



中南大學  
CENTRAL SOUTH UNIVERSITY

## 系统设计与分析期末报告

姓名： 邓晓姗

学号： 8210231029

学院： 商学院

班级： 信管 T2301

课程： 系统分析与设计

日期： 2025 年 12 月 17 日

指导教师： 毕文杰教授

## 摘要

随着全球化贸易的深入发展，中小微进出口企业面临着日益复杂的财务核算与供应链管理挑战。传统的进销存系统往往与财务软件割裂，导致数据孤岛现象严重，库存成本核算滞后。本报告详细阐述了“Cash Online”——一款面向现代贸易企业的 Web ERP 系统的全生命周期设计与实现。

系统采用前后端分离架构，基于 Vue 3 构建响应式前端，利用 Node.js (Express) 提供高性能 RESTful API，并采用 MySQL 8.0 作为数据持久层。针对核心痛点，本系统设计了一套严格遵循复式记账原则 (Double-Entry Bookkeeping) 的财务引擎。创新性地引入了基于有理数 (Rational Number) 的高精度计算模型以消除浮点误差，并通过数据库事务 (Transaction) 保证了业务单据 (PO/SO) 与财务凭证的强一致性 (ACID)。报告通过 UML 建模方法，全面展示了系统的用例模型、业务序列流程、数据库 ER 设计以及关键算法实现，为系统的工程落地提供了详实的依据。

**系统 ip 地址：**<http://115.190.225.93/>

**代码仓库：**<https://github.com/Elnor-chan/Cash-Online-main>

## 目录

<b>I</b>	<b>系统概述</b>	3
<b>II</b>	<b>系统愿景</b>	3
II-A	当前痛点与机会 . . . . .	3
II-B	产品建设目标 . . . . .	3
II-C	核心价值思路 . . . . .	4
<b>III</b>	<b>业务用例图</b>	5
<b>IV</b>	<b>业务序列图</b>	6
IV-A	改进前 . . . . .	6
IV-B	改进后 . . . . .	11
<b>V</b>	<b>系统用例图</b>	16
<b>VI</b>	<b>用例规约</b>	17
VI-A	基础资料配置 . . . . .	17
VI-B	采购商务管理 . . . . .	20
VI-C	销售商务管理 . . . . .	22
VI-D	库存物流管理 . . . . .	24
VI-E	财务核算中心 . . . . .	26
<b>VII</b>	<b>类图</b>	28
<b>VIII</b>	<b>E-R 图</b>	28
VIII-A	数据库逻辑架构说明 . . . . .	29
VIII-A1	核心账务引擎模块 . . . . .	29
VIII-A2	供应链与单据模块 . . . . .	29
VIII-A3	商业实体模块 . . . . .	29
VIII-A4	基础设置模块 . . . . .	29

<b>IX</b>	<b>数据库表结构字段列表</b>	30
<b>X</b>	<b>序列图</b>	33
<b>XI</b>	<b>结论</b>	37

## I. 系统概述

本系统名为”Cash online 财务一体化系统”。Cash Online 旨在解决传统进销存系统与财务软件脱节的问题。通过将采购、销售、库存与总账模块深度集成，系统实现了“业务驱动财务”的自动化流程，消除了企业内部的数据孤岛。

系统采用现代化的前后端分离架构，确保了高并发下的响应速度与未来的水平扩展能力。

- **前端 (Presentation Layer)**: 基于 Vue 3 + Vite 构建单页应用 (SPA)。利用 Pinia 进行全局状态管理，确保多组件间的数据同步；采用 Element Plus 组件库，提供符合企业级标准的交互体验。
- **后端 (Application Layer)**: 采用 Node.js + Express 搭建高性能 RESTful API 服务。后端负责处理核心业务逻辑、执行 JWT 身份鉴权，并作为网关统一管理数据请求。
- **数据层 (Persistence Layer)**: 使用 MySQL 8.0 关系型数据库。利用 InnoDB 引擎的事物特性确保 ACID (原子性、一致性、隔离性、持久性)，特别是在金额存储上，使用了高精度整数 (分子/分母格式) 以彻底避免浮点计算误差。

## II. 系统愿景

本项目的核心愿景是打造一个专为现代中小微进出口贸易企业量身定制的、**业务财务深度融合**的数字化管理平台。Cash Online 不仅仅是一个记录数据的工具，更致力于成为企业经营决策的“单一事实来源 (Single Source of Truth)”。

通过技术手段重构传统的业务流程，系统旨在实现以下三个维度的战略目标：

### A. 当前痛点与机会

在传统的企业管理模式中，业务部门（采购、销售、仓储）与财务部门往往使用异构系统，导致了显著的管理瓶颈：

- **数据孤岛效应**: 业务单据（如发货单）需要人工传递并重复录入财务系统，导致信息流转滞后，且极易产生人为录入错误。
- **库存成本黑箱**: 由于缺乏实时的进销存核算，企业管理者难以准确掌握当前的库存资金占用情况，导致存货积压或断货风险。
- **核算精度缺失**: 通用软件在处理跨境贸易复杂汇率及分摊计算时，常因浮点数精度问题导致“一分钱误差”，增加了财务对账的难度。

### B. 产品建设目标

Cash Online 旨在通过“自动化”与“强一致性”的技术架构解决上述问题，构建一个闭环的生态系统：

- 1) **流程自动化 (Process Automation)**: 实现“业务单据即凭证”。当采购入库或销售出库发生时，系统自动根据预设会计准则生成对应的财务凭证，彻底由“事后核算”转向“事中控制”。
- 2) **数据实时化 (Real-time Visibility)**: 为管理者提供实时的财务报表（资产负债表、利润表）。每一笔交易的发生都能即时反映在总账数据中，支持快速的市场响应与决策。
- 3) **计算精准化 (Computational Precision)**: 建立基于有理数 (Rational Number) 的底层计算模型，确保在任何复杂的折扣、税率及分摊计算场景下，资金数据均能保持绝对的数学精确性。

### C. 核心价值思路

本系统的设计遵循 **ACID** 原则，旨在为企业提供以下核心价值：

- **合规性 (Compliance)**: 内置符合会计准则的复式记账引擎，规范企业的财务作业流程。
- **可追溯性 (Traceability)**: 提供全链路的单据穿透查询能力，从财务报表的某一科目余额可直接追溯至原始的业务单据（如采购订单），满足审计需求。
- **扩展性 (Scalability)**: 基于前后端分离的微服务化设计理念，支持未来对接第三方支付网关、税务系统或物流平台，伴随企业共同成长。

### III. 业务用例图

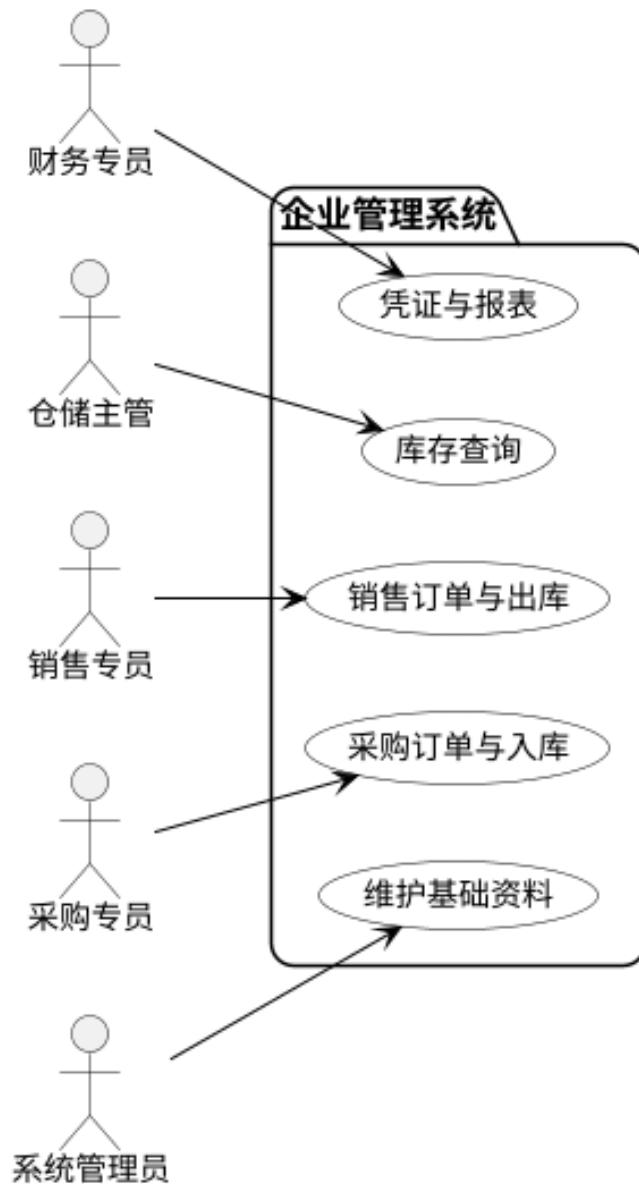


图 1: 业务用例图

#### IV. 业务序列图

A. 改进前

### 业务序列1（改进前）：基础设置与档案维护

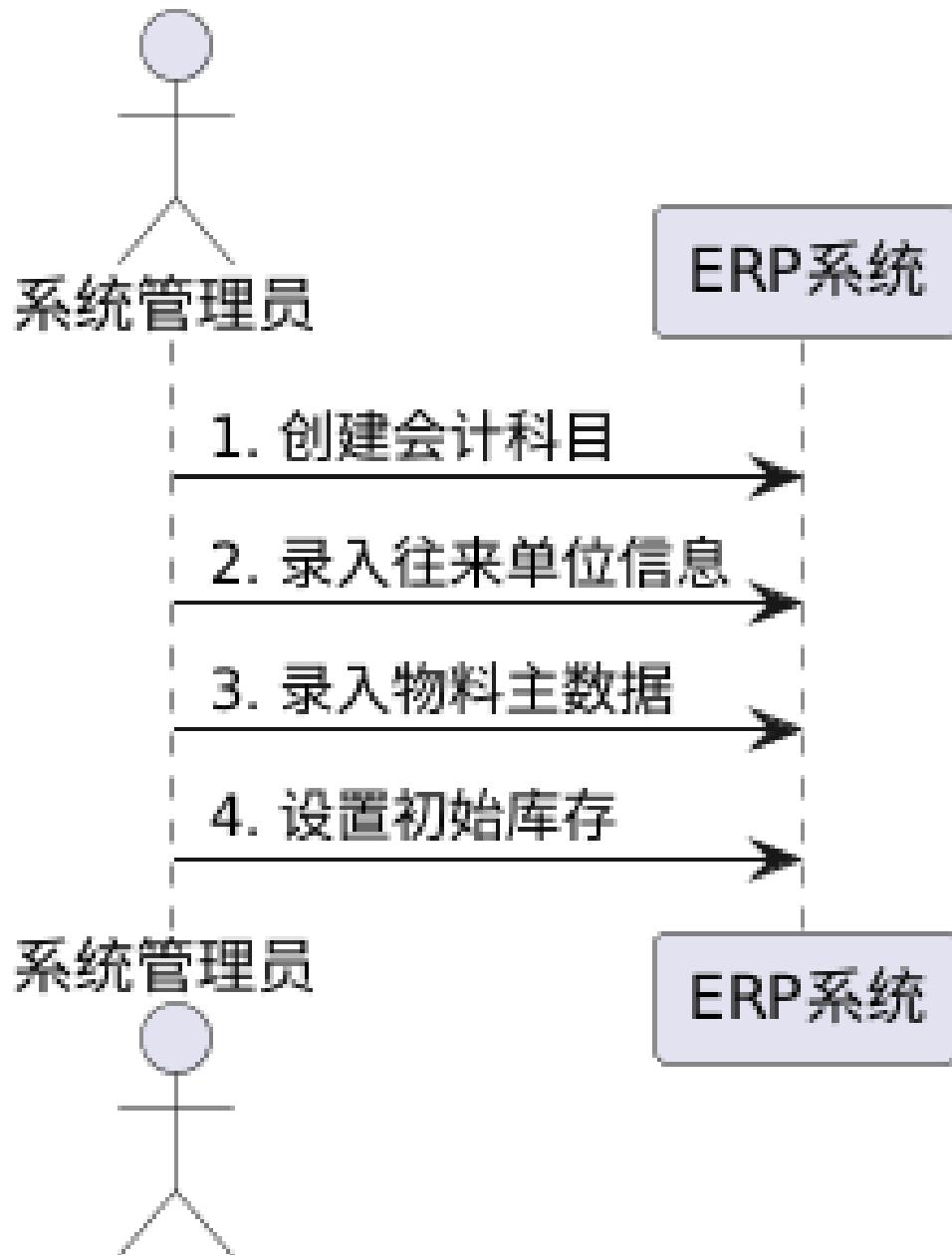


图 2: 改进前基础设置业务序列图

## 业务序列2（改进前）：采购订单与入库

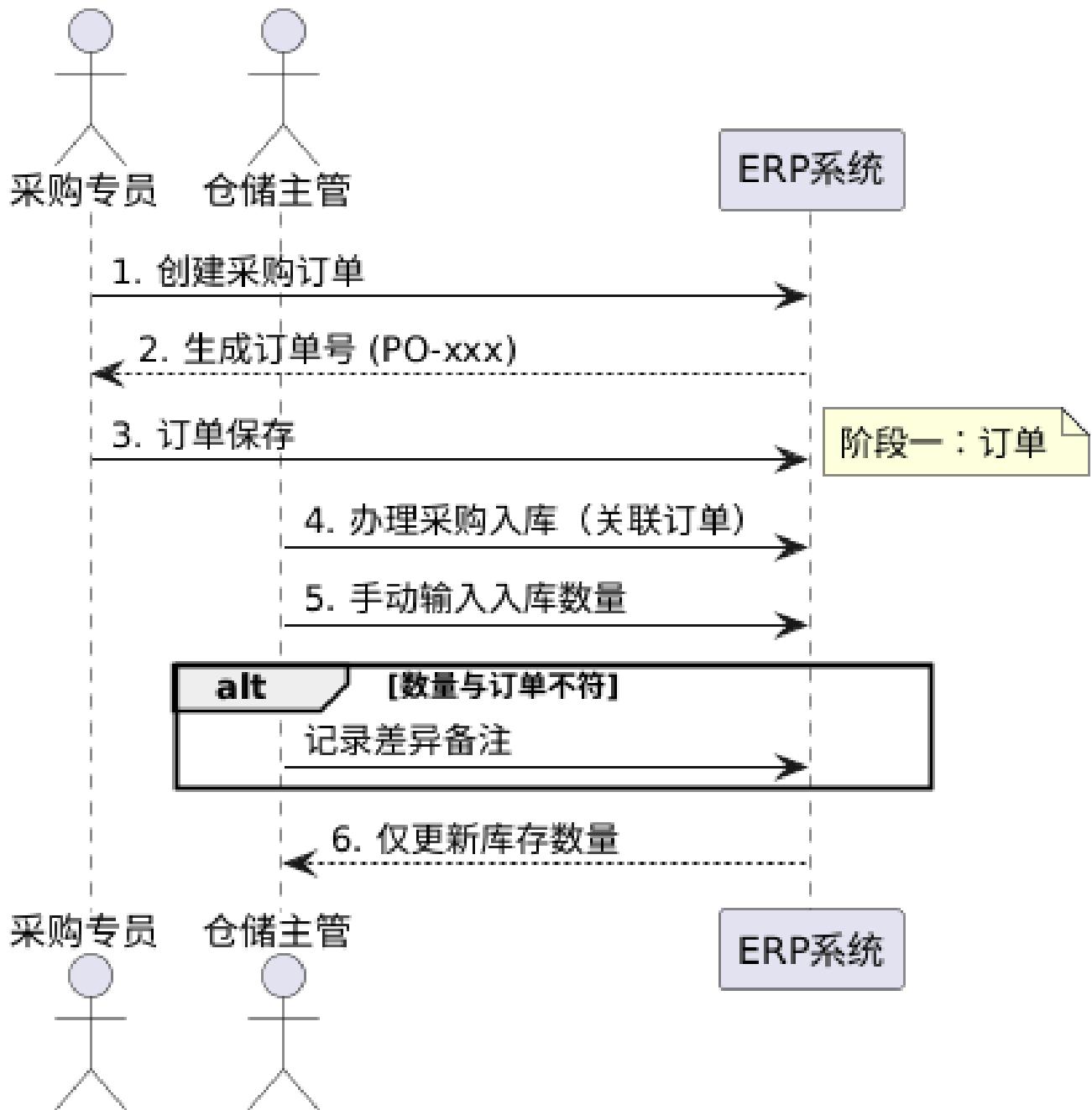


图 3: 改进前采购业务序列图

### 业务序列3（改进前）：销售订单与出库

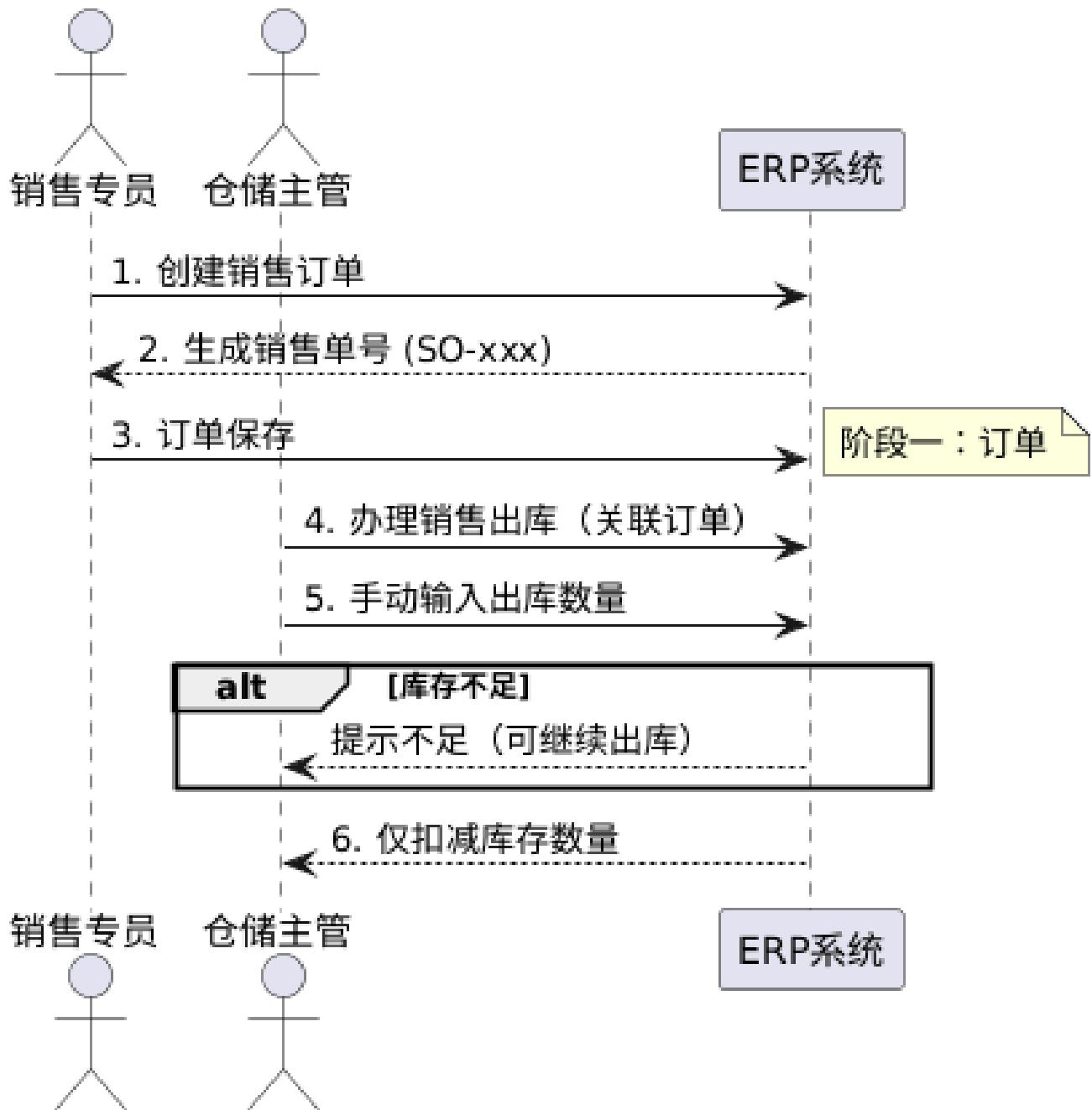


图 4: 改进前销售业务序列图

## 业务序列4（改进前）：库存查询与监控

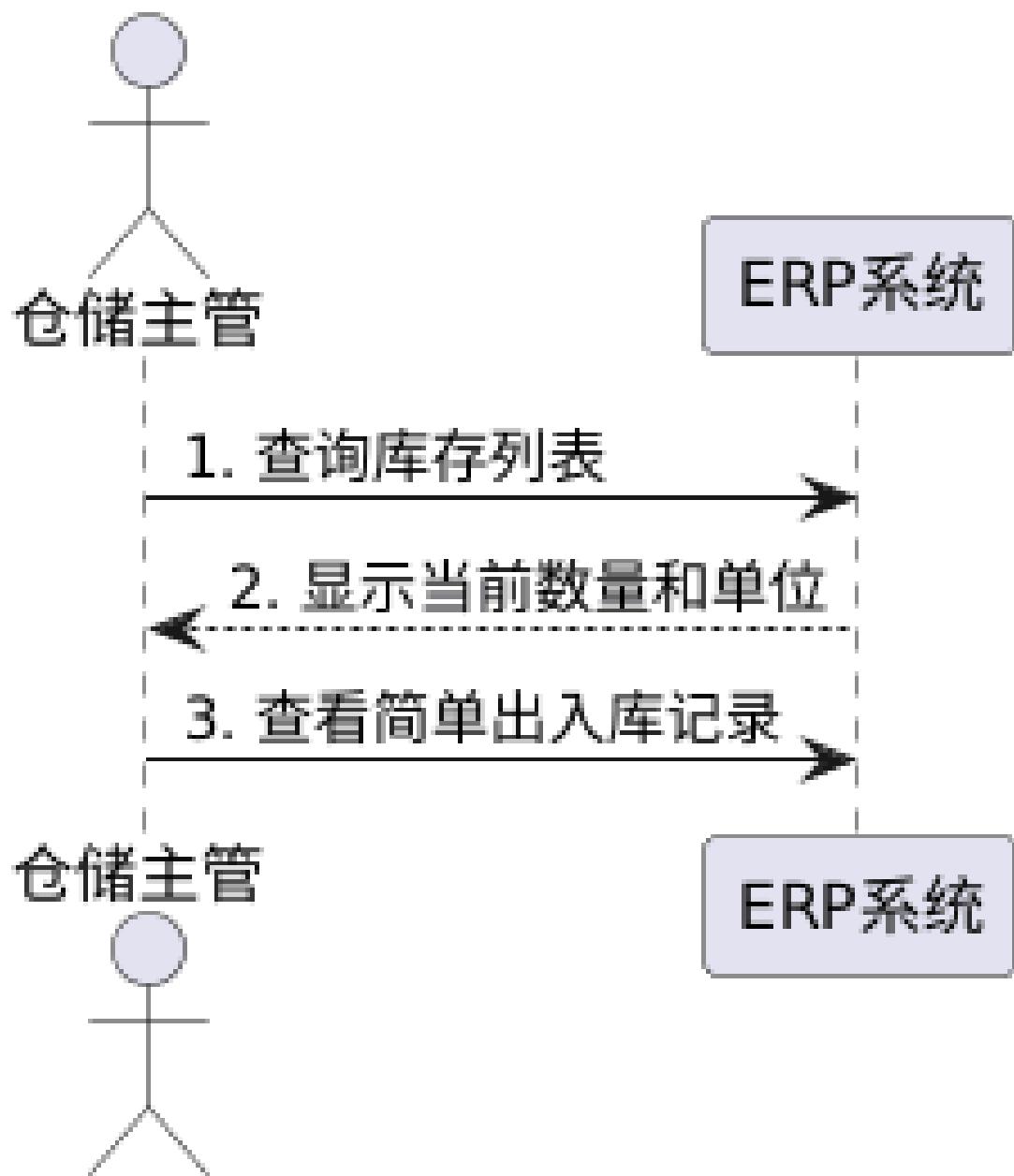


图 5: 改进前库存管理业务序列图

## 业务序列5（改进前）：凭证管理与报表

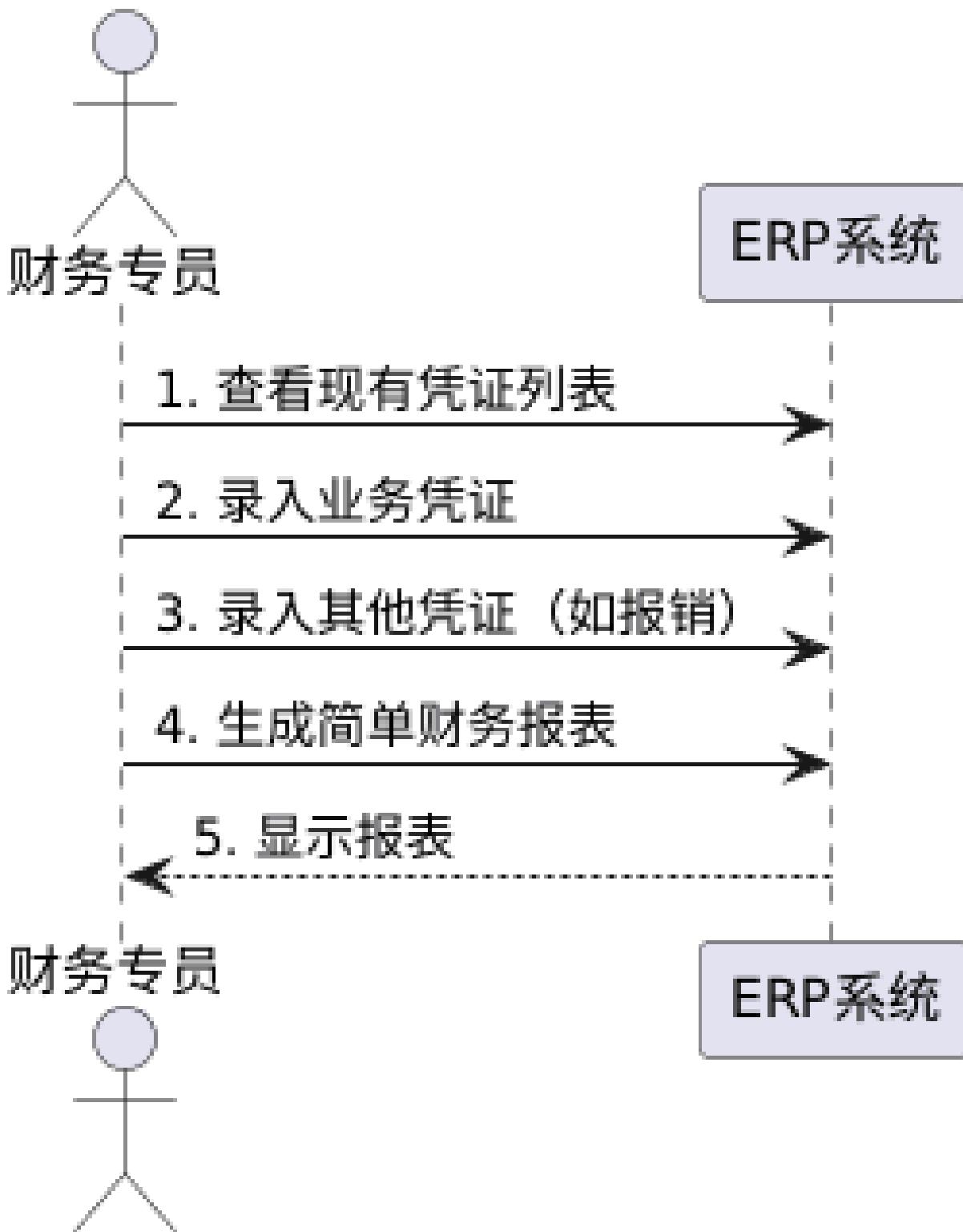


图 6: 改进前财务管理业务序列图

## B. 改进后

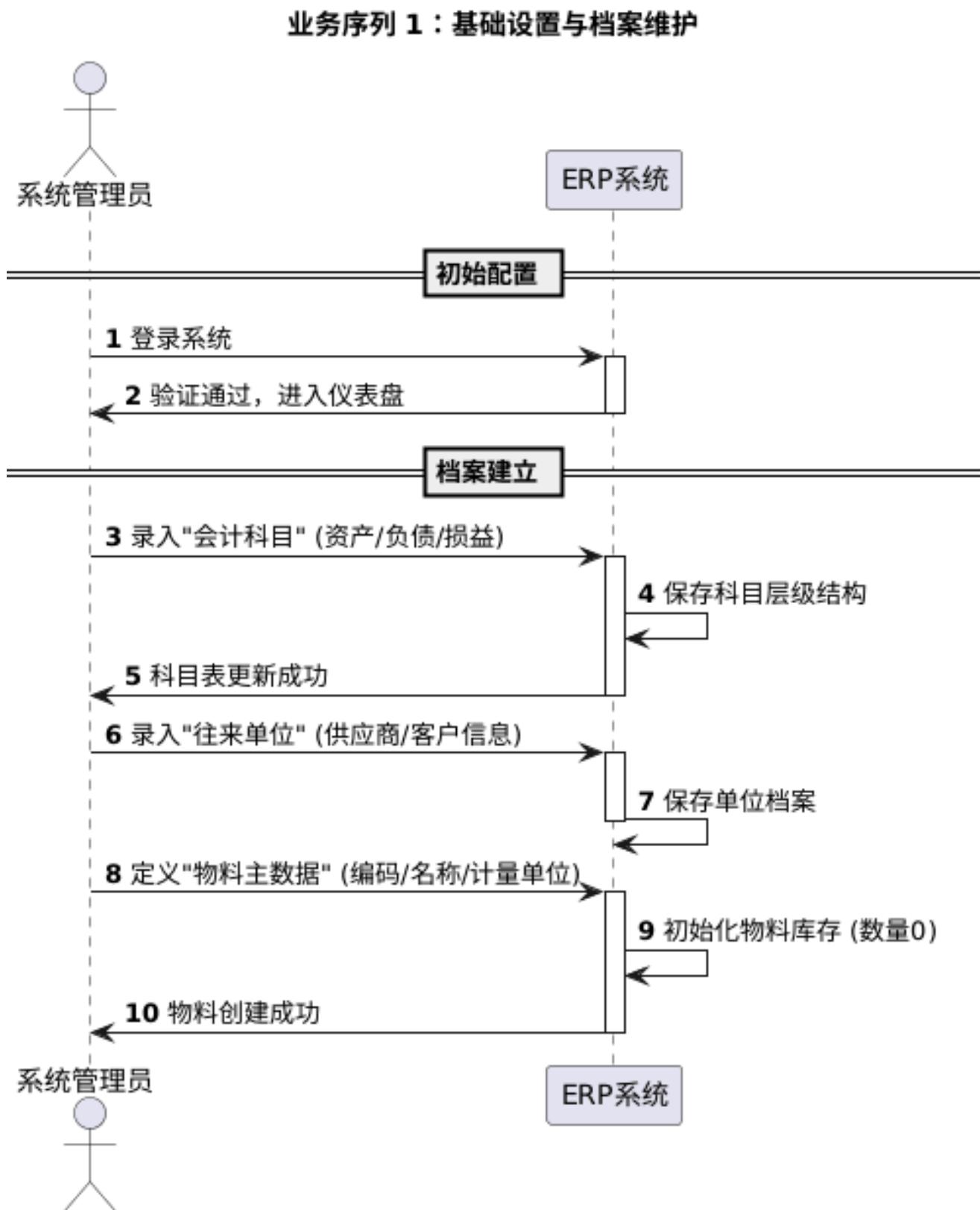


图 7: 改进后基础设置业务序列图

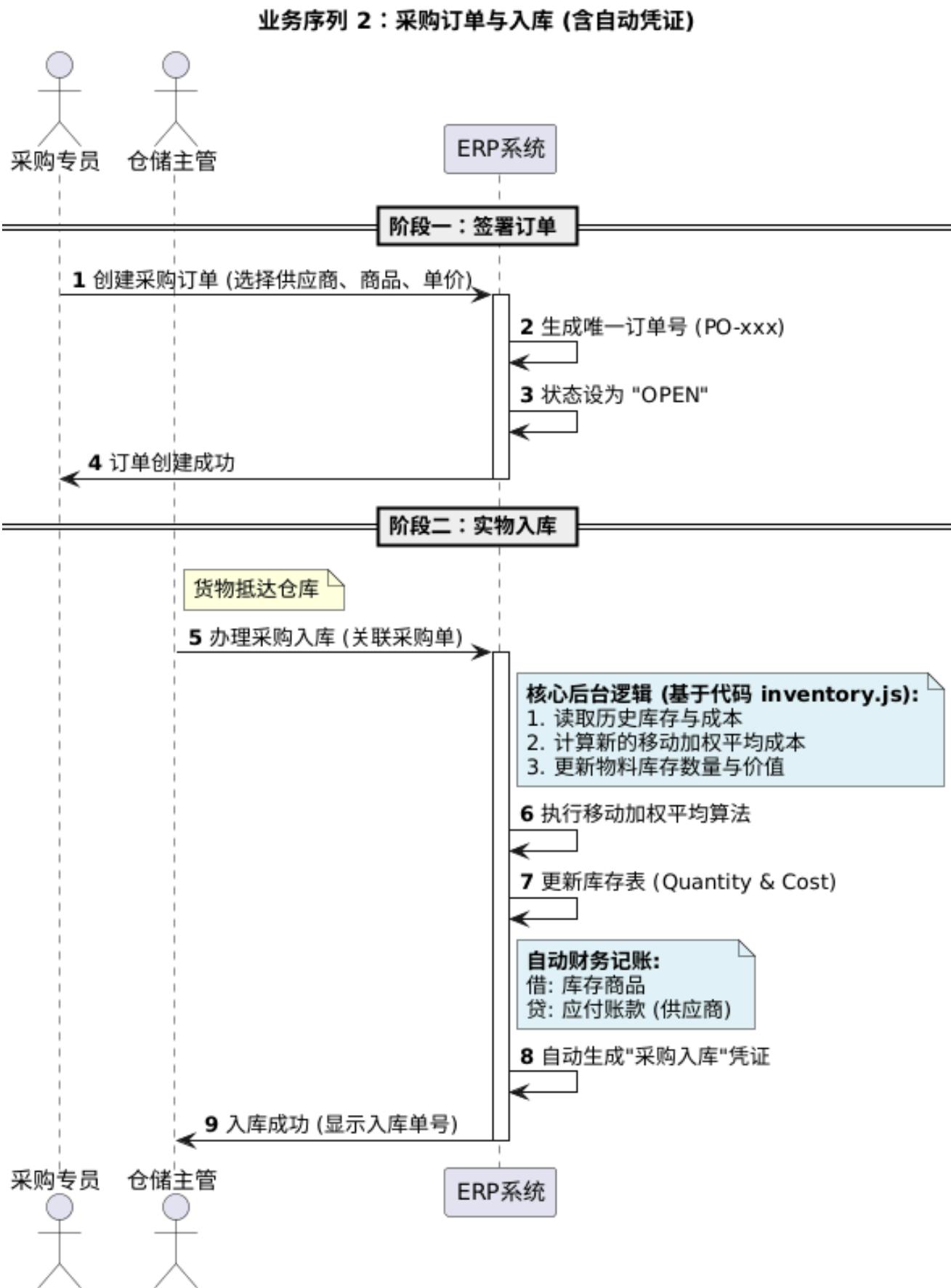


图 8: 改进后采购业务业务序列图

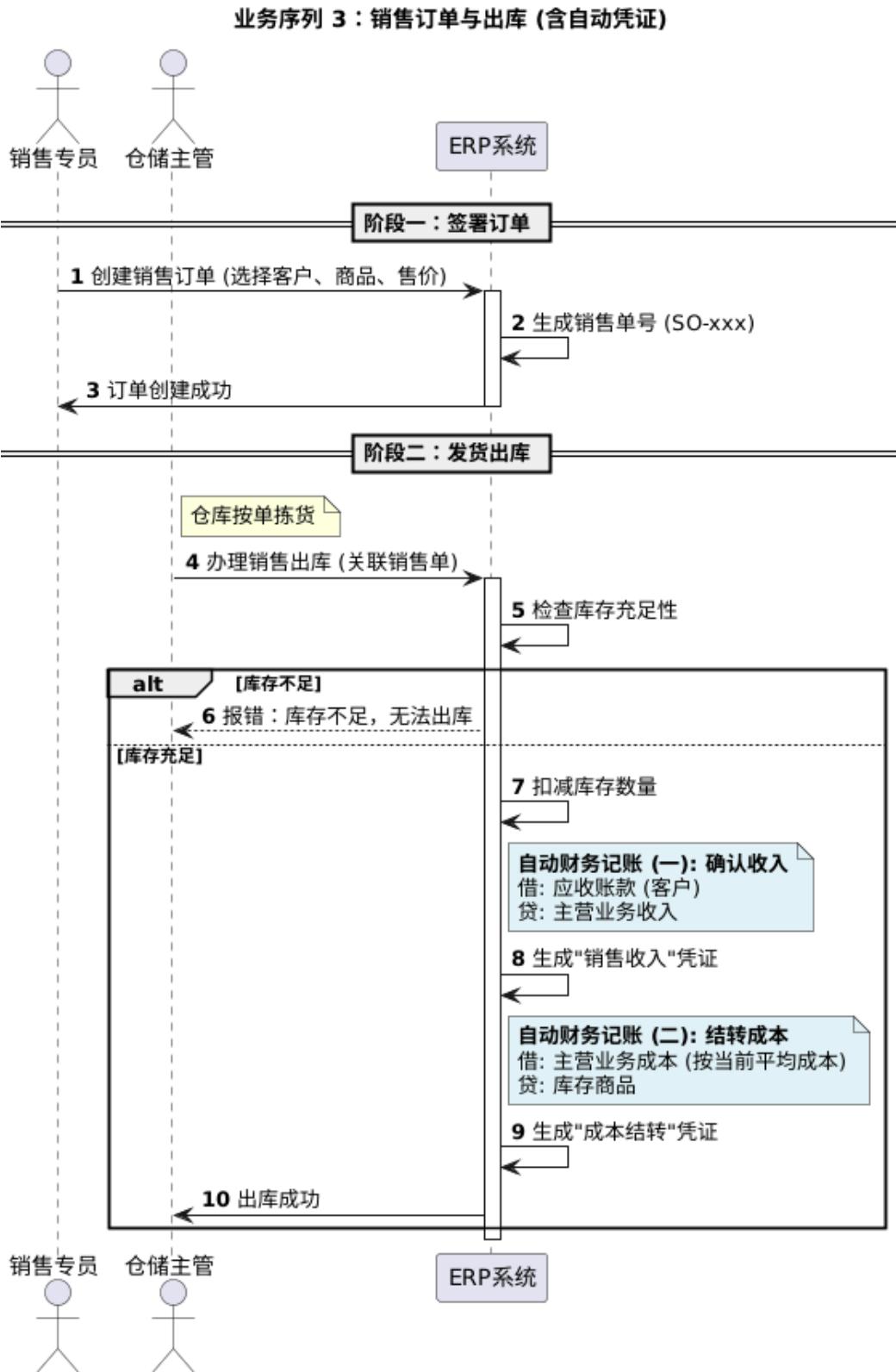


图 9: 改进后销售业务业务序列图

### 业务序列 4：库存查询与监控

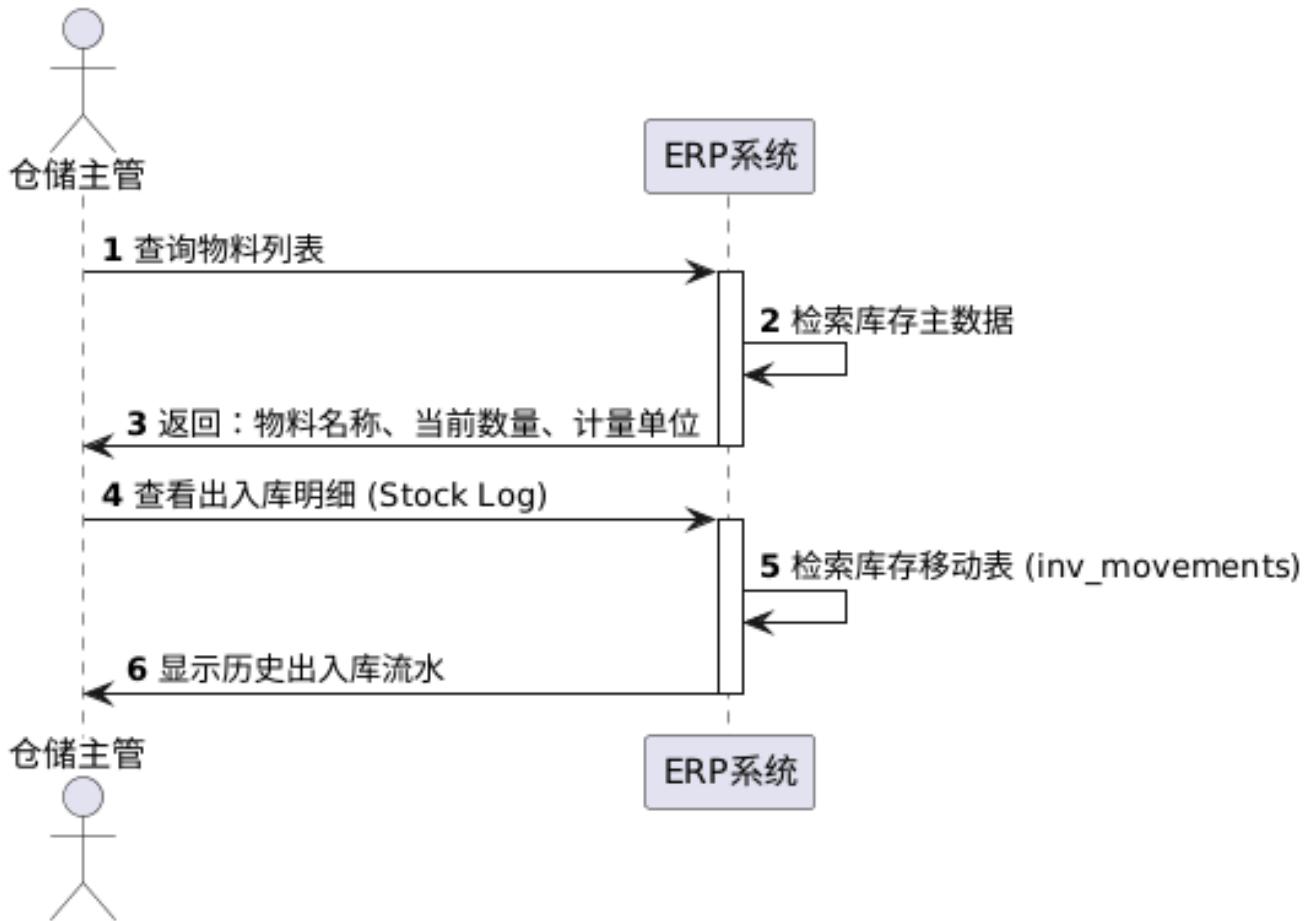


图 10: 改进后库存管理业务序列图

## 业务序列 5：凭证管理与报表

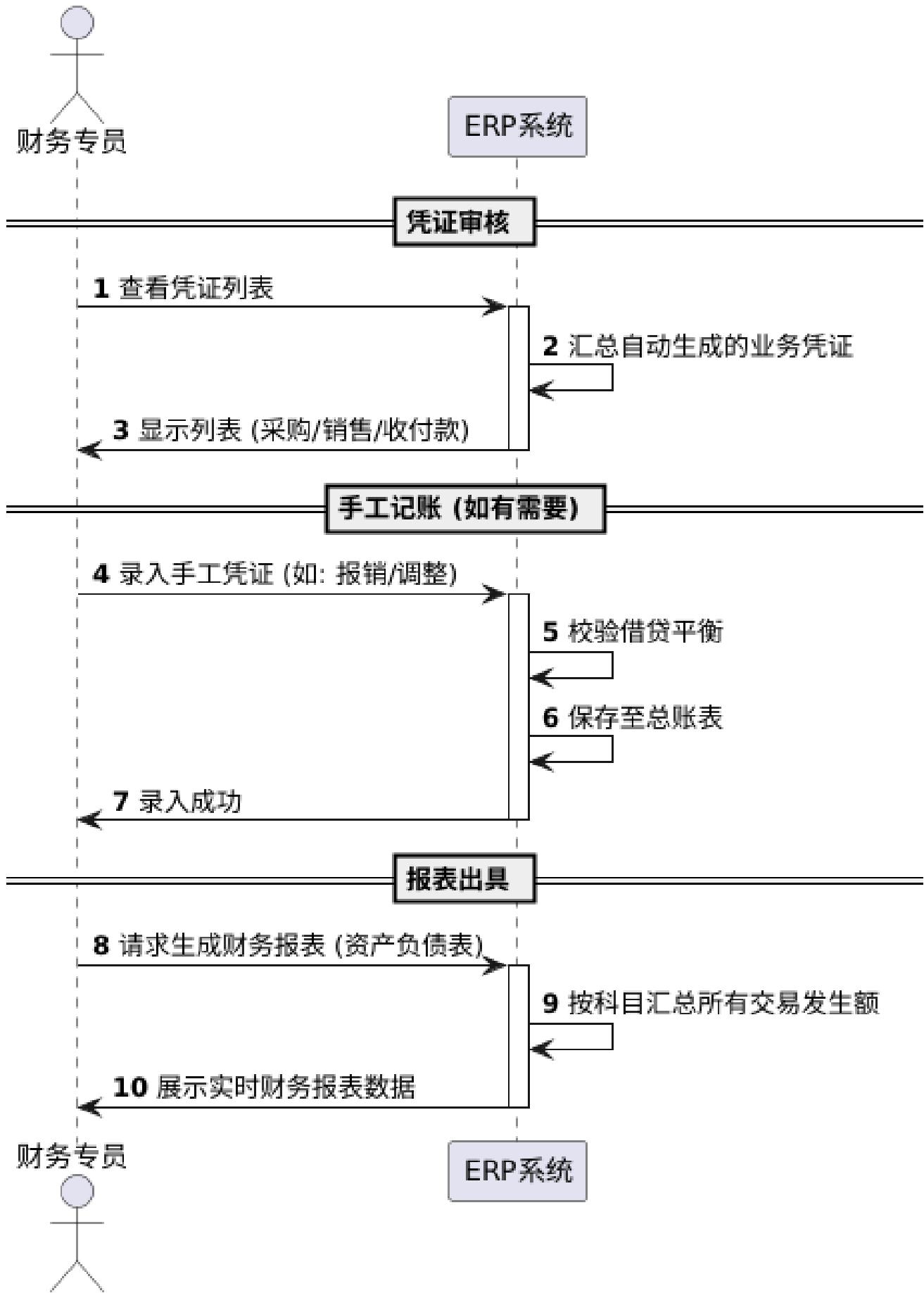


图 11: 改进后财务管理业务序列图

## V. 系统用例图

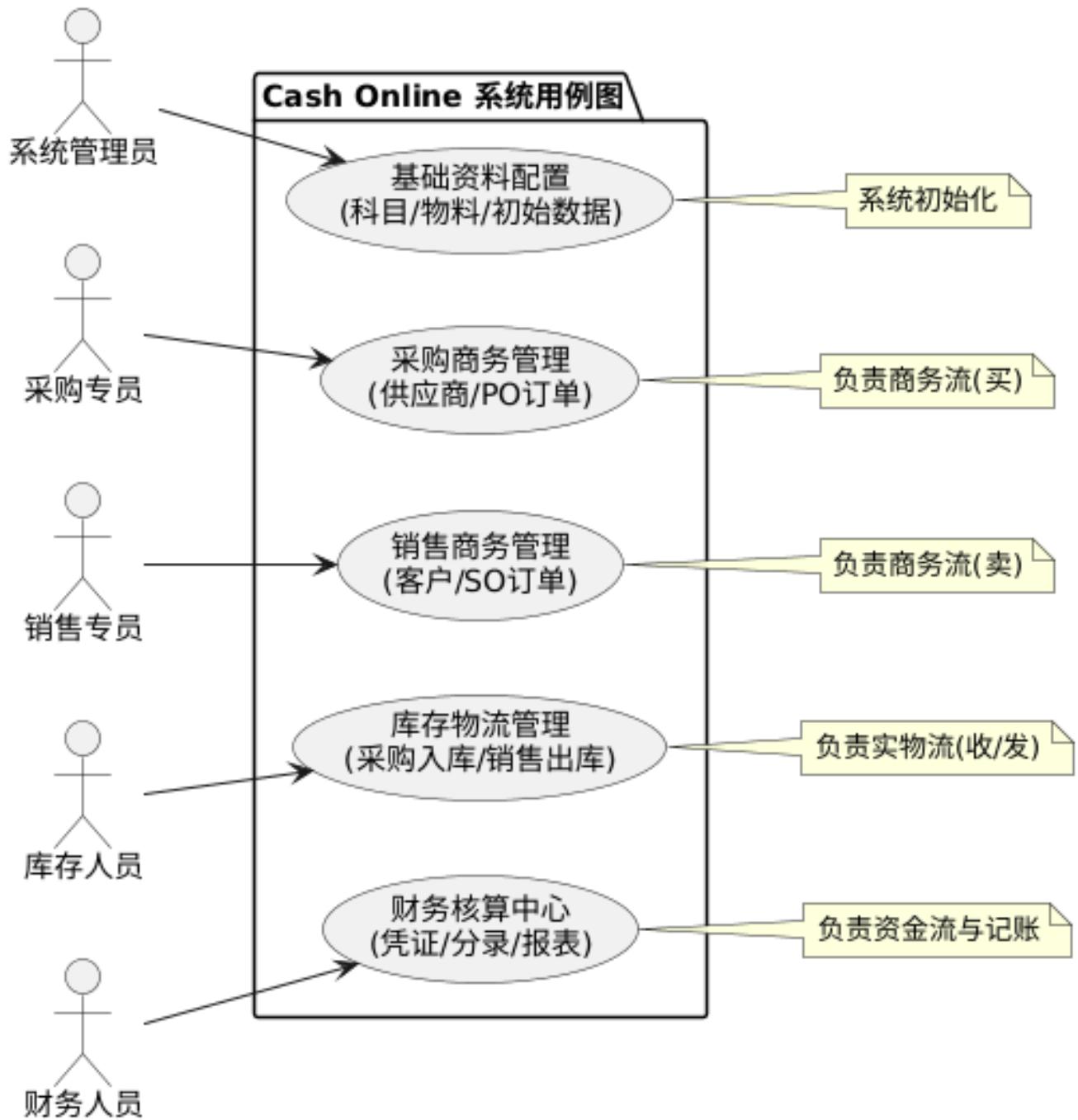


图 12: 系统用例图

## VI. 用例规约

本章详细阐述系统五大核心模块的业务逻辑。所有的业务规则与数据处理流程均基于数据库设计的完整性约束 (Referential Integrity)，确保每一笔业务操作都能准确映射到底层的物理数据表中。

### A. 基础资料配置

#### **UC1：多币种与商品体系构建 (Commodity & Currency Setup)**

用例编号	SYS-CONF-001
用例名	多币种与商品体系构建 (Commodity & Currency Setup)
执行者	系统管理员（主）
前置条件	<ul style="list-style-type: none"> <li>• 系统数据库已完成初始化</li> <li>• 系统管理员拥有最高配置权限 (System Admin 角色)</li> <li>• 基础数据配置模块已启用</li> </ul>
后置条件	<ul style="list-style-type: none"> <li>• 系统成功保存新增或修改的商品/货币定义 (commodities 表)</li> <li>• 系统成功构建或更新会计科目树结构 (accounts 表)</li> <li>• 系统成功保存历史汇率或商品价格记录 (prices 表)</li> <li>• 所有配置数据通过一致性校验，确保后续财务核算可正常进行</li> </ul>
涉众利益	<ul style="list-style-type: none"> <li>• 系统管理员：能够快速、准确地完成基础数据配置，减少后期核算错误</li> <li>• 会计人员：获得清晰、一致的货币与商品定义，保证多币种账务处理和库存估值准确</li> <li>• 财务报表使用者 (老板、股东、审计机构)：资产负债表、利润表等报表数据可靠，汇率与价格来源可追溯</li> <li>• 系统开发与维护人员：高精度、无浮点误差的设计降低后期 bug 发生概率</li> </ul>
基本路径	<ol style="list-style-type: none"> <li>1) 系统管理员登录系统，进入“基础数据配置”模块，选择“商品与货币管理”子模块</li> <li>2) 管理员点击“新增商品/货币”按钮，系统显示商品/货币定义表单</li> <li>3) 管理员选择命名空间 (Namespace)：             <ul style="list-style-type: none"> <li>• 若为法定货币，选择“CURRENCY”</li> <li>• 若为实物商品，选择“PRODUCT”</li> </ul> </li> <li>4) 管理员录入助记码 (mnemonic)：             <ul style="list-style-type: none"> <li>• 货币类遵循 ISO 4217 标准 (如 CNY、USD)</li> <li>• 商品类录入内部 SKU 编码 (如 IPHONE-15-PRO)</li> </ul> </li> <li>5) 管理员设置最小单位精度 (fraction)：             <ul style="list-style-type: none"> <li>• 例如货币设置为 100 (支持两位小数)，商品设置为 1 (不可分割)</li> </ul> </li> <li>6) 系统校验输入合法性，通过后保存至 commodities 表，提示“商品/货币定义成功”</li> <li>7) 管理员进入“会计科目管理”子模块，构建或编辑科目树</li> <li>8) 管理员为每个科目设置：             <ul style="list-style-type: none"> <li>• 科目名称、代码</li> </ul> </li> </ol>

- 科目类型 (account\_type: ASSET、LIABILITY、EQUITY、INCOME、EXPENSE)
  - 上级科目 (parent\_guid, 实现层级嵌套)
  - 本位币绑定 (通过 commodity\_guid 关联已定义的货币)
- 9) 系统保存科目信息至 accounts 表, 提示“科目保存成功”
- 10) 管理员进入“汇率与价格维护”子模块
- 11) 管理员选择基准货币与目标商品/货币, 录入汇率或价格
- 12) 系统要求以分数形式输入 (value\_num / value\_denom), 例如汇率 7.1234 录入为 71234/10000
- 13) 管理员选择来源 (source: 如 CentralBank、UserEntry、MarketAverage)
- 14) 系统保存记录至 prices 表, 提示“价格记录保存成功”
- 15) 管理员完成所有配置后退出, 系统记录操作日志

**扩展路径 (出现)** 1) 助记码 (mnemonic) 重复

**问题处理** A: 系统提示“助记码已存在, 请使用其他编码”, 管理员修改后重新提交

- 2) 命名空间选择错误导致后续无法区分货币与商品

A: 系统在保存前强制校验命名空间与助记码格式一致性 (如 CURRENCY 命名空间必须为三位 ISO 代码), 不一致时拒绝保存并给出明确错误提示

- 3) 分数形式汇率输入不规范 (例如分母为 0)

A: 系统前端校验分母不能为 0, 后端再次校验, 拒绝保存并提示“分母不能为零”

- 4) 科目关联了尚未定义的 commodity\_guid

A: 系统在外键保存时校验 commodity\_guid 是否存在, 不存在则提示“请先定义对应的货币/商品”

- 5) 循环科目层级 (子科目直接或间接指向自身或上级形成环)

A: 保存时系统检测层级结构是否有环, 若有则拒绝保存并提示“检测到循环层级, 请检查上级科目设置”

**字段列表**

- commodities 表主要字段:

guid (主键) + namespace (CURRENCY/PRODUCT) + mnemonic (助记码) + fullname (全称) + fraction (最小单位精度) + description (描述)

- accounts 表主要字段:

guid (主键) + name (科目名称) + code (科目代码) + account\_type (科目类型) + parent\_guid (上级科目) + commodity\_guid (本位币外键) + description (描述)

- prices 表主要字段:

guid (主键) + commodity\_guid (目标商品/货币) + currency\_guid (基准货币) + date (生效日期) + value\_num (分子) + value\_denom (分母) + source (来源, 如 CentralBank/UserEntry)

**业务规则**

- 同一 namespace 内 mnemonic 必须全局唯一

- CURRENCY 命名空间的 mnemonic 必须符合 ISO 4217 三位字母代码

- fraction 必须为正整数, 且对于常见法定货币建议默认为 100、1000 或 10000

- 所有价格与汇率必须以有理数 (value\_num/value\_denom) 形式存储, 禁止使

### 用浮点数

- 资产、负债、权益类科目必须绑定货币类型的 commodity；收入、费用类科目可根据需要绑定货币或商品
  - 科目树不允许出现循环依赖
- 质量需求**
- 可靠性：配置数据保存前必须通过完整性校验，支持多人并发配置时加锁防止冲突
  - 性能：单条记录保存时间  $\leq 500\text{ms}$ ；科目树全量加载 ( $\leq 1000$  条科目)  $\leq 2$  秒
  - 可支持性：关键配置错误需提供明确中文错误提示，便于快速定位；重大配置变更需记录操作日志，便于回滚与审计
- 设计约束**
- 采用网页版前端（支持主流浏览器）
  - 后端使用精确有理数存储价格与汇率，彻底避免浮点精度问题
  - 所有配置操作需记录审计日志（谁、在何时、改了什么）

## B. 采购商务管理

### UC2：采购订单全生命周期管理 (Purchase Order Lifecycle)

<b>用例编号</b>	SCM-PUR-001
<b>用例名</b>	采购订单全生命周期管理 (Purchase Order Lifecycle)
<b>执行者</b>	采购专员（主）、供应商（辅）
<b>前置条件</b>	<ul style="list-style-type: none"> <li>• 系统中已保存供应商信息 (vendors 表)</li> <li>• 采购专员在系统中有账号及相应权限</li> <li>• 账期条款 (billterms 表) 已预先配置</li> </ul>
<b>后置条件</b>	<ul style="list-style-type: none"> <li>• 系统保存完整的采购订单头 (orders 表) 和行项目 (entries 表)</li> <li>• 订单关联供应商、币种、账期等信息正确绑定</li> <li>• 预计付款日期已计算并可用于后续生成应付账款</li> </ul>
<b>涉众利益</b>	<ul style="list-style-type: none"> <li>• 采购专员：采购流程便捷高效，减少手动计算账期和税率错误</li> <li>• 供应商：订单信息清晰透明，便于对账和交付</li> <li>• 会计人员：自动继承账期和税率，确保应付账款入账准确及时</li> <li>• 出纳/财务：可依据预计付款日安排资金计划</li> <li>• 老板/管理层：采购与资金流初步协同，提升资金使用效率</li> </ul>
<b>基本路径</b>	<ol style="list-style-type: none"> <li>1) 采购专员登录系统，进入“采购管理”模块，选择“新建采购订单”</li> <li>2) 系统显示供应商列表</li> <li>3) 采购专员搜索或选择供应商</li> <li>4) 系统自动加载供应商信息（默认币种、账期条款、税务覆盖规则）</li> <li>5) 采购专员录入订单头信息（订单日期、参考号、说明等）</li> <li>6) 系统创建订单头记录 (orders 表)，设置 owner_type 为“Vendor”，owner_guid 指向供应商</li> <li>7) 采购专员逐行添加采购商品/服务</li> <li>8) 对于每行项目，采购专员录入：           <ul style="list-style-type: none"> <li>• 商品描述、支出科目</li> <li>• 数量（支持分数形式：quantity_num / quantity_denom）</li> <li>• 单价（支持分数形式：price_num / price_denom）</li> <li>• 税率（如被供应商税务覆盖规则影响，则自动填充）</li> </ul> </li> <li>9) 重复第 7-8 步，直到录入完所有行项目</li> <li>10) 系统根据供应商账期条款 (billterms 表) 自动计算预计付款日（订单日期 + due_days）</li> <li>11) 系统保存订单头及所有行项目信息</li> <li>12) 系统提示“采购订单已保存成功”</li> </ol>
<b>扩展路径（出 现问题处理）</b>	<ol style="list-style-type: none"> <li>1) 供应商不在选择列表中           <ol style="list-style-type: none"> <li>A: 新增供应商（参考用例 UC-供应商管理或类似新增供应商用例）</li> </ol> </li> <li>2) 供应商未设置账期条款 (terms 为空)           <ol style="list-style-type: none"> <li>A: 系统提示“该供应商未设置账期条款，请先维护或手动选择账期”，允许手动选择默认账期或跳过自动计算</li> </ol> </li> <li>3) 行项目数量或单价输入非正数</li> </ol>

A: 系统前端实时校验，提示“数量和单价必须大于 0”，拒绝保存

#### 4) 税率继承冲突（供应商覆盖与系统默认不一致）

A: 系统以供应商税务覆盖规则优先，并高亮显示“已应用供应商特定税率”以提醒用户

#### 5) 保存时数据库并发冲突（多人同时编辑同一订单）

A: 系统采用乐观锁机制，提示“订单已被他人修改，请刷新后重试”

### 字段列表

- 订单信息 (orders 表): 订单 ID + 日期 + vendor\_guid (供应商) + currency (币种) + terms (账期条款) + reference (外部参考号) + 说明 + owner\_type + owner\_guid
- 订单行项目信息 (entries 表): entry\_guid + order\_guid + 描述 + 支出科目 + quantity\_num + quantity\_denom + price\_num + price\_denom + 税率
- 账期条款 (billterms 表): terms\_id + due\_days (到期天数) + discount\_days (折扣天数) + description

### 业务规则

- 账期: 普通供应商默认 30 天, 信用额度高于一定金额的供应商可设置为 60 天或更长
- 供应商特定税率覆盖 (tax\_override 不为空时) 优先于系统全局税率
- 数量和单价必须以有理数 (分子/分母) 形式存储, 确保精度无损
- 预计付款日 = 订单日期 + due\_days, 仅用于参考, 不影响实际付款操作
- 订单保存后不可随意删除, 仅允许有权限用户作废

### 质量需求

- 可用性: 采购专员经过简单培训 (30 分钟内) 即可熟练创建和维护采购订单, 提供向导式行项目录入和自动税率/账期继承功能
- 可靠性: 所有金额、数量、单价必须采用有理数存储, 保存前进行完整性与精度校验, 支持多人并发编辑时防止数据冲突
- 性能: 50 行以内订单保存时间  $\leq 800\text{ms}$ ; 供应商列表加载  $\leq 1$  秒
- 可支持性: 关键业务错误 (如账期缺失、精度丢失) 提供详细中文提示与建议操作; 重大变更记录完整审计日志, 便于问题追溯

### 设计约束

- 前端采用网页版, 支持主流浏览器和手机端响应式布局, 便于移动场景查看与简单编辑
- 所有货币金额、数量、单价字段必须使用精确有理数 (分子/分母) 存储, 彻底避免浮点精度误差
- 供应商税务覆盖与账期自动继承逻辑在后端实现, 确保业务规则一致性

### C. 销售商务管理

#### UC3: 智能销售开单与实时风控 (Sales Invoicing & Risk Control)

用例编号	SCM-SAL-001
用例名	智能销售开单与实时风控 (Sales Invoicing & Risk Control)
执行者	销售专员 (主)、客户 (辅)
前置条件	<ul style="list-style-type: none"> <li>• 系统中已保存客户信息 (customers 表), 包括信用额度 (credit_limit)</li> <li>• 销售专员在系统中有账号及开单权限</li> <li>• 税率表 (tax_tables) 及项目档案 (jobs 表) 已预先配置</li> </ul>
后置条件	<ul style="list-style-type: none"> <li>• 系统成功保存销售发票头 (invoices 表) 和行项目</li> <li>• 信用风险检查通过或已进入审批流程</li> <li>• 项目关联 (如适用) 正确绑定, 便于后续项目损益核算</li> <li>• 多级税额明细准确计算并写入 taxable_entries 表</li> </ul>
涉众利益	<ul style="list-style-type: none"> <li>• 销售专员: 开单高效, 系统自动完成信用检查、税额计算和项目关联, 减少手动错误</li> <li>• 客户: 开票准确及时, 税额透明</li> <li>• 财务/会计: 应收账款数据可靠, 税费明细可追溯, 支持项目维度损益分析</li> <li>• 风控人员/管理层: 实时信用超限拦截, 降低坏账风险</li> <li>• 老板/股东: 收入入口严格受控, 项目盈利能力清晰可见</li> </ul>
基本路径	<ol style="list-style-type: none"> <li>1) 销售专员登录系统, 进入“销售管理”模块, 选择“新建销售发票”</li> <li>2) 系统显示客户列表</li> <li>3) 销售专员搜索或选择客户</li> <li>4) 系统自动加载客户信息 (信用额度、默认税率区域、币种等)</li> <li>5) 如为项目制销售, 销售专员选择关联项目 (job_id)</li> <li>6) 系统创建发票头记录 (invoices 表), 设置 owner_type 为"Customer", owner_guid 指向客户</li> <li>7) 销售专员逐行添加销售商品/服务</li> <li>8) 对于每行项目, 销售专员录入:           <ul style="list-style-type: none"> <li>• 商品描述、收入科目</li> <li>• 数量 (支持分数形式: quantity_num / quantity_denom)</li> <li>• 单价 (支持分数形式: price_num / price_denom)</li> <li>• 是否应税 (taxable 标记)</li> </ul> </li> <li>9) 重复第 7-8 步, 直到录入完所有行项目</li> <li>10) 系统实时计算当前订单总额</li> <li>11) 系统执行信用风控检查: 计算现有未结应收 + 当前订单总额, 若超过客户信用额度, 则弹出警告 (可配置为阻断或转审批)</li> <li>12) 系统根据客户区域自动匹配税率表, 多级计算税额并生成税费明细</li> <li>13) 系统保存发票头、行项目及税费明细</li> <li>14) 系统提示“销售发票已保存成功”</li> </ol>
扩展路径 (出)	1) 客户不在选择列表中
现问题处理)	A: 新增客户 (参考用例 UC-客户管理或类似新增客户用例)

	<p>2) 客户未设置信用额度 (credit_limit 为 null 或 0)            A: 系统提示“该客户未设置信用额度, 请先维护或手动确认风险”, 允许强制继续或转审批</p> <p>3) 信用检查超限            A: 系统弹出“信用额度超限”警告, 根据配置阻断保存或提交至信用审批流程</p> <p>4) 税率表未匹配到客户区域            A: 系统 fallback 到默认税率, 并高亮提示“使用系统默认税率, 请确认”</p> <p>5) 保存时并发冲突 (多人同时编辑同一发票)            A: 系统采用乐观锁, 提示“发票已被他人修改, 请刷新后重试”</p>
字段列表	<ul style="list-style-type: none"> <li>发票信息 (invoices 表): 发票 ID + 日期 + customer_guid (客户) + job_id (项目) + currency (币种) + credit_limit (客户额度缓存) + 总额 + is_paid + owner_type + owner_guid</li> <li>发票行项目信息 (entries 表): entry_guid + invoice_guid + 描述 + 收入科目 + quantity_num + quantity_denom + price_num + price_denom + taxable (是否应税)</li> <li>税费明细 (taxable_entries 表): tax_entry_guid + invoice_guid + tax_type (税种) + tax_rate (税率) + tax_base (税基) + tax_amount</li> </ul>
业务规则	<ul style="list-style-type: none"> <li>风险敞口 = 客户所有未付发票余额之和 + 当前发票总额</li> <li>若风险敞口 &gt; 客户信用额度, 则触发风控 (阻断或审批, 可配置)</li> <li>税额计算支持多级父子税率结构, 每一级税额单独记录以便审计</li> <li>项目关联可选, 但一旦选择则必须与 jobs 表有效记录绑定</li> <li>数量和单价必须以有理数 (分子/分母) 形式存储, 确保精度无损</li> <li>发票保存后不可随意删除, 仅允许有权限用户作废或红冲</li> </ul>
质量需求	<ul style="list-style-type: none"> <li>可用性: 销售专员经过简单培训 (30 分钟内) 即可熟练开具发票, 支持向导式行项目录入、实时税额预览和信用风险提示</li> <li>可靠性: 信用检查与税额计算必须在保存前实时完成, 所有金额使用有理数存储, 支持并发开单时防止数据冲突</li> <li>性能: 信用风险检查 ≤ 500ms (即使客户有数百笔历史发票); 50 行以内发票保存时间 ≤ 1 秒</li> <li>可支持性: 风控超限、税率异常等关键场景提供详细中文提示与可操作建议; 所有开单操作记录完整审计日志, 便于事后追溯</li> </ul>
设计约束	<ul style="list-style-type: none"> <li>前端采用网页版, 支持主流浏览器和手机端响应式布局, 便于外出销售现场开单</li> <li>信用风控与多级税额计算逻辑在后端实现, 确保规则一致性和计算准确性</li> <li>所有金额、数量、单价、税额字段使用精确有理数存储, 彻底避免浮点精度误差</li> </ul>

## D. 库存物流管理

### UC4: 批次级库存追踪与成本结转 (Lot-Level Inventory Tracking)

用例编号	LOG-INV-001
用例名	批次级库存追踪与成本结转 (Lot-Level Inventory Tracking)
执行者	仓库管理员（主）、采购专员/销售专员（辅）
前置条件	<ul style="list-style-type: none"> <li>系统已配置商品基础数据 (commodities 表)</li> <li>采购订单或销售订单已生成并进入库存变动流程</li> <li>仓库管理员拥有库存操作权限</li> </ul>
后置条件	<ul style="list-style-type: none"> <li>系统在 lots 表中创建或更新批次记录</li> <li>每次库存变动均生成对应的 splits 记录，并正确绑定 lot_guid</li> <li>批次剩余数量为零时自动闭环 (is_closed = 1)</li> <li>库存余额与批次明细保持一致，可用于精确成本结转和审计追踪</li> </ul>
涉众利益	<ul style="list-style-type: none"> <li>仓库管理员：库存操作简单明了，出入库自动关联批次，减少手动错误</li> <li>会计人员：每笔出库成本可精确追溯到具体采购批次，支持 FIFO 或特定辨认法结转</li> <li>审计/税务机构：批次全流程可追溯，满足高标准存货审计要求</li> <li>管理层：消除“幽灵库存”，库存数据真实可靠，提升资产管理精度</li> <li>老板/股东：成本核算准确，避免利润虚增或隐亏</li> </ul>
基本路径	<ol style="list-style-type: none"> <li>仓库管理员登录系统，进入“库存管理”模块，选择“入库确认”或“出库确认”</li> <li>系统根据关联的采购订单或销售订单加载待处理商品明细</li> <li>对于入库操作： <ul style="list-style-type: none"> <li>系统自动在 lots 表创建新批次记录</li> <li>记录批次入库日期、source_guid (采购订单)、初始数量 (有理数形式)</li> <li>生成对应的 splits 记录 (正数量，表示入库)</li> </ul> </li> <li>对于出库操作： <ul style="list-style-type: none"> <li>系统按配置策略 (FIFO 或手动选择) 显示可用批次列表</li> <li>仓库管理员选择出库批次 (支持多批次部分出库)</li> <li>系统生成对应的 splits 记录 (负数量，表示出库)，绑定选中 lot_guid</li> </ul> </li> <li>系统使用有理数精确计算批次剩余数量</li> <li>若某批次剩余数量为零，系统自动将该批次标记为 is_closed = 1</li> <li>系统更新库存总余额并保存所有记录</li> <li>系统提示“库存操作完成，批次已更新”</li> </ol>
扩展路径 (出现问题处理)	<ol style="list-style-type: none"> <li>出库时可用批次库存不足 <ul style="list-style-type: none"> <li>A: 系统提示“选中批次库存不足，请选择其他批次或调整数量”，拒绝保存</li> </ul> </li> <li>出库数量导致有理数计算出现分母不一致 <ul style="list-style-type: none"> <li>A: 系统统一转换为最小公分母进行运算，确保剩余数量精确为零时闭环</li> </ul> </li> <li>历史已闭环批次被误选 <ul style="list-style-type: none"> <li>A: 系统自动过滤已闭环批次不出现在可选列表中，仅在查询时显示</li> </ul> </li> <li>并发操作导致同一批次超扣</li> </ol>

A: 系统采用行级锁或乐观锁机制，检测到冲突时提示“批次已被他人操作，请刷新后重试”

### 5) 手动强制闭环或重新打开批次（管理员特权）

A: 提供高级维护界面，允许有权限用户手动调整 `is_closed` 状态，并记录审计日志

## 字段列表

- 批次信息 (`lots` 表): `lot_guid` (主键) + `commodity_guid` (商品) + `date` (入库日期) + `source_guid` (来源单据) + `initial_quantity_num` + `initial_quantity_denom` + `is_closed` (是否闭环)
- 库存变动明细 (`splits` 表): `split_guid` + `lot_guid` (批次) + `date` (变动日期) + `quantity_num` + `quantity_denom` (正入负出) + `reference_guid` (业务单据)
- 商品库存汇总 (可选视图): `commodity_guid` + `total_quantity` (计算字段)

## 业务规则

- 每笔采购入库必须生成唯一批次 (`lot`)
- 所有库存变动必须通过 `splits` 记录并绑定具体批次，不允许无批次变动
- 出库默认采用 FIFO 策略，可配置为手动选择特定批次
- 库存数量全程使用有理数 (分子/分母) 运算，确保无精度丢失
- 当批次累计出库数量等于初始入库数量时，自动闭环 (`is_closed = 1`)
- 已闭环批次不再参与出库选择与库存余额计算

## 质量需求

- 可用性：仓库管理员经过简单培训 (30 分钟内) 即可熟练完成出入库批次操作，提供批次选择向导和剩余库存实时预览
- 可靠性：所有库存计算采用有理数精确运算，防止幽灵库存；批次闭环逻辑自动触发，支持高并发操作时数据一致性
- 性能：单笔出入库 (含批次处理) 保存时间  $\leq 600\text{ms}$ ；查询可用批次 (万级批次)  $\leq 1$  秒，通过闭环过滤优化
- 可支持性：关键错误 (如库存超扣、精度异常) 提供详细中文提示与纠正建议；所有库存变动记录完整审计日志，便于问题追溯与审计

## 设计约束

- 前端采用网页版，支持主流浏览器和移动端响应式布局，便于仓库现场扫码操作
- 批次管理与 `splits` 绑定逻辑在后端实现，确保数据一致性和事务完整性
- 库存数量统一使用精确有理数存储与运算，彻底消除浮点精度问题

## E. 财务核算中心

### UC5: 全面预算控制与自动化财务 (Budgeting & Automation)

用例编号	FIN-GL-001
用例名	全面预算控制与自动化财务 (Budgeting & Automation)
执行者	财务经理 (主)、部门负责人 (辅)
前置条件	<ul style="list-style-type: none"> <li>系统已完成会计科目配置 (accounts 表)</li> <li>财务经理拥有预算编制与自动化任务管理权限</li> <li>预算年度或周期已设定</li> </ul>
后置条件	<ul style="list-style-type: none"> <li>系统成功保存预算方案头及明细 (budgets、budget_amounts 表)</li> <li>周期性事务模板配置完成 (recurrences 表)</li> <li>后台自动生成符合条件的会计凭证 (transactions 表)</li> <li>任意业务对象可通过 slots 表灵活扩展附加信息</li> </ul>
涉众利益	<ul style="list-style-type: none"> <li>财务经理：预算编制高效、执行自动，实时掌握预算 vs 实际差异</li> <li>部门负责人：清晰了解本部门预算额度，避免超支风险</li> <li>会计人员：固定费用、折旧等重复分录自动生成，极大降低工作量与错误率</li> <li>管理层/老板：事前预算控制严格，财务数据可扩展性强，支持复杂审计与决策</li> <li>审计机构：预算执行可追溯，附加信息完整保留，便于合规检查</li> </ul>
基本路径	<ol style="list-style-type: none"> <li>1) 财务经理登录系统，进入“财务管理”模块，选择“预算编制”</li> <li>2) 系统显示预算方案列表，财务经理点击“新建预算方案”</li> <li>3) 财务经理录入预算头信息（方案名称、年度、周期数、描述等）</li> <li>4) 系统创建 budgets 表记录</li> <li>5) 财务经理为各会计科目逐一设置预算金额：           <ul style="list-style-type: none"> <li>• 选择科目 (account_guid)</li> <li>• 指定周期 (period_num)</li> <li>• 输入预算金额 (支持有理数形式 amount_num / amount_denom)</li> </ul> </li> <li>6) 系统保存预算明细至 budget_amounts 表</li> <li>7) 财务经理进入“自动化任务”子模块，点击“新建周期性事务”</li> <li>8) 财务经理配置模板：           <ul style="list-style-type: none"> <li>• 选择周期类型 (Weekly/Monthly/Quarterly/Yearly)</li> <li>• 设置起始日期、结束日期 (如需)</li> <li>• 定义分录模板 (借贷科目、金额、说明)</li> </ul> </li> <li>9) 系统保存至 recurrences 表</li> <li>10) 后台守护进程每日自动扫描并生成到期凭证至 transactions 表</li> <li>11) 财务经理 (或有权限用户) 在任意业务单据页面点击“附加信息”</li> <li>12) 用户添加键值对扩展字段 (Key + Value，支持整数、字符串、浮点)</li> <li>13) 系统保存至 slots 表，挂载至对应 obj_guid</li> <li>14) 系统提示操作成功</li> </ol>
扩展路径 (出 现问题处理)	<ol style="list-style-type: none"> <li>1) 预算金额输入导致有理数格式错误           <p>A: 系统前端实时校验分母不为 0，后端再次验证，提示“预算金额格式错误”，</p> </li> </ol>

请检查分子分母”

- 2) 同一科目同一周期已存在预算明细
 

A: 系统提示“该科目该周期预算已存在”，允许修改或拒绝新增
- 3) 周期性任务模板分录借贷不平衡
 

A: 保存前系统自动校验借贷平衡，不平衡时拒绝保存并高亮提示不平衡金额
- 4) 后台自动生成凭证时科目已被禁用
 

A: 生成失败并记录错误日志，邮件通知财务经理手动处理
- 5) slots 表扩展 Key 重复
 

A: 系统允许同一对象同一 Key 多次存在（支持历史版本），或根据配置拒绝重复并提示“Key 已存在”

#### 字段列表

- 预算方案 (budgets 表): budget\_guid (主键) + name (方案名称) + fiscal\_year (年度) + num\_periods (周期数) + description
- 预算明细 (budget\_amounts 表): amount\_guid + budget\_guid + account\_guid (科目) + period\_num (周期) + amount\_num + amount\_denom
- 周期性任务 (recurrences 表): recurrence\_guid + period\_type + period\_start + period\_end + template\_json (分录模板) + is\_active
- 扩展信息 (slots 表): slot\_guid + obj\_guid (挂载对象) + obj\_type (对象类型) + key + int\_val + string\_val + double\_val

#### 业务规则

- 预算金额必须以有理数形式存储，支持任意精度
- 同一预算方案内，同一科目同一周期只能存在一条预算记录
- 周期性任务模板必须保证借贷平衡方可激活
- 自动生成凭证时若遇冲突（如科目冻结），记录异常并通知负责人
- slots 表扩展支持任意业务对象挂载，Key 不强制唯一（允许版本化）
- 预算执行分析必须支持“预算 vs 实际”多维度对比报表

#### 质量需求

- 可用性：财务经理经过简单培训（40 分钟内）即可熟练编制多周期预算与配置自动化任务，提供预算复制、模板导入等快捷功能
- 可靠性：预算校验与凭证模板借贷平衡必须在保存前强制检查，自动化任务支持异常回滚与通知机制
- 性能：年度预算（含 12 周期、500 条明细）保存时间 ≤ 1 秒；后台每日扫描千级周期任务 ≤ 5 秒
- 可支持性：预算超支、自动化异常等关键场景提供详细中文提示与处理建议；所有预算与自动化操作记录完整审计日志

#### 设计约束

- 前端采用网页版，支持主流浏览器和移动端响应式布局，便于部门负责人移动端查看预算执行情况
- 预算计算与自动化凭证生成逻辑全部在后端实现，确保事务一致性与安全性
- 所有金额字段使用精确有理数存储，彻底避免浮点精度误差
- 业务扩展统一使用 slots 键值对机制，避免频繁修改核心表结构

## VII. 类图

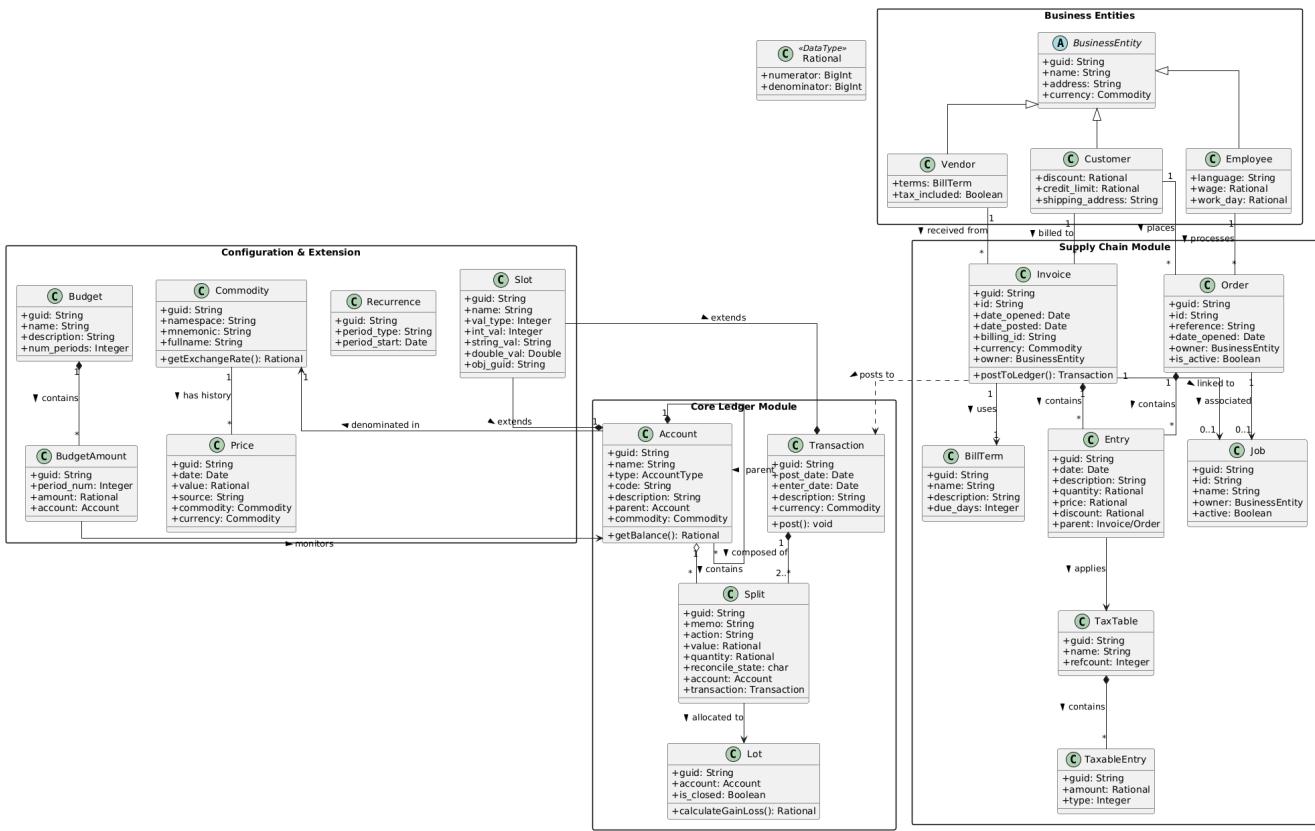


图 13: 类图

## VIII. E-R 图

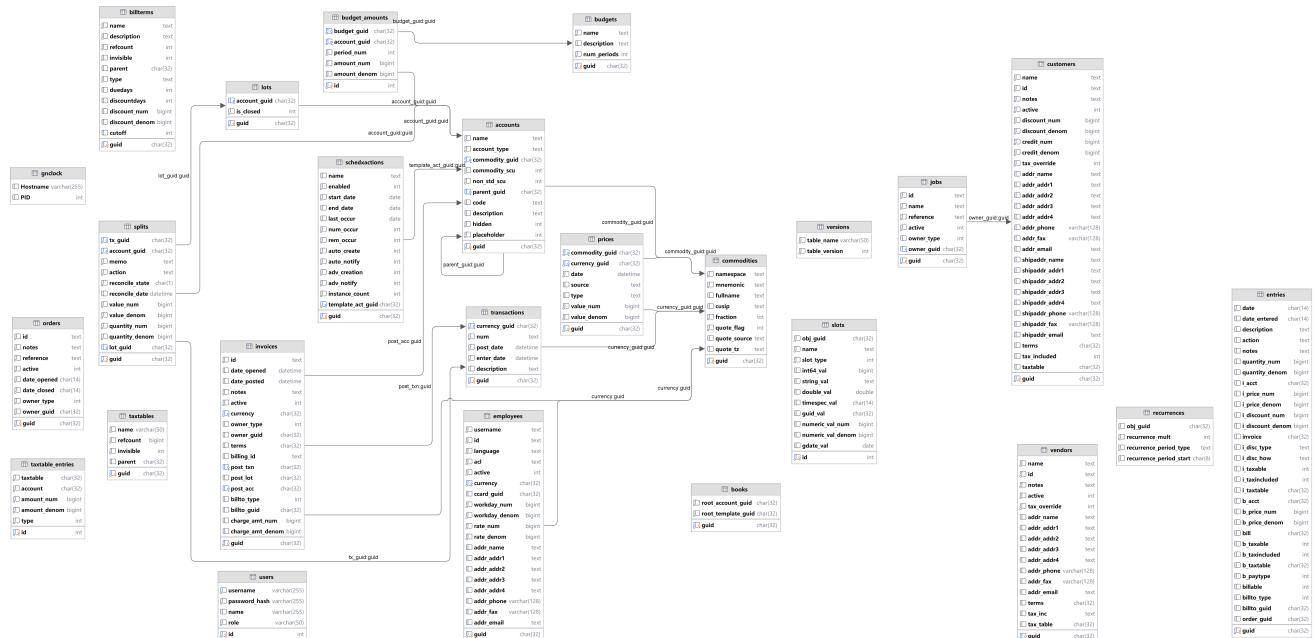


图 14: er 图

## A. 数据库逻辑架构说明

本系统的数据库设计遵循第三范式（3NF），以确保数据的一致性并减少冗余。ER 图核心模型由以下四个关键模块组成，共同支撑起“业财一体化”的业务逻辑：

1) 核心账务引擎模块：这是系统的核心，负责存储所有的财务凭证与资金流向。

- **accounts (科目表)**：采用树状结构存储会计科目（如资产、负债、权益、收入、费用）。通过 `parent_guid` 字段实现无限层级的子科目嵌套。
- **transactions (交易表)**：代表一笔“业务事件”或“凭证头”。它记录了交易发生的时间、描述及货币类型，但不直接存储金额。
- **splits (分录表)**：这是复式记账的灵魂所在。表与 `transactions` 呈  $N : 1$  关系。每一笔交易至少包含两个分录（一借一贷），且所有属于同一交易的 `splits` 金额之和必须为零（遵循“有借必有贷，借贷必相等”原则）。
- **核心亮点**：在 `splits` 表中，金额字段未采用传统的 `float` 或 `decimal` 类型，而是设计了 `value_num`（分子）和 `value_denom`（分母）两个整型字段。这种基于有理数的数据存储方式，从底层彻底解决了浮点数运算的精度丢失问题。

2) 供应链与单据模块：该模块负责管理企业的进销存业务，并与财务模块强关联。

- **orders (订单表)**：管理采购订单 (PO) 和销售订单 (SO)，记录交易的契约状态。
- **invoices (发票/单据表)**：作为业务与财务的连接点。当发票被“过账 (Posted)”时，系统会自动在 `transactions` 和 `splits` 表中生成对应的应收/应付凭证，实现业务驱动财务的自动化。
- **entries (单据明细表)**：记录发票或订单中的具体商品行项目，包括数量、单价及折扣信息。
- 3) 商业实体模块：用于管理与企业发生经济往来的对象，支持 CRM 与 SRM 功能。
- **customers (客户表) & vendors (供应商表)**：存储合作伙伴的基础信息、信用额度及默认结算货币。
- **employees (员工表)**：用于管理销售人员或采购专员，以便进行绩效归属分析。

4) 基础设置模块：

- **commodities (商品/货币表)**：统一管理系统中的流通单位，既包括法币 (CNY, USD)，也包括库存商品 (SKU)。这种设计使得“库存”也可以被视为一种特殊的“账户”，从而实现库存金额的实时核算。
- **prices (价格表)**：记录商品的历史价格及不同货币之间的汇率快照。

## IX. 数据库表结构字段列表

### 1. app\_users (系统用户)

字段名	类型	允许空	默认值	主键	外键	说明
user_id	INT	否	AUTO_INCREMENT	PK	-	用户 ID
login_name	VARCHAR(100)	否	-	-	-	登录名 (唯一)
password_token	VARCHAR(255)	否	-	-	-	密码哈希
display_name	VARCHAR(100)	是	NULL	-	-	显示名称
access_level	VARCHAR(50)	是	'member'	-	-	权限级别

### 2. biz\_commodities (货币/商品定义)

字段名	类型	允许空	默认值	主键	外键	说明
commodity_id	CHAR(36)	否	-	PK	-	主键 ID
namespace	VARCHAR(50)	否	-	-	-	命名空间 (如'CURRENCY')
symbol	VARCHAR(20)	否	-	-	-	符号 (如'CNY')
full_name	VARCHAR(100)	是	NULL	-	-	全称
precision_val	INT	否	-	-	-	精度 (如 100 表示 2 位小数)

### 3. biz\_partners (往来单位)

字段名	类型	允许空	默认值	主键	外键	说明
partner_id	CHAR(36)	否	-	PK	-	主键 ID
partner_code	VARCHAR(50)	否	-	-	-	单位编码
legal_name	VARCHAR(255)	否	-	-	-	法定名称
partner_type	ENUM('customer','vendor','employee')	否	-	-	-	类型 (客户/供应商/员工)
contact_email	VARCHAR(128)	是	NULL	-	-	联系邮箱
contact_phone	VARCHAR(50)	是	NULL	-	-	联系电话
shipping_address	TEXT	是	NULL	-	-	发货地址
is_enabled	INT	是	1	-	-	是否启用 (1= 启用)

### 4. fin\_accounts (会计科目)

字段名	类型	允许空	默认值	主键	外键	说明
account_id	CHAR(36)	否	-	PK	-	主键 ID
account_code	VARCHAR(50)	是	NULL	-	-	科目编码
title	VARCHAR(255)	否	-	-	-	科目名称
category	VARCHAR(50)	否	-	-	-	类别 (ASSET 等)
commodity_ref	CHAR(36)	是	NULL	-	biz_commodities(commodity_id)	计价货币
parent_account_id	CHAR(36)	是	NULL	-	-	父科目 ID
is_placeholder	INT	是	0	-	-	是否占位 (1= 仅分类)
account_description	TEXT	是	NULL	-	-	描述

### 5. fin\_transactions (总账交易头)

字段名	类型	允许空	默认值	主键	外键	说明
txn_id	CHAR(36)	否	-	PK	-	交易 ID
currency_ref	CHAR(36)	否	-	-	biz_commodities(commodity_id)	交易币种
doc_number	VARCHAR(100)	是	NULL	-	-	单据编号
posting_date	DATETIME	否	-	-	-	过账日期

entry_date	DATETIME	是	CURRENT_TIMESTAMP	-	-	录入时间
summary	TEXT	是	NULL	-	-	摘要

## 6. fin\_journal\_entries (凭证分录)

字段名	类型	允许空	默认值	主键	外键	说明
entry_id	CHAR(36)	否	-	PK	-	分录 ID
txn_ref	CHAR(36)	否	-	-	fin_transactions(txn_id)	所属交易
account_ref	CHAR(36)	否	-	-	fin_accounts(account_id)	会计科目
memo	TEXT	是	NULL	-	-	备注
val_num	BIGINT	否	-	-	-	金额分子
val_denom	BIGINT	否	-	-	-	金额分母 (精度)
qty_num	BIGINT	否	-	-	-	数量/原币金额分子
qty_denom	BIGINT	否	-	-	-	数量/原币金额分母

## 7. inv\_items (物料主数据)

字段名	类型	允许空	默认值	主键	外键	说明
item_id	CHAR(36)	否	-	PK	-	主键 ID
item_code	VARCHAR(50)	否	-	-	-	物料编码 (唯一)
item_name	VARCHAR(255)	否	-	-	-	物料名称
unit_of_measure	VARCHAR(20)	否	-	-	-	计量单位
costing_method	ENUM('FIFO','AVERAGE')	否	'AVERAGE'	-	-	成本计算方法
qty_on_hand	BIGINT	是	0	-	-	当前库存数量
avg_cost_num	BIGINT	是	0	-	-	平均成本分子
avg_cost_denom	BIGINT	是	100	-	-	平均成本分母
is_active	INT	是	1	-	-	是否启用
inventory_account_ref	CHAR(36)	是	NULL	-	fin_accounts(account_id)	库存科目
cogs_account_ref	CHAR(36)	是	NULL	-	fin_accounts(account_id)	销售成本科目
sales_account_ref	CHAR(36)	是	NULL	-	fin_accounts(account_id)	销售收入科目

## 8. inv\_movements (库存移动记录)

字段名	类型	允许空	默认值	主键	外键	说明
move_id	CHAR(36)	否	-	PK	-	移动 ID
item_ref	CHAR(36)	否	-	-	inv_items(item_id)	物料
move_type	ENUM('IN','OUT','ADJUST')	否	-	-	-	移动类型
quantity	BIGINT	否	-	-	-	数量
unit_cost_num	BIGINT	否	-	-	-	单位成本分子
unit_cost_denom	BIGINT	否	-	-	-	单位成本分母
move_date	DATETIME	是	CURRENT_TIMESTAMP	-	-	移动时间
related_txn_ref	CHAR(36)	是	NULL	-	fin_transactions(txn_id)	关联交易

## 9. bus\_purchase\_orders (采购订单)

字段名	类型	允许空	默认值	主键	外键
po_id	CHAR(36)	否	-	PK	-
po_number	VARCHAR(50)	否	-	-	-
partner_ref	CHAR(36)	否	-	-	biz_partners(partner_id)
order_date	DATE	否	-	-	-
expected_delivery_date	DATE	是	NULL	-	-
currency_ref	CHAR(36)	否	-	-	biz_commodities(commodity_id)

total_amount_num	BIGINT	否	-	-	-
total_amount_denom	BIGINT	否	-	-	-
order_status	ENUM('DRAFT','OPEN','CLOSED','CANCELED')	否	'DRAFT'	-	-
created_at	DATETIME	是	CURRENT_TIMESTAMP	-	-

### 10. bus\_po\_items (采购订单明细)

字段名	类型	允许空	默认值	主键	外键	说明
line_id	CHAR(36)	否	-	PK	-	行 ID
po_ref	CHAR(36)	否	-	-	bus_purchase_orders(po_id)	所属采购订单
item_ref	CHAR(36)	否	-	-	inv_items(item_id)	物料
quantity	BIGINT	否	-	-	-	数量
unit_price_num	BIGINT	否	-	-	-	单价分子
unit_price_denom	BIGINT	否	-	-	-	单价分母

### 11. bus\_sales\_orders (销售订单)

字段名	类型	允许空	默认值	主键	外键
so_id	CHAR(36)	否	-	PK	-
so_number	VARCHAR(50)	否	-	-	-
partner_ref	CHAR(36)	否	-	-	biz_partners(partner_id)
order_date	DATE	否	-	-	-
expected_shipment_date	DATE	是	NULL	-	-
currency_ref	CHAR(36)	否	-	-	biz_commodities(commodity_id)
total_amount_num	BIGINT	否	-	-	-
total_amount_denom	BIGINT	否	-	-	-
order_status	ENUM('DRAFT','OPEN','CLOSED','CANCELED')	否	'DRAFT'	-	-
created_at	DATETIME	是	CURRENT_TIMESTAMP	-	-

### 12. bus\_so\_items (销售订单明细)

字段名	类型	允许空	默认值	主键	外键	说明
line_id	CHAR(36)	否	-	PK	-	行 ID
so_ref	CHAR(36)	否	-	-	bus_sales_orders(so_id)	所属销售订单
item_ref	CHAR(36)	否	-	-	inv_items(item_id)	物料
quantity	BIGINT	否	-	-	-	数量
unit_price_num	BIGINT	否	-	-	-	单价分子
unit_price_denom	BIGINT	否	-	-	-	单价分母

### 13. biz\_exchange\_rates (汇率表)

字段名	类型	允许空	默认值	主键	外键	说明
rate_id	INT	否	AUTO_INCREMENT	PK	-	自增 ID
from_currency	CHAR(36)	否	-	-	biz_commodities(commodity_id)	源货币
to_currency	CHAR(36)	否	-	-	biz_commodities(commodity_id)	目标货币
rate	DECIMAL(10,4)	否	-	-	-	汇率 (1 源货币 =rate 目标货币)

## X. 序列图

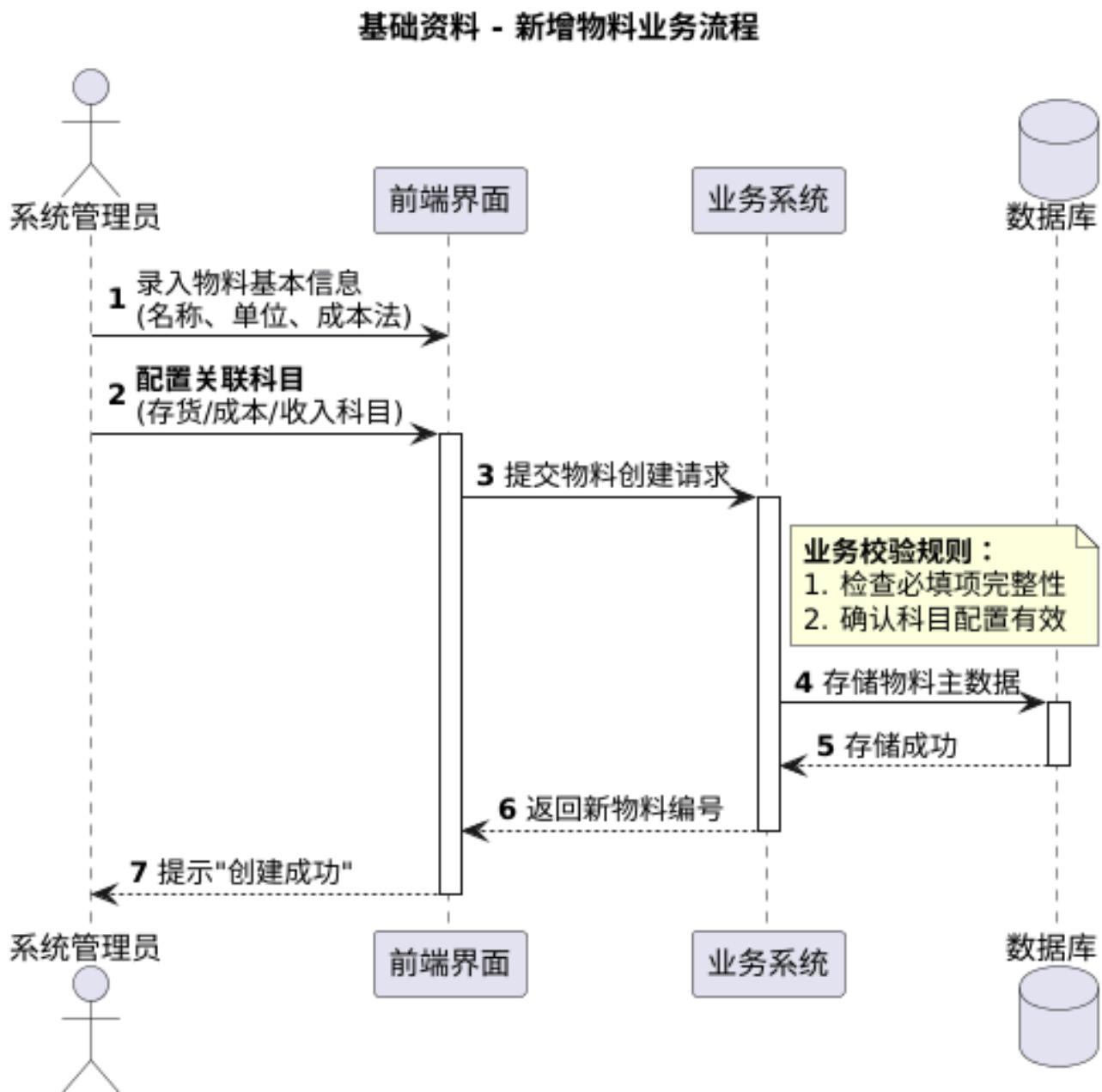


图 15: 基础资料

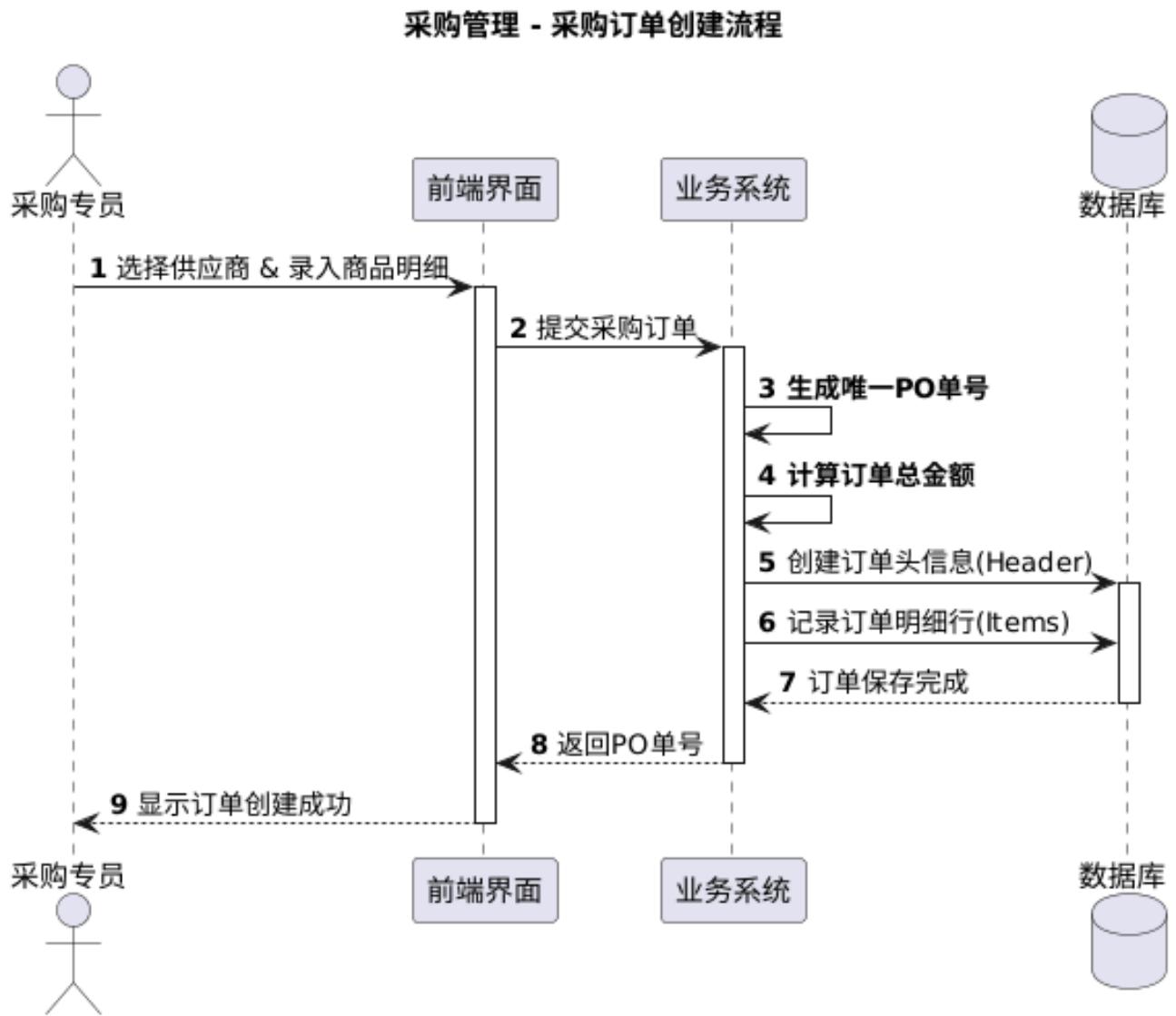


图 16: 采购商务

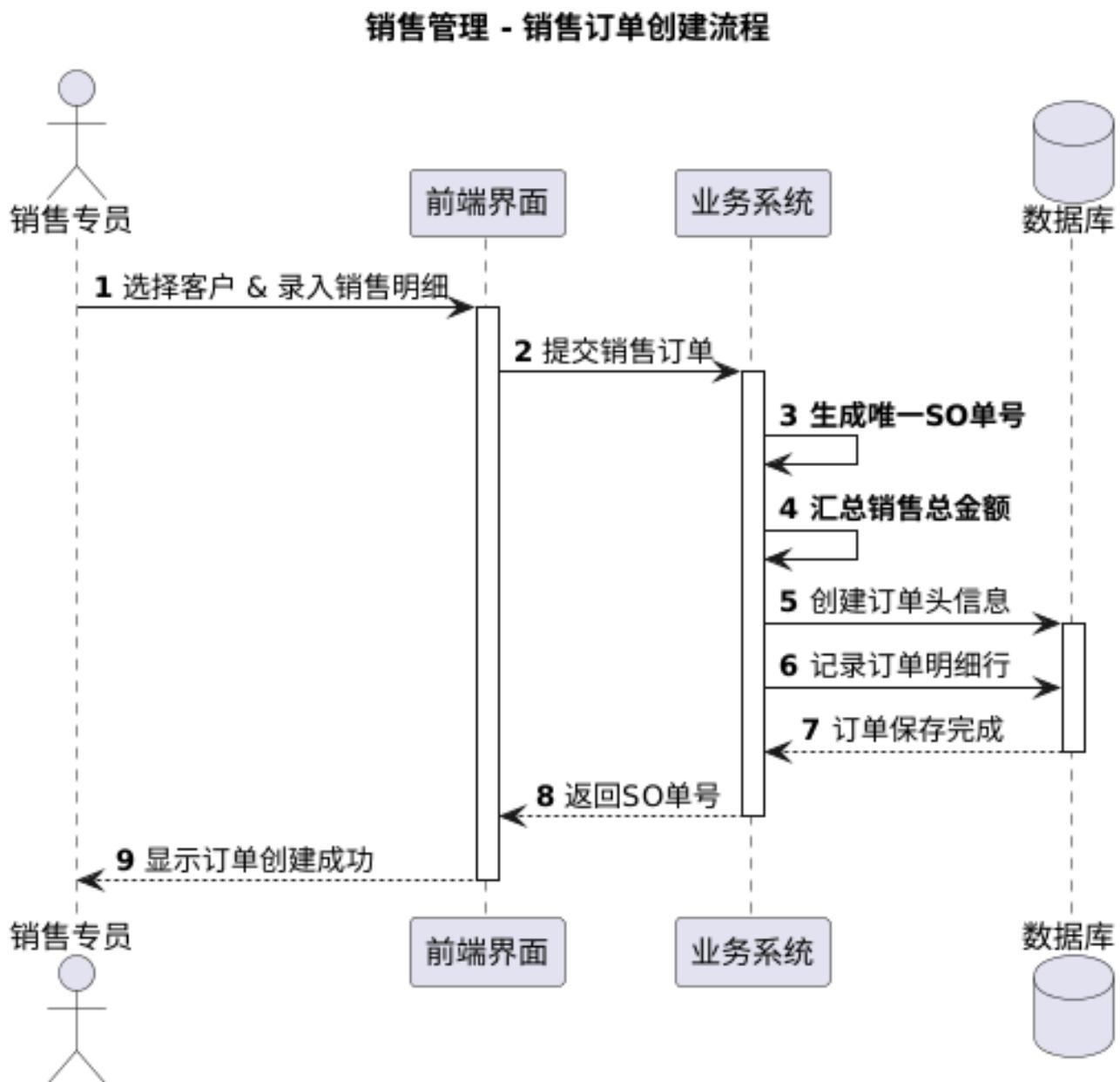


图 17: 销售服务

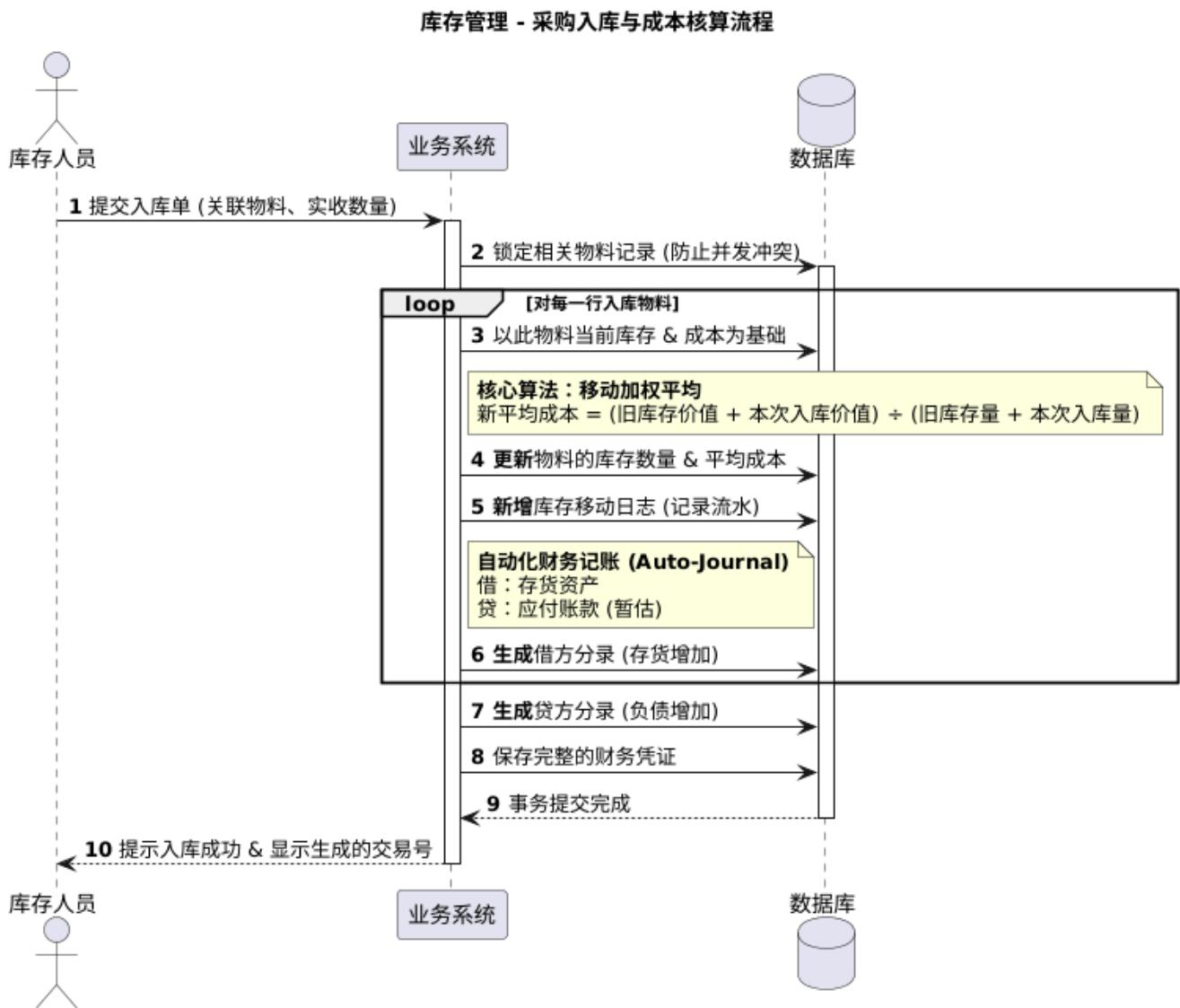


图 18: 库存管理时序图

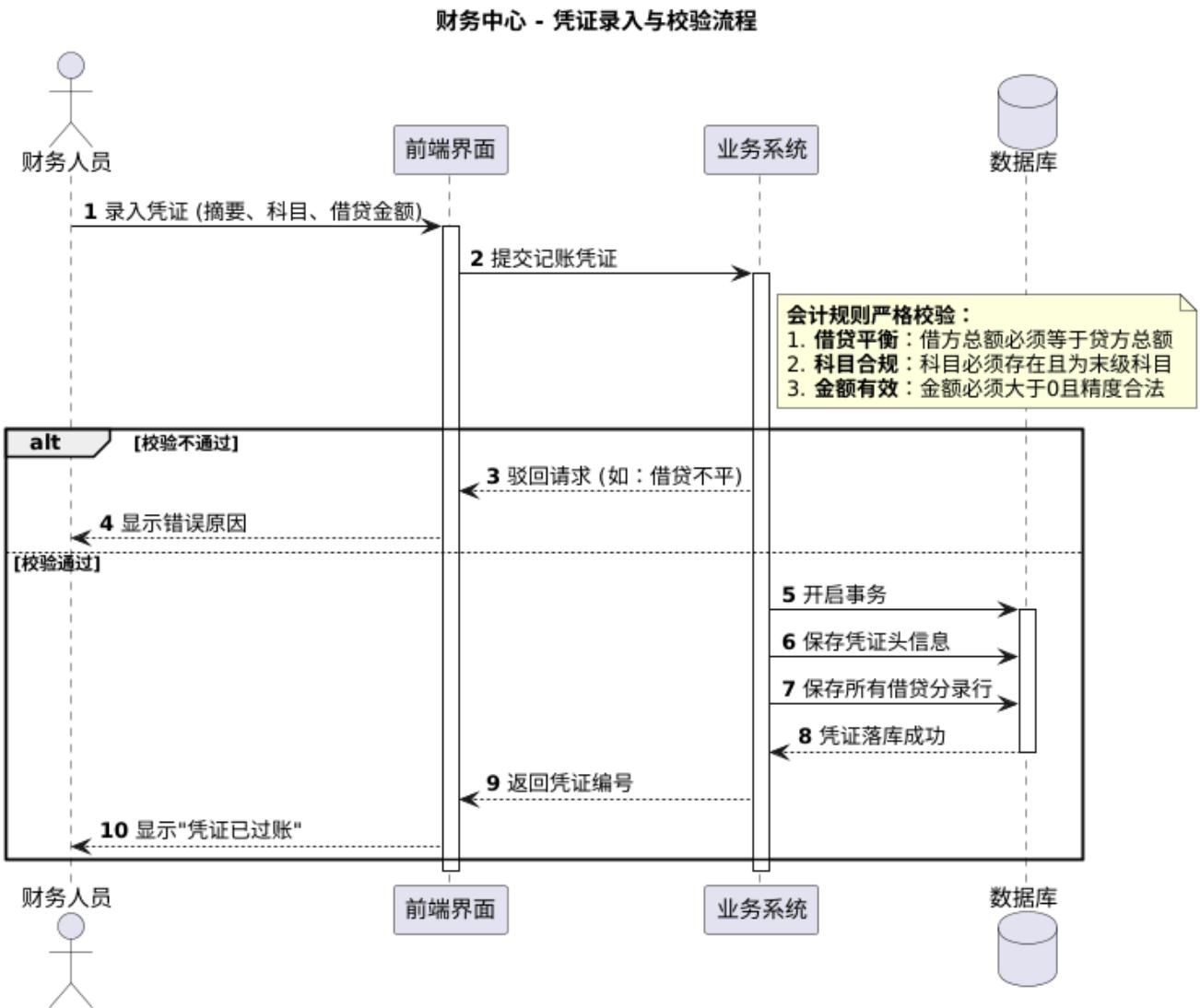


图 19: 财务核算

## XI. 结论

该系统通过深度的业务财务一体化设计，成功打通了企业内部的信息壁垒。系统不仅实现了采购、销售、库存的数字化管理，更通过自动化的会计引擎，大幅降低了财务人员的手工录入成本。

在技术实现上，通过引入 MySQL 事务处理、悲观锁并发控制以及有理数高精度计算模型，解决了传统 Web ERP 中常见的超卖与资金误差问题。该系统的设计方案具有较强的鲁棒性与可扩展性，能够满足中小型进出口贸易企业的核心管理需求。