POS point of sale

PHẦN 1.4

Tổng quan BFD

|  |  |  |
| --- | --- | --- |
| Tên chức năng | Mô tả | Đánh giá khả năng thực hiện (nhân lực, thời gian, công nghệ, môi trường) |
| 0.0 Hệ thống quản lý logistics & vận hành Amart | Hệ thống phần mềm tập trung hỗ trợ quản lý mua hàng, kho, vận hành cửa hàng, tài chính và báo cáo cho toàn chuỗi Amart. | Khả thi. Quy mô hệ thống vừa, có thể triển khai theo từng module. Yêu cầu đội phát triển phần mềm + phối hợp với bộ phận IT và nghiệp vụ. |

## Quản lý mua hàng & nhà cung cấp

|  |  |  |
| --- | --- | --- |
| Tên chức năng | Mô tả | Đánh giá |
| 1.0 Quản lý mua hàng & nhà cung cấp | Nhóm chức năng phục vụ quản trị viên logistic trong việc quản lý NCC, bảng giá, tạo PO và nhập kho tổng. | Khả thi (JavaFX + Java + MySQL) |
| 1.1 Quản lý thông tin nhà cung cấp | Lưu trữ, cập nhật thông tin NCC: tên, địa chỉ, MST, người liên hệ, điều khoản thanh toán… | Khả thi, đơn giản. Cần tối thiểu 1–2 dev, thời gian ngắn. Công nghệ CRUD cơ bản, môi trường không phức tạp. |
| 1.2 Quản lý bảng giá nhập hàng | Quản lý báo giá/ bảng giá nhập cho từng NCC, từng mặt hàng, theo thời gian hiệu lực. | Khả thi. Phức tạp hơn CRUD do có quản lý hiệu lực theo thời gian nhưng vẫn trong tầm hệ thống web chuẩn. |
| 1.3 Lập đơn đặt hàng (PO) | Cho phép chọn NCC, sản phẩm, số lượng, đơn giá theo bảng giá để tạo PO và gửi cho NCC. | Khả thi. Thời gian phát triển trung bình, |
| 1.4 Tiếp nhận hàng & lập phiếu nhập kho tổng | Ghi nhận hàng nhận từ NCC, đối chiếu với PO và hóa đơn, lập phiếu nhập kho tổng. | Khả thi. Cần kết nối dữ liệu với PO và tồn kho. Yêu cầu nghiệp vụ rõ, công nghệ không khó, chủ yếu là xử lý nghiệp vụ đúng. |

## Quản lý kho & điều phối nội bộ

|  |  |  |
| --- | --- | --- |
| Tên chức năng | Mô tả | Đánh giá |
| 2.0 Quản lý kho & điều phối nội bộ | Nhóm chức năng theo dõi tồn kho tổng, xử lý yêu cầu hàng từ cửa hàng và điều chuyển nội bộ. | Khả thi. Quan trọng nhưng mô hình chuẩn (2-level warehouse). Cần thiết kế dữ liệu tồn kho cẩn thận. |
| 2.1 Quản lý tồn kho tổng | Theo dõi số lượng tồn từng mặt hàng tại kho tổng, cập nhật khi nhập/xuất/điều chỉnh. | Khả thi. Yêu cầu cập nhật realtime hoặc gần realtime; công nghệ DB + transaction phổ biến đáp ứng được. |
| 2.2 Tiếp nhận yêu cầu hàng từ cửa hàng | Nhận và lưu các phiếu yêu cầu hàng hóa nội bộ do cửa hàng gửi lên. | Khả thi, đơn giản. Chủ yếu là form nhập liệu và luồng trạng thái “chờ duyệt/đã duyệt”. |
| 2.3 Duyệt yêu cầu & lập lệnh điều chuyển | Quản trị viên logistic duyệt yêu cầu, xác định số lượng xuất và lập phiếu xuất/ lệnh điều chuyển. | Khả thi. Cần thêm logic kiểm tra tồn kho tổng. Thời gian phát triển trung bình. |
| 2.4 Điều chuyển hàng tổng → kho cơ sở | Quản lý quá trình xuất hàng khỏi kho tổng, vận chuyển và giao nhận cho kho cơ sở, giao giữa các kho cơ sở với nhau. | Khả thi. Nếu chỉ quản lý trên giấy tờ (phiếu điều chuyển) thì công nghệ đơn giản. Nếu tích hợp GPS/định vị thì phức tạp hơn (nhưng không bắt buộc trong bài). |
| 2.5 Cập nhật tồn kho tổng và cơ sở | Tự động cập nhật tồn kho tổng (giảm) và tồn kho cơ sở (tăng) sau mỗi điều chuyển/nhập kho. | Khả thi nhưng đòi hỏi thiết kế chuẩn về transaction, tránh lệch tồn. Công nghệ DB giao dịch phổ biến đáp ứng tốt. |

## Quản lý vận hành cửa hàng

|  |  |  |
| --- | --- | --- |
| Tên chức năng | Mô tả | Đánh giá |
| 3.1 Nhập kho cơ sở từ kho tổng | Cửa hàng xác nhận nhận hàng theo lệnh điều chuyển, lập phiếu nhập kho cơ sở và cập nhật tồn. | Khả thi. Nghiệp vụ rõ ràng, chỉ cần form nhập và liên kết với phiếu điều chuyển. |
| 3.2 Ghi log biến động tồn kho | Ghi nhận các điều chỉnh tồn (hàng hỏng, mất, hủy…) kèm lý do bắt buộc; lưu log để kế toán/kiểm toán tra cứu. | Khả thi. Chủ yếu là form nhập và bảng log, không yêu cầu công nghệ đặc biệt. |
| 3.3 Quản lý quỹ chi tiêu lặt vặt | Ghi nhận các khoản chi nhỏ tại cửa hàng (sửa chữa, vật tư nhỏ…), trừ vào quỹ và cập nhật số dư hiện tại. | Khả thi. Dạng mini-kế toán đơn giản, có thể làm nhanh bằng chức năng thu–chi cơ bản. |
| 3.4 Lập báo cáo thu – chi quỹ cửa hàng | Tổng hợp các khoản thu/chi quỹ theo ngày/tháng, kết xuất báo cáo gửi cho kế toán. | Khả thi. Chỉ cần chức năng tổng hợp và xuất báo cáo (PDF/Excel), thời gian phát triển ngắn. |
| 3.5 Quản lý thông tin khách hàng (CRM) | Lưu thông tin khách hàng (SĐT, tên…), lịch sử mua hàng và điểm tích lũy phục vụ chăm sóc khách hàng. | Khả thi. Cần thiết kế bảng dữ liệu hợp lý để truy vấn nhanh, nhưng công nghệ vẫn là CRUD + truy vấn thông thường. |
| 3.6 Tra cứu sản phẩm & giá bán | Cho phép nhân viên tìm kiếm sản phẩm theo mã vạch/mã hàng/tên và xem giá bán hiện tại. | Khả thi. Yêu cầu tìm kiếm nhanh, có thể hỗ trợ barcode; với quy mô vừa, DB quan hệ đáp ứng tốt. |
| 3.7 Tạo đơn bán hàng tại cửa hàng (POS) | Cho phép nhân viên bán hàng tạo đơn bán như siêu thị: quét mã vạch/ chọn sản phẩm, nhập số lượng, áp dụng khuyến mãi/giảm giá, tính tổng tiền, chọn phương thức thanh toán, in hóa đơn; sau khi thanh toán sẽ cập nhật tồn kho và lịch sử mua của khách hàng. | Khả thi nhưng là chức năng trọng tâm nên cần thiết kế tốt giao diện (POS), luồng thao tác nhanh, ổn định. Công nghệ kết hợp thiết bị đọc mã vạch, máy in hóa đơn là phổ biến, có thể triển khai với đội ngũ dev thông thường trong thời gian trung bình. |

## Quản lý tài chính & ngân sách

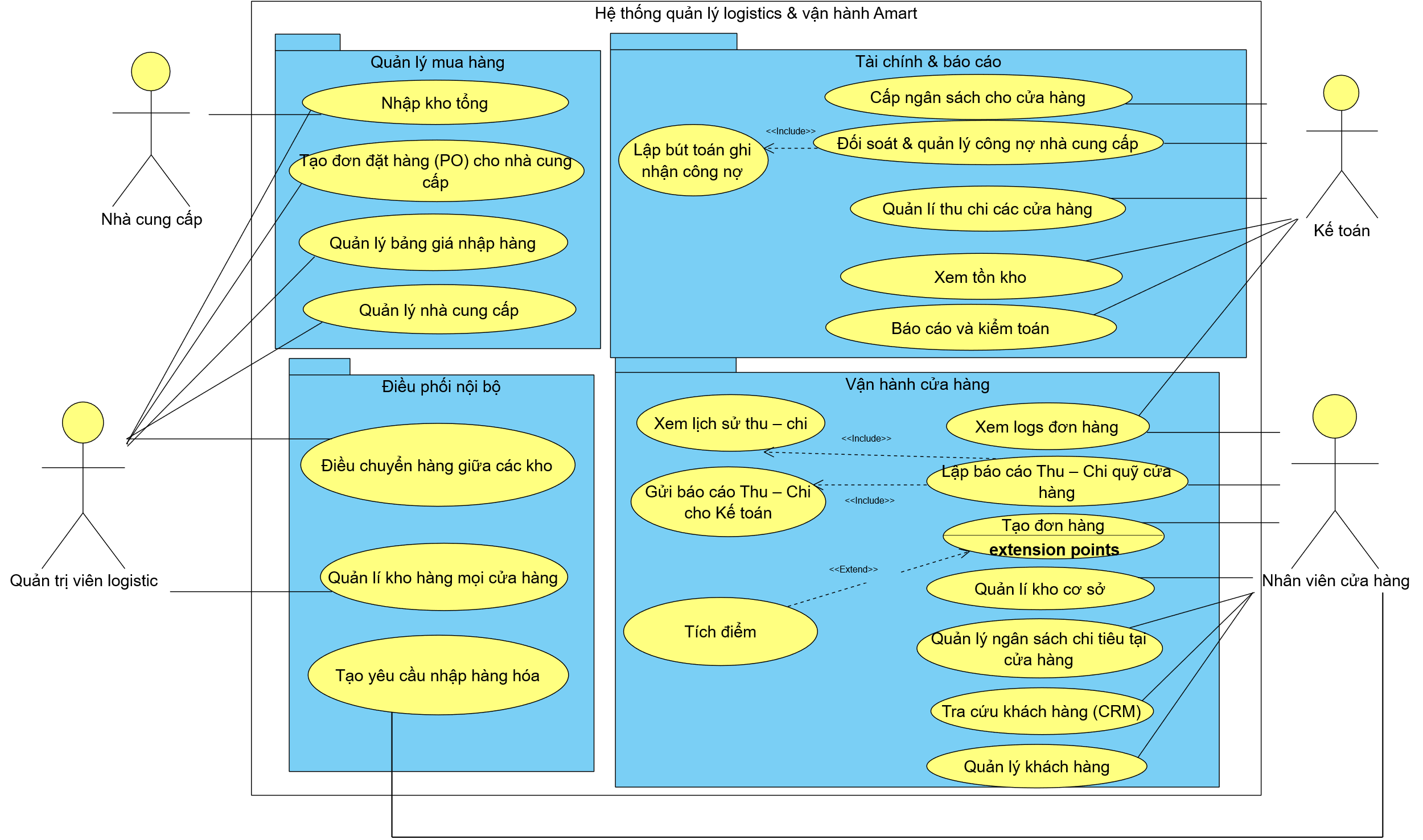
|  |  |  |
| --- | --- | --- |
| Tên chức năng | Mô tả | Đánh giá |
| 4.0 Quản lý tài chính & ngân sách | Nhóm chức năng cho kế toán: sổ cái, công nợ, thanh toán, cấp ngân sách cho cửa hàng. | Khả thi. Nếu chỉ phục vụ nội bộ, phạm vi hẹp hơn phần mềm kế toán chuyên dụng nên làm được. |
| 4.1 Quản lý sổ cái & bút toán kế toán | Ghi nhận các bút toán thu/chi liên quan đến mua hàng, quỹ cửa hàng, thanh toán NCC… | Khả thi ở mức đơn giản (phục vụ quản trị nội bộ). Nếu muốn chuẩn theo chế độ kế toán VN thì phức tạp hơn nhưng vẫn khả thi. |
| 4.2 Đối soát & quản lý công nợ NCC | Đối chiếu phiếu nhập kho tổng với hóa đơn NCC, lập công nợ phải trả từng nhà cung cấp. | Khả thi. Cần tích hợp chặt với module mua hàng và kho. Tốn thời gian phân tích nghiệp vụ nhưng công nghệ không khó. |
| 4.3 Quản lý thanh toán công nợ | Ghi nhận các lần thanh toán công nợ, cập nhật trạng thái công nợ (đã/đang/ quá hạn). | Khả thi. Chỉ cần form nhập và logic kiểm soát số tiền còn nợ. |
| 4.4 Cấp ngân sách/quỹ cho cửa hàng | Lập phiếu cấp ngân sách cho từng cửa hàng dựa trên báo cáo thu–chi, cập nhật quỹ. | Khả thi. Nghiệp vụ đơn giản, kết nối với dữ liệu quỹ cửa hàng và báo cáo. |
| 4.5 Theo dõi số dư quỹ từng cửa hàng | Hiển thị, theo dõi số dư quỹ cho từng cơ sở, đối chiếu với báo cáo thu–chi. | Khả thi. Chủ yếu là truy vấn và hiển thị dữ liệu, có thể kèm biểu đồ, công nghệ web tiêu chuẩn. |

## Báo cáo & kiểm toán

|  |  |  |
| --- | --- | --- |
| Tên chức năng | Mô tả | Đánh giá |
| 5.0 Báo cáo & kiểm toán | Nhóm chức năng báo cáo tổng hợp cho kế toán/ban lãnh đạo: tồn kho, log biến động, công nợ, quỹ. | Khả thi. |
| 5.1 Báo cáo tồn kho toàn hệ thống | Tổng hợp tồn kho theo kho tổng và từng cửa hàng, theo mặt hàng/thời điểm. | Khả thi. |
| 5.2 Báo cáo log biến động kho | Liệt kê các biến động tồn kho (nhập, xuất, hủy, hỏng, điều chỉnh) để phục vụ kiểm toán nội bộ. | Khả thi. |
| 5.3 Báo cáo công nợ nhà cung cấp | Thống kê tổng công nợ, công nợ theo từng NCC, tình trạng thanh toán. | Khả thi. Dựa trên dữ liệu bút toán và công nợ có sẵn, chủ yếu là xử lý báo cáo. |
| 5.4 Báo cáo quỹ cửa hàng | Tổng hợp số dư quỹ, số tiền đã cấp, đã chi của từng cửa hàng theo kỳ. | Khả thi. Cần dữ liệu chuẩn từ module quỹ, nhưng công nghệ làm báo cáo không khó. |
| 5.5 Báo cáo giá nhập & lịch sử giá | Theo dõi lịch sử giá nhập theo thời gian, theo NCC, hỗ trợ phân tích chi phí và đàm phán. | Khả thi. Cần lưu đầy đủ lịch sử giá trong quá trình vận hành. Truy vấn nhiều nên có thể tối ưu index DB, nhưng vẫn trong khả năng kỹ thuật phổ thông. |

# PHẦN 2.2

# BIỂU ĐỒ UC TỔNG QUAN



# PHẦN 3.4

# ERD

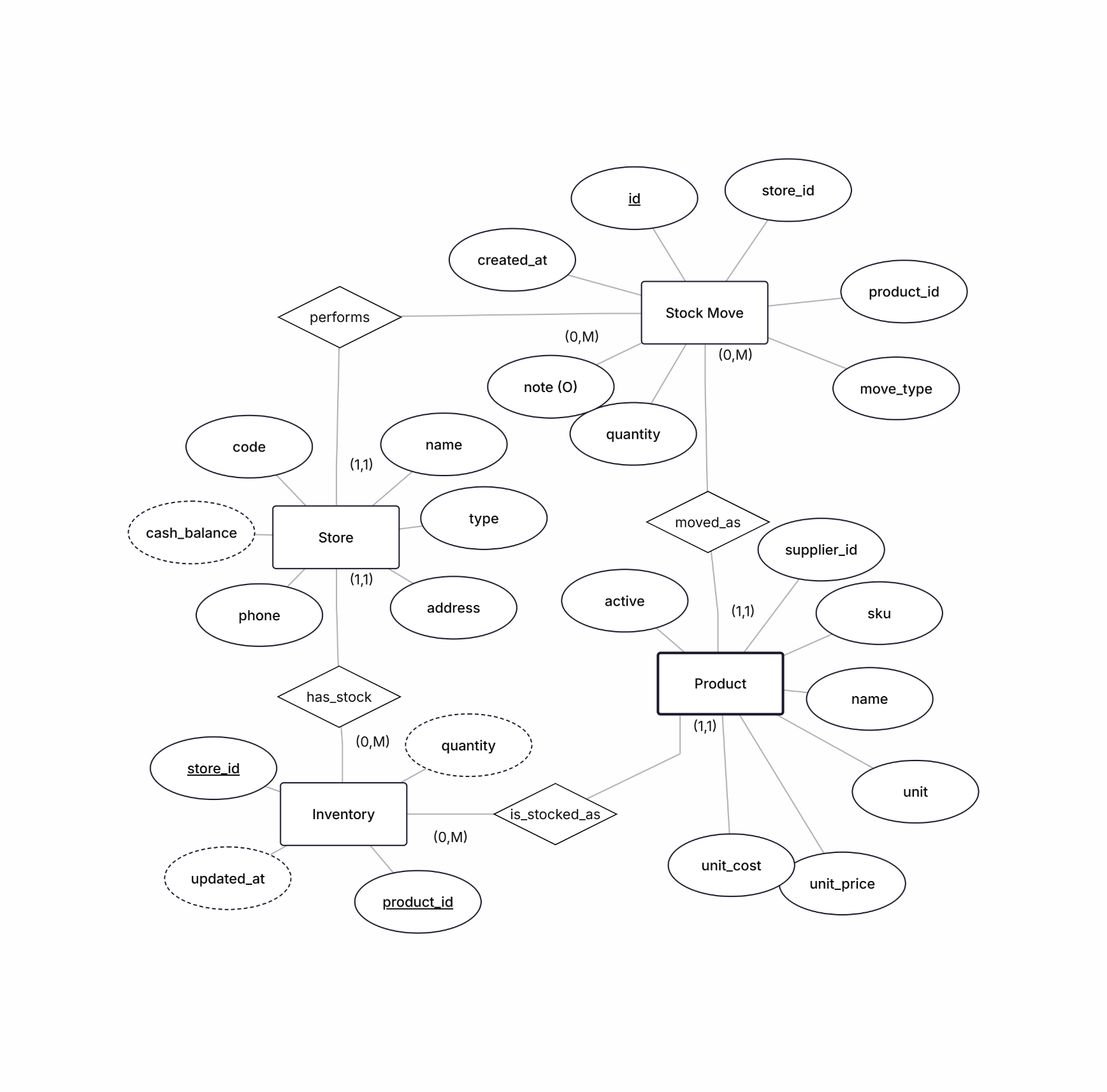
**Tổng quan hệ thống**

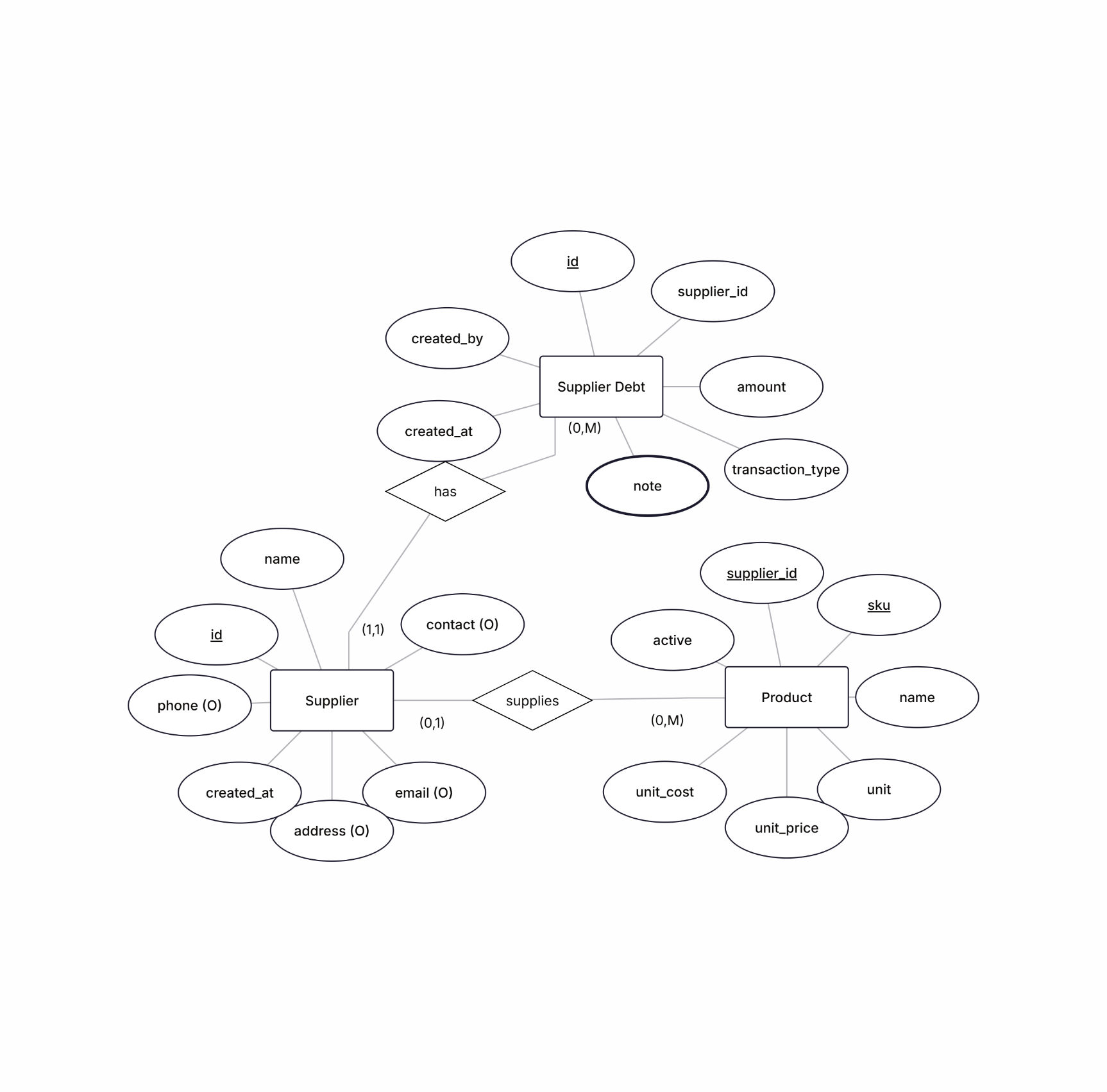
A diagram of a company

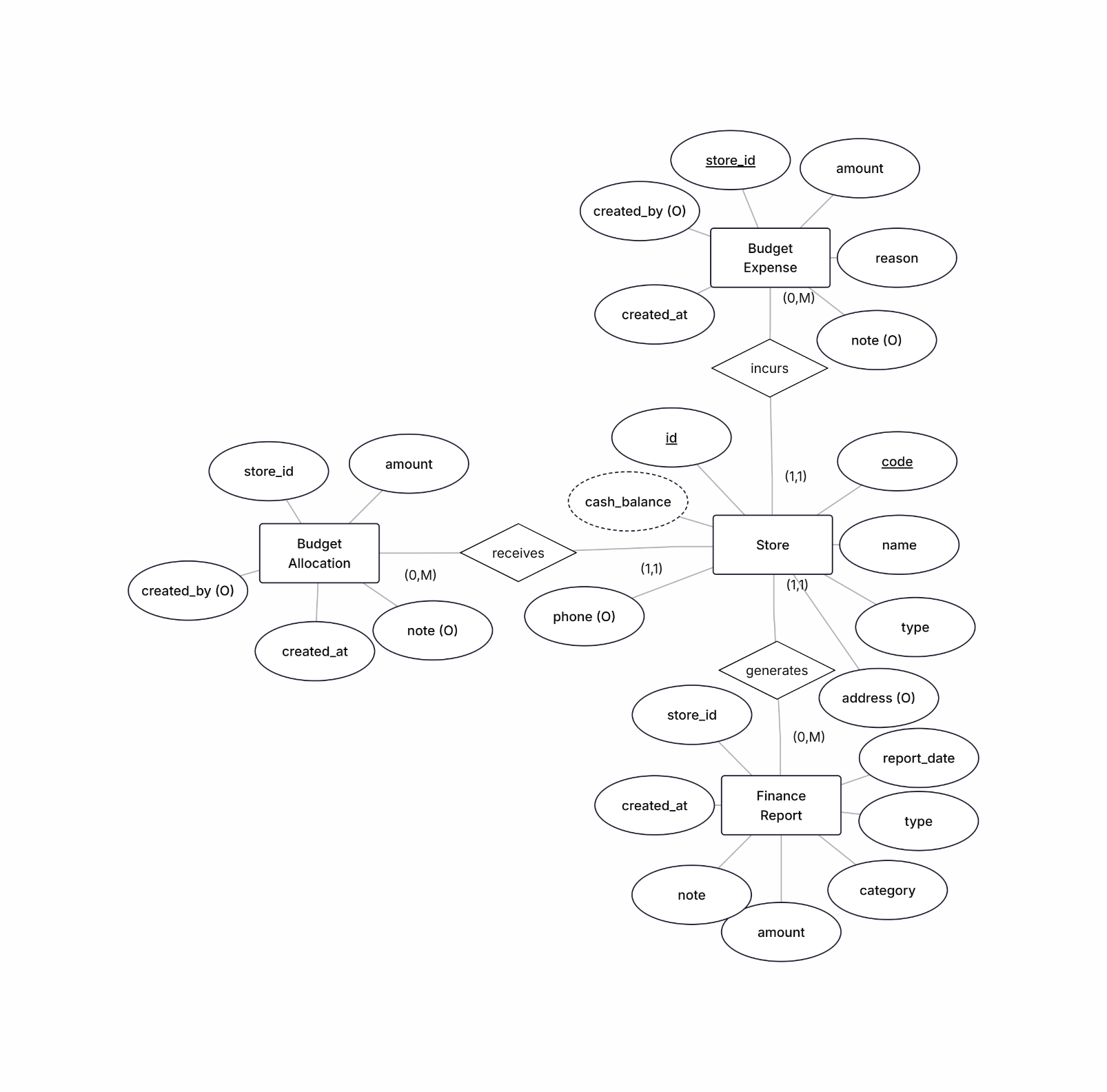
AI-generated content may be incorrect.

**Bán hàng & khách hàngA diagram of a company

AI-generated content may be incorrect.**

**Tồn kho & kho hàng**

**Nhà cung cấp & công nợ**

**Ngân sách, chi tiêu & báo cáo**

**Người dùng & cửa hàng**

# 4.2

**Tổng quan hệ thốngA computer screen shot of a computer

AI-generated content may be incorrect.**

**Bán hàng & khách hàngA screenshot of a computer

AI-generated content may be incorrect.**

**Tồn kho & kho hàngA screenshot of a computer

AI-generated content may be incorrect.**

**Nhà cung cấp & công nợA screen shot of a computer

AI-generated content may be incorrect.**

**Ngân sách, chi tiêu & báo cáoA screenshot of a computer

AI-generated content may be incorrect.**

**Người dùng & cửa hàng**A screenshot of a computer

AI-generated content may be incorrect.

**Tổng quan quan hệ chính (cardinality)**

* **Store (1) – (N) Order: mỗi order thuộc 1 store.**
* **Customer (0..1) – (N) Order: đơn có thể không gắn khách; khách có nhiều đơn.**
* **Order (1) – (N) OrderItem; OrderItem (N) – (1) Product.**
* **Customer (1) – (N) PointsLedger; Order (0..1) – (N) PointsLedger.**
* **Order (1) – (N) OrderEvent.**
* **Supplier (0..1) – (N) Product.**
* **Store (1) – (N) BudgetAllocation; Store (1) – (N) BudgetExpense; Store (1) – (N) FinanceReport.**
* **Store (1) – (N) StockMove; Product (1) – (N) StockMove; Inventory là bảng giao của Store–Product (PK kép).**
* **Supplier (1) – (N) SupplierDebt.**

**1) Bảng stores**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Tên trường | Kiểu dữ liệu | Kích thước | Ràng buộc toàn vẹn | Khuôn dạng (Coding) | Ghi chú |
| id | **BIGINT** | **8B** | **PK, AUTO\_INCREMENT, NOT NULL** | **–** | **Khóa chính** |
| code | **VARCHAR** | **16** | **UNIQUE, NOT NULL** | **Chữ in HOA/ số: ^[A-Z0-9]+$ (ví dụ S001)** | **Mã cửa hàng** |
| name | **VARCHAR** | **128** | **NOT NULL** | **Chuỗi UTF-8** | **Tên hiển thị** |
| type | **ENUM** | **–** | **NOT NULL, giá trị {CENTRAL,RETAIL}** | **–** | **Phân loại store** |
| address | **VARCHAR** | **255** | **NULL** | **–** | **Địa chỉ** |
| phone | **VARCHAR** | **32** | **NULL** | **^[0-9+\- ]{7,32}$** | **SĐT liên hệ** |
| cash\_balance | **DECIMAL** | **(14,2)** | **NOT NULL, DEFAULT 0.00** | **–** | **Derived bởi trigger từ allocations/expenses** |
| created\_at | **TIMESTAMP** | **–** | **NOT NULL, DEFAULT CURRENT\_TIMESTAMP** | **ISO-8601** | **Ngày tạo** |

**Ownership / Maintenance: cash\_balance chỉ thay đổi qua ghi budget\_allocations/budget\_expenses (trigger). Các field khác do Admin quản lý.**

**2) Bảng customers**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Tên trường | Kiểu dữ liệu | Kích thước | Ràng buộc toàn vẹn | Khuôn dạng | Ghi chú |
| id | **BIGINT** | **–** | **PK, AUTO\_INCREMENT, NOT NULL** | **–** |  |
| name | **VARCHAR** | **128** | **NOT NULL** | **–** |  |
| phone | **VARCHAR** | **32** | **UNIQUE, NULL** | **^[0-9+\- ]{7,32}$** | **Có thể bỏ trống** |
| email | **VARCHAR** | **128** | **UNIQUE, NULL** | **Email hợp lệ** | **Có thể bỏ trống** |
| points | **INT** | **–** | **NOT NULL, DEFAULT 0** | **–** | **Derived (tổng delta ledger), nhưng vẫn lưu để nhanh** |
| created\_at | **TIMESTAMP** | **–** | **NOT NULL, DEFAULT CURRENT\_TIMESTAMP** | **–** |  |

**Tham chiếu: được dùng bởi orders.customer\_id, points\_ledger.customer\_id.**

**3) Bảng suppliers**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Trường | Kiểu | Kích thước | Ràng buộc | Khuôn dạng | Ghi chú |
| id | **BIGINT** | **–** | **PK, AUTO\_INCREMENT, NOT NULL** | **–** |  |
| name | **VARCHAR** | **160** | **NOT NULL** | **–** |  |
| contact | **VARCHAR** | **128** | **NULL** | **–** | **Người liên hệ** |
| phone | **VARCHAR** | **32** | **NULL** | **^[0-9+\- ]{7,32}$** |  |
| email | **VARCHAR** | **128** | **NULL** | **Email** |  |
| address | **VARCHAR** | **255** | **NULL** | **–** |  |
| created\_at | **TIMESTAMP** | **–** | **NOT NULL, DEFAULT CURRENT\_TIMESTAMP** | **–** |  |

**Tham chiếu: products.supplier\_id (ON DELETE SET NULL), supplier\_debts.supplier\_id (RESTRICT).**

**4) Bảng products**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Trường | Kiểu | Kích thước | Ràng buộc | Khuôn dạng | Ghi chú |
| id | **BIGINT** | **–** | **PK, AUTO\_INCREMENT** | **–** |  |
| supplier\_id | **BIGINT** | **–** | **FK → suppliers.id, NULL, ON DELETE SET NULL** | **–** | **Tuỳ chọn** |
| sku | **VARCHAR** | **32** | **UNIQUE, NOT NULL** | **^[A-Z0-9\-]+$ (ví dụ SKU-TEA-200)** | **Mã SP** |
| name | **VARCHAR** | **160** | **NOT NULL** | **–** |  |
| unit | **VARCHAR** | **32** | **NOT NULL, DEFAULT 'pcs'** | **–** | **Đơn vị** |
| unit\_price | **DECIMAL** | **(12,2)** | **NOT NULL, DEFAULT 0.00** | **–** | **Giá bán đề xuất** |
| unit\_cost | **DECIMAL** | **(12,2)** | **NOT NULL, DEFAULT 0.00** | **–** | **Giá vốn** |
| active | **TINYINT(1)** | **–** | **NOT NULL, DEFAULT 1** | **0/1** | **Trạng thái** |
| created\_at | **TIMESTAMP** | **–** | **NOT NULL, DEFAULT CURRENT\_TIMESTAMP** | **–** |  |

**Tham chiếu: được tham chiếu bởi order\_items.product\_id, inventory.product\_id, stock\_moves.product\_id.**

**5) Bảng inventory**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Trường | Kiểu | Kích thước | Ràng buộc | Khuôn dạng | Ghi chú |
| store\_id | **BIGINT** | **–** | **PK (phần 1), FK → stores.id, NOT NULL, ON DELETE CASCADE** | **–** |  |
| product\_id | **BIGINT** | **–** | **PK (phần 2), FK → products.id, NOT NULL, ON DELETE CASCADE** | **–** |  |
| quantity | **INT** | **–** | **NOT NULL, DEFAULT 0** | **>= 0 (trigger đảm bảo không âm)** | **Derived từ stock\_moves** |
| updated\_at | **TIMESTAMP** | **–** | **NOT NULL, DEFAULT CURRENT\_TIMESTAMP, ON UPDATE CURRENT\_TIMESTAMP** | **–** |  |

**Integrity rule: Trigger trg\_moves\_after\_insert không cho thành số âm.**

**6) Bảng stock\_moves**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Trường | Kiểu | Kích thước | Ràng buộc | Khuôn dạng | Ghi chú |
| id | **BIGINT** | **–** | **PK, AUTO\_INCREMENT** | **–** |  |
| store\_id | **BIGINT** | **–** | **FK → stores.id, NOT NULL, ON DELETE CASCADE** | **–** |  |
| product\_id | **BIGINT** | **–** | **FK → products.id, NOT NULL, ON DELETE CASCADE** | **–** |  |
| move\_type | **ENUM** | **–** | **NOT NULL, giá trị {IMPORT,EXPORT,ADJUST}** | **–** | **Loại dịch chuyển** |
| quantity | **INT** | **–** | **NOT NULL** | **Dương với IMPORT/EXPORT; ADJUST có thể âm** |  |
| note | **VARCHAR** | **255** | **NULL** | **–** |  |
| created\_at | **TIMESTAMP** | **–** | **NOT NULL, DEFAULT CURRENT\_TIMESTAMP** | **–** |  |

**References: cập nhật inventory qua trigger. Maintenance: chỉ Logistic/Admin tạo/sửa.**

**7) Bảng orders**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Trường | Kiểu | Kích thước | Ràng buộc | Khuôn dạng | Ghi chú |
| id | **BIGINT** | **–** | **PK, AUTO\_INCREMENT** | **–** |  |
| order\_code | **VARCHAR** | **32** | **UNIQUE, NULL** | **Mẫu gợi ý ^ORD-[0-9]{4,}$** | **Mã đơn tự sinh** |
| store\_id | **BIGINT** | **–** | **FK → stores.id, NOT NULL, ON DELETE RESTRICT** | **–** |  |
| customer\_id | **BIGINT** | **–** | **FK → customers.id, NULL, ON DELETE SET NULL** | **–** | **Đơn có thể không gắn khách** |
| total\_amount | **DECIMAL** | **(14,2)** | **NOT NULL, DEFAULT 0.00** | **–** | **Derived = SUM(line\_total)** |
| discount | **DECIMAL** | **(12,2)** | **NOT NULL, DEFAULT 0.00** | **–** | **Giảm giá đơn** |
| status | **ENUM** | **–** | **NOT NULL, DEFAULT CONFIRMED; {CONFIRMED,PAID,CANCELED,REFUNDED}** | **–** |  |
| note | **VARCHAR** | **255** | **NULL** | **–** |  |
| paid\_at | **TIMESTAMP** | **–** | **NULL** | **–** | **Thời điểm thanh toán** |
| created\_at | **TIMESTAMP** | **–** | **NOT NULL, DEFAULT CURRENT\_TIMESTAMP** | **–** |  |

**Maintenance: Seller/Admin tạo; cập nhật status, paid\_at. Tổng tiền do trigger items cập nhật.**

**8) Bảng order\_items**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Trường | Kiểu | Kích thước | Ràng buộc | Khuôn dạng | Ghi chú |
| id | **BIGINT** | **–** | **PK, AUTO\_INCREMENT** | **–** |  |
| order\_id | **BIGINT** | **–** | **FK → orders.id, NOT NULL, ON DELETE CASCADE** | **–** |  |
| product\_id | **BIGINT** | **–** | **FK → products.id, NOT NULL, ON DELETE RESTRICT** | **–** |  |
| quantity | **INT** | **–** | **NOT NULL, > 0** | **–** |  |
| unit\_price | **DECIMAL** | **(12,2)** | **NOT NULL** | **–** | **Giá tại thời điểm bán** |
| line\_total | **DECIMAL** | **(14,2)** | **GENERATED STORED, NOT NULL** | **= quantity × unit\_price** | **Derived; không cho sửa tay** |

**Trigger: thêm/sửa/xoá item sinh stock\_moves tương ứng & cập nhật orders.total\_amount.**

**9) Bảng finance\_reports**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Trường | Kiểu | Kích thước | Ràng buộc | Khuôn dạng | Ghi chú |
| id | **BIGINT** | **–** | **PK, AUTO\_INCREMENT** | **–** |  |
| store\_id | **BIGINT** | **–** | **FK → stores.id, NOT NULL, ON DELETE CASCADE** | **–** |  |
| report\_date | **DATE** | **–** | **NOT NULL** | **YYYY-MM-DD** | **Ngày ghi nhận** |
| type | **ENUM** | **–** | **NOT NULL, {INCOME,EXPENSE}** | **–** |  |
| category | **VARCHAR** | **64** | **NULL** | **–** | **Phân loại** |
| amount | **DECIMAL** | **(14,2)** | **NOT NULL** | **–** | **Số tiền** |
| note | **VARCHAR** | **255** | **NULL** | **–** |  |
| created\_at | **TIMESTAMP** | **–** | **NOT NULL, DEFAULT CURRENT\_TIMESTAMP** | **–** |  |

**10) Bảng users**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Trường | Kiểu | Kích thước | Ràng buộc | Khuôn dạng | Ghi chú |
| id | **INT** | **–** | **PK, AUTO\_INCREMENT** | **–** |  |
| username | **VARCHAR** | **64** | **UNIQUE, NOT NULL** | **^[a-zA-Z0-9\_.-]{3,64}$** | **Tên đăng nhập** |
| email | **VARCHAR** | **128** | **UNIQUE, NULL** | **Email** | **Có thể trống** |
| password | **VARCHAR** | **128** | **NOT NULL** | **Băm (BCrypt/PBKDF2)** | **Lưu hash** |
| role | **ENUM** | **–** | **NOT NULL, DEFAULT user; {admin,user,seller,logistic,accountant}** | **–** | **Vai trò** |
| store\_id | **BIGINT** | **–** | **FK → stores.id, NULL, ON DELETE SET NULL** | **–** | **Nhân viên có thể gắn store** |
| created\_at | **TIMESTAMP** | **–** | **NOT NULL, DEFAULT CURRENT\_TIMESTAMP** | **–** |  |

**Ownership: Admin quản trị; người dùng chỉ sửa thông tin cá nhân phù hợp chính sách.**

**11) Bảng points\_ledger**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Trường | Kiểu | Kích thước | Ràng buộc | Khuôn dạng | Ghi chú |
| id | **BIGINT** | **–** | **PK, AUTO\_INCREMENT** | **–** |  |
| customer\_id | **BIGINT** | **–** | **FK → customers.id, NOT NULL, ON DELETE CASCADE** | **–** |  |
| order\_id | **BIGINT** | **–** | **FK → orders.id, NULL, ON DELETE SET NULL** | **–** | **Có thể điều chỉnh tay** |
| delta | **INT** | **–** | **NOT NULL** | **Dương/âm** | **Biến động điểm** |
| reason | **VARCHAR** | **128** | **NOT NULL** | **–** | **PURCHASE/REFUND/MANUAL\_ADJUST…** |
| note | **VARCHAR** | **255** | **NULL** | **–** |  |
| created\_at | **TIMESTAMP** | **–** | **NOT NULL, DEFAULT CURRENT\_TIMESTAMP** | **–** |  |

**12) Bảng order\_events**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Trường | Kiểu | Kích thước | Ràng buộc | Khuôn dạng | Ghi chú |
| id | **BIGINT** | **–** | **PK, AUTO\_INCREMENT** | **–** |  |
| order\_id | **BIGINT** | **–** | **FK → orders.id, NOT NULL, ON DELETE CASCADE** | **–** |  |
| event\_type | **ENUM** | **–** | **NOT NULL, {CREATED,CONFIRMED,PAID,CANCELED,REFUNDED,NOTE}** | **–** |  |
| data\_json | **JSON** | **–** | **NULL** | **JSON hợp lệ** | **Payload tuỳ chọn** |
| note | **VARCHAR** | **255** | **NULL** | **–** |  |
| created\_at | **TIMESTAMP** | **–** | **NOT NULL, DEFAULT CURRENT\_TIMESTAMP** | **–** |  |

**13) Bảng supplier\_debts**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Trường | Kiểu | Kích thước | Ràng buộc | Khuôn dạng | Ghi chú |
| id | **BIGINT** | **–** | **PK, AUTO\_INCREMENT** | **–** |  |
| supplier\_id | **BIGINT** | **–** | **FK → suppliers.id, NOT NULL, ON DELETE RESTRICT** | **–** |  |
| amount | **DECIMAL** | **(14,2)** | **NOT NULL** | **> 0** | **Số tiền ghi nợ/chi trả** |
| transaction\_type | **ENUM** | **–** | **NOT NULL, {ADD\_DEBT,PAY\_DEBT,ADJUSTMENT}** | **–** |  |
| note | **VARCHAR** | **255** | **NULL** | **–** |  |
| created\_at | **TIMESTAMP** | **–** | **NOT NULL, DEFAULT CURRENT\_TIMESTAMP** | **–** |  |
| created\_by | **VARCHAR** | **64** | **NULL** | **–** | **User tạo (không FK)** |

**14) Bảng budget\_allocations**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Trường | Kiểu | Kích thước | Ràng buộc | Khuôn dạng | Ghi chú |
| id | **BIGINT** | **–** | **PK, AUTO\_INCREMENT** | **–** |  |
| store\_id | **BIGINT** | **–** | **FK → stores.id, NOT NULL, ON DELETE RESTRICT** | **–** |  |
| amount | **DECIMAL** | **(14,2)** | **NOT NULL** | **> 0** | **Cấp quỹ** |
| note | **VARCHAR** | **255** | **NULL** | **–** |  |
| created\_at | **TIMESTAMP** | **–** | **NOT NULL, DEFAULT CURRENT\_TIMESTAMP** | **–** |  |
| created\_by | **VARCHAR** | **64** | **NULL** | **–** | **User tạo (không FK)** |

**Trigger: sau khi INSERT làm tăng stores.cash\_balance.**

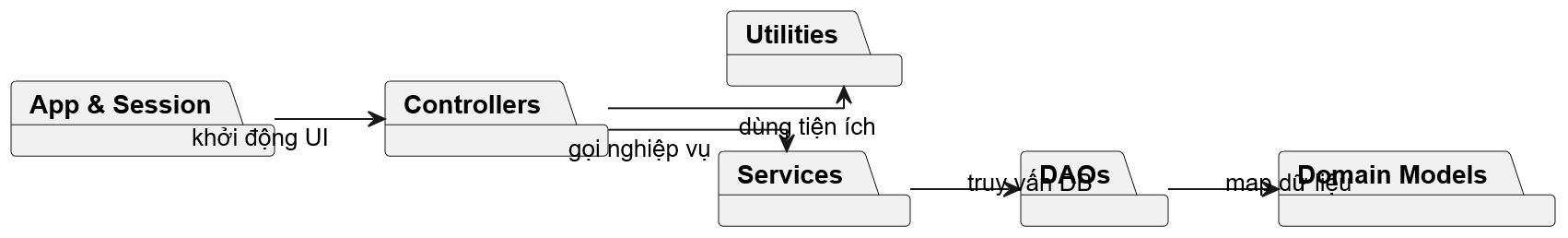
**15) Bảng budget\_expenses**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Trường | Kiểu | Kích thước | Ràng buộc | Khuôn dạng | Ghi chú |
| id | **BIGINT** | **–** | **PK, AUTO\_INCREMENT** | **–** |  |
| store\_id | **BIGINT** | **–** | **FK → stores.id, NOT NULL, ON DELETE RESTRICT** | **–** |  |
| amount | **DECIMAL** | **(14,2)** | **NOT NULL** | **> 0** | **Khoản chi** |
| reason | **VARCHAR** | **255** | **NOT NULL** | **–** | **Lý do chi** |
| note | **VARCHAR** | **500** | **NULL** | **–** |  |
| created\_at | **TIMESTAMP** | **–** | **NOT NULL, DEFAULT CURRENT\_TIMESTAMP** | **–** |  |
| created\_by | **VARCHAR** | **64** | **NULL** | **–** | **User tạo (không FK)** |

**Trigger: sau khi INSERT giảm stores.cash\_balance; nếu số dư không đủ → ROLLBACK.**

# PHẦN 4.3 Thiết kế chi tiết các gói

## Biểu đồ 1 – Kiến trúc các tầng



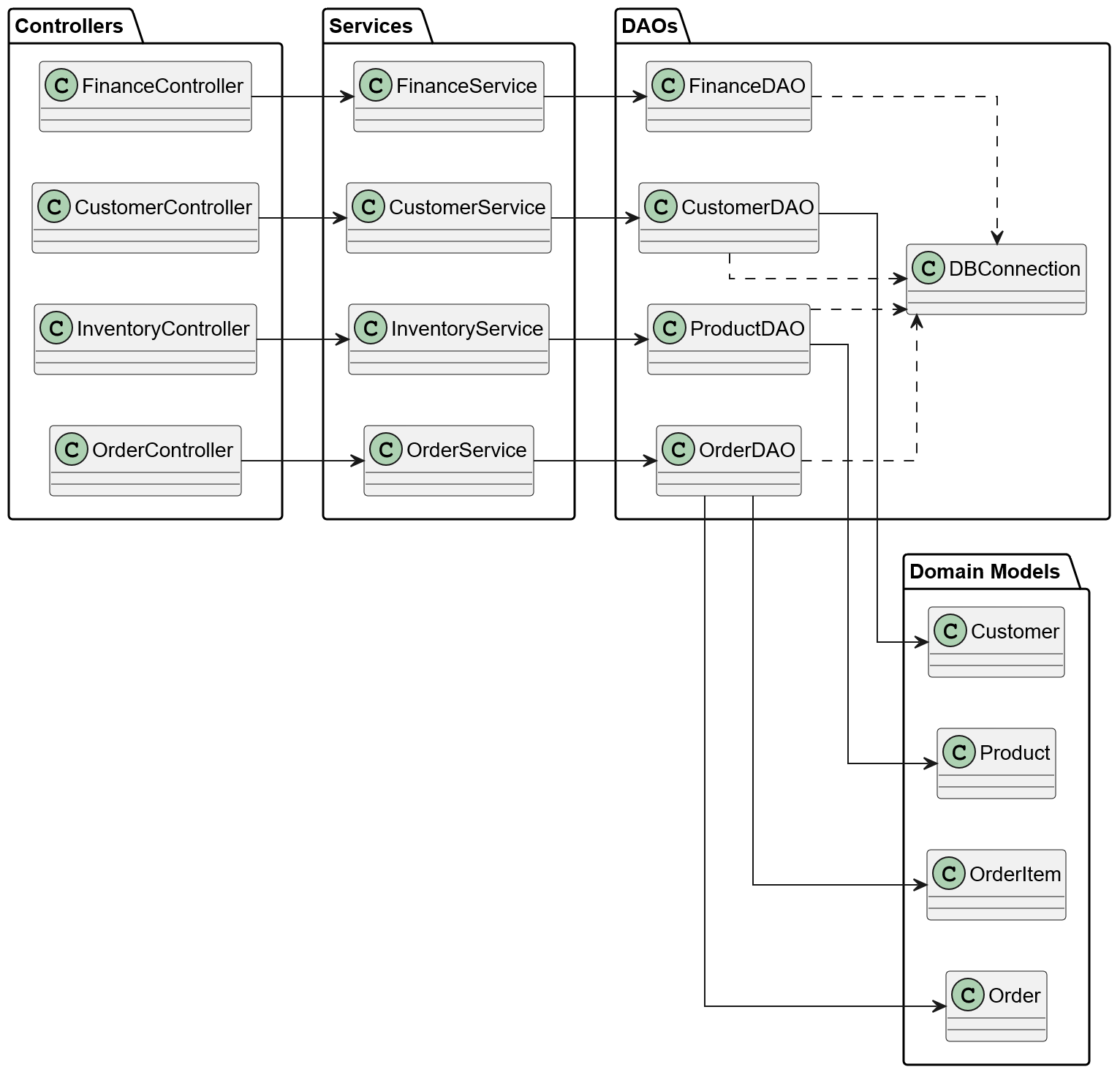
Biểu đồ này mô tả kiến trúc theo tầng của ứng dụng. Tầng App & Session chịu trách nhiệm khởi động giao diện và quản lý phiên làm việc, Controllers xử lý tương tác UI và gọi sang Services để thực hiện nghiệp vụ. Services lại sử dụng DAOs để truy vấn cơ sở dữ liệu và ánh xạ sang các Domain Models. Utilities cung cấp các hàm tiện ích dùng chung cho controller. Cách chia tầng như vậy giúp hệ thống rõ ràng, dễ bảo trì và dễ mở rộng.

## Biểu đồ 2 – Khởi động ứng dụng & quản lý session

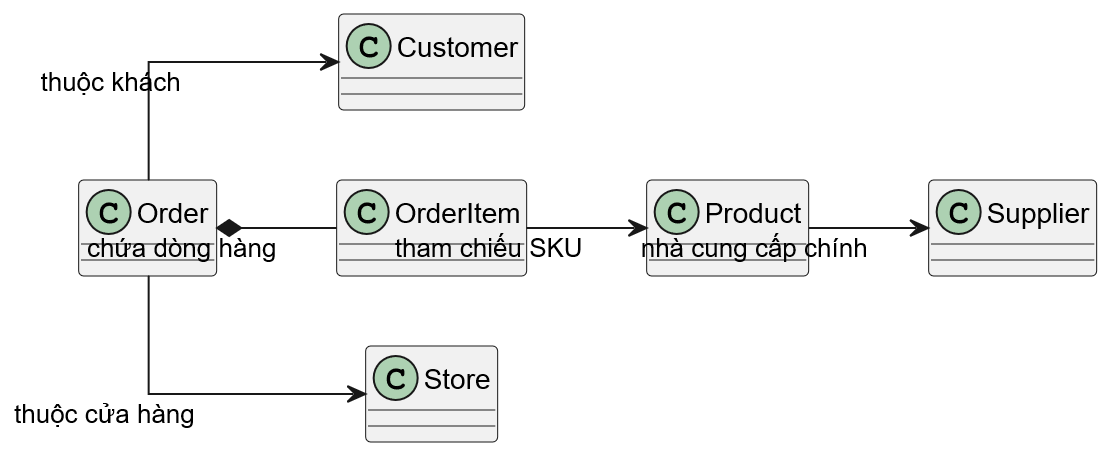


Biểu đồ này tập trung vào **luồng khởi động ứng dụng và quản lý phiên đăng nhập**. Lớp Main chịu trách nhiệm load FXML và hiển thị LoginController hoặc MainController. LoginController sử dụng PasswordUtils để kiểm tra mật khẩu và cập nhật Session khi đăng nhập thành công. MainController đọc thông tin từ Session và dùng AlertUtils để hiển thị thông báo cho người dùng. Thiết kế này tách rõ phần khởi động UI, logic login và trạng thái session.

## Biểu đồ 3 – Luồng nghiệp vụ: Controller → Service → DAO → DB & Model

 Biểu đồ minh hoạ **luồng xử lý nghiệp vụ đi từ giao diện xuống cơ sở dữ liệu**. Các controller như CustomerController, OrderController chỉ gọi sang các service tương ứng (CustomerService, OrderService...), service lại gọi các DAO (CustomerDAO, OrderDAO...) để truy vấn DB thông qua DBConnection. Kết quả truy vấn được ánh xạ vào các Domain Models như Customer, Order, OrderItem, Product. Nhờ vậy, controller không phải thao tác trực tiếp với DB, còn DAO chịu trách nhiệm riêng về truy xuất dữ liệu.

## Biểu đồ 4 – Quan hệ domain model bán hàng (Order, Item, Product, Supplier, Customer, Store)



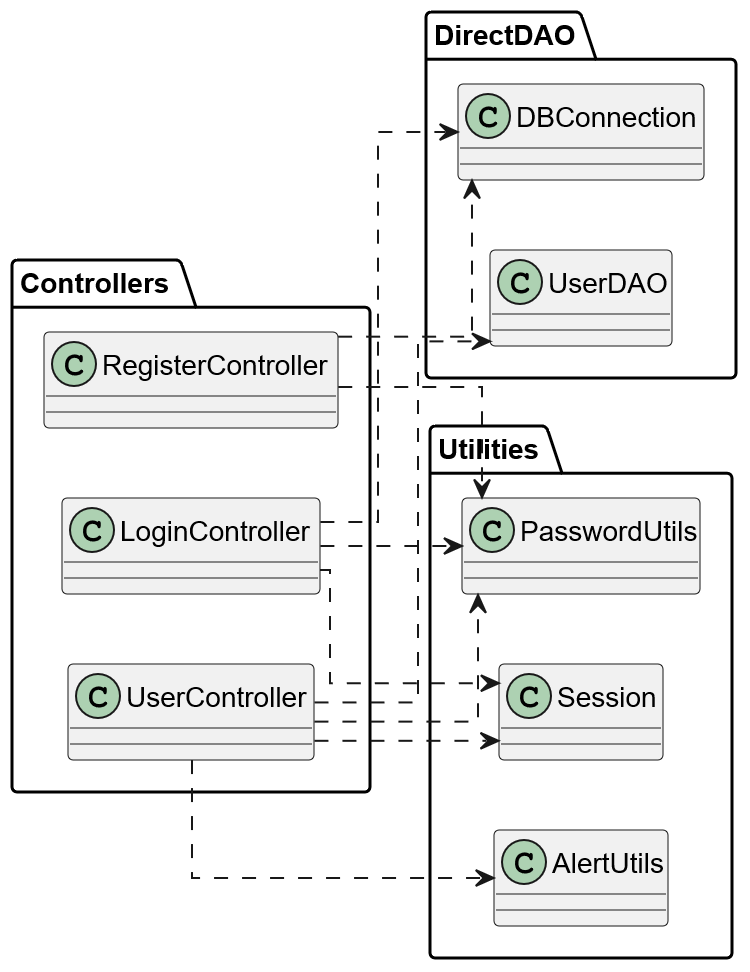
Biểu đồ thể hiện các **thực thể lõi trong nghiệp vụ bán hàng**. Order (đơn hàng) là trung tâm, chứa nhiều OrderItem biểu diễn từng dòng sản phẩm trong đơn. Mỗi đơn gắn với một Customer (khách mua) và một Store (cửa hàng thực hiện đơn). Một OrderItem tham chiếu tới Product để lấy thông tin SKU, giá, tồn kho, còn Product liên kết với Supplier là nhà cung cấp chính. Thiết kế này cho thấy rõ các quan hệ 1-n giữa đơn hàng–dòng hàng và mối liên hệ giữa khách, cửa hàng, sản phẩm, nhà cung cấp.

## Biểu đồ tổng quan kiến trúc Controller–Service–DAO–Utilities (rất đơn giản, 4 package).

## 

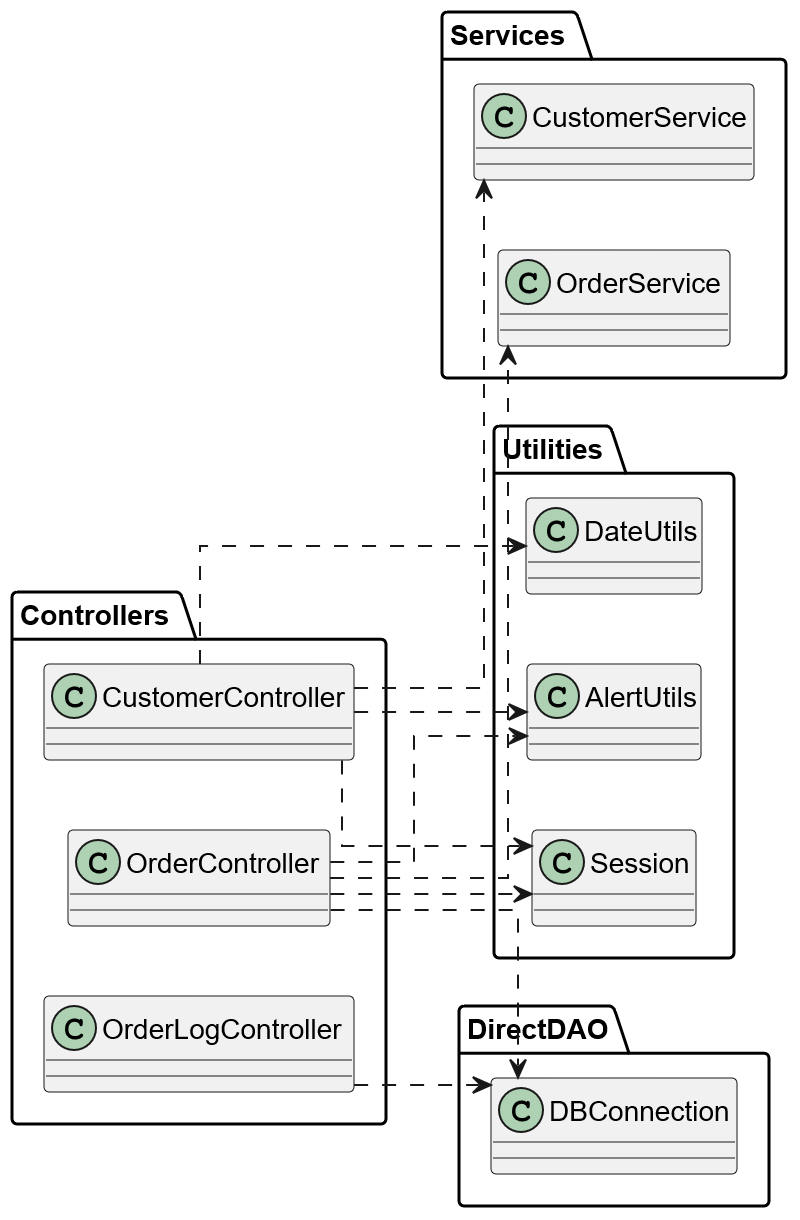
Biểu đồ thể hiện kiến trúc 4 tầng chính của hệ thống: Controllers xử lý tương tác giao diện, Services chứa nghiệp vụ, DirectDAO truy xuất dữ liệu và Utilities cung cấp các hàm tiện ích dùng chung. Các mũi tên phụ thuộc cho thấy Controller không tự xử lý nghiệp vụ nặng mà chủ yếu gọi Service và Utility, chỉ truy cập trực tiếp DAO trong một số trường hợp đơn giản. Cách tách lớp như vậy giúp mã nguồn dễ bảo trì, dễ mở rộng và rõ ràng vai trò từng tầng.

### Biểu đồ Nhóm 1: Đăng nhập, đăng ký & quản lý user



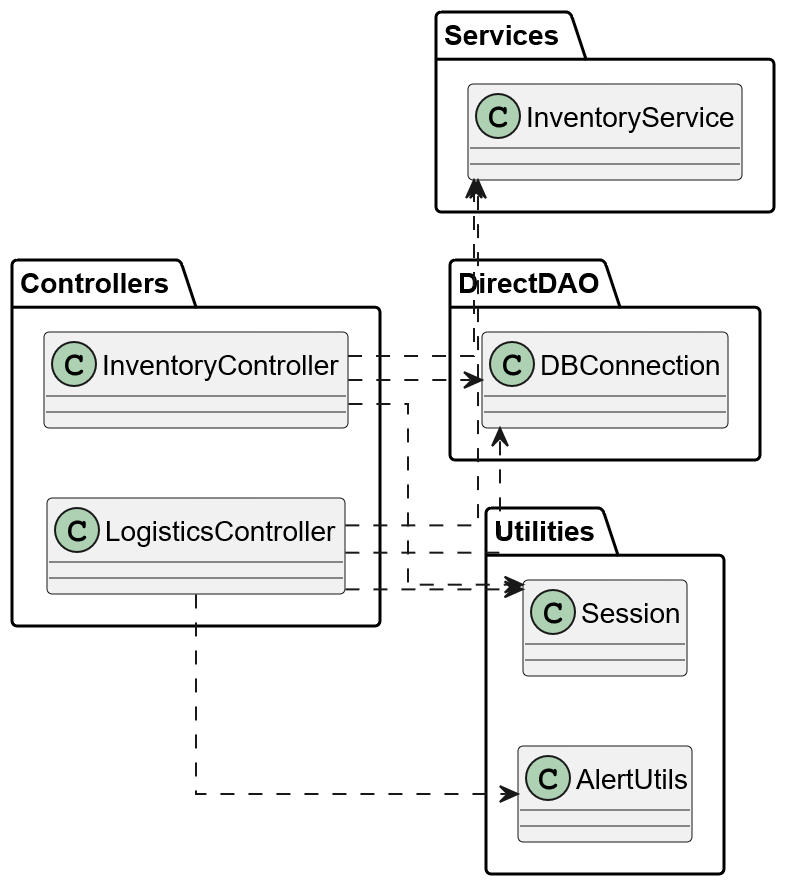
Biểu đồ này tập trung vào luồng xác thực và quản lý người dùng. LoginController và RegisterController chỉ nhận dữ liệu từ giao diện, sau đó sử dụng PasswordUtils để xử lý mật khẩu và dùng DBConnection/UserDAO để kiểm tra hoặc lưu thông tin người dùng. Lớp Session được dùng để lưu trạng thái đăng nhập, còn AlertUtils hiển thị thông báo cho người dùng. Thiết kế đảm bảo controller mỏng, logic xử lý được tách sang lớp tiện ích và DAO, giúp dễ kiểm thử và thay đổi về sau.

## Biểu đồ Nhóm 2: Khách hàng & đơn hàng



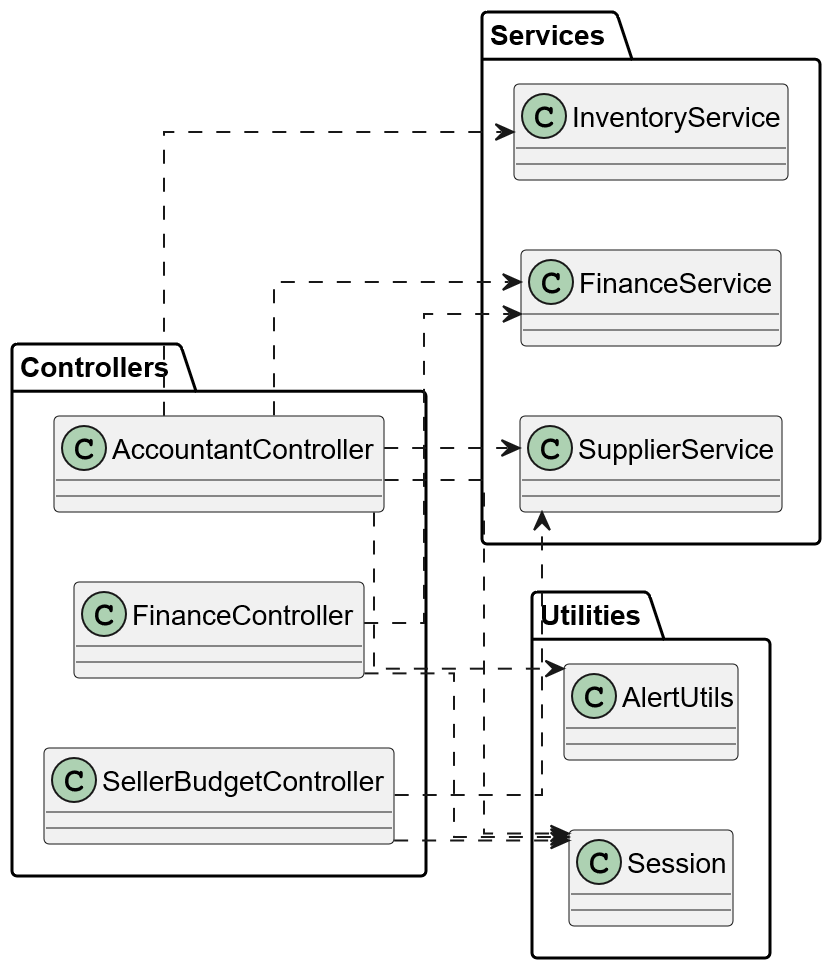
Biểu đồ mô tả các controller phục vụ quản lý khách hàng và đơn hàng. CustomerController và OrderController gọi lần lượt CustomerService và OrderService để thực hiện nghiệp vụ như tạo/sửa/xoá khách hàng, tạo đơn, tính tổng tiền… Các tiện ích AlertUtils và DateUtils hỗ trợ hiển thị thông báo và xử lý ngày tháng. Session được dùng để gắn thông tin người thao tác hoặc phân quyền. OrderLogController truy vấn log qua DBConnection để hiển thị lịch sử đơn hàng. Cách tách Controller – Service giúp thay đổi nghiệp vụ mà không ảnh hưởng giao diện.

## Biểu đồ chi tiết Nhóm 3 – Inventory & Logistics.



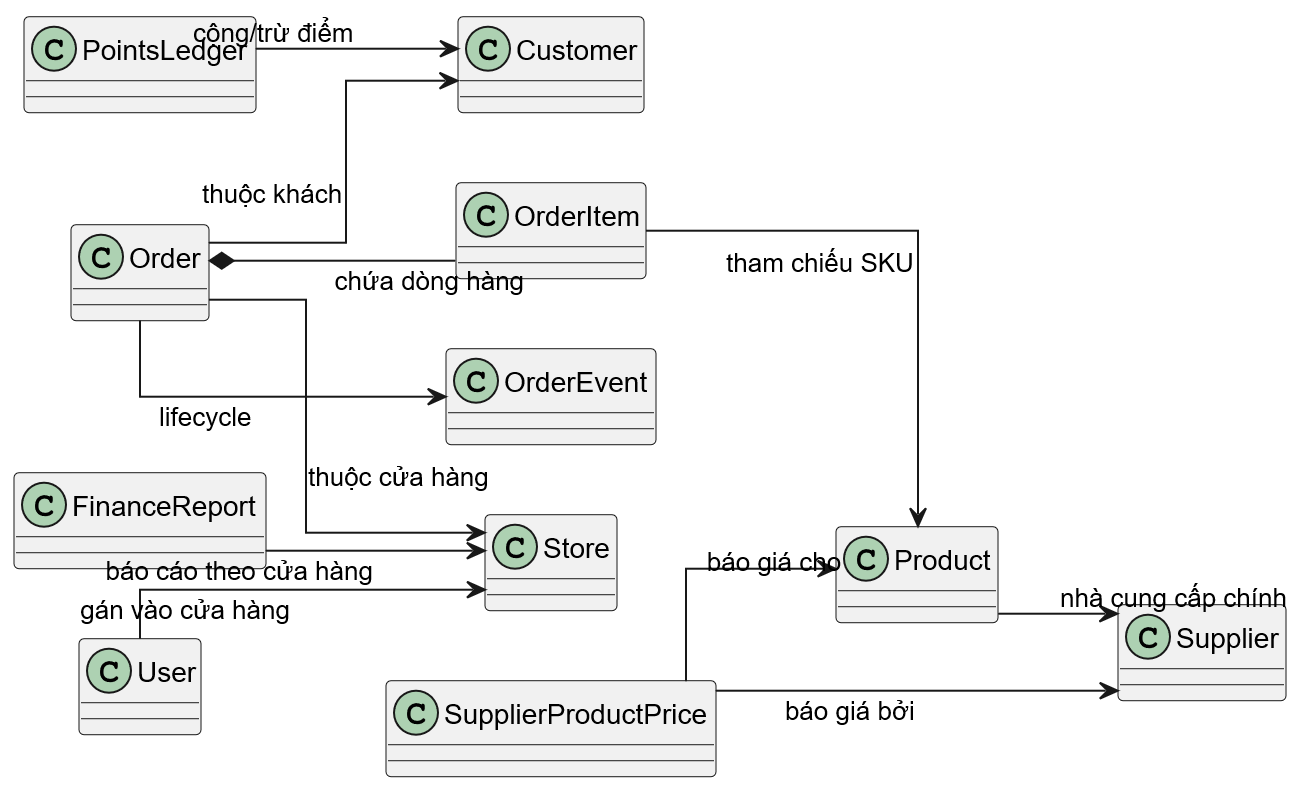
Biểu đồ thể hiện nhóm chức năng quản lý tồn kho và vận chuyển. InventoryController và LogisticsController không truy xuất DB trực tiếp mà chủ yếu gọi InventoryService để kiểm tra tồn kho, nhập/xuất hàng hoặc cập nhật trạng thái vận chuyển. DBConnection chỉ được dùng trong một số thao tác đơn giản. Session giúp xác định người dùng/chi nhánh đang thao tác, còn AlertUtils (nếu dùng) để báo kết quả cho người dùng. Thiết kế này gom các luồng liên quan đến kho và logistics vào cùng một nhóm, giúp nhìn rõ trách nhiệm và phụ thuộc giữa các lớp.

## Biểu đồ chi tiết Nhóm 4 – Finance & Accounting.

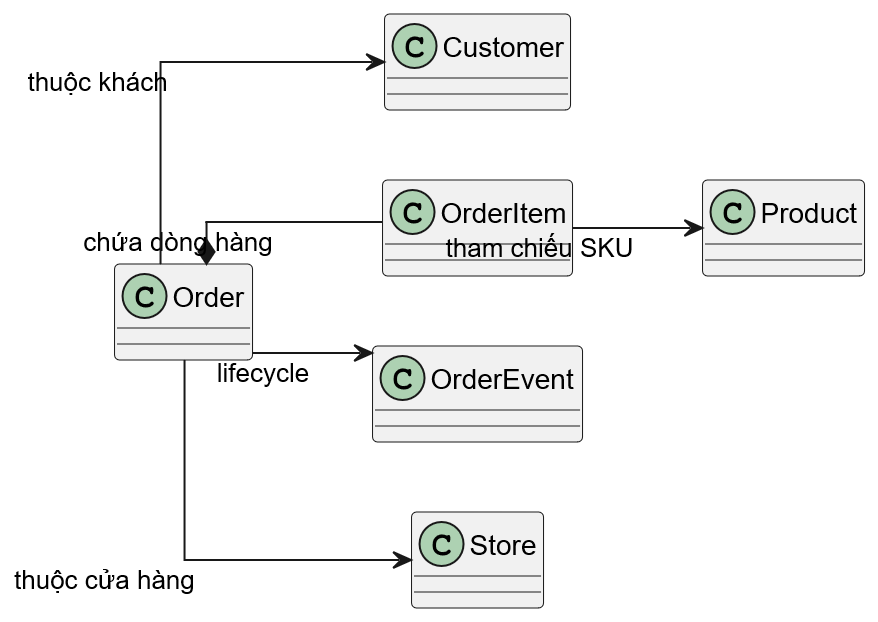


Biểu đồ nhóm các controller làm việc với báo cáo tài chính, kế toán và ngân sách nhà cung cấp. FinanceController dùng FinanceService để lấy số liệu doanh thu, chi phí, lợi nhuận. SellerBudgetController và AccountantController phối hợp nhiều service: SupplierService (nhà cung cấp), InventoryService (giá trị tồn kho) và FinanceService (số liệu tài chính) để thực hiện các nghiệp vụ kế toán phức tạp. Session dùng để kiểm tra quyền (ví dụ vai trò kế toán), còn AlertUtils hiển thị cảnh báo và thông báo lỗi. Thiết kế cho thấy rõ tầng controller chỉ điều phối giữa nhiều service, còn logic tài chính nằm ở tầng service giúp dễ mở rộng nghiệp vụ sau này.

# Biểu đồ tổng quan Model

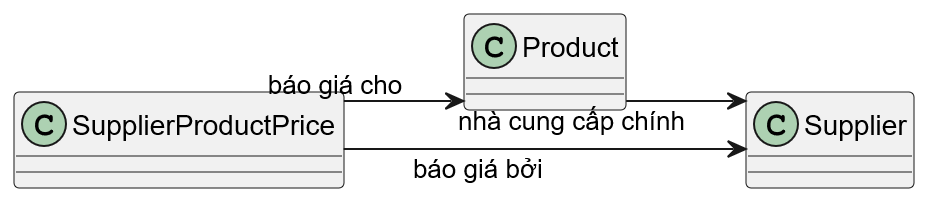


## Biểu đồ Nhóm 1 – Đơn hàng & khách hàng



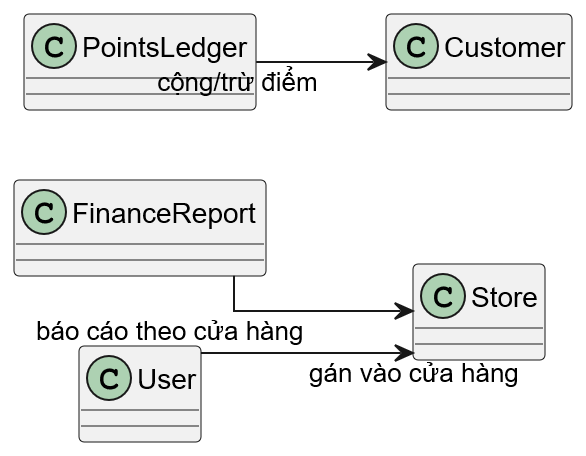
Biểu đồ này tập trung vào phần **bán hàng cốt lõi**. Order (đơn hàng) là thực thể trung tâm, **chứa nhiều** OrderItem thể hiện các dòng hàng chi tiết. Mỗi đơn gắn với một Customer (khách mua) và một Store (cửa hàng bán), đồng thời liên kết với OrderEvent để mô tả lifecycle/trạng thái của đơn (tạo, xác nhận, giao thành công, huỷ...). Từng OrderItem tham chiếu tới Product để lấy thông tin sản phẩm/SKU, tránh trùng lặp dữ liệu. Thiết kế này thể hiện rõ mối quan hệ 1-n giữa đơn hàng và dòng hàng, cũng như cách gắn đơn hàng với khách và cửa hàng.

## Biểu đồ Nhóm 2 – Sản phẩm & báo giá nhà cung cấp.



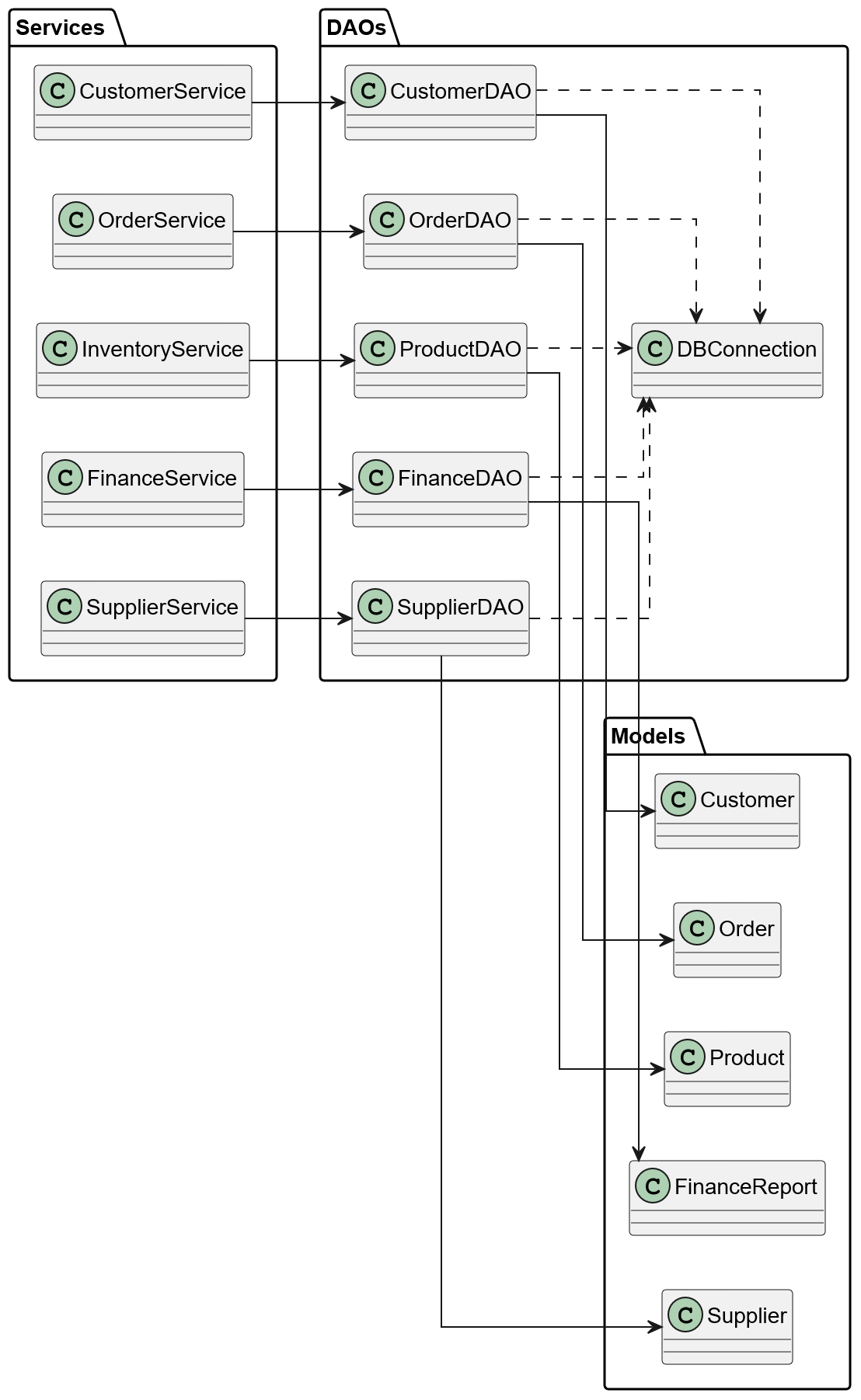
Biểu đồ này mô tả **quan hệ giữa sản phẩm và nhà cung cấp**. SupplierProductPrice đóng vai trò bảng báo giá trung gian, liên kết tới cả Supplier và Product để lưu nhiều mức giá/mốc thời gian khác nhau. Nhờ đó, một sản phẩm có thể được báo giá bởi nhiều nhà cung cấp, phục vụ nghiệp vụ so sánh giá nhập. Đồng thời, Product vẫn tham chiếu trực tiếp tới một Supplier như **nhà cung cấp chính**, giúp các thao tác nhập hàng nhanh hoặc hiển thị thông tin nhà cung cấp mặc định trở nên đơn giản hơn.

## Biểu đồ Nhóm 3 – Cửa hàng, báo cáo tài chính, user & tích điểm.



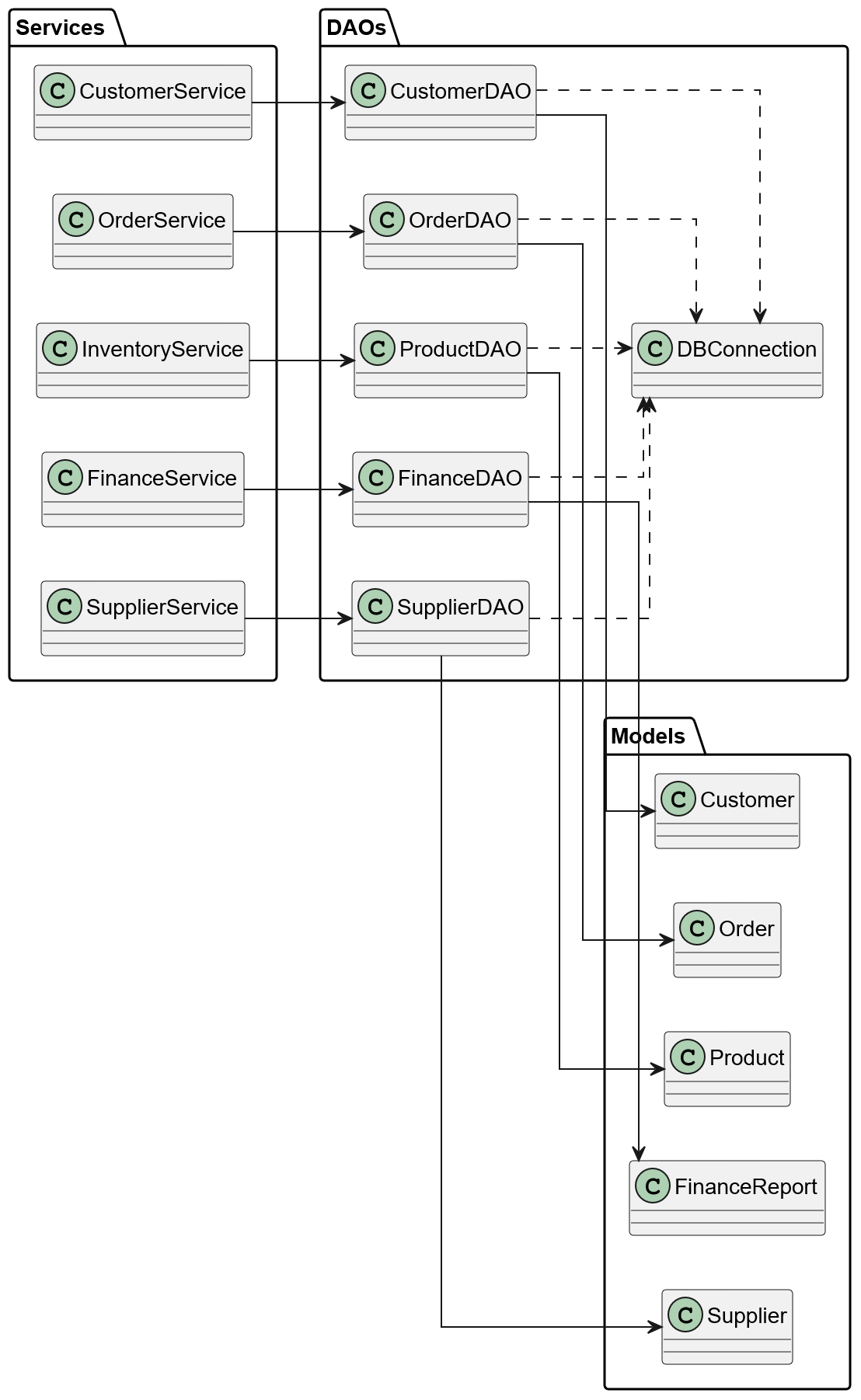
Biểu đồ này tập trung vào **quản lý theo cửa hàng và hệ thống tích điểm khách hàng**. Store là đơn vị kinh doanh, mỗi User (nhân viên/tài khoản đăng nhập) được gán vào một cửa hàng cụ thể để phân quyền và lọc dữ liệu. FinanceReport liên kết với Store để tạo báo cáo tài chính theo từng chi nhánh. Phần tích điểm được tách thành PointsLedger, là bảng nhật ký cộng/trừ điểm cho Customer, giúp truy vết lịch sử điểm thưởng độc lập với đơn hàng. Thiết kế này hỗ trợ tốt mô hình multi-store và dễ mở rộng các chính sách loyalty sau này.

## Biểu đồ 1 – Tổng quan Service–DAO–DBConnection–Model



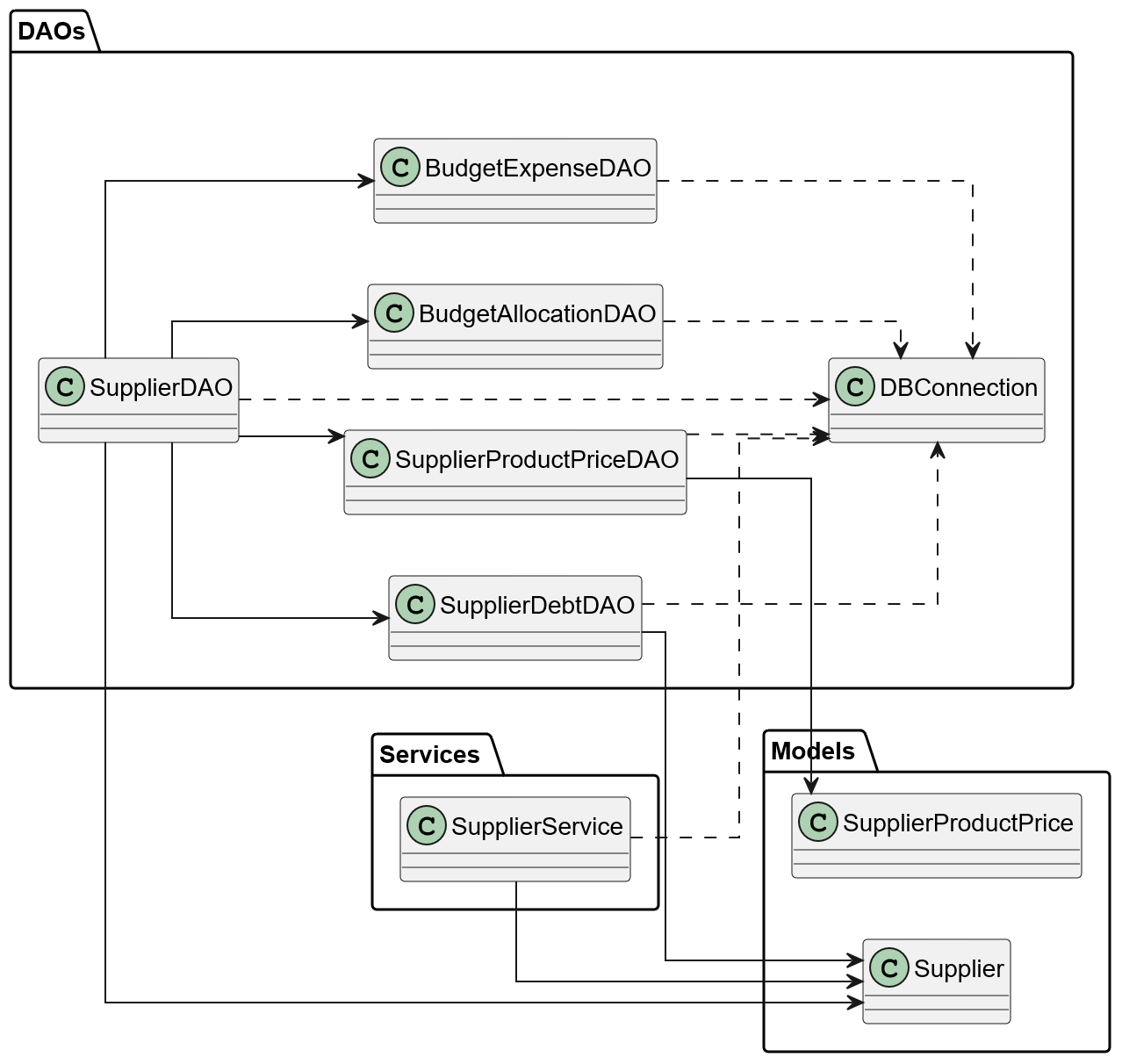
Biểu đồ này mô tả **kiến trúc tổng quan** của tầng Service/DAO. Mỗi Service (Customer, Order, Inventory, Finance, Supplier) chỉ làm việc với DAO tương ứng, còn các DAO mới trực tiếp dùng DBConnection để truy cập cơ sở dữ liệu. Kết quả truy vấn được ánh xạ vào các Models như Customer, Order, Product, FinanceReport, Supplier. Thiết kế giúp tách rõ **nghiệp vụ** (Service) và **truy xuất dữ liệu** (DAO), giảm phụ thuộc giữa các tầng.

## Biểu đồ 2 – Nhóm Customer, Order, Inventory (nghiệp vụ bán hàng chính)



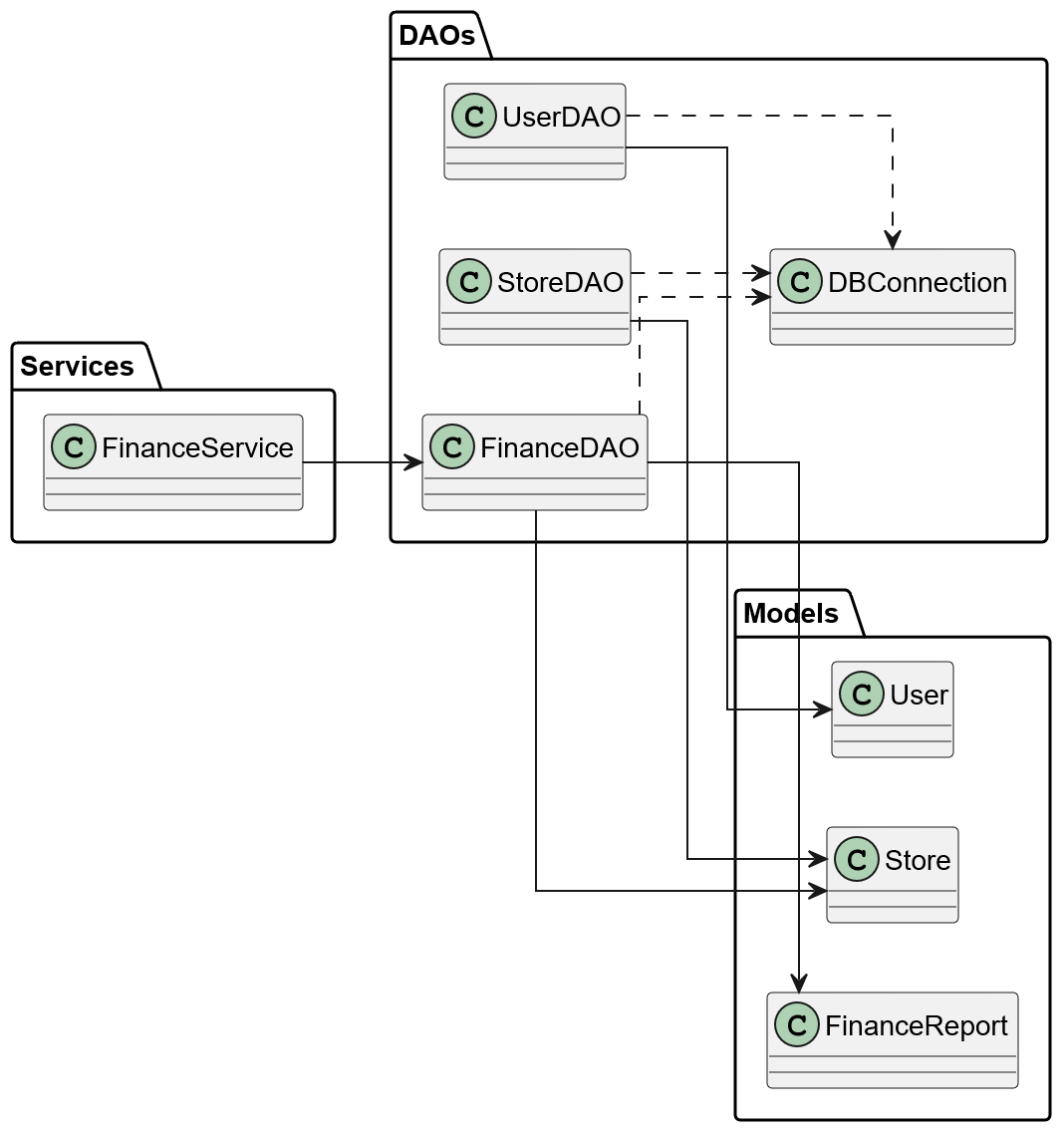
Biểu đồ này tập trung vào các **nghiệp vụ bán hàng cốt lõi**: khách hàng, đơn hàng và tồn kho. CustomerService, OrderService, InventoryService lần lượt làm việc với CustomerDAO, OrderDAO, ProductDAO để CRUD khách, xử lý đơn hàng và sản phẩm. OrderDAO và OrderEventDAO quản lý Order, OrderItem, OrderEvent để lưu chi tiết đơn và lịch sử trạng thái. ProductDAO gắn sản phẩm với Supplier và Store, cho phép kiểm soát tồn kho theo từng cửa hàng. Tất cả DAO đều dùng chung DBConnection, đảm bảo tập trung hoá việc kết nối DB.

## Biểu đồ 3 – Nhóm Supplier & ngân sách/báo giá nhà cung cấp



Biểu đồ này mô tả các lớp liên quan đến **nhà cung cấp và ngân sách**. SupplierService làm việc với Supplier và các DAO phía sau để phục vụ nghiệp vụ nhập hàng, quản lý công nợ và báo giá. SupplierDAO là “hub” trung tâm, điều phối sang SupplierProductPriceDAO, SupplierDebtDAO, BudgetAllocationDAO, BudgetExpenseDAO để truy vấn bảng báo giá, công nợ và phân bổ/chi tiêu ngân sách. Mỗi DAO dùng DBConnection riêng để truy cập DB, còn các model Supplier và SupplierProductPrice lưu thông tin nhà cung cấp và báo giá chi tiết. Thiết kế này gom tất cả logic liên quan supplier/budget vào một cụm thống nhất.

## Biểu đồ 4 – Nhóm Finance, Store & User (báo cáo và thông tin hệ thống)



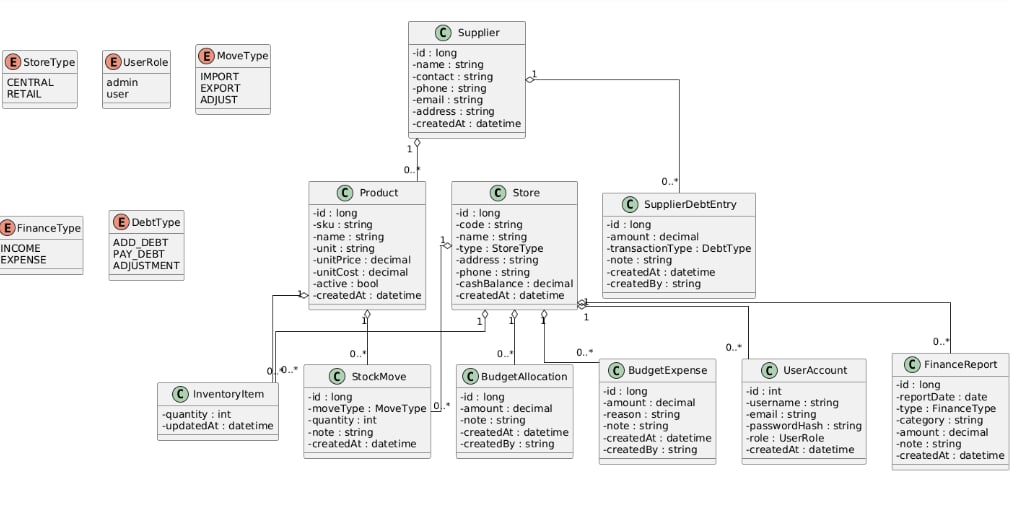
Biểu đồ này gom các lớp phục vụ **báo cáo tài chính và thông tin hệ thống**. FinanceService chỉ làm việc với FinanceDAO để lấy dữ liệu và lập FinanceReport theo từng Store. StoreDAO và UserDAO quản lý thông tin cửa hàng và người dùng, đều dùng chung DBConnection để truy vấn DB. Việc tách riêng nhóm Finance/Store/User giúp rõ ràng phần báo cáo và cấu hình hệ thống, tách khỏi các nghiệp vụ bán hàng và nhà cung cấp ở các biểu đồ khác.

## 4.4 Thiết kế chi tiết lớp

# A blue box with black text AI-generated content may be incorrect. A blue screen with black text AI-generated content may be incorrect. A blue rectangular object with black text AI-generated content may be incorrect. A blue rectangular object with black text AI-generated content may be incorrect. A blue rectangular object with black text AI-generated content may be incorrect. A blue box with black text AI-generated content may be incorrect. A blue card with black text AI-generated content may be incorrect. A blue screen with text AI-generated content may be incorrect. A blue box with black text AI-generated content may be incorrect. A screenshot of a computer AI-generated content may be incorrect. A blue screen with black text AI-generated content may be incorrect. A blue box with black text AI-generated content may be incorrect.

# 4.5 Sơ dồ lớp chi tiết

**0) Tổng quan**

****

**1) Lớp & quan hệ đăng nhập/điều hướng**

A screenshot of a computer program

AI-generated content may be incorrect.

**2) Lớp luồng bán hàng & kho**

A screenshot of a computer

AI-generated content may be incorrect.A blue paper with black text

AI-generated content may be incorrect.A screenshot of a computer program

AI-generated content may be incorrect.

**3) tài chính**

A screenshot of a computer program

AI-generated content may be incorrect.