CA386p茶易管（TeaEase）奶茶门店连锁管理系统设计方案PrivateVersion（PHP7.1+MYSQL8）

CA386p TeaEase - Milk tea store chain management system Private Version

项目编号：CA386p

目录

[CA391茶易管（TeaEase）奶茶门店连锁管理系统设计方案（PHP7.1+MYSQL8） 1](#_Toc190438968)

[CA391 TeaEase - Milk tea store chain management system 1](#_Toc190438969)

[项目编号：CA391 1](#_Toc190438970)

[1. 背景与目标 4](#_Toc190438971)

[2. 技术选型分析 4](#_Toc190438972)

[3. 系统需求与功能概述 4](#_Toc190438973)

[4. 资源需求与投入评估 5](#_Toc190438974)

[5. 风险与对策 5](#_Toc190438975)

[6. 预期收益与效益分析 6](#_Toc190438976)

[7.设计提示 6](#_Toc190438977)

[8、核心功能模块： 6](#_Toc190438978)

[8.1. 门店管理 6](#_Toc190438979)

[8.2. 销售管理 7](#_Toc190438980)

[8.3. 供应链管理 7](#_Toc190438981)

[8.4. 财务统计 7](#_Toc190438982)

[8.5. 用户与权限管理 7](#_Toc190438983)

[9．扩展功能模块： 8](#_Toc190438984)

[9.1. 数据可视化模块 8](#_Toc190438985)

[9.2. 多语言与多区域支持模块 8](#_Toc190438986)

[9.3. API接口模块 8](#_Toc190438987)

[9.4. 通知与提醒模块 8](#_Toc190438988)

[9.5. 系统设置与维护模块 9](#_Toc190438989)

[10.PHP程序文件架构 9](#_Toc190438990)

[11、MYSQL数据库设计 12](#_Toc190438991)

[11.1. 门店管理 (store) 13](#_Toc190438992)

[11.2. 销售管理 (sales) 14](#_Toc190438993)

[11.3. 库存管理 (inventory) 16](#_Toc190438994)

[11.4. 财务报表 (financial) 16](#_Toc190438995)

[11.5. 用户与权限管理 (user & permissions) 17](#_Toc190438996)

[11.6. 促销活动 (promotion) 18](#_Toc190438997)

**茶易管（TeaEase）奶茶门店连锁管理系统**

# 1. 背景与目标

随着奶茶行业的持续扩张，传统的单门店管理模式已经难以满足快速扩展的需求。构建一套统一、高效的连锁门店管理系统，可以为多门店提供集中管理、提高运营效率，同时为决策层提供更全面的数据支持。

# 2. 技术选型分析

 编程语言与框架：PHP 7.1

 优势：PHP 7.1 相较于早期版本，性能提升显著，同时具备稳定性和较大的社区支持。

 可用性：市场上已有丰富的PHP开发人员资源，降低了开发成本。

 数据库：MySQL 8

 优势：支持事务、视图、触发器等高级功能，性能优化和安全性显著提升。

 扩展性：支持分区、分布式部署，能够满足未来可能的业务增长需求。

# 3. 系统需求与功能概述

 核心模块：

 门店管理：门店信息录入与维护

 销售管理：订单记录、会员积分管理、促销活动支持

 供应链管理：原材料采购、库存监控

 财务统计：销售报表、成本分析

 权限管理：角色划分与操作权限控制

 扩展功能：

 数据可视化：提供多维度图表，辅助门店运营分析

 多语言支持：适应不同地区门店需求

 移动端支持：开发API接口，支持APP或小程序接入

# 4. 资源需求与投入评估

 人员：

 1名项目经理

 2名PHP后端开发工程师

 1名前端工程师

 1名数据库管理员

 硬件需求：

 开发环境：普通开发者工作站即可

 测试与生产环境：一台独立服务器或云主机，具备稳定网络连接和适当的存储空间

 时间成本：

 需求分析与设计：2周

 开发周期：810周

 测试与上线：2周

# 5. 风险与对策

 技术风险：

 早期设计不当导致后期扩展困难

 数据库性能瓶颈

 对策：采用MVC架构，进行代码模块化设计；优化数据库查询，必要时使用索引与分区技术

 团队协作风险：

 成员分工不明确，进度延迟

 对策：明确项目职责和时间表，定期同步进展

 运营风险：

 系统上线后运行不稳定

 对策：上线前进行充分测试，预留维护和调整时间

# 6. 预期收益与效益分析

 运营效率提升： 集中化管理使得门店间的资源调配更灵活，减少重复劳动。

 数据支持决策： 实时数据统计和报表帮助管理层快速了解运营情况，及时调整策略。

 客户满意度提高： 标准化的会员积分、促销活动等功能增强客户粘性，促进消费增长。

# 7.设计提示

基于PHP 7.1和MySQL 8的技术选型，以及清晰的功能模块划分和风险控制措施，该奶茶门店连锁管理系统的开发是可行的。该项目能够通过技术手段提升管理效率，为企业的连锁化发展提供强有力的支持。

# 8、核心功能模块：

## 8.1. 门店管理

 门店信息登记与维护

 门店营业状态管理

 门店间资源调配

## 8.2. 销售管理

 销售订单处理

 收银管理与结算

 会员积分管理

 促销活动创建与跟踪

## 8.3. 供应链管理

 原材料采购管理

 供应商信息维护

 库存监控与预警

 物料消耗统计

## 8.4. 财务统计

 销售报表生成

 成本核算与利润分析

 多维度财务数据展示

## 8.5. 用户与权限管理

 用户角色划分

 权限组与功能分配

 操作日志与安全审计

# 9．扩展功能模块：

## 9.1. 数据可视化模块

 销售趋势图表

 库存状态图表

 门店业绩排行榜

## 9.2. 多语言与多区域支持模块

 支持不同地区门店设置本地语言

 支持区域化促销活动与价格设置

## 9.3. API接口模块

 对接移动端APP

 对接第三方营销工具

 提供开放接口，便于后期集成

## 9.4. 通知与提醒模块

 库存不足提醒

 会员到期提醒

 促销活动即将开始或结束提醒

## 9.5. 系统设置与维护模块

 系统参数设置

 数据备份与恢复

系统升级与更新管理

# 10.PHP程序文件架构

root/

├── config/ # 配置文件目录

│ ├── database.php # 数据库连接配置

│ ├── app.php # 应用全局配置

│ └── routes.php # 路由配置

│

├── public/ # 对外访问的文件目录

│ ├── index.php # 入口文件

│ └── assets/ # 静态资源

│ ├── css/ # 样式表

│ ├── js/ # 脚本

│ └── images/ # 图片

│

├── src/ # 应用源码目录

│ ├── Controllers/ # 控制器目录

│ │ ├── StoreController.php # 门店管理控制器

│ │ ├── SalesController.php # 销售管理控制器

│ │ ├── InventoryController.php # 库存管理控制器

│ │ ├── FinancialController.php # 财务报表控制器

│ │ ├── UserController.php # 用户管理控制器

│ │ └── PromotionController.php # 促销活动控制器

│ │

│ ├── Models/ # 模型目录

│ │ ├── Store.php # 门店模型

│ │ ├── Sale.php # 销售订单模型

│ │ ├── Inventory.php # 库存管理模型

│ │ ├── FinancialReport.php # 财务报表模型

│ │ ├── User.php # 用户模型

│ │ ├── Product.php # 产品模型

│ │ ├── StoreEmployee.php # 门店员工模型

│ │ └── Promotion.php # 促销活动模型

│ │

│ ├── Views/ # 视图目录

│ │ ├── store/ # 门店管理页面

│ │ │ ├── index.php # 门店列表页面

│ │ │ ├── add.php # 添加门店页面

│ │ │ └── edit.php # 编辑门店页面

│ │ │

│ │ ├── sales/ # 销售管理页面

│ │ │ ├── index.php # 销售订单列表

│ │ │ ├── add.php # 添加销售订单页面

│ │ │ └── details.php # 销售订单详情页面

│ │ │

│ │ ├── inventory/ # 库存管理页面

│ │ │ ├── index.php # 库存列表页面

│ │ │ ├── add.php # 入库记录页面

│ │ │ └── edit.php # 修改库存页面

│ │ │

│ │ ├── financial/ # 财务报表页面

│ │ │ ├── index.php # 财务报表列表

│ │ │ └── details.php # 财务报表详情

│ │ │

│ │ ├── user/ # 用户管理页面

│ │ │ ├── index.php # 用户列表

│ │ │ ├── add.php # 添加用户页面

│ │ │ └── edit.php # 编辑用户页面

│ │ │

│ │ └── promotion/ # 促销活动页面

│ │ ├── index.php # 促销活动列表页面

│ │ ├── add.php # 添加促销活动页面

│ │ └── edit.php # 编辑促销活动页面

│ │

│ ├── Services/ # 服务层目录

│ │ ├── StoreService.php # 门店业务逻辑处理

│ │ ├── SalesService.php # 销售业务逻辑处理

│ │ ├── InventoryService.php # 库存业务逻辑处理

│ │ ├── FinancialService.php # 财务报表业务逻辑

│ │ ├── UserService.php # 用户管理业务逻辑

│ │ ├── PromotionService.php # 促销活动业务逻辑

│ │ └── ReportService.php # 数据报表生成与导出

│ │

│ ├── Middleware/ # 中间件目录

│ │ ├── AuthMiddleware.php # 身份验证中间件

│ │ ├── PermissionMiddleware.php # 权限验证中间件

│ │ └── LoggerMiddleware.php # 日志记录中间件

│ │

│ └── Helpers/ # 辅助工具目录

│ ├── CommonHelper.php # 公共辅助类

│ ├── DateHelper.php # 日期处理工具

│ ├── FileHelper.php # 文件操作工具

│ ├── Validator.php # 数据验证工具

│ └── ReportHelper.php # 报表生成工具

│

├── tests/ # 测试目录

│ ├── Unit/ # 单元测试

│ ├── Feature/ # 功能测试

│ └── Integration/ # 集成测试

│

├── logs/ # 日志目录

│ ├── app.log

│ └── error.log

│

└── composer.json # PHP依赖管理配置文件

# 11、MYSQL数据库设计

根据您提供的程序架构，以下是基于MySQL 8的数据库架构设计，涵盖门店管理、销售管理、库存管理、财务报表、用户管理、促销活动等模块。

 数据库架构设计

 数据库名称：`tea\_shop\_chain`

 字符集：`utf8mb4`（支持多语言和表情符号）

##  11.1. 门店管理 (store)

 `stores` 表 (门店信息)

```sql

CREATE TABLE `stores` (

 `store\_id` INT AUTO\_INCREMENT PRIMARY KEY, 门店ID

 `store\_name` VARCHAR(255) NOT NULL, 门店名称

 `store\_location` VARCHAR(255) NOT NULL, 门店地址

 `store\_phone` VARCHAR(15), 门店电话

 `store\_status` ENUM('active', 'inactive') DEFAULT 'active', 门店状态

 `created\_at` TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

 `updated\_at` TIMESTAMP DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP

);

```

 `store\_employees` 表 (门店员工信息)

```sql

CREATE TABLE `store\_employees` (

 `employee\_id` INT AUTO\_INCREMENT PRIMARY KEY, 员工ID

 `store\_id` INT NOT NULL, 门店ID（外键）

 `employee\_name` VARCHAR(255) NOT NULL, 员工姓名

 `employee\_role` VARCHAR(255) NOT NULL, 员工角色（如：店长、收银员）

 `employee\_phone` VARCHAR(15), 员工电话

 `hire\_date` DATE, 入职日期

 `status` ENUM('active', 'inactive') DEFAULT 'active', 员工状态

 `created\_at` TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

 `updated\_at` TIMESTAMP DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,

 FOREIGN KEY (`store\_id`) REFERENCES `stores`(`store\_id`) ON DELETE CASCADE

);

```

##  11.2. 销售管理 (sales)

 `sales\_orders` 表 (销售订单)

```sql

CREATE TABLE `sales\_orders` (

 `order\_id` INT AUTO\_INCREMENT PRIMARY KEY, 订单ID

 `store\_id` INT NOT NULL, 门店ID（外键）

 `customer\_id` INT, 顾客ID（可以为NULL）

 `order\_total` DECIMAL(10, 2) NOT NULL, 订单总金额

 `order\_status` ENUM('pending', 'completed', 'cancelled') DEFAULT 'pending', 订单状态

 `payment\_status` ENUM('paid', 'unpaid') DEFAULT 'unpaid', 支付状态

 `order\_date` TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

 `updated\_at` TIMESTAMP DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,

 FOREIGN KEY (`store\_id`) REFERENCES `stores`(`store\_id`) ON DELETE CASCADE

);

```

 `sales\_order\_items` 表 (订单项)

```sql

CREATE TABLE `sales\_order\_items` (

 `order\_item\_id` INT AUTO\_INCREMENT PRIMARY KEY, 订单项ID

 `order\_id` INT NOT NULL, 订单ID（外键）

 `product\_id` INT NOT NULL, 产品ID（外键）

 `quantity` INT NOT NULL, 数量

 `unit\_price` DECIMAL(10, 2) NOT NULL, 单价

 `total\_price` DECIMAL(10, 2) NOT NULL, 总价

 FOREIGN KEY (`order\_id`) REFERENCES `sales\_orders`(`order\_id`) ON DELETE CASCADE,

 FOREIGN KEY (`product\_id`) REFERENCES `products`(`product\_id`) ON DELETE CASCADE

);

```

 `products` 表 (产品管理)

```sql

CREATE TABLE `products` (

 `product\_id` INT AUTO\_INCREMENT PRIMARY KEY, 产品ID

 `product\_name` VARCHAR(255) NOT NULL, 产品名称

 `product\_description` TEXT, 产品描述

 `price` DECIMAL(10, 2) NOT NULL, 售价

 `stock\_quantity` INT NOT NULL DEFAULT 0, 当前库存

 `created\_at` TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

 `updated\_at` TIMESTAMP DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP

);

```

##  11.3. 库存管理 (inventory)

 `inventory\_transactions` 表 (库存变动记录)

```sql

CREATE TABLE `inventory\_transactions` (

 `transaction\_id` INT AUTO\_INCREMENT PRIMARY KEY, 交易ID

 `store\_id` INT NOT NULL, 门店ID（外键）

 `product\_id` INT NOT NULL, 产品ID（外键）

 `quantity` INT NOT NULL, 库存变化数量（可以是负数，表示出库）

 `transaction\_type` ENUM('purchase', 'sale', 'restock') NOT NULL, 交易类型

 `transaction\_date` TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

 FOREIGN KEY (`store\_id`) REFERENCES `stores`(`store\_id`) ON DELETE CASCADE,

 FOREIGN KEY (`product\_id`) REFERENCES `products`(`product\_id`) ON DELETE CASCADE

);

```

##  11.4. 财务报表 (financial)

 `financial\_reports` 表 (财务报表)

```sql

CREATE TABLE `financial\_reports` (

 `report\_id` INT AUTO\_INCREMENT PRIMARY KEY, 报表ID

 `store\_id` INT NOT NULL, 门店ID（外键）

 `report\_date` DATE NOT NULL, 报表日期

 `total\_sales` DECIMAL(10, 2) NOT NULL, 销售总额

 `total\_cost` DECIMAL(10, 2) NOT NULL, 成本总额

 `total\_profit` DECIMAL(10, 2) AS (`total\_sales` `total\_cost`) STORED, 利润

 `created\_at` TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

 FOREIGN KEY (`store\_id`) REFERENCES `stores`(`store\_id`) ON DELETE CASCADE

);

```

##  11.5. 用户与权限管理 (user & permissions)

 `users` 表 (用户信息)

```sql

CREATE TABLE `users` (

 `user\_id` INT AUTO\_INCREMENT PRIMARY KEY, 用户ID

 `username` VARCHAR(255) NOT NULL UNIQUE, 用户名

 `password\_hash` VARCHAR(255) NOT NULL, 密码哈希值

 `role` ENUM('admin', 'manager', 'staff') DEFAULT 'staff', 角色（如：管理员、店长、员工）

 `status` ENUM('active', 'inactive') DEFAULT 'active', 用户状态

 `created\_at` TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

 `updated\_at` TIMESTAMP DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP

);

```

 `user\_permissions` 表 (用户权限)

```sql

CREATE TABLE `user\_permissions` (

 `permission\_id` INT AUTO\_INCREMENT PRIMARY KEY, 权限ID

 `user\_id` INT NOT NULL, 用户ID（外键）

 `permission` VARCHAR(255) NOT NULL, 权限项（如：订单管理、财务查看等）

 FOREIGN KEY (`user\_id`) REFERENCES `users`(`user\_id`) ON DELETE CASCADE

);

```

##  11.6. 促销活动 (promotion)

 `promotions` 表 (促销活动)

```sql

CREATE TABLE `promotions` (

 `promotion\_id` INT AUTO\_INCREMENT PRIMARY KEY, 促销ID

 `store\_id` INT NOT NULL, 门店ID（外键）

 `promotion\_name` VARCHAR(255) NOT NULL, 促销名称

 `promotion\_description` TEXT, 促销描述

 `discount\_percentage` DECIMAL(5, 2) NOT NULL, 折扣百分比

 `start\_date` DATE NOT NULL, 促销开始日期

 `end\_date` DATE NOT NULL, 促销结束日期

 `created\_at` TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

 FOREIGN KEY (`store\_id`) REFERENCES `stores`(`store\_id`) ON DELETE CASCADE

);

```

 总结：

1. 门店管理 (store)：包括门店信息和员工管理。

2. 销售管理 (sales)：包含销售订单和订单项，记录每笔销售交易。

3. 库存管理 (inventory)：记录每个门店的库存变动情况。

4. 财务报表 (financial)：生成和存储门店的财务报表。

5. 用户与权限管理 (user & permissions)：管理系统的用户和用户权限。

6. 促销活动 (promotion)：管理促销活动的相关信息。

每个表之间都有外键约束，确保数据的一致性和关联性。通过这些表的设计，您可以实现高效的门店管理、销售处理、库存跟踪、财务报表生成等功能。

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