**植保精灵需求规约文档**

**修订历史：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编写日期 | SEPG | 版本 | 说明 | 作者 |
| 2024.11.11 |  | v1.0 | 完成项目背景、需求调研、功能性需求 | 宋宇然，许经宝 |
| 2024.11.21 |  | v1.1 | 根据杜庆峰老师建议进行修改和扩充 | 宋宇然 |
| 2024.11.23 |  | v1.2 | 加入非功能性需求 | 陈晓坤 |
|  |  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **评审时间** | **评审参与人员** | **评审后修改批准日期** | **确认签字人员** |
| 2024.11.12 | 宋宇然，陈晓坤，许经宝 | 2024.11.12 | 宋宇然，陈晓坤，许经宝 |
| 2024.11.22 | 宋宇然，陈晓坤，许经宝 | 2024.11.22 | 宋宇然，陈晓坤，许经宝 |
| 2024.11. |  | 2024.11. |  |

目录

[一． 需求调研 3](#__RefHeading___Toc1018_120661697)

[1.1背景 3](#__RefHeading___Toc1020_120661697)

[1.2竞品调研 3](#__RefHeading___Toc1022_120661697)

[1.3目标人群 3](#__RefHeading___Toc1024_120661697)

[1.4目标人群特点 3](#__RefHeading___Toc1026_120661697)

[1.5 项目目标 4](#__RefHeading___Toc1028_120661697)

[1.6 系统范围 4](#__RefHeading___Toc1030_120661697)

[二． 功能性需求 5](#__RefHeading___Toc1032_120661697)

[2.1用例图 5](#__RefHeading___Toc1034_120661697)

[2.2用例描述 5](#__RefHeading___Toc1036_120661697)

[2.2.1用户管理子系统(UM) 5](#__RefHeading___Toc1038_120661697)

[2.2.2地块管理子系统(PM) 7](#__RefHeading___Toc1040_120661697)

[2.2.3 气候预告子系统(WD) 8](#__RefHeading___Toc1042_120661697)

[2.2.4智能检测子系统(PD) 8](#__RefHeading___Toc1044_120661697)

[2.2.5 会员服务子系统(MS) 9](#__RefHeading___Toc1046_120661697)

[2.2.6 统计分析子系统(AS) 9](#__RefHeading___Toc1048_120661697)

[三． 非功能性需求 10](#__RefHeading___Toc1050_120661697)

[3.1 易用性需求 10](#__RefHeading___Toc1052_120661697)

1. **需求调研**

**1.1背景**

在当今时代，传统农业正逐步向智能化、精准化转型，以适应资源高效利用与可持续发展的需求。长期以来，农业生产多依赖于经验积累与人力投入，面对复杂多变的病虫害挑战，往往缺乏有效的即时诊断手段。与此同时，随着现代生活条件的不断提升，越来越多的人开始拥有自己的小块土地，用于满足个人的种植爱好和绿色生活的追求。然而，对于这部分种植爱好者而言，缺乏专业的农业知识和经验，常常在面对农作物病虫害等问题时感到束手无策。

在这个背景下，图像识别、传感器等前沿科技为农业种植领域带来了新的曙光。这些技术不仅能够帮助农民实现精准农业，提高生产效率，还能够为种植爱好者提供强有力的支持。通过运用图像识别算法和大模型技术，可以实现对农作物病虫害等问题的快速、准确识别，为种植者提供及时的解决方案。这类农作物识别软件，能够服务于农民、种植爱好者、科研人员和政府机构等多个群体，帮助他们快速识别农作物及其潜在问题，并提出科学合理的解决方案，从而推动农业领域的智能化、精准化发展。

**1.2竞品调研**

经过深入的竞品调研，我们发现现有同类型产品在满足用户需求方面均存在不同程度的缺陷。

首先，以PestNet(美国)为代表的开放社区软件，虽然提供了农业病虫害互助的平台，但诊断结果往往受限于社区用户的经验水平，存在较大的偏差风险。同时，由于问题的响应速度依赖于其他用户的在线时间和活跃度，因此无法保证即时性。此外，社区中的信息质量参差不齐，重复度高，使得有效信息难以快速查找。

还有如加拿大的FieldClimate应用，这类基于实时数据的分析软件，虽然结合了气象站、传感器网络和数据分析功能，为用户提供了更为专业的建议，但大部分功能依赖于特定厂家的传感器等硬件设备，导致初期投入成本高昂。此外，系统涉及多种传感器，要求农户具备一定的技术知识以完成设备的维护和校准，增加了使用难度。

最后就是移动端图像识别软件，德国公司PEAT GmbH的产品 Plantix作为一款成熟的图像识别软件，在检测植物病害方面表现出色，但遗憾的是目前不支持中文，存在阅读障碍。而与之对标的国内软件，如慧植农当家、识农、耘眼等，虽然能够利用图像识别技术检测植物病虫害，但每次使用都需要用户先选择作物再拍照识别，缺乏对于同一用户地块作物的跟踪记录功能，用户体验有待提升。

针对以上问题，我们的系统将基于YOLO图像识别技术，提供离线的即时识别功能。用户仅需拍照即可获取准确的建议，操作简便易用。同时，系统还将可视化跟踪记录用户地块，更好地贴合用户需求。

**1.3目标人群**

农民：从事包括作物的种植、管理和收获在内的农业生产的个体或者家庭。

种植爱好者：有花园等小块土地，爱好种植且在自己的土地上种植植物的爱好者。

农业研究人员：进行农业科学研究的学者，关注作物生长和病虫害管理。

农业管理机构：负责政策制定和农业发展的政府或组织机构。

**1.4目标人群特点**

**农民：**

​ 技术水平：农民群体技术水平差异较大，有些农民具有基本的计算机操作能力，能够使用智能手机和简单的应用程序，而有些农民则可能对现代技术了解较少。同时农民群体对新技术的接受程度也不一，部分农民可能愿意尝试新技术以提高产量，但是也有一些人对技术持保留态度，以来传统方法。

​ 需求与关注点：因为农民群体的技术水平差异较大，所以需要易于操作，直观明了的系统，能够提供简单直接的解决方案。同时也希望能够实时获取作物生长和病虫害的监测数据，以便及时采取管理措施。

**种植爱好者：**

​ 技术水平：很多种植爱好者都刚刚接触种植不久，少有对相关专业知识和技能的了解。但是他们普遍对现代技术接受度较高，且平均受教育程度更高。

​ 需求与关注点：种植爱好者大多并不会投入过多的时间在种植上，往往利用闲暇时间照顾植物。比起产量等经济效益，他们更关心如何让植物生长的更健康。我们的系统应该照顾到非专业人士的需求，为他们提供便捷专业的服务。

**农业研究人员**

​ 技术水平：农业研究人员通常在高校或科研机构工作，专注于农业科学、植物病理学、生态学等领域的研究。关注农业发展趋势，作物生长机制和病虫害防治等问题。

​ 需求与关注点：农业研究人员需要系统提供高质量的数据支持，便于进行研究和发表论文。同时希望系统能够实现长期对作物生长和环境变化的监测，积累数据方便进行深入分析。

**农业管理机构**

​ 角色与职责：包括政府农业部门、农业合作社及相关组织，负责农业政策制定、推广和管理。需要关注农业生产的整体情况，旨在促进农业可持续发展和提高农民收入。

​ 需求与关注点：农业管理机构希望能够通过系统汇总农民的生产数据，以分析整体农业生产趋势和潜在问题。其需要基于数据提供科学的决策支持，制定有效的农业政策和管理措施，希望通过系统对农民进行技术推广和培训，提高他们的技术水平。

**1.5 项目目标**

本项目旨在通过移动端平台，本地部署先进的YOLO图像识别技术，全面辅助农民、农业研究人员及种植爱好者，高效完成农田作物与园艺作物的病虫害精准检测及地块智能监管。我们致力于打造一个低成本、易操作且功能强大的系统，为不同技术水平的用户提供即时、准确的诊断与建议。通过实时跟踪分析与可视化地块监管，帮助用户实现精准农业管理，优化种植效果，同时促进种植知识的交流与分享，共同推动农业领域的智能化、可持续发展。

**1.6 系统范围**

**作物跟踪监测：**

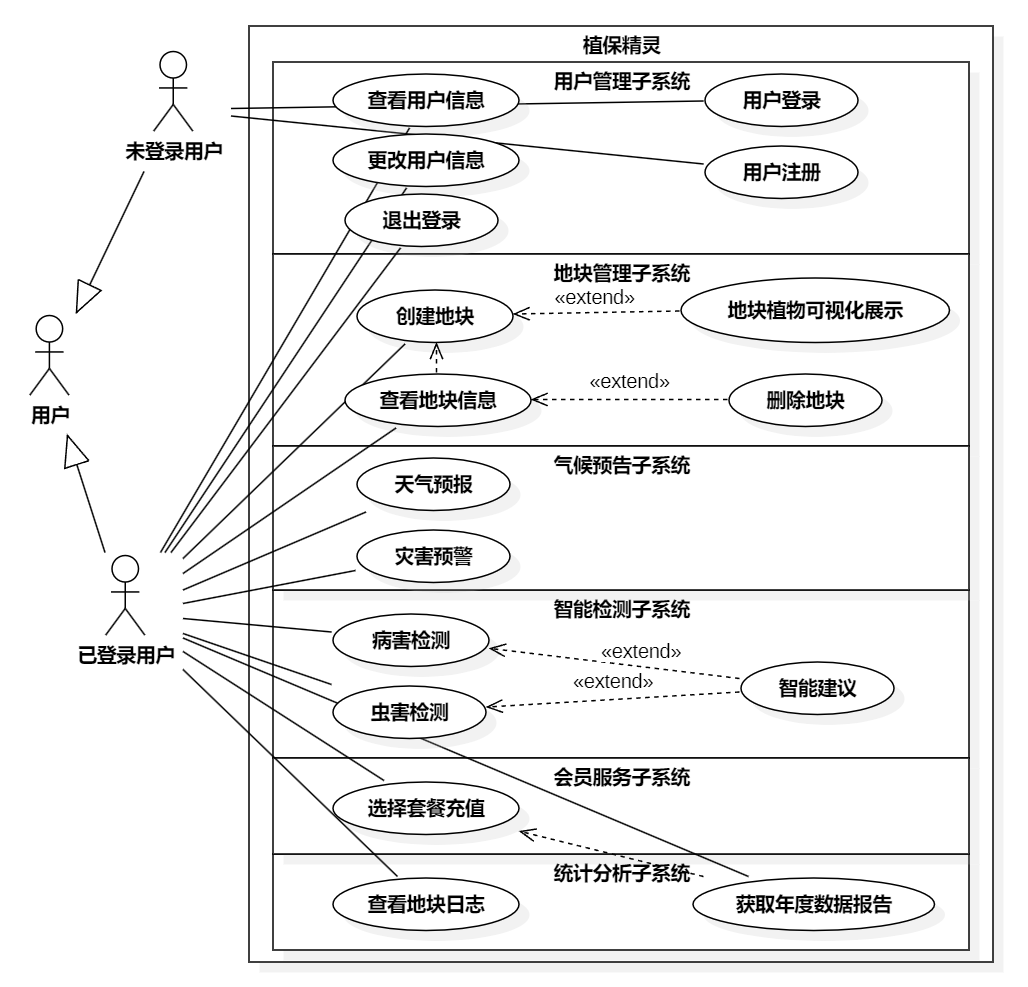
系统提供作物生长阶段跟踪，实时记录作物的生长阶段，如播种、发芽、开花、成熟等，提供生长进度的可视化展示。同时针对作物不同阶段，给出在对应阶段应该做的事情，给作物更好的生长环境。

**病虫害防治：**

通过上传作物的图片到系统上，系统使用相应模型进行图像分析，自动识别作物是否感染病虫害，提供快速准确的识别结果。同时针对不同的病虫害，给出相应的建议和对策，帮助更好地应对病虫害。

**二． 功能性需求**

**2.1用例图**



**2.2用例描述**

**2.2.1用户管理子系统(UM)**

**用例：用户注册(Sign Up)**

**ID:** UM01

**参与者：**未登录用户

**前置条件：**用户未登录，访问注册登录页面

**后置条件：**用户成功注册，账户信息存储在系统中

**基本流程：**

1. 用户点击注册选项，进入注册页面
2. 用户提交用户基本信息(用户名，密码，种植地)
3. 系统将用户信息存储在数据库中
4. 系统向用户发送注册成功的确认信息

**异常流程：**

1. 用户取消注册，点击返回，返回注册登陆页面

**用例：用户登录(Sign In)**

**ID:** UM02

**参与者：**未登录用户

**前置条件：**用户未登录，访问注册登录页面

**后置条件：**用户成功登录，进入系统主界面

**基本流程：**

1. 用户点击登录选项，进入登录页面
2. 用户输入用户名和密码
3. 系统验证用户名和密码
4. 若验证通过，系统将用户重定向至主界面

**异常流程：**

1. 用户名不存在，提示用户名无效，等待用户重新输入
2. 密码错误，提示密码错误，等待用户重新输入
3. 用户取消登录，点击返回，返回注册登录界面

**用例：查看用户信息(Check User Info)**

**ID:** UM03

**参与者：**已登录用户

**前置条件：**用户处于登陆状态

**后置条件：**用户获取个人基本信息和会员信息

**基本流程：**

1. 用户在主界面，点击访问用户信息界面

**用例：更改用户信息(Change User Info)**

**ID:** UM04

**参与者：**已登录用户

**前置条件：**用户处于登陆状态，访问用户信息界面

**后置条件：**用户信息更改成功，返回用户信息界面并显示新的信息

**基本流程：**

1. 用户在用户信息界面点击更改信息
2. 用户在更改信息界面，直接更改原有信息
3. 改完后点击保存
4. 在弹出的窗口点击确定

**异常流程：**

1. 用户更改信息错误，点击了保存，提示错误信息
2. 用户不点保存直接退出，更改取消，不改变原有信息
3. 用户在弹出的窗口点击取消，取消保存，回到更改信息界面继续更改

**用例：退出登录(Log out)**

**ID:** UM05

**参与者：**已登录用户

**前置条件：**用户处于登陆状态

**后置条件：**用户退出登录，返回注册登录界面

**基本流程：**

1. 用户在主界面，点击访问用户信息界面
2. 在用户信息界面，点击退出登录
3. 在弹出的窗口点击确定

**异常流程：**

1. 用户在弹出的窗口点击取消，取消退出登录，弹窗关闭

**2.2.2地块管理子系统(PM)**

**用例：创建地块(Create Plot)**

**ID:** GM01

**参与者：**已登录用户

**前置条件：**用户已登陆

**后置条件：**成功创建地块

**基本流程：**

1. 用户在主界面点击创建地块按钮，进入创建地块界面
2. 用户在创建地块界面输入地块植物基本信息
3. 用户点击创建地块按钮

**异常流程：**

1. 未输入足够的信息时，点击创建地块，提示缺失信息，继续输入信息

**用例：地块植物可视化展示(Plot Visual)**

**ID:** GM02

**参与者：**已登录用户

**前置条件：**用户已登录，有未删除的地块

**后置条件：**显示地块植物生长情况的卡通画面

**基本流程：**

1. 系统读取用户上传的地块植物信息，自动生成地块卡通形象
2. 用户在主界面查看，可以看到所有创建地块的卡通形象

**用例：查看地块信息(Plot Info)**

**ID:** GM03

**参与者：**已登录用户

**前置条件：**用户已登录，有未删除的地块

**后置条件：**系统返回地块的详细信息

**基本流程：**

1. 用户在主界面点击地块图标，进入地块信息页面
2. 系统在数据库中找到此地块详细信息，呈现给用户

**用例：删除地块(Delete Plot)**

**ID:** GM04

**参与者：**已登录用户

**前置条件：**用户已登录，进入地块信息页面

**后置条件：**删除此地块及其所有信息，返回主界面

**基本流程：**

1. 用户在地块信息页面，点击删除地块按钮
2. 在弹窗中选择确定

**异常流程：**

1. 在弹窗中选择取消，弹窗关闭，回到地块信息页面

**2.2.3 气候预告子系统(WD)**

**用例：天气预报(Weather Forecast)**

**ID:** WD01

**参与者：**已登录用户

**前置条件：**用户已登录，设备联网

**后置条件：**显示此地一周的天气和温度信息

**基本流程：**

1. 用户在主界面，点击进入天气界面
2. 系统通过API获取所在地的气象信息，呈现给用户

**异常流程：**

1. 设备未联网，提示未连接互联网

**用例：灾害预警(Disaster Waring)**

**ID:** WD02

**参与者：**已登录用户

**前置条件：**用户已登录，设备联网

**后置条件：**有灾害时提示灾害信息

**基本流程：**

1. 用户在主界面，点击进入天气界面
2. 系统通过API查看近期若有自然灾害，就呈现给用户

**异常流程：**

1. 设备未联网，提示未连接互联网
2. 近期没有自然灾害，显示近期无自然灾害

**2.2.4智能检测子系统(PD)**

**用例：病害检测(Disease Detection)**

**ID:** PD01

**参与者：**已登录用户

**前置条件：**用户已登录，有未删除的地块

**后置条件：**系统返回病害识别结果

**基本流程：**

1. 用户在主界面点击要检测的地块图标，进入地块信息页面
2. 用户点击病害检测按钮，进入病害检测页面
3. 用户上传拍摄的视频，点击开始检测
4. 算法对视频进行截图分析，同时将有效截图上传日志
5. 系统使用病害检测算法处理图片，返回病害的识别结果

**用例：虫害检测(Pest Detection)**

**ID:** PD02

**参与者：**已登录用户

**前置条件：**用户已登录，有未删除的地块

**后置条件：**系统返回虫害识别结果

**基本流程：**

1. 用户在主界面点击要检测的地块图标，进入地块信息页面
2. 用户点击虫害检测按钮，进入虫害检测页面
3. 用户上传拍摄的视频，点击开始检测
4. 算法对视频进行截图分析，同时将有效截图上传日志
5. 系统使用虫害检测算法处理图片，返回虫害的识别结果

**用例：智能建议(Smart Suggest)**

**ID:** PD03

**参与者：**已登录用户

**前置条件：**用户已进行病虫害检测

**后置条件：**系统返回大模型给出的病虫害防治建议

**基本流程：**

1. 用户在检测完病虫害后系统会根据检测结果给出病虫害的防治建议

**2.2.5 会员服务子系统(MS)**

**用例：选择套餐充值(Recharge)**

**ID:** MS01

**参与者：**已登录用户

**前置条件：**用户已登录，进入用户信息界面

**后置条件：**成功充值会员，获得总结次数

**基本流程：**

1. 用户在用户信息页面，滑动选择套餐
2. 用户点击想购买的套餐
3. 用户完成付款

**异常流程：**

1. 用户在付款界面点×，取消付款，返回用户信息页面

**2.2.6 统计分析子系统(AS)**

**用例：查看地块日志(Plot Log)**

**ID:** AS01

**参与者：**已登录用户

**前置条件：**用户已登录，进入地块信息页面

**后置条件：**显示此地块的历史图片及日志信息

**基本流程：**

1. 用户在地块信息页面，点击查看日志按钮
2. 系统调取此地块的历史识别记录，生成地块日志

**异常流程：**

1. 用户暂无日志信息，显示暂无日志，点击返回返回地块信息界面

**用例：获取年度数据报告(Annual Report)**

**ID:** AS02

**参与者：**已登录用户

**前置条件：**用户已登录，且还有总结次数

**后置条件：**显示有记录以来每年的病虫害数据，并给出分析建议

**基本流程：**

1. 用户在主页面，点击获取年度数据报告
2. 在弹窗选择确认生成
3. 系统扣除用户总结次数，进入总结页面
4. 显示有记录以来每年的病虫害数据，并给出分析建议

**异常流程：**

1. 用户暂无信息，显示暂无病虫害数据，点击返回返回地块信息界面
2. 用户总结次数耗尽，提示总结次数已用完，充值获得总结次数

**三． 非功能性需求**

**3.1 易用性需求**

该系统的主要目标用户是农民等农业工作者，农业工作者一般年龄较大，对新技术的熟悉程度较低。为了能让农业工作者快速上手该系统，系统的用户界面必须简单明了，符合常识，易于理解，而无须经过培训或说明。

**3.2 扩展性需求**

该系统的开发采取敏捷开发原则，先从少数几种植物的病虫检测触发，逐步扩展检测的范围并提高检测的精度。该系统应该具有高度可扩展性，使得能够灵活调整系统的组件与功能。系统使用微服务框架，将不同模块之间解耦合。

**3.3 安全性需求**

系统必须要有安全性。具体而言：该系统应该使用加密、匿名化等技术，以及有效的访问控制机制，以保护用户的隐私；该系统应该具有可靠性，使用可靠的服务框架和算法技术，给用户带来稳定的使用体验；该系统应该验证数据的完整性。通过哈希等方法确保数据没有发生损坏，避免带来非预期的影响。

**3.4 响应时间需求**

为了给用户带来更好的使用体验，系统的响应时间应该尽可能地短。为此，系统应该使用微服务架构，将负载均衡分布，提高整体响应能力。同时，使用消息队列和缓存机制，减少重复请求。使用异步框架和批处理请求也能使系统的资源得到最大化利用。