

病虫害识别防治系统

详细设计文档



组长 ：XXX XXXXXXXX

组员 ： XXX XXXXXXXX

XXX XXXXXXXX

XXX XXXXXXXX

目录

[1文档概述 5](#_Toc90735619)

[1.1文档目的和范围 5](#_Toc90735620)

[1.2参考文档 5](#_Toc90735621)

[2系统业务逻辑图 5](#_Toc90735622)

[3功能设计 6](#_Toc90735623)

[3.1登录注册模块 6](#_Toc90735624)

[3.1.1登录注册模块功能定义 6](#_Toc90735625)

[3.1.2登录注册模块结构 7](#_Toc90735626)

[3.2虫害管理模块 7](#_Toc90735627)

[3.2.1虫害管理模块功能定义 7](#_Toc90735628)

[3.2.2虫害管理模块结构 8](#_Toc90735629)

[3.3用户管理模块 8](#_Toc90735630)

[3.3.1用户管理模块功能定义 8](#_Toc90735631)

[3.3.2用户管理模块结构 9](#_Toc90735632)

[3.4虫害报警信息处理管理模块 9](#_Toc90735633)

[3.4.1虫害报警信息处理管理功能定义 9](#_Toc90735634)

[3.4.2虫害报警信息处理管理模块结构 10](#_Toc90735635)

[4数据库设计 10](#_Toc90735636)

[4.1数据库引擎概述 10](#_Toc90735637)

[4.2数据库概要设计 10](#_Toc90735638)

[4.2.1功能实现 10](#_Toc90735639)

[4.2.2 E-R图 11](#_Toc90735640)

[4.2.3关系模式 11](#_Toc90735641)

[4.2.4数据库结构设计 12](#_Toc90735642)

[5接口设计 13](#_Toc90735643)

[5.1界面详细设计 13](#_Toc90735644)

[5.1.1病虫害识别界面 13](#_Toc90735645)

[5.1.2病虫害预防与检索界面 13](#_Toc90735646)

[5.1.3消息通知与处理界面 14](#_Toc90735647)

[5.1.4用户管理界面 14](#_Toc90735648)

[5.2模块间接口设计 14](#_Toc90735649)

[5.2.1登录注册模块 14](#_Toc90735650)

[5.2.2虫害检索模块 16](#_Toc90735651)

[5.2.3虫害报警信息管理模块 19](#_Toc90735652)

[5.2.4用户管理功能模块 20](#_Toc90735653)

# 1文档概述

## 1.1文档目的和范围

对病虫害识别管理系统建立起总体流程及系统总体编码规范等，作为设计人员、编程人员及测试人员工作的基础。

## 1.2参考文档

《需求说明文档》

《概要设计文档》

《系统实现文档》

《用户使用手册》

# 2功能设计

此系统实现的主要功能是基于深度学习的病虫害检测，每个注册的用户可以在系统中进行图片识别，同时还可以检索数据库中的常见虫害信息。管理员在虫害超标时可以收到警告，同时可以在后台进行虫害信息维护与用户信息的维护。此系统根据功能划分以下几个功能模块：登录注册模块、虫害检索模块、用户管理模块、信息管理模块等。

## 2.1登录注册模块

登录注册模块中，包含了用户登录、注册功能。通过此模块的功能，用户可以实现注册系统账号，具体如表2.1所示。

**表2.1 登录注册模块功能定义表**

|  |  |  |
| --- | --- | --- |
| 序号 | 功能点 | 功能点详细说明 |
| 1 | 登录 | 输入账号和密码进行登录，如果账号和密码无法从数据库中匹配则无法登录。 |
| 2 | 注册 | 填写注册信息。注册信息填写包括了用户名、密码、地区。 |

## 2.2害管理模块

虫害管理模块分为两部分，一部分是只有管理员可以使用的更新病虫害的信息与删除新增虫害的信息，另一部分是普通用户和管理员共同可以使用的检索虫害信息功能。具体如表2.2所示。

**表2.2 虫害管理模块功能定义表**

|  |  |  |
| --- | --- | --- |
| 序号 | 功能点 | 功能点详细说明 |
| 1 | 展示所有虫害信息 | 用户打开界面，可以查看所有虫害信息。 |
| 2 | 检索虫害信息 | 用户可以根据虫害名检索它们的属性。 |
| 3 | 更新虫害信息 | 如果虫害的属性有变化，则管理员进行更新。 |
| 4 | 新增虫害信息 | 如果发现新的虫害，则新增虫害信息。 |
| 5 | 删除虫害信息 | 如果某种虫害不再常见，则删除。 |

## 2.3用户管理模块

管理员可以利用此功能，添加用户或者删除用户，也可以更新用户的信息。具体如表2.3所示。

**表2.3 用户管理模块功能定义表**

|  |  |  |
| --- | --- | --- |
| 序号 | 功能点 | 功能点详细说明 |
| 1 | 添加管理员 | 添加新的管理员 |
| 2 | 删除用户 | 移除用户 |
| 3 | 添加用户 | 在后台添加用户 |
| 4 | 更新用户信息 | 更新用户的信息 |

## 2.4虫害报警信息处理管理模块

在进行基于深度学习的虫害检测中，如发现虫害数量超过阈值，则发送告警信息给管理员，管理员可以依次处理该信息。具体如表2.4所示。

**表2.4 报警信息管理模块功能定义表**

|  |  |  |
| --- | --- | --- |
| 序号 | 功能点 | 功能点详细说明 |
| 1 | 查看所有信息 | 查看所有的报警信息 |
| 2 | 更新所有信息 | 经管理员处理后，更新信息的状态。 |

# 3数据库设计

## 3.1数据库引擎概述

MySQL是当今最流行的关系型数据库管理系统之一，在WEB应用方面MySQL是最好的RDBMS：关系数据库管理系统)应用软件之一。由瑞典MySQL AB公司开发，目前属于Oracle公司。MySQL是一种关联数据库管理系统，关联数据库将数据保存在不同的表中，而不是将所有数据放在一个大仓库内，这样就增加了速度并提高了灵活性。由此，本小组使用MySQL作为本系统的数据库。

## 3.2数据库概要设计

### 3.2.1功能实现

1）存储数据，将所需要的数据存入对应数据表中

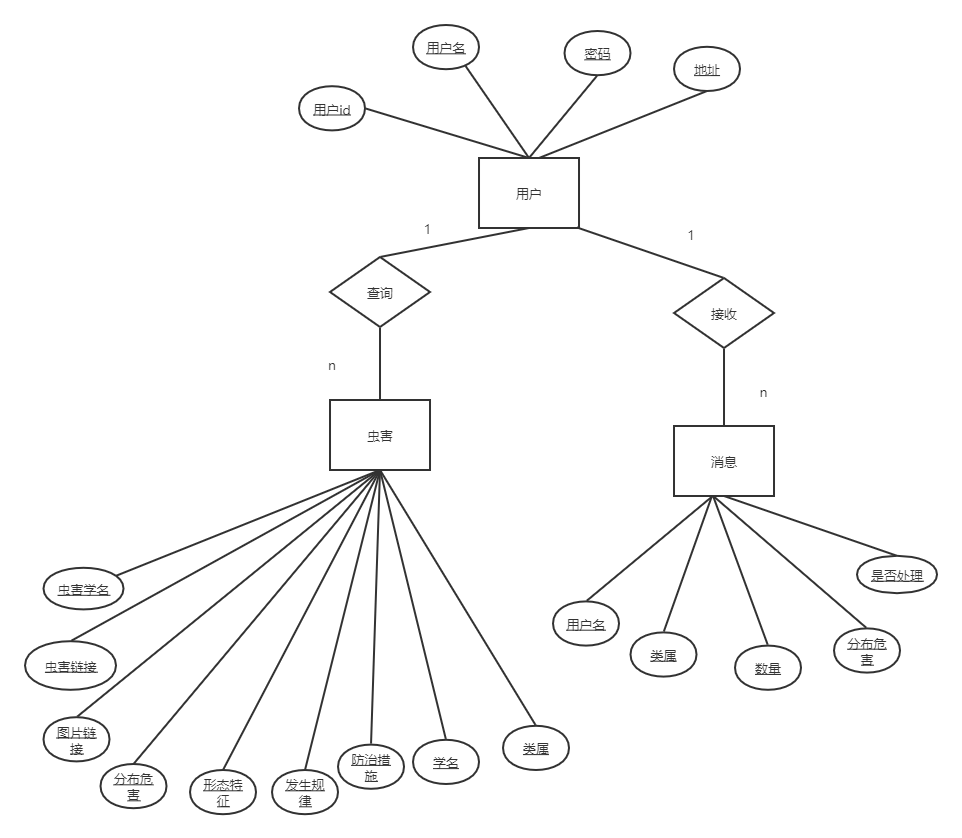
2）更新数据，将需要修改的数据在对应数据表中修改

3）删除数据，将需要删除的数据在对应数据表中删除

4）查找数据，在特定表查找所需要的数据

### 3.2.2 E-R图

项目涉及三个数据表，分别为用户表、病虫信息表和报警消息表，具体如图3-1所示。



**图3-1 E-R图**

### 3.2.3关系模式

1）用户表（- id - username - password - address）;

2）虫害表（- worm\_name - worm\_link - img\_link - location - character - discipline - prevent - worm\_sname - category）;

3）消息表（- username - category - number - location - done）;

### 3.2.4数据库结构设计

用户表如表3.1所示。

**表3.1 用户表**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **用户表** | | | | | |
| 序号 | 字段名 | 字段类型 | 主键 | 属性限定 | 字段说明 |
| 1 | ID | Varchar | 是 | Not null | 编号 |
| 2 | username | Varchar |  | Not null | 用户名 |
| 3 | password | Varchar |  | Not null | 登录密码 |
| 4 | address | Varchar |  | Not null | 地区 |
| 5 | Type | Varchar |  | Not null | 用户身份，管理员还是用户 |

病虫信息表如表3.2所示。

**表3.2 病虫表**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **病虫表** | | | | | |
| 序号 | 字段名 | 字段类型 | 主键 | 属性限定 | 字段说明 |
| 1 | worm\_name | Varchar | 是 | Not null | 虫害名 |
| 2 | worm\_link | Varchar |  |  | 虫害链接 |
| 3 | img\_link | Mediumtext |  |  | 图片链接 |
| 4 | location | Mediumtext |  |  | 分布与危害 |
| 5 | character | Mediumtext |  |  | 形态特征 |
| 6 | discipline | Mediumtext |  |  | 发生规律 |
| 7 | prevent | Mediumtext |  |  | 防治措施 |
| 8 | worm\_sname | Varchar |  |  | 学名 |
| 9 | category | Varchar |  |  | 类属 |

报警信息表如表3.3所示。

**表3.3 报警信息表**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **关系表** | | | | | |
| 序号 | 字段名 | 字段类型 | 主键 | 属性限定 | 字段说明 |
| 1 | username | Varchar |  |  | 用户名 |
| 2 | category | Varchar |  |  | 种属 |
| 3 | number | bool |  |  | 数量 |
| 4 | location | Varchar |  |  | 分布危害 |
| 5 | done | char |  |  | 是否处理 |

# 4接口设计

数据库接口、通信接口和界面接口功能大致相同，以下介绍其中几个函数。

## 4.1登录注册模块

1. loginCheck

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **函数名** | public Object loginCheck | | | |
| **文件名** | logincontroller | | | |
| **功能概要** | 用户名密码登录 | | | |
| **参数** | | | | |
| **类型** | **变量名** | | **I/O** | **说明** |
| String | Username、password | | O | 将这些信息传递给数据库 |
| **返回值** | **类型** | Object | **说明** | |
| **值** | retMsg.Set(MsgType.SUCCESS,user) | 数据传递成功 | |
| retMsg.Set(MsgType.ERROR) | 数据传递失败 | |
| **详细说明** | | | | |
| 验证该用户的密码是否正确 | | | | |

1. checkUserByName

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **函数名** | public Object checkUserByName | | | |
| **文件名** | logincontroller | | | |
| **功能概要** | 注册验证用户是否存在 | | | |
| **参数** | | | | |
| **类型** | **变量名** | | **I/O** | **说明** |
| Qstring | username | | O |  |
| **返回值** | **类型** | Object | **说明** | |
| **值** | retMsg.Set(MsgType.ERROR) | 已经存在该用户，不能创建 | |
| retMsg.Set(MsgType.SUCCESS,pwd) | 可以创建该用户 | |
| **详细说明** | | | | |
| 注册时，如用户名已经存在，则不能创建用户 | | | | |

1. updateUser

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **函数名** | public Object updateUser | | | |
| **文件名** | logincontroller | | | |
| **功能概要** | 插入新的用户 | | | |
| **参数** | | | | |
| **类型** | **变量名** | | **I/O** | **说明** |
| Qstring | username | | I | 将 |
| **返回值** | **类型** | Object | **说明** | |
| **值** | retMsg.Set(MsgType.ERROR) | 注册失败 | |
| retMsg.Set(MsgType.SUCCESS,tmp) | 注册成功 | |
| **详细说明** | | | | |
| 注册新用户 | | | | |

## 4.2虫害检索模块

1. getAllWormData

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **函数名** | public Object getAllWormData | | | |
| **文件名** | Wormcontroller | | | |
| **功能概要** | 展示所有病虫害信息 | | | |
| **参数** | | | | |
| **类型** | **变量名** | | **I/O** | **说明** |
| String | worm\_name、category | | O |  |
| **返回值** | **类型** | Object | **说明** | |
| **值** | retMsg.Set(MsgType.SUCCESS,map) | 成功获取全部信息 | |
|  |  | |

1. updateWormInfo

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **函数名** | public Object updateWormInfo | | | |
| **文件名** | Wormcontroller | | | |
| **功能概要** | 更新虫害数据 | | | |
| **参数** | | | | |
| **类型** | **变量名** | | **I/O** | **说明** |
| String | worm\_name, worm\_sname, category, location, character, discipline, prevent | | I |  |
| **返回值** | **类型** | Bool | **说明** | |
| **值** | retMsg.Set(MsgType.SUCCESS,num) | 更新数据成功 | |
| retMsg.Set(MsgType.ERROR) | 更新数据失败 | |
| **详细说明** | | | | |
| 通过此功能可以更新虫害的数据 | | | | |
| **使用注意事项** | | | | |
| 该功能仅由管理员使用 | | | | |

1. deleteWorm

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **函数名** | public Object deleteWorm | | | |
| **文件名** | Wormcontroller | | | |
| **功能概要** | 删除该虫害的信息 | | | |
| **参数** | | | | |
| **类型** | **变量名** | | **I/O** | **说明** |
| String | worm\_name | | I |  |
| **返回值** | **类型** | Object | **说明** | |
| **值** | retMsg.Set(MsgType.SUCCESS,num) | 删除成功 | |
| retMsg.Set(MsgType.ERROR) | 删除失败 | |
| **详细说明** | | | | |
| 该功能用于删除该虫害的信息 | | | | |
| **使用注意事项** | | | | |
| 该功能仅由管理员使用 | | | | |

1. addWorm

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **函数名** | public Object addWorm | | | |
| **文件名** | Wormcontroller | | | |
| **功能概要** | 添加新的虫害信息 | | | |
| **参数** | | | | |
| **类型** | **变量名** | | **I/O** | **说明** |
| String | worm\_name, worm\_sname, category, img\_link, location, character, discipline, prevent | | I |  |
| **返回值** | **类型** | Object | **说明** | |
| **值** | retMsg.Set(MsgType.SUCCESS,num) | 添加成功 | |
| retMsg.Set(MsgType.ERROR) | 添加失败 | |
| **详细说明** | | | | |
| 该功能用于添加新的虫害信息 | | | | |
| **使用注意事项** | | | | |
| 该功能仅由管理员使用 | | | | |

1. getInfoByName

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **函数名** | public Object getInfoByName | | | |
| **文件名** | Wormcontroller | | | |
| **功能概要** | 按名称查找虫害信息 | | | |
| **参数** | | | | |
| **类型** | **变量名** | | **I/O** | **说明** |
| String | worm\_name | | O |  |
| **返回值** | **类型** | Object | **说明** | |
| **值** | retMsg.Set(MsgType.SUCCESS,dataList); | 查找成功 | |
|  |  | |
| **详细说明** | | | | |
| 该功能用于搜索的虫害信息 | | | | |
| **使用注意事项** | | | | |
| 该功能普通用户与管理员均可使用 | | | | |

## 4.3虫害报警信息管理模块

1. getAllMessage

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **函数名** | getAllMessage | | | |
| **文件名** | Messagecontroller | | | |
| **功能概要** | 管理员查看所有信息 | | | |
| **参数** | | | | |
| **类型** | **变量名** | | **I/O** | **说明** |
| String | worm\_name,address, done | | O |  |
| **返回值** | **类型** | Object | **说明** | |
| **值** | retMsg.Set(MsgType.SUCCESS,map) | 查询成功 | |
|  |  | |
| **详细说明** | | | | |
| 查看所有的信息 | | | | |
| **使用注意事项** | | | | |
| 该功能仅由管理员使用 | | | | |

1. updateDone

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **函数名** | updateDone | | | |
| **文件名** | Messagecontroller | | | |
| **功能概要** | 管理员更新信息处理状态 | | | |
| **参数** | | | | |
| **类型** | **变量名** | | **I/O** | **说明** |
| String | username, address, worm\_name | | I |  |
| **返回值** | **类型** | Object | **说明** | |
| **值** | retMsg.Set(MsgType.SUCCESS,tmp) | 更新成功 | |
| return retMsg.Set(MsgType.ERROR) | 更新失败 | |
| **详细说明** | | | | |
| 更新该消息的处理状态 | | | | |
| **使用注意事项** | | | | |
| 该功能仅由管理员使用 | | | | |

## 4.4用户管理功能模块

1. addAdmin

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **函数名** | addAdmin | | | |
| **文件名** | usercontroller | | | |
| **功能概要** | 添加管理员 | | | |
| **参数** | | | | |
| **类型** | **变量名** | | **I/O** | **说明** |
| String | id,username,password,address,type | | I | 填写管理员信息 |
| **返回值** | **类型** | Object | **说明** | |
| **值** | return retMsg.Set(MsgType.SUCCESS,tmp) | 添加成功 | |
| retMsg.Set(MsgType.ERROR) | 添加失败 | |
| **详细说明** | | | | |
| 增加新的管理员用户 | | | | |
| **使用注意事项** | | | | |
| 该功能仅由管理员使用 | | | | |

1. deleteUser

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **函数名** | deleteUser | | | |
| **文件名** | usercontroller | | | |
| **功能概要** | 删除用户 | | | |
| **参数** | | | | |
| **类型** | **变量名** | | **I/O** | **说明** |
| String | id | | I |  |
| **返回值** | **类型** | Object | **说明** | |
| **值** | return retMsg.Set(MsgType.SUCCESS,tmp) | 删除成功 | |
| retMsg.Set(MsgType.ERROR) | 删除失败 | |
| **详细说明** | | | | |
| 删除普通用户 | | | | |
| **使用注意事项** | | | | |
| 该功能仅由管理员使用 | | | | |

1. getAllUser

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **函数名** | getAllUser | | | |
| **文件名** | usercontroller | | | |
| **功能概要** | 获取全部用户 | | | |
| **参数** | | | | |
| **类型** | **变量名** | | **I/O** | **说明** |
| String | address,type | | O |  |
| **返回值** | **类型** | Object | **说明** | |
| **值** | return retMsg.Set(MsgType.SUCCESS,map) | 获取成功 | |
|  |  | |
| **详细说明** | | | | |
| 查看所有用户信息 | | | | |
| **使用注意事项** | | | | |
| 该功能仅由管理员使用 | | | | |

1. updateUser

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **函数名** | public Object updateUser | | | |
| **文件名** | usercontroller | | | |
| **功能概要** | 添加用户 | | | |
| **参数** | | | | |
| **类型** | **变量名** | | **I/O** | **说明** |
| String | id,username,address | | I | 填写用户信息 |
| **返回值** | **类型** | Object | **说明** | |
| **值** | return retMsg.Set(MsgType.SUCCESS,tmp) | 添加成功 | |
| retMsg.Set(MsgType.ERROR) | 添加失败 | |
| **详细说明** | | | | |
| 增加新的用户 | | | | |
| **使用注意事项** | | | | |
| 该功能仅由管理员使用 | | | | |