

# 小米商城项目文档

## 项目选题说明和功能结构图

### 项目背景和目的

小米商城项目基于**EmarketMall电子超市系统**进行改造<sup>①</sup>。EmarketMall 是一个采用传统 JSP/Servlet 技术构建的简易商城系统示例<sup>②</sup>。选择以小米商城作为改造主题，是因为小米作为知名电子产品品牌，其商城功能场景丰富，包括商品浏览购买、用户账户以及售后服务等，具有较好的实践意义。通过将 EmarketMall 系统定制为小米商城，本项目旨在巩固 JSP+Servlet 开发技能，学习 Gradle 构建项目的配置与部署，熟悉 MVC 分层架构设计，并在现有基础上添加“设备服务”功能模块以贴近真实商城需求。

### 功能结构

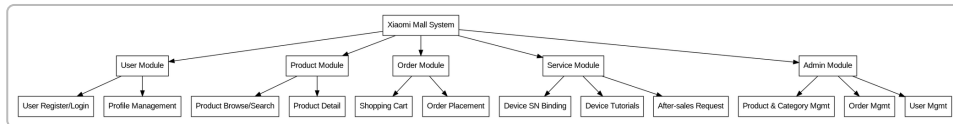


图1：小米商城功能模块结构图。本系统主要包括用户、商品、订单、售后服务和后台管理五大功能模块，各模块下设若干子功能。箭头层次关系表示功能模块的隶属与调用关系。

如图1所示，小米商城划分为前台和后台两部分功能模块。其中，**用户模块**提供用户注册、登录及个人信息管理等功能；**商品模块**涵盖商品分类浏览、搜索以及商品详情查看；**订单模块**包括购物车管理、订单提交与查询等购物流程；**售后服务模块**是本项目新增的特色，支持用户通过设备SN码绑定已购设备，绑定成功后可查看对应设备的使用教程，并在线申请售后服务；**后台管理模块**提供基础的数据维护功能（如商品及分类管理、订单管理、用户管理等），便于管理员维护商城内容。

## 项目进度计划及甘特图

本项目团队由前端开发李源和后端开发宋沛航两名成员组成，采用 JSP 技术结合 Gradle 构建进行开发。数据库设计基于原 EmarketMall 项目的SQL脚本<sup>③</sup>，在此基础上增补售后服务相关的表结构。开发工作计划在6月20日前完成，6月21日至23日留用于完善文档及答辩准备。项目进度划分为多个阶段，涵盖从需求设计到编码实现、测试和答辩准备。各阶段的时间安排和负责人如下所示。

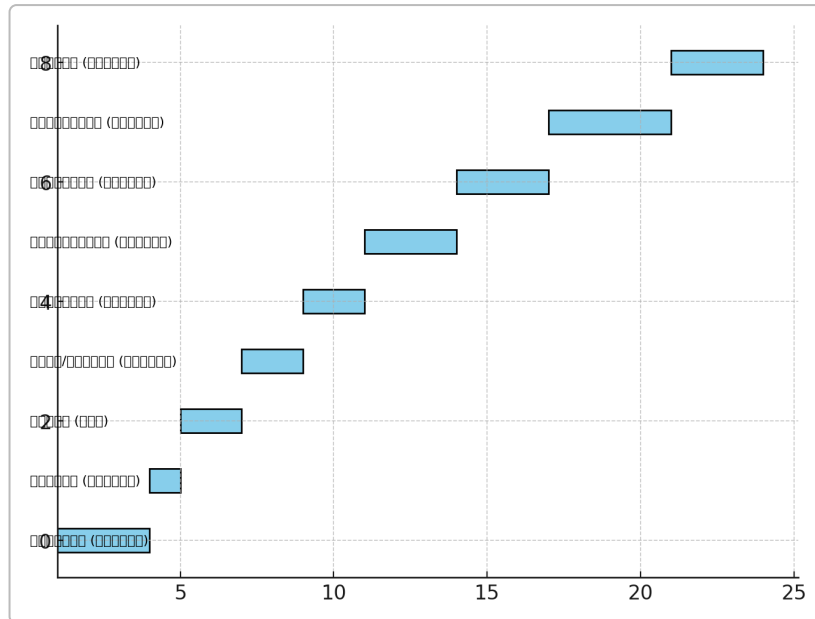


图2：项目进度甘特图（横轴为6月份日期，纵轴为任务编号1-9）。该甘特图直观展示了项目各阶段任务的起止时间及持续天数，任务编号对应下文的任务列表。

甘特图中任务编号及计划安排说明如下：

- 需求分析与设计（6月1日 - 6月3日，负责人：宋沛航、李源）** – 明确项目改造的功能需求，确定小米商城的模块划分和页面流程，包括新增“售后服务”模块的功能定义。完成系统原型设计和技术方案选型，为开发做好准备。
- 开发环境搭建（6月4日，负责人：宋沛航、李源）** – 配置项目的开发环境，包括安装配置IDE及Tomcat服务器，导入Gradle项目依赖，初始化数据库（导入原型SQL脚本）等。确保整个团队的开发和数据库环境一致，为编码工作打下基础。
- 数据库设计（6月5日 - 6月6日，负责人：宋沛航）** – 在原有 EmarketMall 数据库结构基础上完善数据模型 <sup>4</sup> <sup>5</sup>。新增用于设备绑定和售后申请的表结构，定义字段和键关系，确保数据库能够支持“服务”模块的需求。输出数据库结构说明文档并更新SQL脚本。
- 用户注册/登录模块开发（6月7日 - 6月8日，负责人：宋沛航（后端）、李源（前端））** – 实现用户账户相关功能，包括用户注册、登录校验、注销等。后端编写用户注册和登录的Servlet控制器和业务逻辑，使用数据库的用户信息表验证用户凭证；前端编写注册登录 JSP 页面和表单校验。保证用户可以成功注册账户并登录进入系统。
- 商品目录模块开发（6月9日 - 6月10日，负责人：宋沛航（后端）、李源（前端））** – 完成商品分类浏览和商品详情查看功能。后台实现商品类别和商品信息的获取接口，前端编写商品列表页面、搜索框以及商品详情展示页面。确保用户能够根据类别浏览商品、查看商品详细信息 <sup>6</sup>。
- 购物车和订单模块开发（6月11日 - 6月13日，负责人：宋沛航（后端）、李源（前端））** – 实现购物车管理、订单提交与订单查询功能。后端完善加入购物车、生成订单、订单列表查询等Servlet，操作相应的购物车、订单表和订单明细表 <sup>7</sup> <sup>8</sup>；前端开发购物车页面和订单确认页。此阶段确保用户可以将商品加入购物车并下单，订单信息持久化存储便于后续查询。

7. 售后服务模块开发（6月14日 - 6月16日，负责人：宋沛航（后端）、李源（前端）） - 实现设备SN码绑定及售后服务申请功能。后台设计设备绑定的Servlet接口，核对用户提交的SN码并绑定到用户账户；同时实现提交售后申请的接口，生成售后申请记录。前端开发“我的设备”页面，提供SN码输入绑定功能，以及设备详情页上提供“一键申请售后”功能入口。确保用户能够成功绑定其设备并提交售后服务单据。
8. 系统集成测试与优化（6月17日 - 6月20日，负责人：宋沛航、李源） - 对以上各模块进行集成测试，重点验证注册登录、购物下单、设备绑定和售后申请等功能的完整流程。模拟不同使用场景进行测试，修复发现的缺陷和错误，优化页面显示和交互细节，提升系统稳定性和用户体验。6月20日进行最终代码冻结，完成部署版本。
9. 项目答辩准备（6月21日 - 6月23日，负责人：宋沛航、李源） - 编写和美化项目文档（包括本文档的完善），制作答辩所需的 PPT 幻灯片与演示视频。团队成员熟悉各自模块的功能实现细节，整理项目亮点和难点，在答辩模拟中分工讲解、演示系统功能流程，确保答辩时能够清晰准确地展示项目成果。

## 数据库设计说明文档

本项目数据库在 EmarketMall 原有数据库结构基础上进行了扩展。原系统数据库采用 MySQL，包含用户、商品、订单等基础表结构<sup>9</sup><sup>5</sup>。在设计小米商城的数据库时，我们尽量沿用原有表，以减少重复开发，仅为新增的“售后服务”模块添加必要的表和字段。所有表统一采用整型自增主键 `id` 作为主键标识，并包含创建时间、修改时间、删除标记等维护字段，以方便管理（这些维护字段在下文不逐一赘述）。各主要数据表的结构和字段说明如下：

- **用户信息表（`user_info`）**：存储注册用户的基本信息。<sup>4</sup> 字段包括：
  - **id** (int) - 用户ID，主键，自增。
  - **name** (varchar) - 用户姓名（真实姓名）
  - **nickname** (varchar) - 用户昵称。
  - **loginName** (varchar) - 登录名，用于用户登录识别。
  - **avatar** (varchar) - 用户头像URL。
  - **email** (varchar) - 邮箱地址。
  - **phone** (varchar) - 联系电话<sup>10</sup><sup>11</sup>。
  - **password** (varchar) - 密码（采用MD5加密存储）<sup>12</sup>。此表用于存放前商城用户的信息，用户注册时插入新记录。表中 `loginName` 应唯一以避免重复注册。
- **商品信息表（`product_info`）**：存储商品的详细信息。<sup>6</sup> 字段包括：
  - **id** (int) - 商品ID，主键。
  - **productCore** (varchar) - 商品编码，用于标识商品SKU。
  - **productName** (varchar) - 商品名称。
  - **oneCategoryId** (int) - 一级分类ID，外键关联 `product_category` 表的主键，用于表示商品所属的主类别。
  - **twoCategoryId** (int) - 二级分类ID，外键关联 `product_category` 表，用于细分类别（如有二级分类）。
  - **price** (decimal) - 商品价格。
  - **publishStatus** (int) - 上下架状态（例如0表示下架，1表示上架）。
  - **productionDate** (datetime) - 生产日期。
  - **shelfLife** (int) - 保质期（月），用于推算商品有效期。

- **description** (varchar) – 商品描述信息。  
该表记录商城中销售的商品详情。通过 `oneCategoryId` 和 `twoCategoryId` 与商品分类表关联，以支持按分类浏览商品。商品信息由管理员通过后台管理模块维护。
- **商品分类表** (`product_category`)：存储商品类别层次信息。主要字段包括：
  - **id** (int) – 分类ID，主键。
  - **categoryName** (varchar) – 分类名称，如“手机”、“笔记本”。
  - **categoryCode** (varchar) – 分类编码。
  - **parentId** (int) – 父分类ID，自关联外键，指向本表中的上级分类（顶级分类的 `parentId` 为 NULL）。
  - **categoryLevel** (int) – 分类级别（如1=一级分类，2=二级分类）。
  - **categoryStatus** (int) – 分类状态（如0不显示，1显示）。  
该表采用自关联结构支持多级分类展示。例如，手机属于电子产品大类时，手机分类的 `parentId` 指向电子产品分类的ID。
- **商品图片表** (`product_pic`)：存储商品图片资源。字段包括：
  - **id** (int) – 图片ID，主键。
  - **productId** (int) – 所属商品ID，外键关联 `product_info.id`。
  - **picDesc** (varchar) – 图片描述。
  - **picUrl** (varchar) – 图片URL地址。  
每件商品可以有多个图片，该表通过 `productId` 与商品信息表形成一对多关系，便于前端展示商品轮播图等。
- **库存表** (`warehouse_product`)：记录商品库存数量。字段包括：
  - **id** (int) – 库存记录ID，主键。
  - **productId** (int) – 商品ID，外键关联 `product_info.id`。
  - **currentAmount** (int) – 当前库存数量。  
该表用于简化库存管理，在订单完成支付或发货后，库存数量应相应减少。Demo系统中可能未实现完整库存扣减流程，但表结构已预留。
- **购物车表** (`order_cart`)：存储用户临时加入购物车的商品项。字段包括：
  - **id** (int) – 购物车记录ID，主键。
  - **productId** (int) – 商品ID（关联商品信息表）。
  - **amount** (int) – 购买数量。
  - **price** (decimal) – 商品加入购物车时的单价。  
购物车表用于暂存用户选购尚未下单的商品。用户登录状态下，可以添加商品到购物车，多次添加同一商品可能产生多条记录或更新数量（具体由实现决定）。当用户提交订单时，相关记录将从购物车表移除或转移到订单明细表中。
- **订单表** (`orders`)：记录用户提交的订单主信息。 7 字段包括：
  - **id** (int) – 订单ID，主键。
  - **orderNum** (varchar) – 订单编号，唯一标识一次订单交易，可由日期+随机数生成。
  - **userId** (int) – 下单用户ID，外键关联 `user_info.id` 表示订单归属的顾客 13。

- **shippingUser** (varchar) – 收货人姓名。
- **address** (varchar) – 收货地址。
- **paymentMethod** (int) – 支付方式（例如1=现金，2=余额，3=网银，4=支付宝，5=微信）。
- **orderMoney** (decimal) – 订单商品总金额。
- **shippingMoney** (decimal) – 运费金额。
- **discountMoney** (decimal) – 优惠金额。
- **paymentMoney** (decimal) – 实际支付金额。
- **payTime** (datetime) – 支付时间。
- **receiveTime** (datetime) – 收货时间。

- **orderStatus** (int) – 订单状态（例如0=未支付，1=已支付待发货，2=已发货待收货，3=已完成，4=取消等）。

订单表保存每笔订单的概要信息。当用户在前台提交订单后，生成一条订单主记录以及若干订单明细记录。订单状态随着流程推进而更新，例如支付完成后从0变为1，发货后变为2等。

- **订单明细表** (**order\_detail**)：记录订单中的具体商品项明细。<sup>14</sup> 字段包括：

- **id** (int) – 订单明细ID，主键。
- **orderId** (int) – 所属订单ID，外键关联 **orders.id**。
- **productId** (int) – 商品ID，关联 **product\_info.id**。
- **productName** (varchar) – 商品名称（冗余存储，记录下单时的商品名称）。
- **amount** (int) – 购买数量。

- **productPrice** (decimal) – 商品单价（冗余存储，记录下单时的单价）。

订单明细表与订单表构成一对多关系，每条订单可对应多条明细记录，用于保存订单中包含的所有商品及数量。通过冗余商品名称和价格，可以防止商品信息日后修改对历史订单的影响。

- **员工信息表** (**staff\_info**)：存储后台管理端工作人员账号信息。主要字段包括：

- **id** (int) – 员工ID，主键。
- **loginName** (varchar) – 登录名（管理员账户名）。
- **name** (varchar) – 员工姓名。
- **staffType** (varchar) – 员工类型（岗位角色）。
- **email** (varchar) – 邮箱。
- **phoneNum** (varchar) – 电话。
- **gender** (varchar) – 性别。
- **avatar** (varchar) – 头像URL。
- **password** (varchar) – 登录密码。
- **status** (int) – 帐号状态（0=正常，1=停用）。

- **deptId** (int) – 部门ID，外键关联 **department.id** 表示所属部门。

该表用于后台管理员登录验证和权限管理。由于本项目主要侧重前台商城功能，**staff\_info** 和 **department** 等表属于原型系统的后台部分，在本次开发中仅作保留，未进行大的改动。

- **部门表** (**department**)：存储后台员工所在部门信息。字段包括：

- **id** (int) – 部门ID，主键。
- **name** (varchar) – 部门名称。
- **leader** (varchar) – 部门负责人姓名。

- **phoneNum** (varchar) – 联系电话。

此表与 `staff_info` 表通过部门ID建立联系，一个部门下可有多名员工。部门表数据由超级管理员维护，用于对员工账号进行组织分类（例如技术部、市场部等）。

- **设备信息表** (`device_info`) (新增) \*\*: 存储用户绑定的设备信息。主要字段设计：

- **id** (int) – 设备记录ID，主键。
- **snCode** (varchar) – 设备SN码，产品序列号，具有唯一性。
- **productId** (int) – 设备对应的商品ID，外键关联 `product_info.id`，表示该设备的产品型号。
- **userId** (int) – 绑定用户ID，外键关联 `user_info.id`，表示该设备当前绑定在哪个用户名下。

- **bindDate** (datetime) – 绑定时间。

设备信息表是“售后服务”模块新增的核心表，用于记录用户已经绑定的设备。用户购买商品后，可以通过输入商品的序列号（SN码）将设备绑定到个人账户下。绑定时系统需验证该 SN码 的有效性（本项目可简化为与预置的数据比对）。一旦绑定成功，`device_info` 中会创建记录并关联到具体用户和商品，表示该用户拥有此设备。在实际场景中，设备表还可能包含购入日期、保修状态等信息，但在本项目中从简设计，只保留必要字段。

- **售后申请表** (`service_request`) (新增) \*\*: 存储用户提交的售后服务申请单。主要字段设计：

- **id** (int) – 申请单ID，主键。
- **userId** (int) – 提交申请的用户ID，外键关联 `user_info.id`。
- **deviceId** (int) – 相关设备ID，外键关联 `device_info.id`，指明申请售后的具体设备。
- **requestType** (varchar) – 服务申请类型，例如“维修”“退换货”等。
- **description** (varchar) – 问题描述，由用户填写设备故障或诉求的描述信息。
- **status** (int) – 处理状态（0=待处理，1=处理中，2=已完成，3=已拒绝等）。
- **requestTime** (datetime) – 申请提交时间。
- **processTime** (datetime) – 申请处理完成时间（若已处理）。

售后申请表记录用户每次提交的售后服务请求。通过 `userId` 和 `deviceId` 字段可以追踪是哪位用户针对哪台设备发起了申请。管理员在后台可查看并处理这些申请，将处理结果更新状态字段。借助该表，系统能够提供售后服务单据的登记与查询功能，实现线上售后流程的闭环。

综上所述，小米商城项目的数据库设计在 EmarketMall 原始结构之上进行了合理扩展，新增的**device\_info**和**service\_request**表与原有表通过外键建立联系，从而支持了设备绑定和售后服务的功能。整个数据库各表设计清晰简洁，遵循第三范式，满足项目功能需求的同时亦保证了学生在有限时间内可以完成开发实现。

---

1 2 3 README.md

<https://github.com/Aftnos/CQUET-JSP/blob/f07289b379b2e523a977abad3bbc21566edf672c/README.md>

4 5 6 7 8 9 13 14 dump-emarket\_mall-202506050955.sql

[https://github.com/Aftnos/CQUET-JSP/blob/f07289b379b2e523a977abad3bbc21566edf672c/server/dump-emarket\\_mall-202506050955.sql](https://github.com/Aftnos/CQUET-JSP/blob/f07289b379b2e523a977abad3bbc21566edf672c/server/dump-emarket_mall-202506050955.sql)

10 11 12 user.jsp

<https://github.com/Aftnos/CQUET-JSP/blob/f07289b379b2e523a977abad3bbc21566edf672c/src/main/webapp/user/user.jsp>