

CA393 智研 LIMS 实验室信息管理系统

SmartLab LIMS -

Laboratory Information Management System

项目编号：CA393

目录

CA393 智研 LIMS 实验室信息管理系统.....	1
项目编号: CA393.....	1
智研 LIMS 实验室信息管理系统.....	3
1.功能设计.....	3
1.1. 用户与权限管理.....	3
1.2. 样本管理.....	3
1.3. 检测与实验管理.....	3
1.4. 设备与仪器管理.....	4
1.5. 试剂与物料管理.....	4
1.6. 质量控制与标准管理.....	4
1.7. 数据统计与报表.....	5
1.8. 业务流程管理.....	5
1.9. 客户与订单管理（适用于第三方检测机构）.....	5
1.10. 系统管理.....	6
2.PHP 程序架构.....	6
3.mysql 数据库架构.....	9
3.1 用户与权限管理（users, roles, permissions）.....	10
3.2 样本管理（samples）.....	11
3.3 检测与实验管理（experiments, experiment_results）.....	12
3.4 设备与仪器管理（equipment）.....	14
3.5 试剂与物料管理（reagents）.....	15
3.6 质量控制（quality_control）.....	15
3.7 数据统计与报表（reports）.....	16
3.8 系统日志与操作记录（audit_logs）.....	17

智研 LIMS 实验室信息管理系统

1. 功能设计

1.1. 用户与权限管理

用户管理（创建、编辑、删除用户）

角色与权限（不同级别的访问权限，如管理员、实验员、审核员）

登录认证（基于账号密码，支持双因素认证）

操作日志（记录用户操作，方便审计）

1.2. 样本管理

样本登记（录入样本编号、类别、来源、接收时间等）

样本存储（存储位置管理，如冰箱编号、层架号）

样本流转（记录样本在实验过程中的移动）

样本状态（状态跟踪，如“待检测”、“检测中”、“已完成”）

样本销毁（实验后样本处理）

1.3. 检测与实验管理

检测项目定义（配置不同的检测项目，如生化、微生物、物理化学）

实验流程管理（实验步骤、SOP 标准操作规程）

实验任务分配（给实验员分配实验任务）

实验记录（实验过程、数据记录、异常情况）

数据校验（自动检测数据异常）

1.4. 设备与仪器管理

设备台账（登记设备信息，如品牌、型号、购买日期）

设备维护（定期维护提醒）

设备校准（记录设备的校准情况）

设备使用记录（实验员使用设备的记录）

1.5. 试剂与物料管理

试剂库存管理（库存数量、批号、有效期）

试剂采购（采购记录、供应商管理）

试剂领用（试剂的领取、使用、归还）

危险品管理（特殊试剂的安全管理）

1.6. 质量控制与标准管理

标准方法管理（存储实验标准方法，如 ISO、GB、FDA 等）

质控样本管理（用于校验实验的标准样本）

实验室审核（审核实验数据，确保合规性）

质量评估（实验误差分析，数据重复性分析）

1.7. 数据统计与报表

实验数据分析（样本检测数据的趋势分析）

设备使用统计（设备的使用频率、故障率）

库存使用报表（试剂、样本的消耗情况）

报告生成（支持 PDF、Excel 导出）

1.8. 业务流程管理

实验任务流程（从样本接收到检测完成的流程）

实验审批流程（实验数据的审核与批准）

客户报告管理（为外部客户提供检测报告）

自动通知（通过邮件或短信通知实验进度）

1.9. 客户与订单管理（适用于第三方检测机构）

- 客户管理（记录客户信息）
- 订单管理（客户提交的检测订单）
- 合同与报价（客户检测服务的收费管理）
- 账单管理（客户付款、发票管理）

1. 10. 系统管理

- 系统配置（实验室名称、Logo、时区等）
- 数据备份与恢复（定期备份数据库）
- 日志管理（系统操作日志）
- API 接口（与其他系统对接，如 ERP、MES）

2. PHP 程序架构

```
/lims
|—— /app
|   |—— /controllers
|   |   |—— AuthController.php
|   |   |—— DashboardController.php
|   |   |—— SampleController.php
|   |   |—— ExperimentController.php
|   |   |—— EquipmentController.php
```

```
|   |   | — ReagentController.php
|   |   | — QualityControlController.php
|   |   | — ReportController.php
|   |   | — WorkflowController.php
|   |   | — UserController.php
|   |   | — SystemController.php
|   |
|   | — /models
|   |   | — User.php
|   |   | — Sample.php
|   |   | — Experiment.php
|   |   | — Equipment.php
|   |   | — Reagent.php
|   |   | — QualityControl.php
|   |   | — Report.php
|   |   | — Workflow.php
|   |   | — System.php
|   |
|   | — /views
|   |   | — /auth
|   |   |   | — login.php
|   |   |   | — register.php
|   |   | — /dashboard
|   |   |   | — index.php
|   |   | — /samples
|   |   |   | — index.php
|   |   |   | — create.php
|   |   |   | — edit.php
|   |   |   | — view.php
```

```
|   |   | — /experiments
|   |   |   | — index.php
|   |   |   | — create.php
|   |   |   | — edit.php
|   |   |   | — view.php
|   |   | — /equipment
|   |   |   | — index.php
|   |   |   | — create.php
|   |   |   | — edit.php
|   |   |   | — view.php
|   |   | — /reports
|   |   |   | — index.php
|   |   |   | — generate.php
|   |   | — /users
|   |   |   | — index.php
|   |   |   | — create.php
|   |   |   | — edit.php
|
| — /config
|   | — config.php
|   | — database.php
|
| — /core
|   | — Controller.php
|   | — Model.php
|   | — View.php
|   | — Database.php
|
| — /public
```

```
|   | — index.php
|   | — /css
|   |   | — style.css
|   | — /js
|   |   | — script.js
|   | — /uploads
|
| — /routes
|   | — web.php
|
| — /storage
|   | — logs/
|   | — backups/
|
| — .htaccess
| — composer.json
| — README.md
```

3. mysql 数据库架构

以下是 LIMS（实验室信息管理系统）适用于 MySQL 8 的数据库架构设计，包括主要的数据库表及字段。

数据库名称: lims_db

```
```sql
CREATE DATABASE lims_db CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci;
USE lims_db;
```
```

3.1 用户与权限管理 (users, roles, permissions)

用户表 (users)

```
```sql
CREATE TABLE users (
 id INT PRIMARY KEY AUTO_INCREMENT,
 username VARCHAR(50) NOT NULL UNIQUE,
 password VARCHAR(255) NOT NULL,
 email VARCHAR(100) NOT NULL UNIQUE,
 full_name VARCHAR(100),
 role_id INT NOT NULL,
 created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
 updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP,
 FOREIGN KEY (role_id) REFERENCES roles(id) ON DELETE CASCADE
);
```
```

角色表 (roles)

```
```sql
CREATE TABLE roles (
 id INT PRIMARY KEY AUTO_INCREMENT,
```

```
role_name VARCHAR(50) NOT NULL UNIQUE,
description TEXT
);
...
```

权限表 (permissions)

```
```sql  
CREATE TABLE permissions (  
    id INT PRIMARY KEY AUTO_INCREMENT,  
    permission_name VARCHAR(50) NOT NULL UNIQUE,  
    description TEXT  
);  
...
```

角色权限关联表 (role_permissions)

```
```sql  
CREATE TABLE role_permissions (
 role_id INT,
 permission_id INT,
 PRIMARY KEY (role_id, permission_id),
 FOREIGN KEY (role_id) REFERENCES roles(id) ON DELETE CASCADE,
 FOREIGN KEY (permission_id) REFERENCES permissions(id) ON DELETE CASCADE
);
...
```

## 3.2 样本管理 (samples)

样本表 (samples)

```
```sql
```

```
CREATE TABLE samples (  
    id INT PRIMARY KEY AUTO_INCREMENT,  
    sample_code VARCHAR(50) NOT NULL UNIQUE,  
    sample_name VARCHAR(100) NOT NULL,  
    sample_type VARCHAR(50) NOT NULL,  
    collected_by INT NOT NULL,  
    received_by INT NOT NULL,  
    received_date DATETIME NOT NULL,  
    storage_location VARCHAR(100),  
    status ENUM('待检测', '检测中', '已完成', '已销毁') DEFAULT '待检测',  
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
    updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE  
CURRENT_TIMESTAMP,  
    FOREIGN KEY (collected_by) REFERENCES users(id) ON DELETE SET NULL,  
    FOREIGN KEY (received_by) REFERENCES users(id) ON DELETE SET NULL  
);  
```
```

### 3.3 检测与实验管理 (experiments, experiment\_results)

实验表 (experiments)

```
```sql
```

```
CREATE TABLE experiments (  

```

```

id INT PRIMARY KEY AUTO_INCREMENT,
experiment_code VARCHAR(50) NOT NULL UNIQUE,
sample_id INT NOT NULL,
assigned_to INT NOT NULL,
experiment_type VARCHAR(100),
start_date DATETIME,
end_date DATETIME,
status ENUM('待处理', '进行中', '已完成') DEFAULT '待处理',
remarks TEXT,
created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
updated_at  TIMESTAMP  DEFAULT  CURRENT_TIMESTAMP  ON  UPDATE
CURRENT_TIMESTAMP,
FOREIGN KEY (sample_id) REFERENCES samples(id) ON DELETE CASCADE,
FOREIGN KEY (assigned_to) REFERENCES users(id) ON DELETE SET NULL
);
...

```

实验结果表（experiment_results）

```

```sql
CREATE TABLE experiment_results (
 id INT PRIMARY KEY AUTO_INCREMENT,
 experiment_id INT NOT NULL,
 result_data TEXT NOT NULL,
 result_status ENUM('正常', '异常') DEFAULT '正常',
 recorded_by INT NOT NULL,
 recorded_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
 FOREIGN KEY (experiment_id) REFERENCES experiments(id) ON DELETE
CASCADE,
 FOREIGN KEY (recorded_by) REFERENCES users(id) ON DELETE SET NULL

```

---

);

...

### 3.4 设备与仪器管理 (equipment)

设备表 (equipment)

```sql

```
CREATE TABLE equipment (  
    id INT PRIMARY KEY AUTO_INCREMENT,  
    equipment_code VARCHAR(50) NOT NULL UNIQUE,  
    equipment_name VARCHAR(100) NOT NULL,  
    model VARCHAR(50),  
    manufacturer VARCHAR(100),  
    purchase_date DATE,  
    last_calibration DATE,  
    next_calibration DATE,  
    status ENUM('正常', '维修', '报废') DEFAULT '正常',  
    location VARCHAR(100),  
    assigned_to INT,  
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
    updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE  
CURRENT_TIMESTAMP,  
    FOREIGN KEY (assigned_to) REFERENCES users(id) ON DELETE SET NULL  
);  
...
```

3.5 试剂与物料管理 (reagents)

试剂库存表 (reagents)

```
```sql
CREATE TABLE reagents (
 id INT PRIMARY KEY AUTO_INCREMENT,
 reagent_code VARCHAR(50) NOT NULL UNIQUE,
 reagent_name VARCHAR(100) NOT NULL,
 manufacturer VARCHAR(100),
 batch_number VARCHAR(50),
 quantity INT NOT NULL DEFAULT 0,
 unit VARCHAR(20),
 expiry_date DATE,
 storage_location VARCHAR(100),
 status ENUM('可用', '过期', '损坏') DEFAULT '可用',
 created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
 updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP
);
```
```

3.6 质量控制 (quality_control)

质量控制记录表 (quality_control)

```
```sql
```

```
CREATE TABLE quality_control (
 id INT PRIMARY KEY AUTO_INCREMENT,
 sample_id INT NOT NULL,
 experiment_id INT NOT NULL,
 qc_type VARCHAR(50) NOT NULL,
 qc_status ENUM('合格', '不合格') DEFAULT '合格',
 reviewed_by INT NOT NULL,
 reviewed_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
 FOREIGN KEY (sample_id) REFERENCES samples(id) ON DELETE CASCADE,
 FOREIGN KEY (experiment_id) REFERENCES experiments(id) ON DELETE
CASCADE,
 FOREIGN KEY (reviewed_by) REFERENCES users(id) ON DELETE SET NULL
);
...
```

## 3.7 数据统计与报表 (reports)

报告表 (reports)

```
```sql
```

```
CREATE TABLE reports (  
    id INT PRIMARY KEY AUTO_INCREMENT,  
    report_code VARCHAR(50) NOT NULL UNIQUE,  
    experiment_id INT NOT NULL,  
    generated_by INT NOT NULL,  
    report_content TEXT NOT NULL,  
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
    FOREIGN KEY (experiment_id) REFERENCES experiments(id) ON DELETE
```



```
CASCADE,  
    FOREIGN KEY (generated_by) REFERENCES users(id) ON DELETE SET NULL  
);  
...
```

3.8 系统日志与操作记录 (audit_logs)

系统日志表 (audit_logs)

```
```sql  
CREATE TABLE audit_logs (
 id INT PRIMARY KEY AUTO_INCREMENT,
 user_id INT NOT NULL,
 action VARCHAR(255) NOT NULL,
 table_name VARCHAR(100),
 record_id INT,
 timestamp TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
 FOREIGN KEY (user_id) REFERENCES users(id) ON DELETE SET NULL
);
...
```

这套 LIMS MySQL8 数据架构:

- 使用 InnoDB, 支持事务
- 严格外键关联, 防止数据孤立
- 日志记录可追溯
- 支持权限管理
- 数据冗余最小化

叶梓阳 总经理

[WWW.CAFFZ.COM](http://WWW.CAFFZ.COM)

广东知周数字科技有限公司