

CA387（鸡场智管）养鸡场管理系统 统一强调追踪管理养殖情况

**ChickTrackSystem-Chicken Farm
Management System - Emphasis on
tracking and managing the breeding
situation**

项目编号：CA387

目录

CA387（鸡场智管）养鸡场管理系统—强调追踪管理养殖情况.....	- 1 -
ChickTrackSystem- Chicken Farm Management System - Emphasis on tracking and managing the breeding situation.....	- 1 -
项目编号：CA387.....	- 1 -
CA387（鸡场智管）养鸡场管理系统—强调追踪管理养殖情况.....	- 6 -
1. 主页（Main.php）.....	- 6 -
1.1 仪表板视图：.....	- 6 -
1.2 快捷入口：.....	- 6 -
1.3 消息通知中心：.....	- 7 -
2. 养殖管理.....	- 7 -
2.1 批次管理：.....	- 7 -
2.1.1 批次信息录入与编辑：.....	- 7 -
2.1.2 批次状态监控：.....	- 8 -
2.1.3 批次生命周期管理：.....	- 8 -
2.2 生长记录：.....	- 8 -
2.2.1 日常记录：.....	- 8 -
2.2.2 健康监测：.....	- 8 -
2.3 存栏管理：.....	- 8 -
2.3.1 实时统计：.....	- 8 -
2.3.2 分组与筛选：.....	- 9 -
2.3.3 自动报警：.....	- 9 -
2.4 养殖计划与提醒：.....	- 9 -
2.4.1 计划设置：.....	- 9 -
2.4.2 自动提醒：.....	- 9 -
2.5 出栏管理：.....	- 9 -
2.5.1 出栏记录：.....	- 10 -
2.5.2 质量追溯：.....	- 10 -
3. 饲料管理.....	- 10 -
3.1 饲料采购管理：.....	- 10 -
3.1.1 采购计划：.....	- 10 -
3.1.2 订单管理：.....	- 10 -
3.2 库存管理：.....	- 11 -
3.2.1 实时库存监控：.....	- 11 -
3.2.2 饲料消耗与分析：.....	- 11 -
3.2.3 利用率分析：.....	- 11 -
3.2.4 饲料配方推荐：.....	- 12 -
4. 疫病防控.....	- 12 -
4.1 疫苗管理：.....	- 12 -
4.1.1 疫苗档案：.....	- 12 -
4.1.2 接种记录：.....	- 13 -
4.2 疾病管理：.....	- 13 -
4.2.1 病例记录：.....	- 13 -

4.2.2 治疗方案库:	- 13 -
4.2.3 统计与分析:	- 13 -
4.3 防疫提醒与报警:	- 13 -
5. 环境监测	- 14 -
5.1 温湿度监测:	- 14 -
5.2 空气质量监测:	- 15 -
5.3 通风与照明控制:	- 15 -
5.4 环境数据分析:	- 15 -
6. 销售管理	- 16 -
6.1 销售订单管理:	- 16 -
6.2 客户管理:	- 16 -
6.3 价格管理:	- 17 -
6.4 销售数据统计与报表:	- 17 -
7. 财务管理	- 18 -
7.1 成本核算:	- 18 -
7.2 利润分析:	- 18 -
7.3 支出管理:	- 19 -
8. 设备管理	- 19 -
8.1 设备档案管理:	- 19 -
9. 用户管理	- 20 -
9.1 角色与权限管理:	- 21 -
9.2 用户信息管理:	- 21 -
9.3 操作日志与审计:	- 21 -
10. 报表与数据分析	- 22 -
10.1 定制化报表生成:	- 22 -
10.2 数据可视化:	- 22 -
10.3 决策支持系统:	- 23 -
11. API 接口	- 23 -
11.1 RESTful API 设计:	- 23 -
11.2 移动端及 IoT 对接:	- 24 -
12. 额外考虑	- 24 -
12.1 多终端支持:	- 25 -
12.2 数据库安全与优化:	- 25 -
13. 页面架构设计部分	- 26 -
13.1. 主页 (Main)	- 28 -
13.1.1. `Main.php`	- 28 -
13.1.2. `HomeController.php`	- 28 -
13.2. 养殖管理 (Farm)	- 29 -
13.2.1. `index.php`	- 29 -
13.2.2. `BatchManagement.php`	- 29 -
13.2.3. `GrowthRecords.php`	- 29 -
13.2.4. `StockManagement.php`	- 30 -
13.2.5. `PlanSchedule.php`	- 30 -
13.2.6. `Outgoing.php` (或 `DisposalManagement.php`)	- 30 -

13.3. 饲料管理 (Feed)	- 30 -
13.3.1. `index.php`	- 31 -
13.3.2. `FeedPurchase.php`	- 31 -
13.3.3. `FeedInventory.php`	- 31 -
13.3.4. `FeedConsumption.php`	- 31 -
13.3.5. `FeedFormula.php`	- 31 -
13.4. 疫病防控 (Disease)	- 32 -
13.4.1. `index.php`	- 32 -
13.4.2. `VaccineManagement.php`	- 32 -
13.4.3. `DiseaseManagement.php`	- 32 -
13.4.4. `DeathRecord.php`	- 32 -
13.4.5. `Alert.php`	- 33 -
13.5. 环境监测 (Environment)	- 33 -
13.5.1. `index.php`	- 33 -
13.5.2. `Sensors.php`	- 33 -
13.5.3. `EnvironmentControl.php`	- 34 -
13.5.4. `EnvironmentAlarm.php`	- 34 -
13.6. 销售管理 (Sales)	- 34 -
13.6.1. `index.php`	- 35 -
13.6.2. `SalesOrder.php`	- 35 -
13.6.3. `ClientManagement.php`	- 35 -
13.6.4. `PriceManagement.php`	- 35 -
13.6.5. `SalesReport.php`	- 35 -
13.7. 财务管理 (Finance)	- 36 -
13.7.1. `index.php`	- 36 -
13.7.2. `CostManagement.php`	- 36 -
13.7.3. `ProfitAnalysis.php`	- 36 -
13.7.4. `Expense.php`	- 36 -
13.7.5. `InvoiceManagement.php`	- 37 -
13.8. 设备管理 (Equipment)	- 37 -
13.8.1. `index.php`	- 37 -
13.8.2. `DeviceArchive.php`	- 37 -
13.8.3. `Maintenance.php`	- 37 -
13.8.4. `FaultAlarm.php`	- 38 -
13.9. 用户管理 (User)	- 38 -
13.9.1. `index.php`	- 38 -
13.9.2. `RolePermission.php`	- 38 -
13.9.3. `UserProfile.php`	- 39 -
13.9.4. `OperationLog.php`	- 39 -
13.9.5. `Auth.php`	- 39 -
13.10. 报表与数据分析 (Report)	- 39 -
13.10.1. `index.php`	- 40 -
13.10.2. `ProductionReport.php`	- 40 -
13.10.3. `CostAnalysis.php`	- 40 -

13.10.4. `SalesAnalysis.php`	- 40 -
13.10.5. `DataVisualization.php`	- 40 -
13.11. API 接口 (API)	- 41 -
13.11.1. `index.php`	- 41 -
13.11.2. `AuthApi.php`	- 41 -
13.11.3. `FarmApi.php`	- 41 -
13.11.4. `FeedApi.php`	- 41 -
13.11.5. `More.php` (更多)	- 42 -
14. 数据库设计部分	- 43 -
14.1. 配置文件: `config.php`	- 44 -
14.2. 养殖管理模块:	- 45 -
#14.2.1 批次管理 (BatchManagement)	- 45 -
#14.2.2 生长记录 (GrowthRecords)	- 46 -
# 14.2.3 存栏管理 (StockManagement)	- 47 -
14.3. 饲料管理模块:	- 47 -
# 14.3.1 饲料采购 (FeedPurchase)	- 47 -
# 14.3.2 饲料库存 (FeedInventory)	- 48 -
# 14.3.3 饲料消耗 (FeedConsumption)	- 49 -
14.4. 疫病防控模块:	- 50 -
# 14.4.1 疫苗管理 (VaccineManagement)	- 50 -
# 14.4.2 疾病记录 (DiseaseManagement)	- 51 -
14.5. 环境监测模块:	- 51 -
# 14.5.1 环境数据 (EnvironmentData)	- 51 -
14.6. 销售管理模块:	- 52 -
# 14.6.1 销售订单 (SalesOrder)	- 52 -
14.7. 财务管理模块:	- 53 -
# 14.7.1 财务支出 (Expense)	- 53 -
# 14.7.2 成本核算 (CostCalculation)	- 54 -
14.8. 用户管理模块:	- 55 -
# 14.8.1 用户 (User)	- 55 -
14.9. 设备管理模块:	- 56 -
# 14.9.1 设备档案 (Device)	- 56 -

CA387（鸡场智管）养鸡场管理系统—— 强调追踪管理养殖情况

1. 主页（Main.php）

功能细化：

1.1 仪表板视图：

关键数据统计图表（饼图、折线图、柱状图等），实时显示存栏数、死亡率、出栏率、饲料消耗情况。

重要指标预警图标，如饲料库存低、疫苗接种超期、环境异常等，点击可跳转到对应模块详情页面。

1.2 快捷入口：

大图标按钮快速进入“养殖管理”、“饲料管理”、“疫病防控”等模块。

最近操作记录及待办事项列表（例如当天需完成的饲料采购、疫苗接种、设备维修任务）。

1.3 消息通知中心：

系统自动生成消息通知，如：

当天计划任务（如疫苗接种提醒）。

环境监测数据异常报警。

新增销售订单和财务数据更新。

自定义仪表盘：

用户可通过拖拽组件调整各统计图表的位置，满足个性化需求。

2. 养殖管理

功能细化：

2.1 批次管理：

2.1.1 批次信息录入与编辑：

录入批次的进场日期、品种、供应商、进场数量、初始健康状况等。

每个批次生成唯一编号，便于追踪记录。

2.1.2 批次状态监控:

自动更新存栏数量, 统计出栏、死亡、淘汰等情况。

2.1.3 批次生命周期管理:

定义不同养殖阶段(育雏期、成长期、采蛋期等), 并记录各阶段的关键指标。

2.2 生长记录:

2.2.1 日常记录:

每日体重记录、饲料消耗量、饮水量、活动量记录(如通过视频或传感器监控)。

自动生成生长曲线图, 便于对比历史数据。

2.2.2 健康监测:

录入疫苗接种情况、疾病诊断、治疗方案和恢复情况。

提供健康评分系统, 自动评估鸡群整体健康状况。

2.3 存栏管理:

2.3.1 实时统计:

显示当前每个批次的存栏数量，标识异常（例如：短期内减少明显）。

2.3.2 分组与筛选：

根据品种、批次、区域等条件进行筛选和排序。

2.3.3 自动报警：

当存栏数据低于预设阈值时，系统自动报警提示养殖户检查原因。

2.4 养殖计划与提醒：

2.4.1 计划设置：

设定各批次的预期出栏时间和目标指标。

通过日历形式展现各批次的关键节点（疫苗接种、换饲料、健康检查等）。

2.4.2 自动提醒：

系统根据预设计划自动发送邮件/短信提醒，确保操作不遗漏。

2.5 出栏管理：

2.5.1 出栏记录:

记录每个批次出栏的时间、数量、销售价格、销售渠道。

提供出栏后统计数据，方便后续利润分析和决策优化。

2.5.2 质量追溯:

生成出栏记录单，包含养殖全过程信息，用于产品质量溯源。

3. 饲料管理

功能细化:

3.1 饲料采购管理:

3.1.1 采购计划:

根据日常消耗量和库存数据，系统自动生成饲料采购计划。

支持供应商信息管理和采购合同附件上传。

3.1.2 订单管理:

记录订单号、进货日期、数量、单价、总金额等信息。

支持订单状态（待发货、在途、已到货）的追踪。

3.2 库存管理：

3.2.1 实时库存监控：

饲料进出库数据实时更新。

设定库存预警阈值，当库存低于一定数量时自动提醒。

批次管理：

针对不同种类饲料，按批次记录进货和出库情况，确保质量追踪。

3.2.2 饲料消耗与分析：

日常记录：

记录每日各批次鸡只的饲料消耗数据。

自动计算每只鸡的平均饲料消耗量。

3.2.3 利用率分析：

根据不同生长阶段，分析饲料利用率，自动推荐优化饲料配比。

历史数据对比：

提供多维度历史数据对比，便于发现异常消耗情况。

3.2.4 饲料配方推荐:

配方管理:

存储不同生长阶段推荐的饲料配比。

可根据实际采集的生长数据反馈进行调整。

智能建议:

基于 AI 或规则引擎, 根据历史数据和当前生长状况推荐最优饲料配比。

4. 疫病防控

功能细化:

4.1 疫苗管理:

4.1.1 疫苗档案:

建立疫苗信息库 (名称、厂家、批次、有效期等)。

自动提醒疫苗过期、库存不足情况。

4.1.2 接种记录:

为每批鸡只记录疫苗接种时间、疫苗种类、接种部位、接种人员。

自动计算下次接种时间并发送提醒。

4.2 疾病管理:

4.2.1 病例记录:

详细记录每次疾病的症状、诊断、治疗方案和效果。

支持上传病例照片、实验室检验报告等附件。

4.2.2 治疗方案库:

建立常见疾病治疗方案库，供养殖人员参考和快速处理。

根据地区流行病史进行数据比对和预警。

4.2.3 统计与分析:

自动统计不同疾病的发生率、病死率，生成图表进行趋势分析。

4.3 防疫提醒与报警:

自动提醒:

根据疫苗接种计划和疾病高发期设置提醒机制。

短信或邮件推送防疫知识、紧急措施等信息。

死亡记录管理：

每次死亡记录详细原因（疾病、意外、设备故障等），进行数据归类和统计分析。

形成死亡率、治疗效果等报告，为养殖策略调整提供依据。

5. 环境监测

功能细化：

5.1 温湿度监测：

数据采集：

与物联网传感器对接，实时采集鸡舍内温度、湿度数据。

支持数据历史存储和趋势图展示。

自动控制：

根据预设阈值自动调节风扇、加湿器或除湿机，确保环境稳定。

报警机制：

当温湿度超出正常范围时自动触发报警（声音、短信、APP 通

知等)。

5.2 空气质量监测：

监控参数：

二氧化碳、氨气、甲烷等有害气体浓度检测。

与传感器数据对接，实时显示空气质量指数。

自动通风：

根据检测数据自动开启或调节通风设备，确保良好空气流通。

5.3 通风与照明控制：

自动调节：

根据日照时间、温度和生长阶段自动调节照明时长和亮度。

智能控制通风设备，根据室内外温度差自动调节风速。

手动覆盖：

用户可手动调整通风、照明设置，适应临时需求。

5.4 环境数据分析：

历史数据对比：

提供周期（如日、周、月）的环境数据报告，便于长期趋势分

析。

根据历史数据预测未来可能出现的环境异常，提前调整预案。

6. 销售管理

功能细化：

6.1 销售订单管理：

订单录入与管理：

录入鸡只、鸡蛋或其他产品的销售订单，包含订单号、产品类型、数量、单价、总金额。

支持订单状态（待发货、在途、已完成、退单）更新和自动提醒功能。

允许批量导入和导出订单数据，便于对接第三方系统（如电商平台）。

6.2 客户管理：

客户档案：

建立客户信息库，包括联系人、联系方式、历史订单、信用记录等。

支持按客户区域、订单量、回购率等维度进行客户分组和标签管理。

客户互动记录：

记录每次客户沟通、反馈、投诉及售后服务记录，形成客户跟踪档案。

6.3 价格管理：

动态定价：

根据市场行情、成本数据和竞争对手价格，提供价格调整建议。

设置不同产品的促销活动、折扣和优惠券管理。

价格历史：

记录历史价格变化，形成价格走势图，便于制定长期销售策略。

6.4 销售数据统计与报表：

收入统计：

实时统计每日、每周、每月销售收入和产品销量。

提供多维度数据分析，如产品种类、客户区域、销售渠道统计。

数据可视化:

自动生成销售图表、趋势分析报告,支持导出 PDF 或 Excel 文件。

7. 财务管理

功能细化:

7.1 成本核算:

数据整合:

自动整合饲料、疫苗、人工、设备维护等各项成本数据。

与销售订单、出栏记录数据联动,实现自动成本核算。

分批次分析:

针对每个批次养殖成本进行单独核算,生成成本明细报表。

预算管理:

设定年度、季度和月度预算,并与实际支出进行对比分析。

7.2 利润分析:

盈亏报表:

汇总销售收入与各项支出,自动计算利润率。

支持按时间、批次、产品种类进行利润对比。

财务预警：

当利润率下降或成本异常时自动预警，提示调整养殖策略。

7.3 支出管理：

费用录入：

手动录入水电费、设备维修、运输费用等其他支出。

分类记录各项费用，便于日后统计和税务报表生成。

发票和凭证管理：

上传和存储电子发票、收据、合同扫描件，便于审计和报税。

8. 设备管理

功能细化：

8.1 设备档案管理：

设备信息录入：

录入设备名称、型号、采购日期、供应商、保修期、安装位置等详细信息。

生成设备唯一识别码，便于后续跟踪。

设备状态监控：

显示设备运行状态、使用时间和维护记录，定期检查设备寿命。

设备维护与保养：

维护计划：

根据设备使用说明和历史维护数据，生成定期保养计划（如每月、每季度维护提醒）。

支持保养记录录入、维护人员分配和反馈记录。

故障记录与报警：

当设备出现故障时，记录故障原因、维修方案、维修时长。

自动向设备管理员和维护人员发送故障报警及处理建议。

9. 用户管理

功能细化：

9.1 角色与权限管理:

角色定义:

定义多个用户角色，如系统管理员、养殖管理员、操作人员、财务、技术支持等。

每个角色拥有不同权限和操作范围，支持细粒度权限控制。

权限分配:

通过模块、功能点、数据记录等维度进行权限分配和审核，支持权限模板管理。

9.2 用户信息管理:

用户档案:

录入用户基本信息、联系方式、登录记录、操作日志。

支持头像、签名、偏好设置等个性化定制。

登录与安全管理:

支持多重身份认证（如密码、短信验证码、二次认证）。

自动记录登录失败次数、操作记录，防止恶意操作。

9.3 操作日志与审计:

详细日志记录:

记录用户每次操作的时间、IP 地址、操作内容、修改数据等，

便于追溯和审计。

支持日志数据的检索、导出和归档。

10. 报表与数据分析

功能细化：

10.1 定制化报表生成：

报表类型：

养殖生产报表、饲料使用报表、疫病统计报表、财务盈亏报表、设备运行报表等。

用户可自定义报表模板，设置显示字段和筛选条件。

数据导出：

支持 PDF、Excel、CSV 格式数据导出，便于共享和归档。

10.2 数据可视化：

图表展示：

提供柱状图、折线图、饼图、热力图等多种数据图形展示方式。

通过交互式图表，用户可以点击细分数据，查看详细信息。

实时数据分析：

结合实时数据源和历史数据，进行趋势预测、异常检测、关键指标监控。

10.3 决策支持系统：

智能分析：

基于 AI 算法和数据挖掘，自动识别影响养殖效果的关键因素，提供优化建议。

与外部市场数据对接，预测市场价格趋势和需求变化，为销售策略调整提供依据。

11. API 接口

功能细化：

11.1 RESTful API 设计：

数据接口：

提供各模块（养殖、饲料、疫病、环境、销售、财务等）的数据读写接口。

使用标准 JSON 格式传输数据，支持 GET、POST、PUT、DELETE 操作。

权限认证：

使用 OAuth2、Token 机制进行接口访问认证，确保数据安全。

11.2 移动端及 IoT 对接：

移动应用支持：

提供轻量级 API，方便手机、平板等终端访问和操作。

支持实时数据推送和操作确认。

智能设备对接：

与温湿度传感器、摄像头、自动饲喂系统等 IoT 设备对接，实时传输设备数据并支持远程控制。

提供设备状态 API 和报警接口，方便开发第三方控制系统。

12. 额外考虑

12.1 多终端支持:

设计响应式页面，确保在 PC、手机、平板等不同设备上均有良好体验。

保持简洁的 UI 设计，减少不必要的 JavaScript，主要依赖 PHP 渲染和表单提交。

12.2 数据库安全与优化:

采用 PDO 或 ORM 框架操作数据库，预防 SQL 注入。

数据库设计应考虑分表、索引优化，确保大数据量时系统流畅运行。

系统扩展性:

模块之间松耦合设计，未来可以方便地添加新的功能模块或与外部系统集成。

支持数据备份、恢复及日志归档，保障数据安全和系统稳定性。

用户培训与文档:

为每个模块提供详细的使用文档和培训视频，降低操作门槛。

内置帮助系统和 FAQ，快速解答用户疑问。

13. 页面架构设计部分

下面给出一个针对 PHP 7.1 环境、按“一个模块一个文件夹”思路规划的项目文件结构示例。此结构既符合每个功能模块独立维护的需求，也便于后续扩展和维护。具体命名可根据团队规范和个人习惯做调整，但整体思路大致如下：

项目整体结构

...

├── index.php	项目的入口文件（可选，看你是否需要单入口）
├── config.php	数据库连接、常量定义等配置文件（可根据需求添加）
├── /modules	存放各功能模块的主目录
│ ├── main	1. 主页（Main.php）
│ └── farm	2. 养殖管理

├── feed	3. 饲料管理
├── disease	4. 疫病防控
├── environment	5. 环境监测
├── sales	6. 销售管理
├── finance	7. 财务管理
├── equipment	8. 设备管理
├── user	9. 用户管理
├── report	10. 报表与数据分析
└── api	11. API 接口 (可选)
├── /assets	前端资源(可选): CSS、JS、 图片等
├── /vendor	Composer 依赖目录(如使用 第三方库)
└── .htaccess	Apache 重写规则(如果需要)
...	

> 说明:

- > ``index.php`` 作为单一入口文件，也可以直接通过 ``modules/xxx/xx.php`` 访问。
- > ``config.php`` 用于数据库连接、常量定义、初始化操作等。
- > ``vendor`` 目录存在的前提是你使用 Composer 来管理依赖。
- > 你也可以增加 ``/includes`` 或 ``/classes`` 放公共函数、类库等。

13.1. 主页 (Main)

文件夹: ``/modules/main``

› 用于系统的“仪表盘”(Dashboard)和核心首页功能。

建议的 PHP 文件:

13.1.1. ``Main.php``

系统主页、仪表盘界面，展示各类实时统计和提醒。

入口: ``/modules/main/Main.php``。

也可命名为 ``Dashboard.php``，根据习惯来定。

13.1.2. ``HomeController.php``

如果使用 MVC 思路，可单独建一个控制器文件处理首页逻辑，``Main.php`` 用于渲染页面。

如果只想简单结构，也可以不需要控制器文件。

13.2. 养殖管理 (Farm)

文件夹: `/modules/farm`

› 负责管理鸡只的生命周期、批次、出栏、健康状况等。

可能需要的 PHP 文件:

. 13.2.1. `index.php`

模块首页, 可列出所有子功能的入口链接。

13.2.2. `BatchManagement.php`

批次管理: 进场时间、品种、供应商、数量等。

13.2.3. `GrowthRecords.php`

生长记录: 每日体重、饲料消耗、健康数据等。

13.2.4. `StockManagement.php`

存栏管理：实时统计存栏数量，异常预警。

13.2.5. `PlanSchedule.php`

养殖计划：设定目标出栏时间、周期提醒等。

13.2.6. `Outgoing.php`（或 `DisposalManagement.php`）

出栏管理：记录出栏时间、数量、去向等。

› 以上文件可根据项目需要合并或再细分。例如批次管理和生长记录可以在一个文件，也可以拆分更细。

13.3. 饲料管理（Feed）

文件夹：`/modules/feed`

› 负责饲料采购、库存、配方和日常消耗记录等。

可能需要的 PHP 文件：

13.3.1. `index.php`

模块首页、子功能列表。

13.3.2. `FeedPurchase.php`

饲料采购：供应商信息、采购订单、进货记录等。

13.3.3. `FeedInventory.php`

库存管理：记录饲料进出库、库存余量、预警提醒等。

13.3.4. `FeedConsumption.php`

饲料消耗：每天对各批次鸡只的饲料消耗进行统计与分析。

13.3.5. `FeedFormula.php`

饲料配方：不同生长阶段的饲料配比、配方库等。

13.4. 疫病防控 (Disease)

文件夹：`/modules/disease`

› 负责疫苗管理、疾病诊断、治疗、死亡记录、防疫提醒等。

可能需要的 PHP 文件：

13.4.1. `index.php`

模块首页、子功能列表。

13.4.2. `VaccineManagement.php`

疫苗管理：疫苗档案、接种记录、提醒机制等。

13.4.3. `DiseaseManagement.php`

疾病记录：症状、诊断、治疗方案、治愈情况等。

13.4.4. `DeathRecord.php`

死亡记录：死亡原因、时间、所属批次、自动汇总死亡率分析。

13.4.5. `Alert.php`

防疫提醒：对即将到期的疫苗接种、潜在疫情等进行预警通知。

13.5. 环境监测（Environment）

文件夹：`/modules/environment`

› 负责实时监测鸡舍环境（温度、湿度、空气质量等），并进行自动/手动控制。

可能需要的 PHP 文件：

13.5.1. `index.php`

模块首页、子功能列表。

13.5.2. `Sensors.php`

传感器数据采集、显示与管理，展示温湿度、气体浓度历史和实时数据。

13.5.3. `EnvironmentControl.php`

通风、照明、加湿/除湿等自动控制逻辑。

与物联网设备联动（如果有），也可拆成 `IoTController.php`。

13.5.4. `EnvironmentAlarm.php`

异常预警：如温度过高、空气质量不达标的报警通知。

13.6. 销售管理（Sales）

文件夹：`/modules/sales`

› 负责鸡只/鸡蛋销售订单、客户管理、价格设置、收益统计等。

可能需要的 PHP 文件：

13.6.1. `index.php`

模块首页、子功能列表。

13.6.2. `SalesOrder.php`

销售订单管理：记录订单信息、数量、价格、发货状态等。

13.6.3. `ClientManagement.php`

客户管理：客户档案、历史订单、信用记录、回访等。

13.6.4. `PriceManagement.php`

价格管理：根据市场行情调整售价，设置优惠活动等。

13.6.5. `SalesReport.php`

销售报表：每日/每周/每月销售统计、收入、销售趋势图等。

13.7. 财务管理 (Finance)

文件夹: ``/modules/finance``

› 负责成本核算、利润分析、各项开支记录、财务报表等。

可能需要的 PHP 文件:

13.7.1. ``index.php``

模块首页、子功能列表。

13.7.2. ``CostManagement.php``

成本核算: 饲料、疫苗、人工、设备维护等整合。

13.7.3. ``ProfitAnalysis.php``

利润分析: 汇总收入与成本, 输出盈亏报表。

13.7.4. ``Expense.php``

其他支出管理: 水电费、运输费、杂项开支等记录管理。

13.7.5. `InvoiceManagement.php`

发票、收据等财务凭证记录与管理。

13.8. 设备管理 (Equipment)

文件夹: `/modules/equipment`

› 负责养殖场设备的档案、维护、故障报警等。

可能需要的 PHP 文件:

13.8.1. `index.php`

模块首页、子功能列表。

13.8.2. `DeviceArchive.php`

设备档案: 设备名称、型号、采购日期、保修期、使用状态等。

13.8.3. `Maintenance.php`

维护记录：维护日期、维护人员、维护内容、备件更换等。

13.8.4. `FaultAlarm.php`

故障报警：设备出现故障后自动生成报警信息，通知维护人员处理。

13.9. 用户管理 (User)

文件夹：`/modules/user`

› 负责系统角色与权限、用户档案、操作日志等。

可能需要的 PHP 文件：

13.9.1. `index.php`

模块首页、子功能列表。

13.9.2. `RolePermission.php`

角色管理与权限分配：定义管理员、养殖户、财务、技术等角色。

13.9.3. `UserProfile.php`

用户基本信息管理：查看/编辑用户资料，修改密码等。

13.9.4. `OperationLog.php`

操作日志：记录用户操作行为，提供审计和追溯。

13.9.5. `Auth.php`

登录、退出、会话管理等；也可放在单独的 `/auth` 文件夹中。

13.10. 报表与数据分析 (Report)

文件夹：`/modules/report`

› 负责生成各类综合报表和数据可视化分析。

可能需要的 PHP 文件：

13.10.1. `index.php`

模块首页、子功能列表。

13.10.2. `ProductionReport.php`

生产报表：汇总养殖情况、批次存栏、死亡率、出栏数据等。

13.10.3. `CostAnalysis.php`

成本报表：饲料、疫苗、人工等各项成本占比图表。

13.10.4. `SalesAnalysis.php`

销售与市场分析：销售额、渠道、客户等多维度报表。

13.10.5. `DataVisualization.php`

集中展示仪表盘式图表、趋势分析、预测数据等。

13.11. API 接口 (API)

文件夹: ``/modules/api``

› 如果需要对接移动端或外部系统, 可在此提供 RESTful API 接口。

可能需要的 PHP 文件:

13.11.1. ``index.php``

模块首页/说明文档, 也可直接跳转到 API 文档页面。

13.11.2. ``AuthApi.php``

处理鉴权、Token 发放、刷新 Token 等功能。

13.11.3. ``FarmApi.php``

对外提供养殖管理的数据接口 (GET、POST、PUT、DELETE)。

13.11.4. ``FeedApi.php``

对外提供饲料管理的数据接口。

13.11.5. `More.php` (更多)

根据各模块需要分别提供接口文件，或使用统一入口+路由的方式集中管理。

其他文件或文件夹（可选建议）

`/includes` 或 `/classes`：公共类库、函数文件等，可在此存放 PDO 连接类、工具函数等。

`/templates` 或 `/views`：若使用模板系统（如 Twig、Blade），可在此存放视图文件。

`/logs`：存放日志文件（若不使用数据库记录，也可在服务器日志中查看）。

`.htaccess`：若使用 Apache 并希望进行 URL 重写，可在根目录放置此文件。

总结

1. “一个模块一个文件夹”的基本思路是将功能划分清晰，每个模块都有自己的入口 `index.php`，并按功能需求拆分为若干文件，如 `xxxManagement.php`、`xxxReport.php` 等。
2. 针对 PHP 7.1 环境，可以在代码中利用 `PDO` 来做数据库操作，做好准备语句以防止 SQL 注入。
3. 如果规模逐渐变大，可以考虑引入 MVC 或更完善的框架（如 Laravel、CodeIgniter 等），但初期也可手工维护简单的结构。
4. 文件命名、模块划分都可灵活调整，核心要保证 可读性、可维护性和 扩展性。

14. 数据库设计部分

以下是根据你的设计方案，为养鸡场管理系统设计的数据库表结构。每个功能模块都对应着相关的数据表，考虑到数据库的关系性、索引、数据完整性等方面。使用的是 MySQL 8 的标准 SQL 语法。

14.1. 配置文件：`config.php`（用于存储数据库配置，连接数据库）

```
```php
<?php

$host = 'localhost'; // 数据库主机

$dbname = 'poultry_farm'; // 数据库名称

$username = 'root'; // 用户名

$password = ''; // 密码

$charset = 'utf8mb4'; // 字符集

// 创建 PDO 实例

$dsn = "mysql:host=$host;dbname=$dbname;charset=$charset";

$options = [

 PDO::ATTR_ERRMODE => PDO::ERRMODE_EXCEPTION,

 PDO::ATTR_DEFAULT_FETCH_MODE => PDO::FETCH_ASSOC,

 PDO::ATTR_EMULATE_PREPARES => false,

];

try {

 $pdo = new PDO($dsn, $username, $password, $options);

} catch (\PDOException $e) {

 throw new \PDOException($e->getMessage()),
```

```
(int)$e->getCode());
}
?>
...

```

## 14. 2. 养殖管理模块:

### #14. 2. 1 批次管理 (BatchManagement)

```
```sql  
CREATE TABLE IF NOT EXISTS `batch_management` (  
    `batch_id` INT AUTO_INCREMENT PRIMARY KEY, -- 批次 ID  
    `entry_date` DATE NOT NULL, -- 进场日期  
    `species` VARCHAR(255) NOT NULL, -- 鸡种  
    `supplier` VARCHAR(255), -- 供应商  
    `quantity` INT NOT NULL, -- 进场数量  
    `status` ENUM('active', 'closed', 'disposal') NOT NULL  
    DEFAULT 'active', -- 批次状态  
    `remarks` TEXT, -- 备注  
    `created_at` TIMESTAMP DEFAULT CURRENT_TIMESTAMP -- 创
```

建时间

);

...

#14.2.2 生长记录 (GrowthRecords)

```sql

```
CREATE TABLE IF NOT EXISTS `growth_records` (
 `record_id` INT AUTO_INCREMENT PRIMARY KEY, -- 记录 ID
 `batch_id` INT NOT NULL, -- 批次 ID
 `weight` DECIMAL(5, 2), -- 体重
 `feed_consumed` DECIMAL(5, 2), -- 饲料消耗
 `water_consumed` DECIMAL(5, 2), -- 饮水消耗
 `activity_level` VARCHAR(255), -- 活动水平
 `health_status` VARCHAR(255), -- 健康状况
 `date_recorded` DATE NOT NULL, -- 记录日期
 FOREIGN KEY (`batch_id`) REFERENCES
 `batch_management` (`batch_id`)
);
...
```

### # 14.2.3 存栏管理 (StockManagement)

```
```sql
CREATE TABLE IF NOT EXISTS `stock_management` (
  `stock_id` INT AUTO_INCREMENT PRIMARY KEY, -- 存栏记录
  ID
  `batch_id` INT NOT NULL, -- 批次 ID
  `current_stock` INT NOT NULL, -- 当前存栏数量
  `date_recorded` DATE NOT NULL, -- 记录日期
  FOREIGN KEY (`batch_id`) REFERENCES
  `batch_management` (`batch_id`)
);
```

```

## 14.3. 饲料管理模块:

### # 14.3.1 饲料采购 (FeedPurchase)

```
```sql
```

```
CREATE TABLE IF NOT EXISTS `feed_purchase` (  
    `purchase_id` INT AUTO_INCREMENT PRIMARY KEY, -- 采购  
ID  
    `supplier` VARCHAR(255) NOT NULL, -- 供应商  
    `feed_type` VARCHAR(255) NOT NULL, -- 饲料类型  
    `quantity` DECIMAL(10, 2) NOT NULL, -- 采购数量  
    `price_per_unit` DECIMAL(10, 2) NOT NULL, -- 单价  
    `total_price` DECIMAL(10, 2) NOT NULL, -- 总价  
    `purchase_date` DATE NOT NULL, -- 采购日期  
    `remarks` TEXT -- 备注  
);  
...
```

14.3.2 饲料库存 (FeedInventory)

```
```sql
```

```
CREATE TABLE IF NOT EXISTS `feed_inventory` (
 `inventory_id` INT AUTO_INCREMENT PRIMARY KEY, -- 库存
ID
 `feed_type` VARCHAR(255) NOT NULL, -- 饲料类型
 `quantity` DECIMAL(10, 2) NOT NULL, -- 当前库存数量
```

```

 `purchase_id` INT, -- 关联的采购记录
 `created_at` TIMESTAMP DEFAULT CURRENT_TIMESTAMP, -- 创建时间
 FOREIGN KEY (`purchase_id`) REFERENCES
`feed_purchase` (`purchase_id`)
);
...

```

### # 14.3.3 饲料消耗 (FeedConsumption)

```

```sql
CREATE TABLE IF NOT EXISTS `feed_consumption` (
    `consumption_id` INT AUTO_INCREMENT PRIMARY KEY, -- 消耗记录 ID
    `batch_id` INT NOT NULL, -- 批次 ID
    `quantity` DECIMAL(10, 2) NOT NULL, -- 消耗数量
    `date_recorded` DATE NOT NULL, -- 记录日期
    FOREIGN KEY (`batch_id`) REFERENCES
`batch_management` (`batch_id`)
);
...

```

14. 4. 疫病防控模块:

14. 4. 1 疫苗管理 (VaccineManagement)

```
```sql
CREATE TABLE IF NOT EXISTS `vaccine_management` (
 `vaccine_id` INT AUTO_INCREMENT PRIMARY KEY, -- 疫苗 ID
 `batch_id` INT NOT NULL, -- 批次 ID
 `vaccine_name` VARCHAR(255) NOT NULL, -- 疫苗名称
 `dose` DECIMAL(5, 2), -- 每剂量
 `date_administered` DATE NOT NULL, -- 接种日期
 `administered_by` VARCHAR(255), -- 接种人
 `remarks` TEXT, -- 备注
 FOREIGN KEY (`batch_id`) REFERENCES
`batch_management` (`batch_id`)
);
```
```

14.4.2 疾病记录 (DiseaseManagement)

```
```sql
CREATE TABLE IF NOT EXISTS `disease_management` (
 `disease_id` INT AUTO_INCREMENT PRIMARY KEY, -- 疾病 ID
 `batch_id` INT NOT NULL, -- 批次 ID
 `disease_name` VARCHAR(255), -- 疾病名称
 `diagnosis_date` DATE NOT NULL, -- 诊断日期
 `treatment` TEXT, -- 治疗方案
 `status` ENUM('treated', 'untreated') NOT NULL, -- 治疗
状态
 `remarks` TEXT, -- 备注
 FOREIGN KEY (`batch_id`) REFERENCES
`batch_management` (`batch_id`)
);
```
---
```

14.5. 环境监测模块:

14.5.1 环境数据 (EnvironmentData)

```
```sql
CREATE TABLE IF NOT EXISTS `environment_data` (
 `data_id` INT AUTO_INCREMENT PRIMARY KEY, -- 数据 ID
 `temperature` DECIMAL(5, 2), -- 温度
 `humidity` DECIMAL(5, 2), -- 湿度
 `co2_level` DECIMAL(5, 2), -- CO2 浓度
 `ammonia_level` DECIMAL(5, 2), -- 氨气浓度
 `timestamp` TIMESTAMP DEFAULT CURRENT_TIMESTAMP -- 数据
记录时间
);
```
---
```

14.6. 销售管理模块:

14.6.1 销售订单 (SalesOrder)

```
```sql
CREATE TABLE IF NOT EXISTS `sales_order` (
 `order_id` INT AUTO_INCREMENT PRIMARY KEY, -- 订单 ID
```

```

`customer_name` VARCHAR(255) NOT NULL, -- 客户名称
`product_type` VARCHAR(255) NOT NULL, -- 产品类型（鸡只、
鸡蛋）
`quantity` INT NOT NULL, -- 销售数量
`price_per_unit` DECIMAL(10, 2) NOT NULL, -- 单价
`total_price` DECIMAL(10, 2) NOT NULL, -- 总价
`order_date` DATE NOT NULL, -- 订单日期
`status` ENUM('pending', 'completed', 'cancelled') NOT
NULL DEFAULT 'pending', -- 订单状态
`remarks` TEXT -- 备注
);
...

```

## 14.7. 财务管理模块:

### # 14.7.1 财务支出 (Expense)

```

```sql
CREATE TABLE IF NOT EXISTS `expense` (
    `expense_id` INT AUTO_INCREMENT PRIMARY KEY, -- 支出 ID
```

```

    `expense_type` VARCHAR(255) NOT NULL, -- 支出类型（如水
电费、设备维护费）
    `amount` DECIMAL(10, 2) NOT NULL, -- 金额
    `date_incurred` DATE NOT NULL, -- 发生日期
    `remarks` TEXT -- 备注
);
...

```

14.7.2 成本核算 (CostCalculation)

```

```sql
CREATE TABLE IF NOT EXISTS `cost_calculation` (
 `cost_id` INT AUTO_INCREMENT PRIMARY KEY, -- 成本 ID
 `batch_id` INT NOT NULL, -- 批次 ID
 `feed_cost` DECIMAL(10, 2), -- 饲料成本
 `vaccine_cost` DECIMAL(10, 2), -- 疫苗成本
 `labor_cost` DECIMAL(10, 2), -- 人工成本
 `other_cost` DECIMAL(10, 2), -- 其他成本
 `total_cost` DECIMAL(10, 2), -- 总成本
 FOREIGN KEY (`batch_id`) REFERENCES
`batch_management` (`batch_id`)

```

```
);


```

## 14.8. 用户管理模块:

### # 14.8.1 用户 (User)

```
```sql  
CREATE TABLE IF NOT EXISTS `users` (  
    `user_id` INT AUTO_INCREMENT PRIMARY KEY, -- 用户 ID  
    `username` VARCHAR(255) NOT NULL UNIQUE, -- 用户名  
    `password` VARCHAR(255) NOT NULL, -- 密码 (存储加密后)  
    `role` ENUM('admin', 'manager', 'worker') NOT NULL, --  
    用户角色  
    `email` VARCHAR(255), -- 邮箱  
    `phone` VARCHAR(15), -- 手机号码  
    `created_at` TIMESTAMP DEFAULT CURRENT_TIMESTAMP -- 创  
    建时间  
);  
```
```

----

## 14.9. 设备管理模块:

### # 14.9.1 设备档案 (Device)

```
```sql
CREATE TABLE IF NOT EXISTS `devices` (
    `device_id` INT AUTO_INCREMENT PRIMARY KEY, -- 设备 ID
    `device_name` VARCHAR(255) NOT NULL, -- 设备名称
    `device_type` VARCHAR(255) NOT NULL, -- 设备类型
    `supplier` VARCHAR(255), -- 供应商
    `purchase_date` DATE, -- 采购日期
    `warranty_period` INT, -- 保修期 (月)
    `status` ENUM('active', 'inactive', 'maintenance') NOT
NULL DEFAULT 'active', -- 状态
    `location` VARCHAR(255) -- 设备位置
);
```
```

----

## 总结

这些 SQL 表设计涵盖了养鸡场管理系统的各个模块，包括批次管理、饲料管理、疫病防控、销售管理、财务管理、用户管理等。根据需要，进一步可以加入索引、优化查询等内容，确保系统在实际运行中表现良好。

叶梓阳 总经理

广东知周数字科技有限公司

官网 [www.caffz.com](http://www.caffz.com)

13826867328