

# 广州大学

## 本科毕业论文(设计)

课题名称	AI 驱动的零售商品全流程管理系统设计与实现
学    院	计算机科学与软件工程学院
专    业	软件工程
班级名称	软件（创）211
学生姓名	张景致
学    号	32106300004
指导教师	闫红洋
完成日期	2025 年 4 月 12 日

教 务 处 制

# AI 驱动的零售商品全流程管理系统设计与实现

软件工程专业 软件（创）211 班 张景致

指导教师：闫红洋

**摘要** 当今时代人工智能技术、人工智能模型训练和推理行业发展迅速，各行各业都在积极利用人工智能技术助力各个产业的数字化、智能化进步，进一步推动社会生产力的创新发展。在这股 AI（人工智能）浪潮之中，本就因为电子商务的兴起而态势疲弱的传统零售行业面临进一步落后的风险。但是具有实际体验并与商品进行各种互动等独有优势的传统零售行业仍有进一步发展的空间、数字化转型的必要。本设计以服务于广大消费者、零售从业者为宗旨，以构建实际零售管理系统为实验方法，在商品文案设计、商品检索、商品推荐与咨询、商品结算、库存管理、销售数据分析等方面，探究并提出了一系列将时代前沿 AI 技术与传统零售行业管理和服系统相结合的具体办法。

**关键词** 人工智能；零售；大语言模型；实体经济

**ABSTRACT** Nowadays the advancement of AI (artificial intelligence) technologies and AI model training and inference industry are significant. Most industries and areas are actively leveraging AI technologies to empower the digitalization efforts, boosting the creative step forward of our society. However, despite the emerging AI innovations, traditional retail businesses, being already behind the waves because of the rising e-commerce industry, face the risk of further decline. But it is also apparent that having the unique feature of allowing end-users to experience and interact with products demonstrates that retail businesses could and should be further advanced and digitalized. This project design, with the consumers and retail owners in mind, in the way of building an functional retail management system, researches and purposes a set of concrete methods to fuse AI and traditional retail together, including the design of product write-ups, product searching, product recommendation and consulting, check-outs, inventory management and analysis of sales data.

**KEY WORDS** Artificial Intelligence, Retail, Large Language Model

# 目 录

<b>1</b>	<b>前 言</b>	<b>5</b>
<b>2</b>	<b>现有方案</b>	<b>7</b>
<b>3</b>	<b>整体架构</b>	<b>8</b>
3.1	服务端	8
3.2	商家端	9
3.3	顾客端	11
<b>4</b>	<b>零售管理系统</b>	<b>12</b>
4.1	服务器	12
4.1.1	数据结构	12
4.1.2	应用程序接口	13
4.1.3	自动部署	14
4.1.4	服务依赖管理	15
4.1.5	图像存储	15
4.2	管理界面	15
4.2.1	高级搜索	17
4.2.2	移动端程序	18
4.3	结算界面	18
4.3.1	条码识别器	18
4.3.2	质量传感器	19
<b>5</b>	<b>商家端 AI 功能</b>	<b>20</b>
5.1	商品设计辅助	20
5.1.1	编写文案	20
5.1.2	建议定价	20
5.1.3	提供额外参考	20
5.2	销售数据分析	20
5.2.1	图表生成	20
5.2.2	图表理解	20
5.2.3	营业策略建议	20
5.3	库存审计	20
<b>6</b>	<b>顾客端 AI 功能</b>	<b>21</b>
6.1	导购助手	21

6.1.1	对话型生成式 AI	21
6.1.2	输出风格	21
6.1.3	搜索关键词提取	21
6.1.4	商品搜索	21
6.2	称重商品识别	21
6.2.1	数据准备	21
6.2.2	图像处理	21
6.2.3	AI 图像分类	21
6.3	模糊搜索	21
6.3.1	商品数据处理	21
6.3.2	AI 近义词搜寻	21
6.3.3	词典构建	21
6.3.4	搜索算法	21
<b>7</b>	<b>实验</b>	<b>22</b>
7.1	商家端	22
7.1.1	桌面应用	22
7.1.2	手机应用	22
7.1.3	数据集管理程序	22
7.2	顾客端	22
7.2.1	手机应用	22
7.2.2	结算程序	22
<b>8</b>	<b>讨论</b>	<b>23</b>
<b>9</b>	<b>结论</b>	<b>24</b>
	<b>参考文献</b>	<b>24</b>
	<b>致谢</b>	<b>25</b>

# 1 前 言

近几年来,人工智能(AI)技术有了前所未有的深入发展。从上世纪 50、60 年代“机器学习(Machine Learning)”<sup>[1]</sup>概念被提出时它初次登上历史舞台,到人们利用图像处理单元(GPU)等专用硬件进行处理<sup>[2]</sup>如 AlexNet<sup>[3]</sup>和 ResNet<sup>[4]</sup>等规模成指数级别增长的深度学习(Deep Learning)模型,再到 2017 年 Transformer 模型<sup>[5]</sup>掀起自然语言处理领域变革,AI 领域已然有了长足的进步。而近几年从 OpenAI ChatGPT<sup>[6]</sup>引爆大语言模型(LLM)热潮到以 DeepSeek 为代表的一众大幅降低部署、使用成本的开源大模型走进人们生活不难看出,AI 技术势不可挡,并且在可以遇见的未来还会进一步发展壮大。

人工智能技术既是一个独立的领域,又是其他行业和领域进一步深入发展<sup>[7]</sup>、进行数字化转型<sup>[8]</sup>的不可或缺的一部分。例如多模态(multi-modal)的大语言模型将图像、音频等不同媒介的信息与一般大语言模型的文字信息连接起来,形成了“看得懂”、“听得懂”的大(语言)模型<sup>[9]</sup>。从智能制造到智慧医疗,人工智能在垂直领域中逐步渗透,与不同行业、工业相结合,使其得到了新的发展力量。

与朝气蓬勃的人工智能产业现成鲜明对比的一个领域是传统实体零售行业。改革开放以来,直到电子商务(电商)产业兴起以前,实体店铺几乎是民众购买不同产品的唯一方式,担任了将商品从设计、生产和批量分发的企业转移到最终用户的桥梁的角色,既是产品供给的终点,也是收益反馈的起点。以淘宝为代表的网上购物平台(也就是面向最终消费者的电子商务平台)兴起之后,产业链成本、消费便利性等种种因素使得消费者愈发青睐网上购物,足不出户便能选购喜爱的产品。即便没有自主选购的意愿,吸引人的商品也会从各种不同的广告推荐渠道来到消费者眼前。如此突出优势,与近年波动的经济环境和不安定的地缘政治情况,化为了许多中小型实体零售企业、个体户的运营压力,甚至使得其中的许多面临不得不终止运营的极端情况。

然而,实体零售行业仍有无法被取代的优势。消费者在参与线下购物的过程中,可以通过与商品的近距离互动来产生对其直观的印象,这种“零距离”的、有着天然信任的购物体验无法复制;网上购物的方法割裂了顾客支付商品价格和接受实体商品的过程,对物流有着一定程度的依赖,而实体零售则可以购买当时直接获得对应实体产品。此外,线下购物的过程同时也可以社交的过程,可以营造独特的社会价值,增强消费者的生活体验。

为了在充满激烈竞争的经济环境下保持甚至提升自身的地位,实体零售行业需要积极进行自身的数字化、智能化转型,通过各种不同方法充分发掘实体零售的独特优势。在这样的情势下,“新零售”<sup>[10-11]</sup>、“智慧零售”<sup>[12]</sup>等概念、预想应运而生,“人-货-场”匹配的最优化<sup>[13]</sup>受到广泛研究探讨。但是,这样的转型目前一般只有市场头部企业开始实施,体量较小的企业和个体户尚无资本和技术能力展开;并且其中较为重要的一个方面,实体零售与人工智能技术的结合,还有待深入开发。

本设计项目立足于传统实体零售行业，尤其是成本上受到较大限制的中小型企业 and 个体户，对产业数字化、智能化的需求日渐急迫的当下，致力于探究在传统零售行业中应用、融入乃至整合人工智能技术，利用人工智能的便利增强经营者的营业能力并降低运营门槛，将更便利的、更有亲和力的实体购物体验带给消费者。具体来说本项目在实现一套基本可用的分布式零售管理基础设施、管理软件的基础上，利用不同类型的人工智能技术实现了以下几个不同的功能模块：

- 服务器部分
  - 智能分词技术、近义词搜寻技术驱动的关键词搜索引擎
- 管理端部分
  - 大模型驱动的智能商品文案编写助手
  - 大模型驱动的智能业务图表分析、运营建议模块
  - 基于智能条码识别、扫描技术的点货功能
- 门店端部分
  - 基于图像分类的智能商品识别功能
- 客户端部分
  - 大模型驱动的多轮对话、搜索推荐智能导购助手

该部分之后的文章内容从介绍和分析该领域（零售管理）的现有方案（章节 2）开始，其后从整体角度对该项目所实现系统的架构设计作出解释说明（章节 3），然后分别对各个模块的具体设计、实现方案进行详细的描述，再之后对该系统相关的测试和运行效果进行列举和说明。最后对该系统的实现效果、未来改进空间等话题进行讨论，进而结合这个项目的情况对该领域的未来作出预测来对该文章收尾。

## 2 现有方案

零售管理的需求是跨越客户关系管理（CRM）、企业资源计划（ERP）、销售时点情报系统（POS）和“进销存”（进货、入库、销售）等多种运营管理领域的综合性的管理需求。同样，目前市面上较为常见的各类零售管理方案因此也覆盖了其中部分或全部的领域。

其中十分具有代表性的“微盟”是以微信平台为依托的，覆盖实体店面收银、库存管理、会员管理等各个功能的一个功能较为广泛的软件即服务（SaaS, Software as a Service）管理产品。在门店、导购数字化方案之外，“微盟”还提供私域电商<sup>[14]</sup>（由一个或多个特定实体主导的私有电子商务平台，有别于“淘宝”等较多不同实体进驻的公共平台）搭建和管理服务，有助于其客户利用平台整合度高的优势推进其自身的数字化和智慧化。

另一个具有一定代表性的是伯俊科技公司的“云 POS”和“BOS Cloud”等零售门店管理服务。该方案以方式种类丰富的 POS 为核心，提供传统桌面式收银台、手持式收银终端、自助收银终端等专用硬件。与“微盟”一样，该方案同样提供了一定程度的在线销售功能。不同的是“云 POS”采用接入一般外卖、直播电商平台的方法实现线下门店的“线上化”，以此促进其客户的数字转型。值得注意的是，该平台具有接入其他第三方管理平台（如“微盟”）的方法，其供应商绑定风险较低。

除了上述的两个方案，还有如“T+Cloud”实体店新零售管理系统、“茗匠”有人或无人零售收银管理系统、“金蝶”小微企业云服务平台和“致心零售”SaaS 零售软件等整合了多个管理维度，提供零售行业一站式管理的各种不同方案。

这些方案都在可以从某个或某些方面上增强对应零售业务的数字化、智能化程度，提高运营便利、自动化程度，进而改善产业生态和客户体验，但是它们都具备相似的问题：注重于将线下门店“线上化”、将电子商务“实体化”，把电子商务的优势整合到传统实体零售行业中，而一定程度上忽视了对这些技术和手段（无论线下还是线上）本身的进一步优化和发展，进而在对经营者门店数字化工作便利性的增强和对消费者核心购物体验改善的工作上有所不足<sup>1</sup>。

而本设计所构建的 AI 驱动的零售商品全流程管理系统，通过利用人工智能这一“通用算法”，实现了以传统商业应用程序及其算法所无法实现的功能和特性，进而探究了从经营者、消费者多角度多层次提高零售行业运作水准的可能性，给出了将人工智能技术和传统零售行业相结合的切实方案、可行方案，为零售行业的数字化、智能化未来添砖加瓦。

---

<sup>1</sup>上海微盟企业发展有限公司在 2024 年前后发布了“WAI”人工智能工具集合。根据其公布的少量资料，这些工具以面向经营者为主，尚未覆盖到零售业务的各方面。此外，这方面可供参考的资料稀缺，故“WAI”产品未被在正文提及。

### 3 整体架构

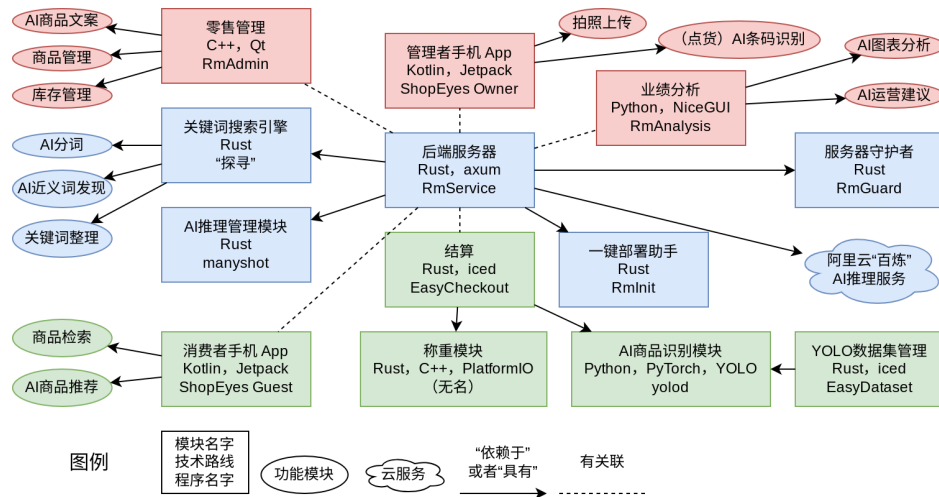


图 3.1 本设计的系统架构图示：其中蓝色部分为服务端，红色部分为商家端、绿色部分为顾客端。图例（不同形状的各自意义）位于图像下侧。

本设计的系统架构如图 3.1 所示，而从图像中不难看出，该系统整体上呈现出客户端-服务器模式的结构，其中客户端部分分为面向零售行业从业者的商家端和面向最终消费者的顾客端。

#### 3.1 服务端

本设计中对服务端功能的期望主要有如下几点：

- 高性能的 HTTP API 服务器
- 商品、库存、订单等运营资料的增删改查
- 高性能、效果良好的智能商品搜索
- AI 大语言模型推理托管

实际部署的情况下，服务器可能需要在短时间内处理来自大量客户端应用程序的服务请求，比如营业高峰期来自许多客人的商品检索、询问 AI 助手索取导购建议的请求，因此服务器本身的效率必须纳入设计的考虑之中。基于这样的考虑，该设计采用 Rust 语言进行开发。Rust 是比较流行的一门强调内存安全的系统编程语言，利用这门语言，服务器的代码执行效率有充足的优化机会，并且开发的便利性得到了一定保证。为了充分利用大部分部署环境都将会具备的多处理器的对称并行处理



(SMP) 执行环境，服务器内主要采用由 tokio 第三方库驱动的异步开发的技术手段。为了实现服务器与客户端的有效、高效、泛用性强的信息双向传输，采用通过（商铺内部）局域网的 HTTP。为了与总体的技术选择相统一，采用依托并行计算技术开发的 axum HTTP 服务器框架。

服务器需要能实际储存并处理在零售运营过程中需要的各种数据。为了满足这个期望，该设计中采用基于 SQLite 关系数据库引擎的 rqlite 分布式关系数据库管理软件作为原始数据存储、查询和管理的手段。为了提供统一可靠的开发接口，rqlite 服务器将只面向服务器软件开放，而客户端任何对实际业务数据的访问都必须通过服务器对其结构化、统一化的封装。

为了实现在众多商品中快速找到顾客所需，服务器需要具备通过（顾客提供的）一段可能和商品本身文字内容不尽相同的搜索语句对商品进行匹配的功能。市面上常见的如 Apache Lucene、Elasticsearch 等各种相关产品与该设计的相关理念并不匹配：因为较为复杂而不适合在该设计所期望的低成本设备上部署；对中文没有比较针对性的优化，并且搜索对象仅限于搜索目标中出现过的字词。为了解决这个问题，该设计中包含一个由多种人工智能技术驱动的，高效、高匹配率、简明易用的中文特化搜索引擎“探寻”。

鉴于该项目中多处利用到了大语言模型，若是服务器可以向各个客户端软件提供统一的推理接口，将不同模型、不同服务商的区别消除，将有利于项目的整体可用性、可维护性。因此，服务器提供一个专门开发的推理模块，具备单次推理（“oneshot”）、多次尝试（“manysot”）和有状态多轮对话管理等功能。

## 3.2 商家端

本设计中对商家端功能的期望主要有如下几点：

- 高性能、便于使用的用户界面
- 功能丰富、易于操作的商品、库存管理界面
- 完整的销售数据查询界面
- 业绩图表生成、展示界面
- 业绩图表 AI 智能分析、运营建议
- AI 条码识别点货
- AI 商品文案自动生成、批量生成
- 商品识别数据集创建和修改

- 从手机上传用于商品的图片

为了在普通的计算机上高效管理营业资料，商家端需要一套对系统要求较小的、对键盘鼠标操作较为友好的、对屏幕尺寸需求灵活的用户界面。基于这样的缘由，该设计的商家桌面端应用程序采用 Kotlin 作为业务逻辑开发语言以最大化开发效率，进而采用受到工业界广泛采用的 Qt Widgets 应用程序开发框架在 Java、Kotlin 语言上的实现 Qt Jambi。

在实际操作的情况下，经营者不可避免地将需要对各类数据进行条件细致的筛选。为了满足这样的需要，商家桌面端应用程序为多种不同零售资料的查询准备了既功能丰富，又简单易懂的图形化高级搜索条件拟写工具。

将销售数据按表格列出是简单易行的，但这样的数据展示方式往往无法满足营业者的营业数据分析的需要，图表可以更好地呈现数据之中潜在的结构性和规律性。为了实现这样的功能，本设计采用业界惯用的 Python 语言作为数据分析的主要语言，利用 HTTP API 与基于 Rust 的服务器进行通讯，并采用 pandas、numpy 等数据科学库展开数据整理工作，利用 matplotlib 库进行图形的绘制。为了简化开发过程，图表部分的用户界面同样采用 Python 编写，同时选择 NiceGUI 用户界面框架实现与上述第三方库更高的整合度。

为了减轻零售从业者观察图形规律的压力，该设计利用来自阿里云“百炼”AI 推理服务的多个 AI 模型，针对不同类型、复杂程度的营业数据图表进行“先观察后思考”的“接力式”智能分析报告编写或者“边观察边思考”的快速图形规律总结，向从业者提供详尽准确的数据分析。

业务资料中的商品图片较为特殊，在手机设备上处理可能相比在桌面型计算机上更加便捷。因此，该项目采用 Jetpack Compose 安卓应用程序开发框架实现面向商家的管理用应用程序。利用该应用程序营业者可以较为简单地将手机中的照片或现场拍摄的照片上传到服务器中。

实体零售行业常常无法避免对仓库中、货架上产品进行审计（统计），以此确认实际产品数量与系统中库存数量对应关系的需要，而手工记录并比照的方法费时费力并且容易出现错误。因此，该设计的移动商家端应用程序采用谷歌 MLKit 的 AI 条码识别功能开发了利用手机自带摄像头的 AI 条码识别并记录、上传的功能。

为了简化用于智能结算房屋中的商品识别模型的训练工作，该项目具备利用 Rust 和 iced 用户界面框架开发的数据集创建和修改功能。数据集准备的过程分为“类别规划”“图片采集”和“图片管理”三个连贯的部分。从业者能够先根据实际的需要在应用程序中输入需要的商品种类，再利用智能商品识别对应的摄像设备进行数据集中图片的采集，最后在采集的图片中筛选出质量较高者。

### 3.3 顾客端

本设计中对顾客端功能的期望主要有如下几点：

- 美观大方、便于使用的用户界面
- 推荐商品浏览
- 商品检索和详情浏览
- AI 导购多轮对话商品推荐
- 基本商品结算
- AI 商品识别-计重结算

为了最大程度提高消费者获取商品信息的便利程度，增强新零售购物体验，该项目采用 Jetpack Compose 开发面向消费者的用于浏览商品情况的应用程序。应用程序分为“推荐”“搜索”和“询问 AI”三个板块，分别通过不同的用户界面、不同的对服务器 API 的请求方式来实现相应的功能。其中推荐功能展示由从业人员在商家端预先设置的一系列推荐商品，而搜索功能顾名思义。

另一个功能是“询问 AI”。在该功能界面下，用户可以通过屏幕底部的搜索框输入需要向 AI 询问的导购相关问题，用户输入的问题将会经由服务器中转发送到后端预先指定的推理服务和选择的大模型。在获取输出之后，顾客端应用程序会利用 AI 的答复再次利用 AI 进行关键词的提取，进而利用归纳得出的关键词进行对商品的检索，进而实现对用户推荐相关的商品的功能。

为了同时满足店员辅助结算和消费者自主结算的需要，该项目需要提供一个用于结算设备的简洁明了的用户界面。因此，该设计采用 Rust 语言及 iced 用户界面开发框架设计开发结算用用户界面。用户可以利用通过人体工学输入设备（HID）方式连接到结算设备的激光条码扫描设备来输入任何按件结算的商品。

某些零售细分领域所提供的商品（如生鲜蔬果、熟食等）是按重量结算的，需要称重才能获取实际价格，并且按实际情况有可能商品表明无法粘贴对应的条形码（比如肉类），需要其他手段辅助才能确定商品的具体类型。为了解决这样的问题，本设计利用 YOLO 图像分类模型，结合前文提及的商家端数据集管理工具，开发 AI 智能商品识别算法，用以简化计重商品结算流程。

## 4 零售管理系统

### 4.1 服务器

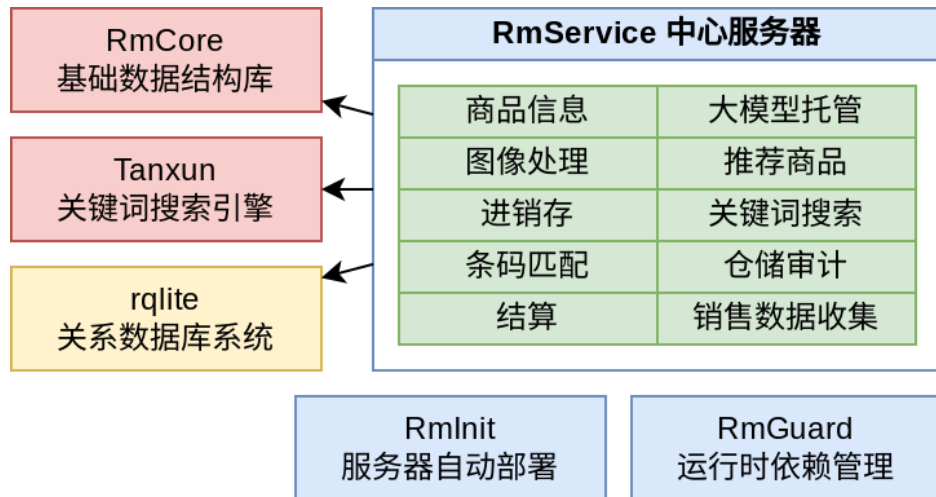


图 4.1 本设计的服务器整体设计图示：其中蓝色表示相互独立的应用程序，绿色表示服务器的功能，红色表示功能库，黄色表示第三方库或应用程序，箭头表示被指方受到依赖。

#### 4.1.1 数据结构

如图 4.1 所示，因为采用了客户端-服务器设计模式，该系统所有业务相关信息统一由 RmService 模块进行管理。因此，为了在系统中体现充分的可维护性、可拓展性，实际对数据进行存储的关系数据库系统 rqlite 原则上无法被客户端（各个商家端、顾客端应用程序）所直接访问，而是由 RmCore 所定义的数据结构进行封装。

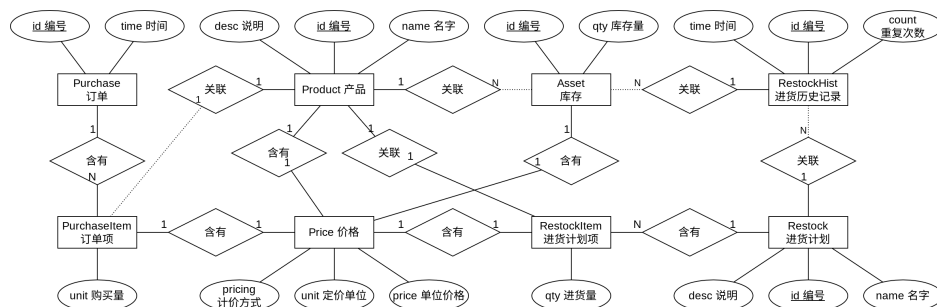


图 4.2 RmCore 模块所定义数据结构的实体关系图

如图 4.2 所示，该模块定义了零售经营过程中会利用到的各种（需要持久化存储的）数据结构，并且已经以面向对象的、有利于代码复用、序列化和反序列化和有利于在不同开发环境下形成统一接口的方式进行封装。为了便于在数据库中以整数

方式存储许多常常为小数的数据，规定任何直接代表金钱数量的值以人民币分为单位，任何直接代表物体质量的值以克为单位。

其中值得注意的一个类型是库存“Asset”。为了便于对库存所对应的进货批次、进货价格等信息进行跟踪，每一个库存项都与一个进货历史记录项目“RestockHist”相关联。因此，同一个产品可以因为多次不同进货并且较老进货批次剩余产品尚未被消化完毕而存在多个库存条目。同样地，只需要统计一个 RestockHist 所对应的 Asset，便可以统计某一个批次进货的消化情况。

另一个较为特殊的类型是价格“Price”。考虑到不同零售商品类型定价策略的区别，该类型的属性计价方式“pricing”为具有“Package”（按件）和“Weight”（按重量）两个可能值的枚举。而属性定价单位“unit”在按件计价时代表属性价格“price”对应的商品件数，反之则是克数。而将属性 price 和 unit 分离（而不是表示为单位价格）有助于避免如“十元三件”（一件  $\frac{10}{3}$  元）此类出现整数或 IEEE 754 浮点类型数字无法准确表示的数值的情况。

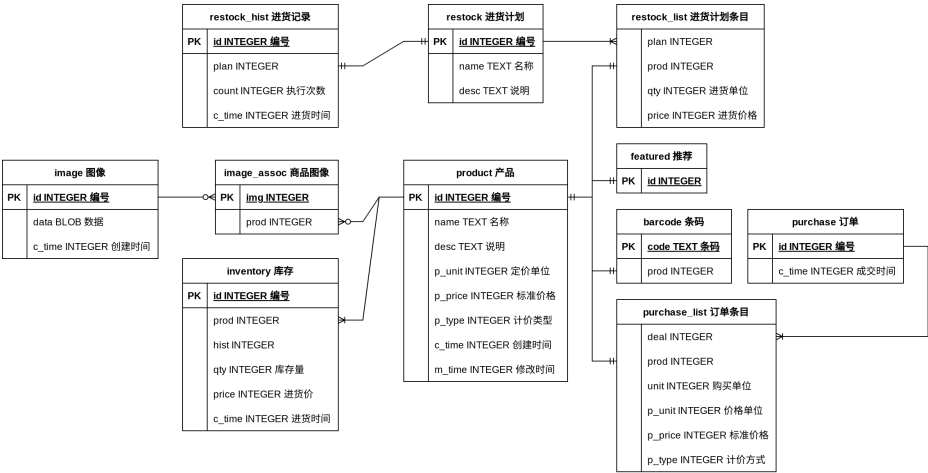


图 4.3 数据库中存储数据格式的实体关系图

图 4.3 所描绘的是 4.2 所示的模型还有少数以其他方式封装的数据的在关系数据库系统中的实际存储方式。除了 Price 类型没有单独表格，而是被扁平化地存储在了各个表中的区别，数据存储的方式基本与封装成一比一对应关系。少部分表格具有额外的创建、修改时间字段（“c\_time”和“m\_time”）留作未来拓展使用。值得注意的是，为了改善大规模访问的效率，商品图像数据并不是存储在外部文件系统，而是和其他资料一同存储在数据库中。

#### 4.1.2 应用程序接口

为了实现各个客户端应用程序与服务器的有效信息交换，服务器采用 HTTP 协议进行信息传递，即服务器实现面向客户端应用程序的 HTTP API。由于 API 覆盖范围较广、数量较大并且功能迥异，故该课题代码开发时大部分 API 端点设计遵从如下规范：

- 期望请求主体（body）部分带有实际数据者，接受 POST 方法的请求；否则接受 GET 方法的请求，并且对方法错误的请求响应“Not Found”状态。
- 期望请求主体带有多于一个字段者，接受使用 JSON 格式序列化的数据，否则按实际情况决定主体数据类型。
- 较短、较少的请求参数，可以通过 HTTP 地址参数而不是请求主体传递。
- 除纯二进制数据（如图片）外，方法一律使用 JSON 格式序列化返回值。
- 在端点调用遇到责任在客户端（调用参数格式错误等）的问题时，返回“Bad Request”状态并且尽可能在回复主体中包含对问题根源进行描述的字符串。
- 同上，但责任在服务器（意外情况）时，返回“Internal Server Error”状态并且在主体中包括导致问题的函数调用返回的相关错误信息。
- 客户端请求的资源或任务结果尚未准备完成时，返回“Accepted”状态并在回复主体中标注具体任务现状。

为了支持商家端应用程序中对某些数据条目的高级搜索功能，服务器对某些数据类型提供了封装程度较低请求的方式，即允许部分客户端应用程序自行合成并向服务器提供用于数据获取的 SQL 查询 SELECT 调用参数中的 WHERE 子句。值得注意的是，该特性因为受到的自动检查、合法性担保较少而具有一定的危险性，不合理的使用可以引发较严重的安全隐患。因此，只有特定商家端的功能将会使用到这个 API 终点，并且子句的合成也会（在客户端）受到一定的规范和限制。

#### 4.1.3 自动部署

不难发现，该设计所实现功能板块领域跨度较大，在原理、技术路线和使用环境要求上不尽相同，甚至在某些情况下相差甚远，所以部署的难度和复杂程度是比较高的。因此，该设计包含一个利用与服务器相似的技术路线开发的自动服务器部署应用程序 RmInit。

该程序具备利用系统命令执行和命令输出检测该设计中各个功能模块在该程序所运行的设备上运行所依赖的第三方软件库或执行环境是否正确安装的功能；具备使用系统自带下载工具（如 `wget`）自动从互联网上下载对应该程序所运行的设备的体系结构的 `sqlite` 应用程序合集，并且在部署的系统上运行数据库初始化脚本的功能；能够按照预先准备的应用程序项目列表逐个运行对应的编译任务并复制编译输出的二进制程序，支持使用 `uv` 托管的 Python 项目、使用 `Gradle` 项目管理工具的 Java-Kotlin 项目、基于 `CMake` 项目管理工具的 C++ 项目和使用 `Cargo` 命令管理的 Rust 可执行程序项目目标。

为了适应在多设备分布式系统中正确分配各设备功能特性、规避在特定平台无法完成编译任务的应用程序，RmInit 采用了 clap 命令行参数分析器，支持利用命令行参数控制大部分功能的包含与否、部分配置文件的覆盖与否。

此外，该应用程序还实现了可选的配置文件升级功能。配置文件升级是一个递归的过程，每一次调用都由新配置和旧配置（或者它们的一部分）参与。若新旧配置均为基本值类型且类型一致，则旧配置受到保留。若新旧配置类型、结构不一致，则使用新配置取代之。若新旧配置为键值表，则利用旧表内容覆盖新表，每个元素的覆盖递归执行该过程。

#### 4.1.4 服务依赖管理

服务器中不同应用程序可能存在一定依赖关系，比如 RmService 依赖于 rqlite 关系数据库管理系统，此时合理安排各个应用程序或后台服务的启动顺序对系统整体稳定程度将会有所帮助。因此，服务器工具软件中包含 RmGuard 服务器运行时依赖管理，支持按网络请求启动或停止相关服务的服务器守护程序。该应用程序内置服务器中不同模块二进制程序之间的依赖关系，并且启动时将会读取对应的配置文件。不论使用任何方式（自启动或按需启动）来唤起任何模块，都会触发对应程序所依赖的模块（若尚未启动）的启动过程。

#### 4.1.5 图像存储

为了顺利在较为受到限制的存储空间内存放潜在的海量商品图像，并且改善各应用程序获取商品图片的速度、降低网络带宽压力，服务器具备自动对图像进行缩放和压缩的功能。服务器所接受的图片（已编码的数据）首先会被解码，然后将被在保持图片原本宽高比的情况下，将图片较长一边的长度限制在 300 以内，并按需调整另一边，并利用较为快速的双线性插值办法进行重采样。程序将逐个尝试 100、95、90...60、55 的 JPEG 压缩率对图片进行编码，直到使用了最大压缩率或图像大小小于配置文件制定阈值为止。

### 4.2 管理界面

为了充分利用面向桌面型计算机开发的应用程序的操作效率高的特性，管理界面模块采用如图 4.4 所示的按标签页分文档的 UI 组织方式，其中大部分标签页采用主要内容板块加侧边栏展示额外信息的布局方式，以增强对键盘鼠标操作的友好性。鉴于许多种类的零售业务资料可以被抽象为同一种类型的多个不同实例，并且具有同样的对理解数据内容较为重要的字段（如名称、说明），故决定以表格方式对其进行整理。管理界面具备如下的功能：

- 商品管理

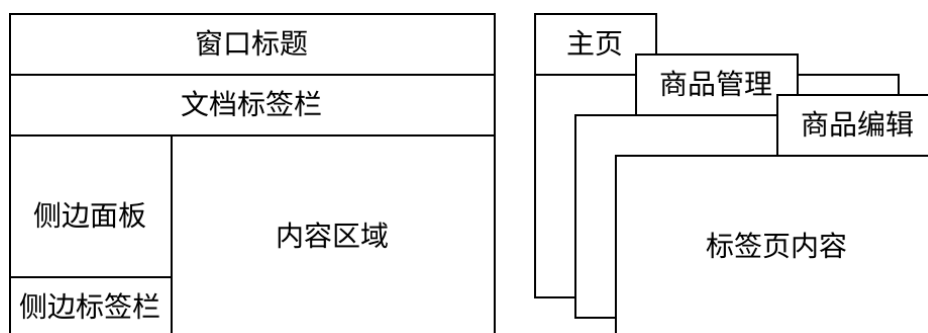


图 4.4 左侧：桌面商家端的用户界面布局；右侧：多个文档标签的实例

- 创建或编辑商品
- 检索、查看和管理既有商品
- 库存管理
  - 管理、预览和执行进货计划
  - 查看仓储情况
  - 管理仓储审计批次
- 编辑推荐商品
- 管理商品图片

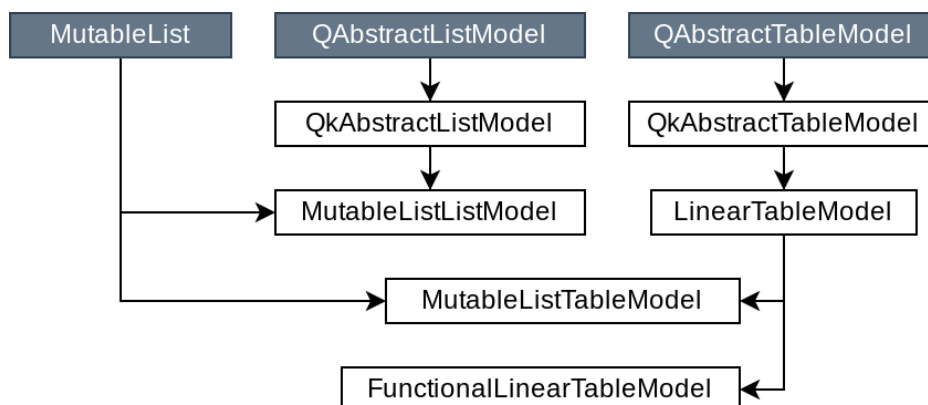


图 4.5 用于表格组件的模型类继承关系：灰色框代表第三方库或开发环境带有的类，箭头指向子类或者实现该接口的类。

不同表格模型（向表格组件提供数据的对象）类之间的继承关系如图 4.5 所示，其中 `MutableListListModel` 和 `MutableListTableModel` 在表格模型中整合了可变不定长数组的特性，可以将开发模型中的对象数组与对应对象集合在 UI 上的展示相同步，避免繁重的界面开发任务，以此简化数据映射的开发复杂程度。



此外，管理界面的许多作为既有信息展示方法的表格组件还具备通过键盘快捷键或右键菜单进行搜索结果整理、复制或粘贴的功能，从业者可以通过在不同的搜索结果、功能模块之间粘贴剪贴板中编码的对象引用来实现较为复杂的查询和修改。例如若需要查询某几种商品的库存情况，可以首先在“商品管理”中对所需商品进行检索，再在该界面内复制所需商品，在库存查询界面粘贴，即可查看对应的库存情况；若需要在编辑商品信息的过程添加图片，可以在图片管理界面检索到所需的图片，在将图片（的引用）复制到商品编辑所对应的标签页。

#### 4.2.1 高级搜索

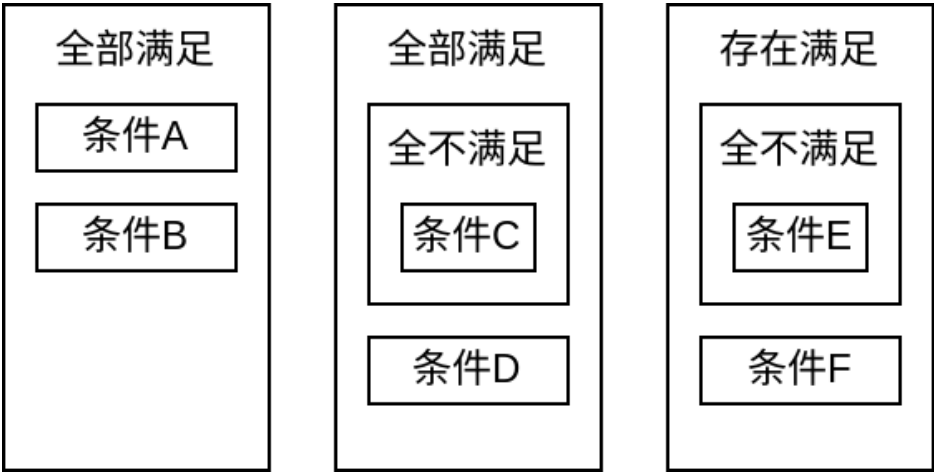


图 4.6 复合查询条件示例：三个条件分别等价于  $A \wedge B$ ， $\neg C \wedge D$  和  $E \vee F$

零售业务运作的时候，从业者往往需要从数量较为庞大的数据中筛选出所需的一小部分。为了满足这样的需要，管理界面对某些数据类型提供基于布尔代数思想的复合查询条件设计工具。因此从业者并不需要编写布尔代数表达式或 SQL WHERE 子句就能进行条件复杂查询操作，通过使用鼠标在应用程序提供的用户界面上操作，从业者可以自由组合以下几种类型的条件（Criterion）：

- 字符串字段：前缀为、后缀为、等价于或包含指定字符串
- 数值字段：等于、大于、小于指定数值
- 日期时间字段：早于、晚于指定日期时间组合
- 编号字段：包含于指定正整数集合
- 以上条件的集合：其中元素全部符合、任意符合、全不符合、任意不符合

输入完成之后，若应用程序没有检测到非法值，将会生成对应的 SQL WHERE 子句并发送到服务器，服务器将按照对应语句运行 SQL 查询并返回结果。例如查询名

称 (name) 包含 “可乐” 二字, 并且于 2025 年 4 月创建 (c\_time) 的商品, 与以下 SQL WHERE 子句近似的语句将会被产生:

```
(name like '%可乐%' and c_time > 1743350399 and c_time < 1746028800)
```

4.2.2 移动端程序

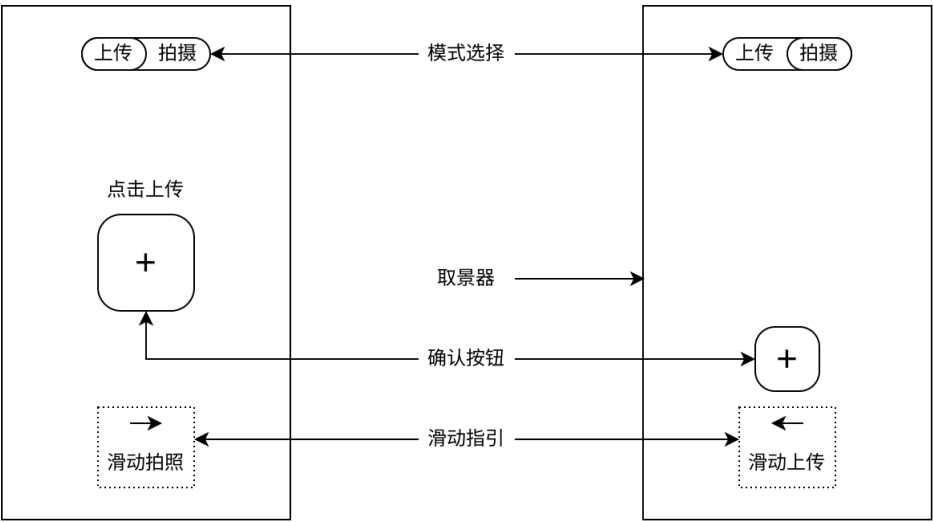


图 4.7 商家端移动应用程序图像上传部分 UI 设计

为了进一步降低营业成本, 该设计包含一个利用 Kotlin 语言和 Jetpack Compose 用户界面库开发的商家端应用程序, 用以简化向系统提供图片的流程。从图 4.7 可见, 用户可以通过屏幕上侧的标签栏或滑动屏幕空白部分来在上传本地图片和拍摄新照片的界面之间切换。在上传界面上, 点击确认按钮可以唤起图片选择界面, 用户可以选择一个或多个图片进行上传; 在拍摄界面, UI 背景部分为取景器画面, 在选择完成拍摄角度之后用户只要点击确认按钮即可拍摄照片并即刻上传。

4.3 结算界面

该设计具有采用 Rust 语言和 iced 实验性用户界面框架构建的用于结算的应用程序。该程序通过与条码识别器、质量传感器和摄像头交互来实现获取商品数据, 其后可以在其用户界面上展示待结算的产品。由于具备相应的软硬件支持, 该程序可以进行不同计价方式的商品 (按件和按质量) 的混合结算, 一定程度减轻了店员结算的压力和用户自主结算的门槛。

4.3.1 条码识别器

为了实现高效识别 (计件) 商品, 结算程序默认通过使用 USB HID 协议通讯的激光条码识别硬件来读取商品条形码。这种硬件将模拟键盘输入条形码内容, 应

用程序在商品结算界面将检测是否在较短时间之内从键盘输入符合 EAN（European Article Number）-13 格式的数据，并在需要时与服务器进行通讯。因为服务器存储条形码的方式是使用字符串而不是数字等其他格式，条形码格式和输入方式方面该设计具有较高的可拓展性，实际部署的情况可以根据不同零售行业细分领域的实际对其进行调整。

#### 4.3.2 质量传感器

通过将质量传感器（带有信息传输接口的“电子秤”）整合到结算设备中，设备将具备直接进行计重商品结算的功能，而不需要额外的硬件或设备来进行称重，也不需要额外编写标签并打印。这样，计重商品的结算也可以完全由顾客自助完成，不需要来自店员的帮助。

该设计理论上对质量传感器的规格和信息传输方式没有要求，只要该硬件提供足以断定价格的精度和稳定程度即可使用。本设计默认使用市面上采用率较高的海芯科技 HX711 电子秤用模数转换器，该芯片成本较为可控，并且在多种平台上具备成熟生态。值得注意的是，HX711 的串行通讯功能对相应频率和延迟的要求较为严格。为了解决此问题，本设计包含 Arduino UNO 3 微控制器开发硬件上采用 C++ 语言和 PlatformIO 嵌入式开发技术编写的驱动程序，将 HX711 的专有输出格式转化为了易于处理的文本格式，并通过 USB 模拟串行接口终端通讯与计算机（和结算程序）相连接和消息传递，以此实现了称重和调零的功能。

## 5 商家端 AI 功能

### 5.1 商品设计辅助

#### 5.1.1 编写文案

#### 5.1.2 建议定价

#### 5.1.3 提供额外参考

### 5.2 销售数据分析

#### 5.2.1 图表生成

#### 5.2.2 图表理解

#### 5.2.3 营业策略建议

### 5.3 库存审计

## 6 顾客端 AI 功能

### 6.1 导购助手

#### 6.1.1 对话型生成式 AI

#### 6.1.2 输出风格

#### 6.1.3 搜索关键词提取

#### 6.1.4 商品搜索

### 6.2 称重商品识别

#### 6.2.1 数据准备

#### 6.2.2 图像处理

#### 6.2.3 AI 图像分类

### 6.3 模糊搜索

#### 6.3.1 商品数据处理

#### 6.3.2 AI 近义词搜寻

#### 6.3.3 词典构建

#### 6.3.4 搜索算法

## 7 实验

### 7.1 商家端

#### 7.1.1 桌面应用

#### 7.1.2 手机应用

#### 7.1.3 数据集管理程序

### 7.2 顾客端

#### 7.2.1 手机应用

#### 7.2.2 结算程序

## 8 讨论

## 9 结论

### 参考文献

- [1] SAMUEL A L. Some Studies in Machine Learning Using the Game of Checkers [J/OL]. IBM Journal of Research and Development, 1959, 3(3):210-229. DOI: 10.1147/rd.33.0210.
- [2] RAINA R, MADHAVAN A, NG A Y. Large-scale deep unsupervised learning using graphics processors[C/OL]//ICML '09: Proceedings of the 26th Annual International Conference on Machine Learning. Montreal, Quebec, Canada: Association for Computing Machinery, 2009: 873-880. <https://doi.org/10.1145/1553374.1553486>. DOI: 10.1145/1553374.1553486.
- [3] KRIZHEVSKY A, SUTSKEVER I, HINTON G E. ImageNet classification with deep convolutional neural networks[J/OL]. Commun. ACM, 2017, 60(6): 84-90. <https://doi.org/10.1145/3065386>. DOI: 10.1145/3065386.
- [4] HE K, ZHANG X, REN S, et al. Deep Residual Learning for Image Recognition [C/OL]//2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2016: 770-778. DOI: 10.1109/CVPR.2016.90.
- [5] VASWANI A, SHAZEER N, PARMAR N, et al. Attention Is All You Need [EB/OL]. 2023. <https://arxiv.org/abs/1706.03762>. arXiv: 1706.03762 [cs.CL].
- [6] Introducing ChatGPT[EB/OL]. (2024-03-13) [2025-04-09]. <https://openai.com/index/chatgpt/>.
- [7] 苏美文, 杨文爽, 李博文, 等. 推动人工智能与实体经济深度融合加快发展新质生产力[J]. 工业技术经济, 2025, 44(04): 32-59.
- [8] 谢捷, 唐声羽, 陈柳钦. 数字化转型视域下人工智能驱动企业新质生产力提升[J]. 南海学刊, 1-15.
- [9] RADFORD A, KIM J W, XU T, et al. Robust Speech Recognition via Large-Scale Weak Supervision: arXiv:2212.04356[EB/OL]. arXiv. (2022-12-06) [2025-04-09]. <http://arxiv.org/abs/2212.04356>. arXiv: 2212.04356[eess].
- [10] 赵树梅, 徐晓红. “新零售”的含义、模式及发展路径[J/OL]. 中国流通经济, 2017, 31(05): 12-20. DOI: 10.14089/j.cnki.cn11-3664/f.2017.05.002.



- [11] 杜睿云, 蒋侃. 新零售: 内涵、发展动因与关键问题[J/OL]. 价格理论与实践, 2017(02): 139-141. DOI: 10.19851/j.cnki.cn11-1010/f.2017.02.038.
- [12] 廖夏, 石贵成, 徐光磊. 智慧零售视域下实体零售业的转型演进与阶段性路径[J]. 商业经济研究, 2019(05): 28-30.
- [13] 王先庆, 雷韶辉. 新零售环境下人工智能对消费及购物体验的影响研究——基于商业零售变革和人货场体系重构视角[J]. 商业经济研究, 2018(17): 5-8.
- [14] 齐欢欢, 惠银银, 郭洋洋. 浅析私域流量时代的直播电商运营[J]. 今日财富, 2020(24): 58-59.

## 致谢