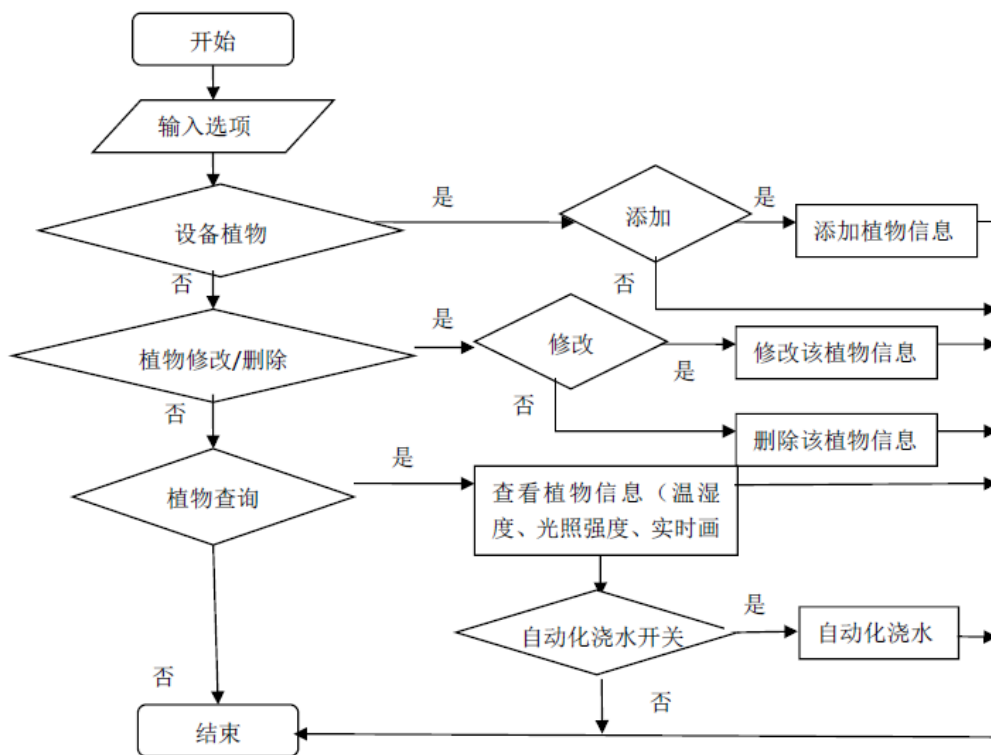
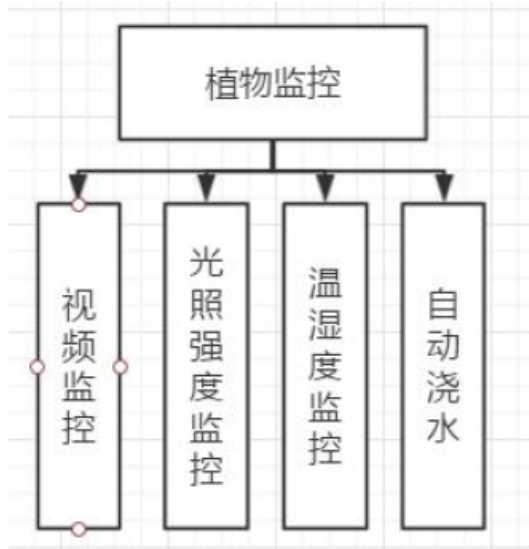
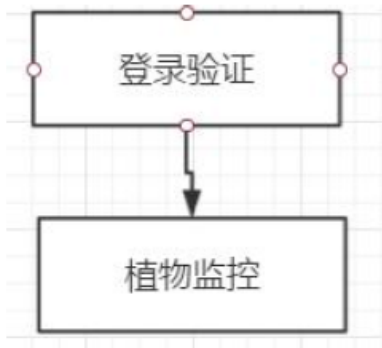


图 4.2 植物监控模块流程图

4.8 接口



4.9 存储分配

所有数据存储于数据库服务器，采用 MySQL 数据库系统。

4.10 注释设计

同本文 3.10。

4.11 限制条件

同本文 3.11。

4.12 测试计划

用例 ID	233333333	用例名称	用户登录使用	
测试用例 ID	场景	测试步骤	预期结果	备注
TC1	用户登录	输入正确账号密码	登录成功	
TC2	选择操作	进入选择界面选择植物监控操作	选择成功	
TC3	添加植物请求	输入添加请求	成功输入	
TC4	正确添加请求	输入正确的添加请求	返回输入请求界面	
TC5	修改植物请求	输入修改请求	成功输入	
TC6	正确修改请求	输入正确的修改请求	返回输入请求界面	
TC7	查看植物请求	输入查看请求	选择植物名称（百合）	
TC8	输出植物（百合）信息	植物监控信息响应	成功显示植物相关信息	

4.13 尚未解决的问题

暂无。

5.设备管理模块设计说明

5.1 程序描述

设备管理模块包括设备添加、修改、删除、查询 4 项内容。实现了设备的全部操作。

5.2 功能

设备信息的添加、修改、删除、查询功能。设备管理模块的 IPO 图如图 5.1 所示。

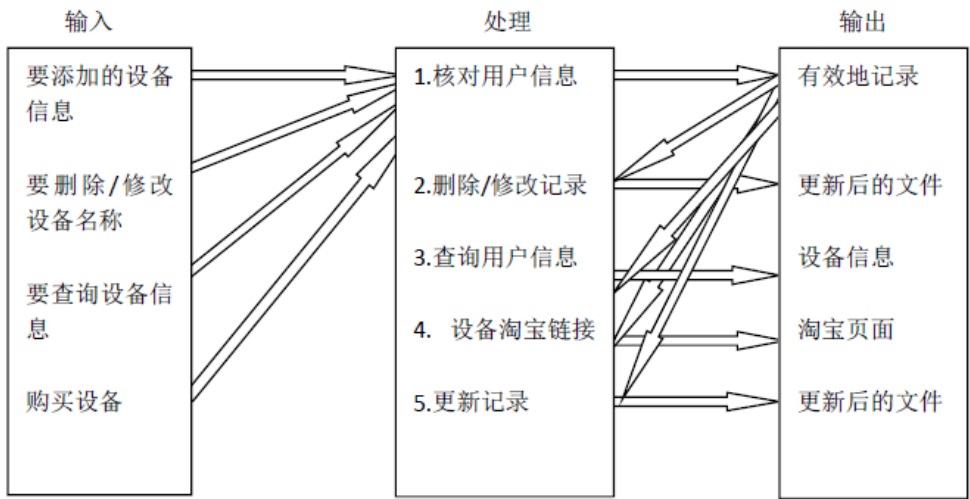


图 5.1 设备管理模块 IPO 图

5.3 性能

本软件中，文件的大小的计量单位最小为 KB（兆比特），其余依次为 MB（兆比特）、GB（千兆比特）以及 TB（千吉比特）

响应时间：一般页面的相应时间不超过 5 秒，正常传输文件的时间与文件大小相对应。

对大多数操作系统保持良好的兼容性，其他无要求。

本软件设定支持中文以及英文的输入输出。

5.4 输入项

名称	数据类型	数据长度	输入方式	标识
用户名	varchar	64	鼠标或键盘键入	username
密码	varchar	64	鼠标或键盘键入	password
验证码	varchar	32	鼠标或键盘键入	Api_key
设备名称	char	64	鼠标或键盘键入	dev_name
设备价格	int	32	鼠标或键盘键入	dev_price
设备所属植物	char	64	鼠标或键盘键入	Plant_name

5.5 输出项

名称	数据类型	数据长度	输入方式	标识
设备名称	char	64	鼠标或键盘键入	dev_name
设备价格	int	32	鼠标或键盘键入	dev_price
设备所属植物	char	64	鼠标或键盘键入	Plant_name

5.6 算法

设备添加:

(1) 输入设备的各项信息, 按提交按钮, 判断用户名是否为空。如果为空则弹出警告对话框。

(2) 用户名不为空时, 将该条记录插入数据库。

设备修改/删除:

(1) 在设备名称处, 直接从数据库中调用已添加的全部设备名称, 用下拉列表显示。

(2) 选好要修改/删除的设备名称后, 点击确定, 下面会显示该设备的全部信息。

(3) 要修改设备信息时, 在本已显示个原信息的基础上对其修改, 点击修改按钮, 更新数据库中的信息。

(4) 要删除设备信息时, 直接按删除按钮, 删除数据库中该设备的全部信息。

设备查询:

(1) 可根据设备名称对其进行查询操作。

(2) 当什么都没选时, 点击查询按钮为全部信息查询。

5.7 流程逻辑

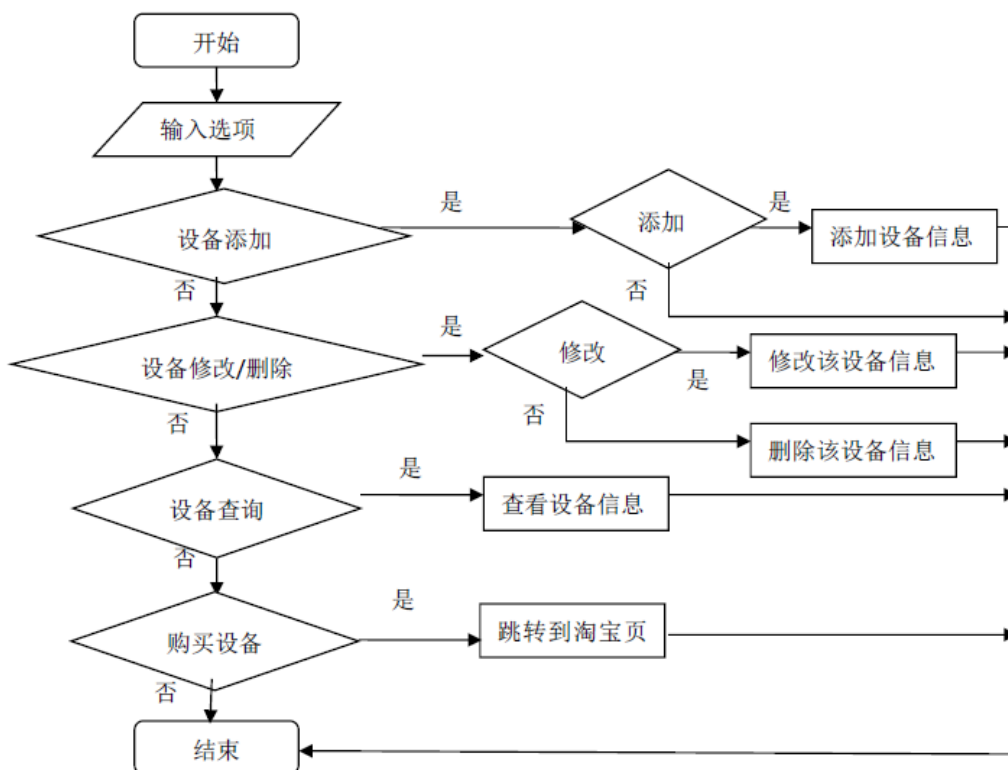
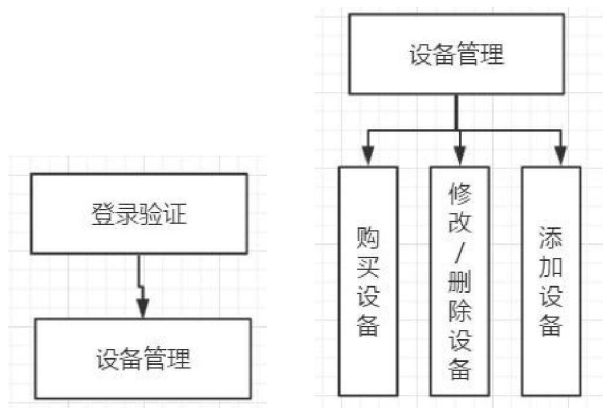


图 5.2 设备管理模块流程图

5.8 接口



5.9 存储分配

所有数据存储于数据库服务器，采用 MySQL 数据库系统。

5.10 注释设计

同本文 3.10。

5.11 限制条件

同本文 3.11。

5.12 测试计划

用例 ID	3-3333	用例名称	用户登录使用	
测试用例 ID	场景	测试步骤	预期结果	备注
TC1	用户登录	输入正确账号密码	登录成功	
TC2	选择操作	进入选择界面选择设备管理操作	选择成功	
TC3	添加设备请求	输入添加请求	成功输入	
TC4	正确添加请求	输入正确的添加请求	返回输入请求界面	
TC5	错误添加请求	输入错误的添加请求	返回输入请求界面	
TC6	修改设备请求	输入修改请求	成功输入	
TC7	正确修改请求	输入正确的修改请求	返回输入请求界面	
TC8	查看设备请求	输入查看请求	选择设备名称	
TC9	输出设备信息	设备信息响应	成功显示设备信息	

5.13 尚未解决的问题

并发时资源占用极大。

6.植物推荐模块设计说明

6.1 程序描述

该模块用于向用户推荐一定时间段内适宜种植的花卉。

6.2 功能

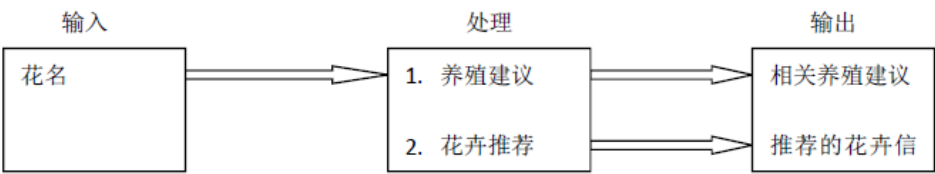


图 6.1 推荐模块 IPO 图

6.3 性能

本软件中，文件的大小的计量单位最小为 KB（兆比特），其余依次为 MB（兆比特）、GB（千兆比特）以及 TB（千吉比特）。

响应时间：一般页面的相应时间不超过 5 秒，正常传输文件的时间与文件大小相对应。

对大多数操作系统保持良好的兼容性，其他无要求。

本软件设定支持中文以及英文的输入输出。

6.4 输入项

名称	数据类型	数据长度	输入方式	标识
用户名	varchar	64	鼠标或键盘键入	username
密码	varchar	64	鼠标或键盘键入	password
验证码	varchar	32	鼠标或键盘键入	Api_key

6.5 输出项

名称	取值类型	长度	输出形式
花名	varchar	64	文字方式
建议	varchar	128	文字方式
适宜季节	varchar	64	文字方式
账号	Char	20	文字数字结合方式

6.6 算法

web 爬虫中需要设计一个广度优先的算法，以控制爬虫爬行网址的先后顺序，这里用一个链表实现，用链表是因为链表的插入速度够快。设计思路：

取下一个地址：从链表的头部取出一个，并将头部元素删除

加入地址池：将 URL 地址加入到适当的位置

为了保证加入的时候能够加入到合适的地址，最容易想到的办法就是遍历那个地址池，但遍历的效率确实不高，当地址池中数量增大的时候，消耗在遍历上的 cpu 资源过多，导致爬

行效率降低。还有一种方法就是记录每一个深度的 URL地址的最后一个元素在地址池中索引，当加入的时候就不需要遍历地址池，只需通过需要加入的 URL 地址，找到同级 URL 地址在地址池中的索引，然后加入到地址池中，加入位置为这个索引的后面一个，并且更新索引表，当然如果没有找到同级别的索引，这依次找上一级的索引，若在回溯的过程中，都没有找到，这可以确定索引为 0，即将该 URL 地址加入到地址池的最前面。

6.7 流程逻辑

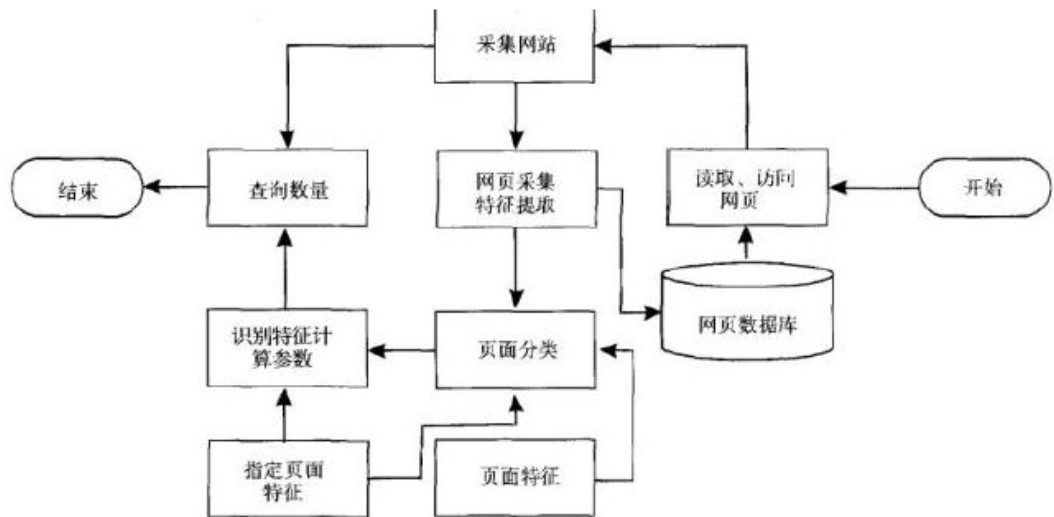
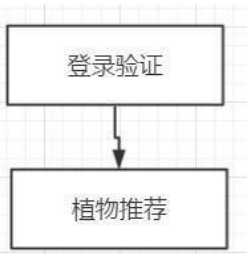


图 6.2 爬虫流程图

6.8 接口



6.9 存储分配

所有数据存储于数据库服务器，采用 MySQL 数据库系统。

6.10 注释设计

同本文 3.10。

6.11 限制条件

同本文 3.11。

6.12 测试计划

无测试需求。

6.13 尚未解决的问题

http 协议请求头部字节占用较多。

7.数据库系统设计

7.1 设计要求

(1) 原子性

一整个事务中的所有操作，要么全部完成，要么全部不成功，不可能停滞在中间某个环节。事务在执行过程中发生错误，会被回复（Rollback）到事务开始前的状态，就像这个事务从来没有执行过一样。

(2) 一致性

在事务开始之前和事务结束以后，数据库的完整性限制没有被破坏。

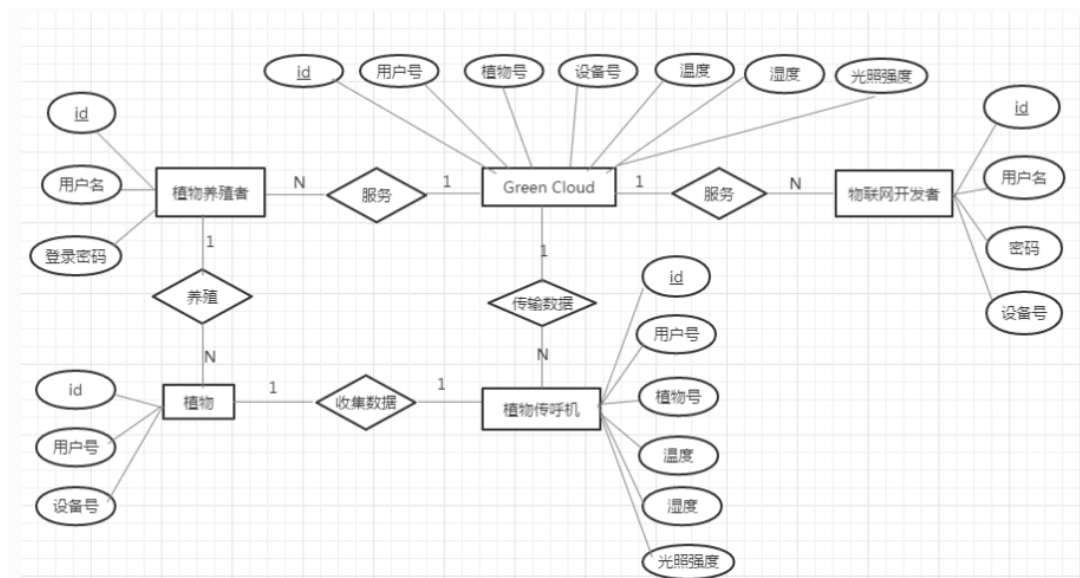
(3) 隔离性

两个事务的执行是互不干扰的，一个事务不可能看到其他事务运行时，中间某一时刻的数据。

(4) 持久性

在事务完成以后，该事务对数据库所作的更改便持久地保存在数据库之中，并不会被回复

7.2 信息模型设计



7.3 数据库设计

设计依据：

数据库设计应参照以下原则进行设计：

- (1) 方便各模块功能的实现和扩展；
- (2) 方便设计开发，增强系统的稳定性和可维护性；
- (3) 保证数据的完整性和安全性；

(4) 提高数据存储效率，在满足功能需求的前提下，使时间开销和空间开销达到优化平衡；

(5) 数据库、数据库对象、表、表空间、视图、索引、序列等的命名都应遵循一定的规则。

数据库种类及特点：

本项目的数据库设计采用Mysql，特点有：

(1) 它使用的核心线程是完全多线程，支持多处理器。

(2) 有多种列类型：1、2、3、4、和8字节长度自有符号 / 无符号整数、FLOAT、DOUBLE、CHAR、VARCHAR、TEXT、BLOB、DATE、TIME、DATETIME、TIMESTAMP、YEAR、和ENUM类型。

(3) 它通过一个高度优化的类库实现SQL函数库并像他们能达到的一样快速，通常在查询初始化后不该有任何内存分配。没有内存漏洞。

(4) 全面支持SQL的GROUP BY和ORDER BY子句，支持聚合函数(COUNT()、COUNT(DISTINCT)、AVG()、STD()、SUM()、MAX()和MIN())。你可以在同一查询中混来自不同数据库的表。

(5) 支持ANSI SQL的LEFT OUTER JOIN和ODBC。

(6) 所有列都有缺省值。你可以用INSERT插入一个表列的子集，那些没用明确给定值的列设置为他们的缺省值。

(7) MySQL可以工作在不同的平台上。支持C、C++、Java、Perl、PHP、Python和TCL API。

数据库逻辑结构设计：

用户表

id	用户的id号
username	用户名
password	用户密码
api_key	用户使用云平台所需要的接口验证序号
state	用户注册后的状态是否激活

设备表

id	设备id
dev_name	设备名称
api_key	用户使用云平台所需要的接口验证序号
sensor_one	设备的第一个传感器的数据
sensor_two	设备的第二个传感器的数据
sensor_three	设备的第三个传感器的数据
sensor_four	设备的第四个传感器的数据
sensor_five	设备的第五个传感器的数据

可视化表

id	可视化的id
dev_id	设备id
sensor_one	设备的第一个传感器的数据
sensor_two	设备的第二个传感器的数据
sensor_three	设备的第三个传感器的数据
sensor_four	设备的第四个传感器的数据
sensor_five	设备的第五个传感器的数据

自动化控制表

id	自动化控制id
watering	浇水状态是否开启
curtain	窗帘状态是否开启
devid	设备id

数据库物理结构设计：

用户表

属性	类型	长度
id	varchar	32
username	varchar	64
password	varchar	64
api_key	varchar	32
state	int	11

设备表

属性	类型	长度
id	varchar	32
dev_name	varchar	64
api_key	varchar	32
sensor_one	varchar	64
sensor_two	varchar	64
sensor_three	varchar	64
sensor_four	varchar	64
sensor_five	varchar	64

可视化表

属性	类型	长度
id	int	11
dev_id	varchar	32
sensor_one	varchar	64
sensor_two	varchar	64
sensor_three	varchar	64
sensor_four	varchar	64
sensor_five	varchar	64

自动化控制表

数据库安全：

数据库的安全措施主要有以下三个方面：

（1）用户标识和鉴定

用户的标识和鉴定提供最外层的安全保护措施，其方法是由系统提供一定的方法让用户标识出自己的名字或身份，系统内部记录了所有的合法用户标识，每次要求进入系统时，由系统进行核实，通过鉴定后才提供使用权。

（2）存取控制

数据库的存取控制就是只确保有资格的用户可以访问数据库的权限，同时令所有未被授权的人无法接近数据。我们采用自主存取控制方式，使得不同的用户有不同的操作权限。

（3）数据加密

为保证用户的个人信息不被盗取，我们对部分数据进行了加密处理。

（4）数据备份

为防止意外故障导致的数据丢失，我们对数据进行备份。

