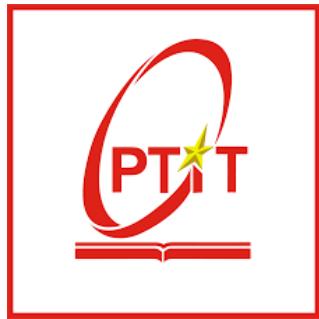


HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG
KHOA CÔNG NGHỆ THÔNG TIN



BÁO CÁO BÀI TẬP LỚN
PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG THÔNG TIN
--- NHÓM LỚP 04---

Đề tài 2: Hệ thống quản lý bán hàng siêu thị

Nhóm 09

Đào Quang Duy	B20DCCN146
Cao Duy Hải	B20DCCN218
Phan Trọng Kiều	B20DCCN362

- Quản lý thông tin mặt hàng
- Thanh toán cho khách hàng tại quầy
- Xem thống kê khách hàng theo doanh thu

I. Requirement

1. Mô tả các chức năng bằng ngôn ngữ tự nhiên

*) Quản lý thông tin mặt hàng:

- Quản lý vào hệ thống => GD đăng nhập: username, password, nút đăng nhập.
- Quản lý nhập username và password + click đăng nhập:
 - Nếu đăng nhập sai => thông báo sai username/password – click ok của thông báo => quay về GD đăng nhập và chứa thông tin cũ.
 - Nếu đăng nhập đúng => GD chính của quản lý xuất hiện:
 - Có ít nhất 1 nút/link quản lý thông tin mặt hàng.
 - Một nút đăng xuất.
- Quản lý click quản lý thông tin mặt hàng => GD quản lý thông tin mặt hàng:
 - Ô tìm kiếm, nút tìm kiếm, một nút thêm, một nút quay lại.
 - Danh sách các mặt hàng mà siêu thị đang bán.
 - Mỗi mặt hàng trên 1 dòng:
 - Mã MH, tên MH, loại MH, kích thước, mô tả, giá bán, số lượng tồn, lô sản xuất, ngày sản xuất, hạn sử dụng.
 - Nút/link xem, xóa, sửa.
- Nếu quản lý nhập vào ô tìm kiếm tên mặt hàng/loại mặt hàng + click tìm kiếm.
 - Nếu mặt hàng/loại mặt hàng không tồn tại => hệ thống hiển thị thông báo không tồn tại mặt hàng/loại mặt hàng – click ok của thông báo => danh sách đầy đủ các mặt hàng mà siêu thị đang bán được hiển thị và vẫn ở nguyên GD quản lý thông tin mặt hàng.
 - Nếu mặt hàng/loại mặt hàng tồn tại, danh sách các mặt hàng liên quan được hiển thị và vẫn ở nguyên GD quản lý thông tin mặt hàng.
- Nếu quản lý click xem 1 dòng mặt hàng => GD xem mặt hàng:
 - Thông tin mặt hàng hiển thị mỗi thứ một dòng: mã MH, tên MH, loại MH, kích thước, mô tả, giá bán, số lượng tồn, lô sản xuất, ngày sản xuất, hạn sử dụng.
 - Nút/link đóng => click đóng – GD xem mặt hàng tắt, quay lại GD quản lý thông tin mặt hàng.
- Nếu quản lý click sửa 1 dòng mặt hàng => GD sửa mặt hàng:
 - Thông tin mặt hàng hiển thị như GD xem mặt hàng.
 - Nút/link để quay lại/lưu.
- Nếu quản lý click quay lại: hệ thống không thay đổi và quay về GD quản lý thông tin mặt hàng.

- Nếu quản lý nhập thông tin cần sửa và click lưu => hệ thống cập nhật thông tin mặt hàng và quay lại GD quản lý thông tin mặt hàng.
- Nếu quản lý click vào xóa 1 dòng mặt hàng => hệ thống hiển thị thông báo xác nhận xóa.
 - Nếu quản lý chọn không xóa, hệ thống không thay đổi và ở nguyên GD quản lý thông tin mặt hàng.
 - Nếu quản lý chọn xóa, hệ thống xóa mặt hàng khỏi DB và ở nguyên GD quản lý thông tin mặt hàng.
- Nếu quản lý click vào thêm mặt hàng => GD thêm mặt hàng:
 - Các ô nhập thông tin mỗi thứ một dòng: mã MH, loại MH, kích thước, mô tả, giá bán.
 - Nút/link để quay lại/lưu.
- Nếu quản lý click quay lại: hệ thống không thay đổi và quay về GD quản lý thông tin mặt hàng.
- Nếu quản lý nhập các thông tin và click lưu => hệ thống lưu thông tin mặt hàng vào DB và quay lại GD quản lý thông tin mặt hàng.
- Nếu quản lý click quay lại ở GD quản lý thông tin mặt hàng => hệ thống quay lại GD chính của quản lý.

*) Thanh toán cho khách hàng tại quầy:

- Thu ngân vào hệ thống => GD đăng nhập: username, password, nút đăng nhập.
- Thu ngân nhập username và password + click đăng nhập:
 - Nếu đăng nhập sai => thông báo sai username/password – click ok của thông báo => quay về GD đăng nhập và chưa thông tin cũ.
 - Nếu đăng nhập đúng => GD chính của thu ngân xuất hiện:
 - Có ít nhất 1 nút/link thanh toán.
 - Một nút đăng xuất.
- Thu ngân click thanh toán => GD thanh toán:
 - Thông tin hóa đơn đang tạo: mã HD, tên siêu thị, địa chỉ siêu thị, mã thu ngân, tên thu ngân, thời gian thanh toán.
 - Một nút/link dùng thẻ thành viên.
 - Một nút/link dùng phiếu giảm giá.
 - Một ô hiển thị mã thẻ thành viên (mã khách hàng).
 - Một ô hiển thị mã phiếu giảm giá.
 - Bảng để điền thông tin cho mặt hàng tiếp theo:

- Thông tin mặt hàng: mã MH, số lượng, đơn giá, ưu đãi, thành tiền.
 - Nút/link để lưu/xóa.
 - Bảng để hiển thị danh sách các mặt hàng khách hàng mua gồm các thông tin: mã MH, tên MH, đơn giá, số lượng, số tiền.
 - Một dòng hiển thị số mặt hàng, tổng tiền trước khi giảm giá.
 - Một dòng hiển thị số tiền giảm giá.
 - Một dòng hiển thị số tiền phải thanh toán.
 - Một nút/link thanh toán bằng tiền mặt.
 - Một nút/link hủy.
- Nếu khách hàng hủy thanh toán, thu ngân click hủy => hệ thống hiển thị thông báo xác nhận hủy:
 - Nếu thu ngân click không hủy của thông báo: hệ thống không thay đổi và vẫn ở nguyên GD thanh toán với các thông tin cũ.
 - Nếu thu ngân click xác nhận của thông báo: hệ thống quay lại GD chính của thu ngân.
 - Thu ngân nhập mã MH và số lượng + click lưu:
 - Nếu mặt hàng chưa được thêm trên hệ thống => thông báo mặt hàng chưa tồn tại – click ok của thông báo và xin lỗi khách hàng.
 - Nếu mặt hàng đã có trên hệ thống: hệ thống lưu mặt hàng vào bảng danh sách mặt hàng khách hàng mua và hiển thị trên GD thanh toán, mỗi mặt hàng trên một dòng.
 - Nếu khách hàng không muốn mua/thay đổi số lượng mặt hàng đã lưu vào bảng, thu ngân chọn dòng của mặt hàng đó:
 - Thông tin mặt hàng đó hiển thị ở bảng thông tin cho mặt hàng tiếp theo.
 - Nếu thu ngân click xóa => thông báo xác nhận xóa:
 - Nếu thu ngân chọn không xóa: hệ thống không thay đổi gì và vẫn ở nguyên GD thanh toán với các thông tin cũ.
 - Nếu thu ngân chọn xóa: vẫn ở nguyên GD thanh toán và hệ thống xóa mặt hàng đó khỏi bảng danh sách mặt hàng khách hàng mua.
 - Nếu thu ngân nhập lại số lượng + click lưu: hệ thống cập nhật lại bảng danh sách mặt hàng khách hàng mua và vẫn ở nguyên GD thanh toán.
 - Nếu khách hàng có thẻ thành viên, thu ngân click thẻ thành viên => GD dùng thẻ thành viên:
 - Một ô để nhập mã thẻ, một nút/link kiểm tra.
 - Một bảng để hiển thị thông tin khách hàng.

- Mỗi thông tin trên một dòng: mã KH, tên KH, số điện thoại, email, địa chỉ.
 - Nút/link xác nhận/quay lại.
- Thu ngân nhập mã thẻ + click kiểm tra:
 - Nếu mã thẻ không tồn tại => thông báo mã thẻ không tồn tại hoặc bị khóa – click ok của thông báo => quay lại GD thanh toán với các thông tin cũ và thông báo với khách hàng.
 - Nếu mã thẻ tồn tại: thông tin khách hàng hiển thị lên GD dùng thẻ thành viên.
 - Thu ngân xác nhận với khách hàng và click xác nhận => quay lại GD thanh toán với các thông tin cũ, có thêm mã thẻ thành viên.
- Nếu khách hàng có mã giảm giá, thu ngân click thêm phiếu giảm giá => GD dùng phiếu giảm giá:
 - Một ô nhập mã phiếu giảm giá, một nút/link kiểm tra.
 - Một bảng để hiển thị thông tin phiếu giảm giá: mã phiếu, giá trị, ngày phát hành, ngày nhận, ngày hết hạn.
 - Nút/link xác nhận/quay lại.
- Thu ngân nhập mã phiếu giảm giá + click kiểm tra:
 - Nếu mã phiếu không tồn tại => thông báo mã phiếu không tồn tại hoặc đã hết hạn – click ok của thông báo => quay lại GD thanh toán với các thông tin cũ và thông báo với khách hàng.
 - Nếu mã phiếu tồn tại: thông tin phiếu giảm giá hiển thị lên GD dùng phiếu giảm giá
 - Thu ngân click xác nhận => quay lại GD thanh toán với các thông tin cũ, có thêm mã thẻ giảm giá, cập nhật số tiền giảm giá, số tiền phải thanh toán.
- Sau khi nhập xong các mặt hàng khách muốn mua, thu ngân thông báo cho khách hàng số tiền phải thanh toán và hỏi khách hàng muốn thanh toán bằng tiền mặt, thẻ ngân hàng hay chuyển khoản.
- Nếu khách hàng thanh toán bằng tiền mặt, thu ngân click thanh toán bằng tiền mặt trong GD thanh toán => GD thanh toán bằng tiền mặt:
 - Một dòng hiển thị tổng tiền phải trả.
 - Một ô để nhập tiền khách hàng đưa.
 - Một dòng hiển thị tiền thối lại cho khách.
 - Nút/link để xác nhận/quay lại.

- Nếu khách hàng muốn thay đổi danh sách các mặt hàng cần thanh toán, thu ngân click quay lại ở GD thanh toán bằng tiền mặt => hệ thống quay lại GD thanh toán với các thông tin cũ.
- Nếu thu ngân nhập số tiền khách đưa và click xác nhận ở GD thanh toán bằng tiền mặt => hệ thống hiển thị thông báo xác nhận thanh toán:
 - Nếu thu ngân click hủy của thông báo: hệ thống không thay đổi và vẫn ở GD thanh toán bằng tiền mặt.
 - Nếu thu ngân click xác nhận của thông báo: hệ thống lưu hóa đơn vào DB và hiển thị thông báo thanh toán thành công.
- Thu ngân thông báo, trả tiền thừa cho khách (nếu có) và click ok của thông báo => quay về GD chính của thu ngân.

*) Xem thống kê khách hàng theo doanh thu:

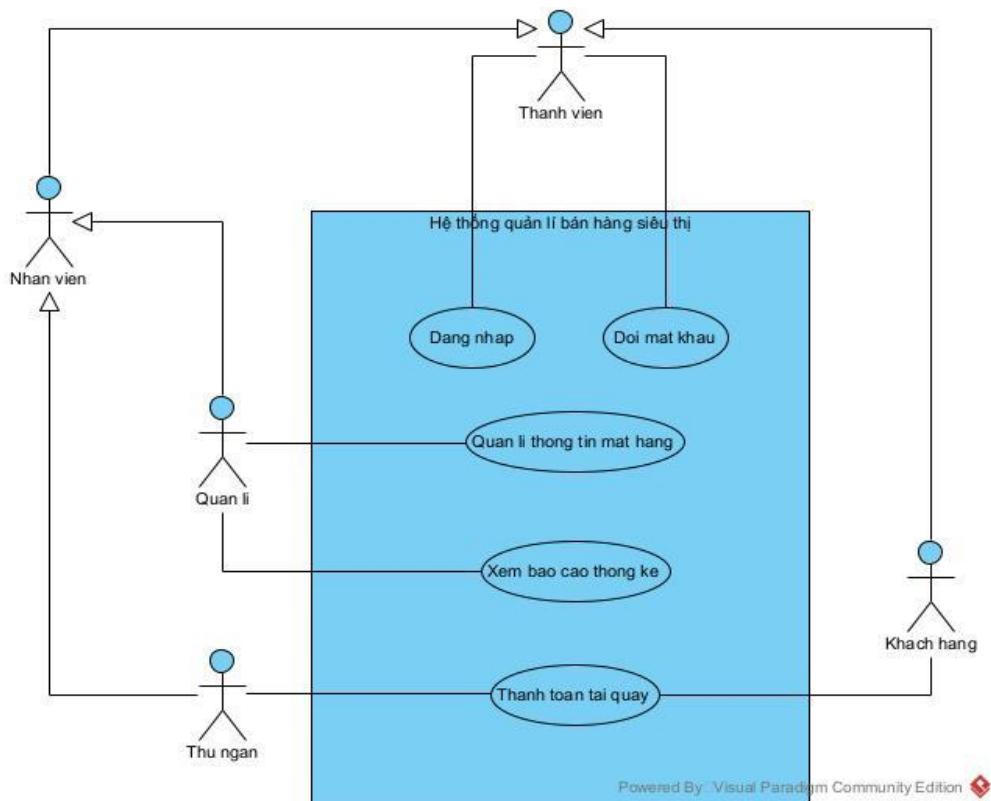
- Quản lý vào hệ thống => GD đăng nhập: username, password, nút đăng nhập.
- Quản lý nhập username và password + click đăng nhập:
 - Nếu đăng nhập sai => thông báo sai username/password – click ok của thông báo => quay về GD đăng nhập và chưa thông tin cũ.
 - Nếu đăng nhập đúng => GD chính của quản lý xuất hiện:
 - Có ít nhất 1 nút/link xem báo cáo thống kê.
 - Một nút đăng xuất.
- Quản lý click xem báo cáo thống kê => GD chọn loại báo cáo:
 - Danh sách chọn đối tượng báo cáo: khách hàng, các loại mặt hàng.
 - Danh sách chọn tiêu chí báo cáo: doanh thu, tỉ lệ mua hàng.
 - Nút/link quay lại/xác nhận.
- Quản lý chọn đối tượng là khách hàng, tiêu chí là doanh thu + click xác nhận => GD xem báo cáo thống kê khách hàng:
 - Hai ô để chọn ngày bắt đầu và ngày kết thúc.
 - Một nút/link kiểm tra, một nút/link quay lại.
 - Một ô để tìm kiếm, một nút/link tìm kiếm.
 - Một danh sách để hiển thị kết quả gồm các thông tin: mã KH, tên KH, số điện thoại, email, địa chỉ, tổng tiền hàng đã mua.
 - Một ô để hiển thị tổng doanh thu trong khoảng thời gian chọn báo cáo.
- Quản lý chọn ngày bắt đầu và ngày kết thúc + click kiểm tra:
 - Nếu ngày bắt đầu chưa diễn ra => thông báo chọn lại ngày bắt đầu sớm hơn – click ok của thông báo => quay lại GD xem báo cáo thống kê khách hàng và chưa các thông tin cũ.

- Nếu ngày bắt đầu lớn hơn ngày kết thúc => thông báo vui lòng chọn ngày bắt đầu nhỏ hơn ngày kết thúc – click ok của thông báo => quay lại GD xem báo cáo thống kê khách hàng và chứa các thông tin cũ.
 - Nếu ngày bắt đầu đã diễn ra và ngày kết thúc lớn hơn hoặc bằng ngày bắt đầu => danh sách các khách hàng có thẻ thành viên hiện ra (sắp xếp giảm dần theo tổng tiền mua hàng trong khoảng thời gian chọn để báo cáo) và ở nguyên GD xem báo cáo thống kê khách hàng.
- Nếu quản lý nhập tên khách hàng/mã khách hàng vào ô tìm kiếm + click tìm kiếm:
 - Nếu khách hàng không tồn tại => thông báo không tồn tại khách hàng – click ok của thông báo => quay lại GD xem báo cáo thống kê khách hàng với các thông tin cũ.
 - Nếu khách hàng tồn tại, danh sách các khách hàng liên quan được hiển thị và vẫn ở nguyên GD xem báo cáo thống kê khách hàng.
- Nếu quản lý chọn vào 1 dòng khách hàng => GD xem báo cáo thống kê khách hàng chi tiết:
 - Thông tin khách hàng mỗi thứ một dòng: mã KH, tên KH, số điện thoại, email, địa chỉ.
 - Danh sách các hóa đơn của khách hàng đó được xếp theo ngày thanh toán.
 - Mỗi hóa đơn trên 1 dòng: mã HD, mã thu ngân, địa chỉ chi nhánh mua sắm, thời gian thanh toán, tổng tiền.
 - Một nút/link quay lại.
- Nếu quản lý chọn 1 dòng hóa đơn => GD xem hóa đơn chi tiết:
 - Thông tin hóa đơn mỗi thứ một dòng: mã HD, tên siêu thị, địa chỉ siêu thị, số điện thoại khách hàng, mã thu ngân, tên thu ngân, thời gian thanh toán.
 - Danh sách các mặt hàng được mua trong hóa đơn đầy.
 - Mỗi mặt hàng trên 1 dòng: mã MH, tên MH, đơn giá, số lượng, số tiền.
 - Một dòng hiển thị tổng tiền trước khi giảm giá.
 - Một dòng hiển thị số lượng mặt hàng.
 - Thông tin phiếu giảm giá (nếu có): mã phiếu giảm giá, giá trị phiếu.
 - Một dòng hiển thị số tiền được giảm.
 - Một dòng hiển thị số tiền phải thanh toán.
 - Một dòng hiển thị phương thức thanh toán.
 - Một dòng hiển thị số tiền khách trả.
 - Một dòng hiển thị số tiền thối lại cho khách.

- Một nút/link quay lại.

2. Usecase tổng quan

- Đề xuất các actor của hệ thống: khách hàng, thu ngân, quản lý. Tất cả đều có chức năng giống thành viên của hệ thống nên kế thừa từ thành viên. Riêng thu ngân, quản lý còn kế thừa từ nhân viên của siêu thị, nhân viên kế thừa trực tiếp từ thành viên.
- Các chức năng tương ứng với từng actor:
 - Thành viên: đăng nhập, đổi mật khẩu.
 - Khách hàng: tham gia gián tiếp vào chức năng: thanh toán tại quầy.
 - Thu ngân: thanh toán tại quầy.
 - Quản lý: quản lý thông tin mặt hàng, xem báo cáo thống kê.

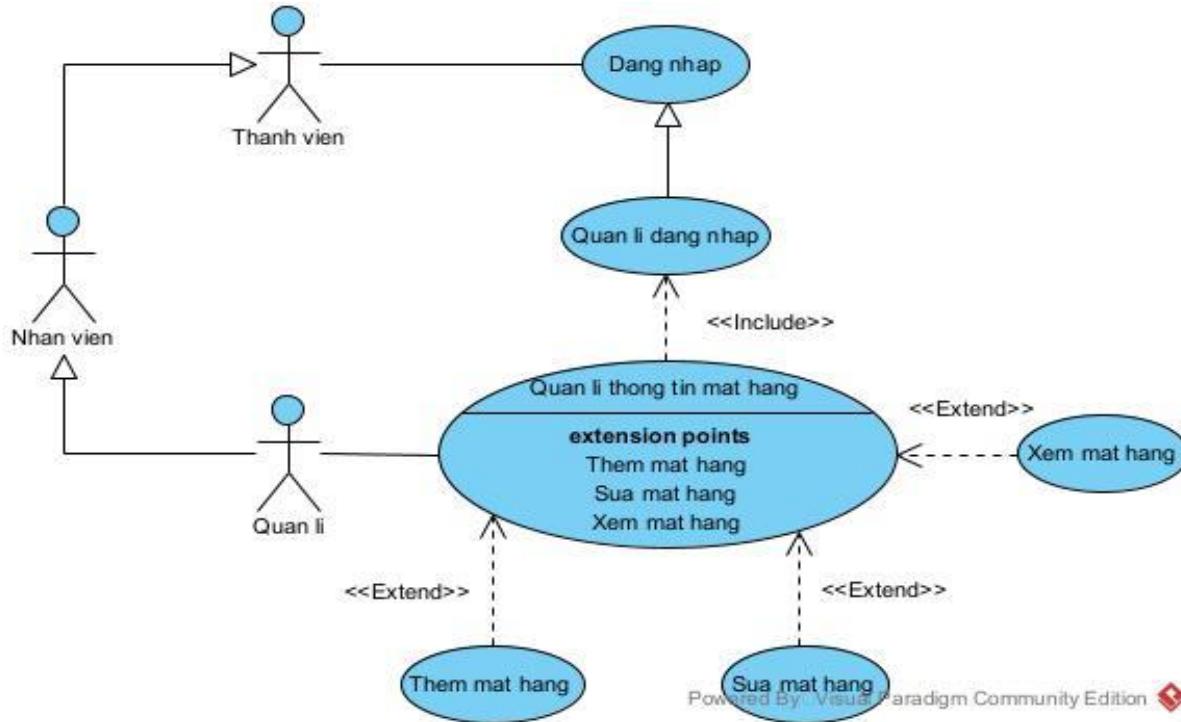


- Mô tả các UC:
 - Đăng nhập: UC cho phép thành viên đăng nhập vào hệ thống.
 - Đổi mật khẩu: UC cho phép thành viên đổi mật khẩu của tài khoản của bản thân trong hệ thống.
 - Quản lý thông tin mặt hàng: UC này cho phép quản lý quản lý thông tin mặt hàng theo thông tin của mặt hàng tương ứng.

- Thanh toán tại quầy: UC này cho phép thanh toán hóa đơn cho khách tại quầy giao dịch.
- Xem báo cáo thống kê: UC này cho phép quản lí xem các loại báo cáo thống kê về doanh thu theo khách, lượng hàng nhập...

3. Usecase chi tiết

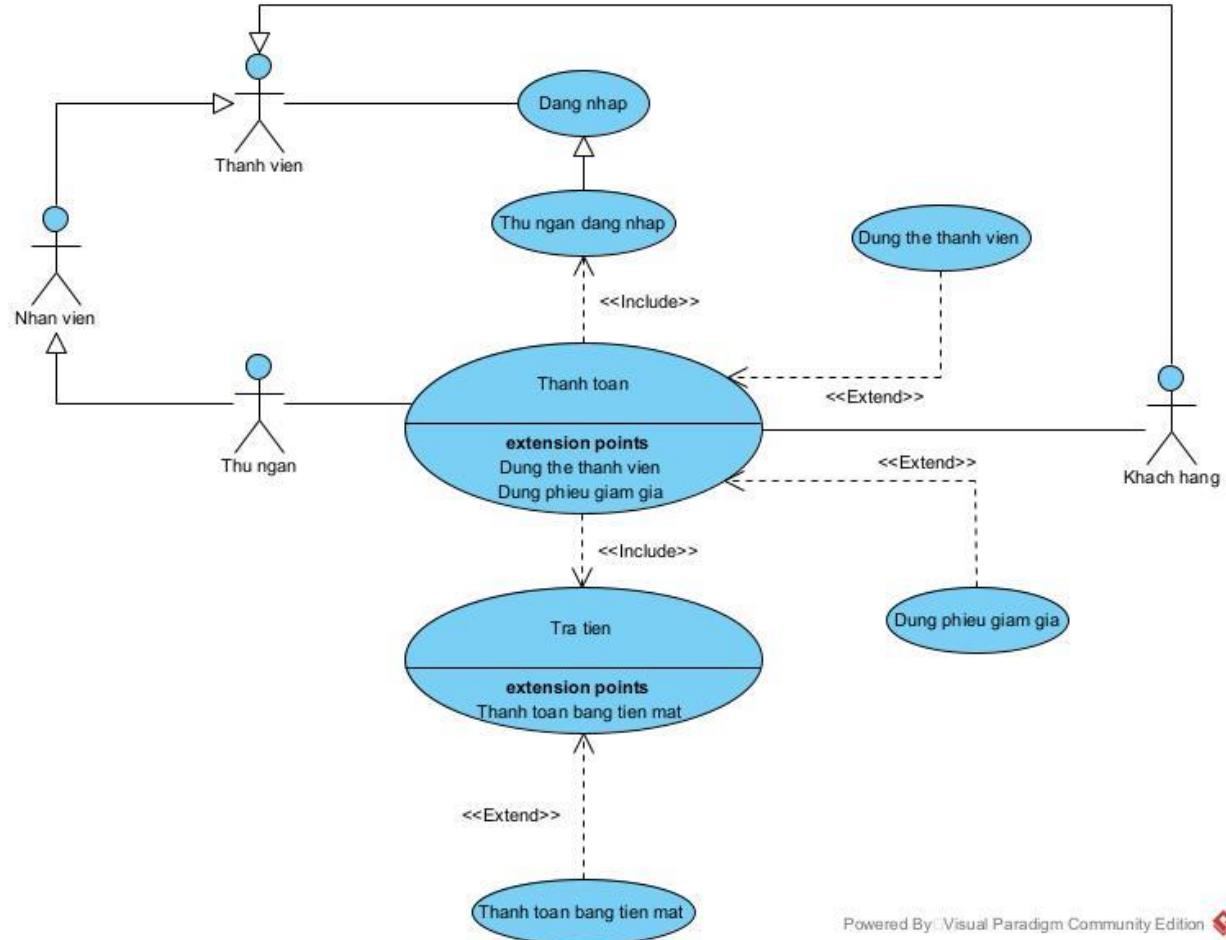
a, Quản lí thông tin mặt hàng



- Actor liên quan: quản lí.
- Chức năng quản lí thông tin mặt hàng có các giao diện tương tác với quản lí:
 - Đăng nhập -> để xuất UC đăng nhập.
 - Quản lí thông tin mặt hàng -> để xuất UC quản lí thông tin mặt hàng.
 - Xem mặt hàng -> để xuất UC xem mặt hàng.
 - Sửa mặt hàng -> để xuất UC sửa mặt hàng.
 - Thêm mặt hàng -> để xuất UC thêm mặt hàng.
- Quan hệ của UC con với UC chính:
 - Quản lí thông tin mặt hàng <<include>> đăng nhập.
 - Quản lí thông tin mặt hàng <<extend>> xem mặt hàng.
 - Quản lí thông tin mặt hàng <<extend>> sửa mặt hàng.
 - Quản lí thông tin mặt hàng <<extend>> thêm mặt hàng.
- Các UC được mô tả như sau:

- Xem mặt hàng: UC này cho phép quản lí xem thông tin chi tiết của một mặt hàng.
- Sửa mặt hàng: UC này cho phép quản lí sửa thông tin chi tiết của một mặt hàng.
- Thêm mặt hàng: UC này cho phép quản lí thêm thông tin chi tiết của một mặt hàng mới.

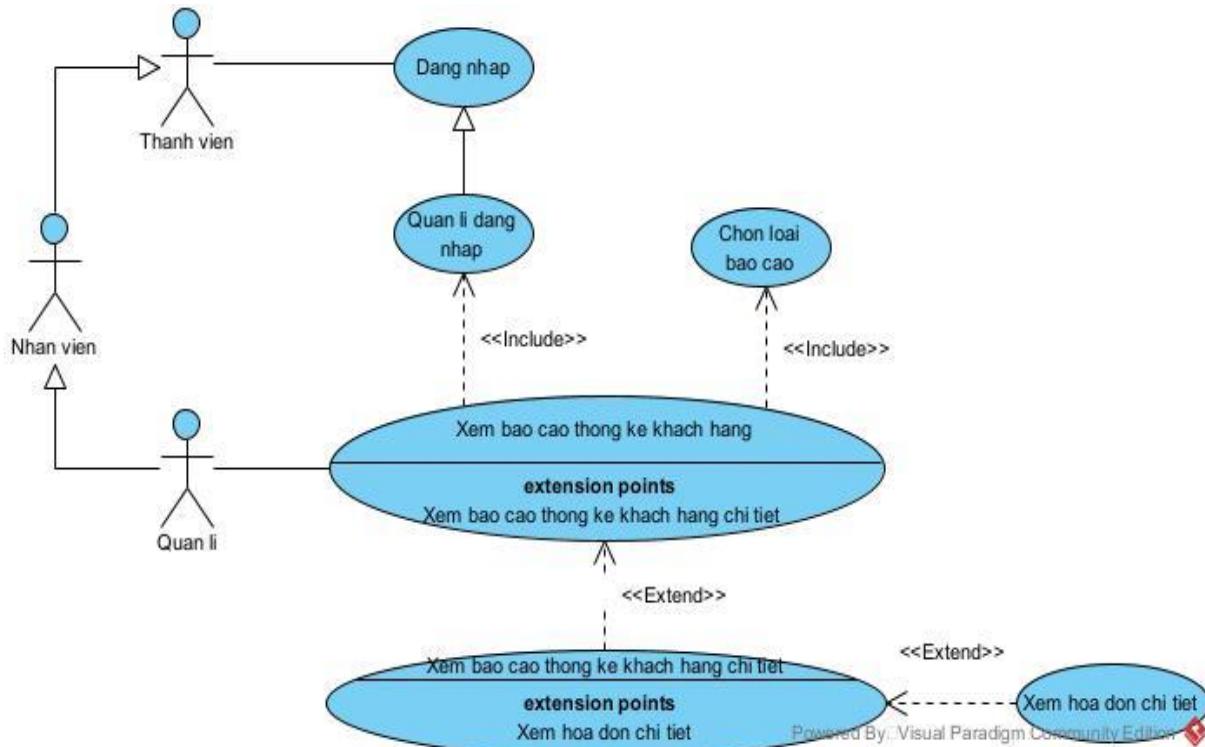
b, Thanh toán cho khách hàng tại quầy



- Actor liên quan: thu ngân, khách hàng.
- Chức năng thanh toán cho khách tại quầy có các giao diện tương tác với thu ngân:
 - Đăng nhập -> đề xuất UC đăng nhập.
 - Thanh toán -> đề xuất UC thanh toán.
 - Dùng thẻ thành viên -> đề xuất UC dùng thẻ thành viên.
 - Dùng phiếu giảm giá -> đề xuất UC dùng phiếu giảm giá.
 - Trả tiền -> đề xuất UC trả tiền.

- Thanh toán bằng tiền mặt -> đề xuất UC thanh toán bằng tiền mặt.
- Quan hệ giữa UC con với UC chính:
 - Thanh toán cho khách hàng tại quầy <<include>> đăng nhập.
 - Thanh toán cho khách hàng tại quầy <<include>> thanh toán.
 - Thanh toán cho khách hàng tại quầy <<extend>> dùng thẻ thành viên.
 - Thanh toán cho khách hàng tại quầy <<extend>> dùng phiếu giảm giá.
 - Thanh toán cho khách hàng tại quầy <<include>> trả tiền.
 - Trả tiền <<extend>> thanh toán bằng tiền mặt.
- Các UC được mô tả như sau:
 - Dùng thẻ thành viên: cho phép thu ngân nhập thẻ thành viên để khách nhận ưu đãi nhiều hơn.
 - Dùng phiếu giảm giá: cho phép thu ngân nhập phiếu giảm giá mà khách hàng sở hữu.
 - Trả tiền: cho phép thu ngân chọn phương thức thanh toán mà khách hàng muốn.
 - Thanh toán bằng tiền mặt: giúp thu ngân biết đã nhận bao nhiêu tiền của khách và phải trả lại bao nhiêu khi khách trả tiền mặt.

c, Xem thống kê khách hàng theo doanh thu



- Actor liên quan: quản lí.
- Chức năng xem thống kê khách hàng theo doanh thu có các giao diện tương tác với quản lí:
 - Đăng nhập -> đề xuất UC đăng nhập.
 - Chọn loại báo cáo -> đề xuất UC chọn loại báo cáo
 - Xem báo cáo thống kê khách hàng -> đề xuất UC xem báo cáo thống kê khách hàng.
 - Xem báo cáo thống kê khách hàng chi tiết -> đề xuất UC xem báo cáo thống kê khách hàng chi tiết.
 - Xem hóa đơn chi tiết -> đề xuất UC xem hóa đơn chi tiết.
- Quan hệ giữa UC con với UC chính:
 - Xem báo cáo thống kê khách hàng <<include>> đăng nhập.
 - Xem báo cáo thống kê khách hàng <<include>> chọn loại báo cáo.
 - Xem báo cáo thống kê khách hàng <<extend>> xem báo cáo thống kê khách hàng chi tiết.
 - Xem báo cáo thống kê khách hàng chi tiết <<extend>> xem hóa đơn chi tiết.
- Các UC được mô tả như sau:
 - Chọn loại báo cáo: cho phép quản lí chọn đối tượng cần được thống kê.
 - Xem báo cáo thống kê khách hàng chi tiết: cho phép quản lí xem chi tiết những đơn hàng (hóa đơn) mà một khách hàng đã thanh toán.
 - Xem hóa đơn chi tiết: cho phép quản lí xem chi tiết hóa đơn của một khách hàng.

II. Analysis

1. Quản lý thông tin mặt hàng

a, Scenario

Scenario	Quản lý thông tin mặt hàng																																							
Actor	Quản lý																																							
Pre-condition	Quản lý có tài khoản cấp quản lý, quản lý muốn sửa thông tin của Snack bí đỏ.																																							
Post-condition	Quản lý sửa thành công thông tin của Snack bí đỏ																																							
Main events	<p>1. Quản lý A vào hệ thống để chỉnh sửa thông tin mặt hàng.</p> <p>2. GD đăng nhập hiện ra:</p> <ul style="list-style-type: none"> • Ô nhập username • Ô nhập password • Nút login <p>3. Quản lý nhập username = ManagerA, password = MA001, click login.</p> <p>4. GD chính của quản lý hiện ra:</p> <ul style="list-style-type: none"> • Nút/link quản lý thông tin mặt hàng. • Một nút đăng xuất. <p>5. Quản lý chọn chức năng quản lý thông tin mặt hàng.</p> <p>6. GD quản lý thông tin mặt hàng hiện ra:</p> <ul style="list-style-type: none"> • Ô tìm kiếm mặt hàng, nút tìm kiếm, một nút thêm, một nút quay lại. <p>7. Quản lý nhập vào ô tìm kiếm từ khóa “Snack” + click tìm kiếm.</p> <p>8. Hệ thống hiển thị danh sách các mặt hàng có tên chứa từ “Snack” như dưới đây:</p> <table border="1"> <thead> <tr> <th>Mã</th><th>Tên</th><th>Loại</th><th>Kích thước</th><th>Mô tả</th><th>Giá</th><th>Tồn kho</th><th>Lô sản xuất</th><th>NSX</th><th>HSD</th><th>Xem</th><th>Sửa</th><th>Xóa</th></tr> </thead> <tbody> <tr> <td>SN01</td><td>Snack bí đỏ</td><td>Ăn vặt</td><td>250 gr</td><td>Ngon</td><td>5000 Vnd</td><td>50 gói</td><td>AV01</td><td>01/05/2023</td><td>01/11/2023</td><td>Xem</td><td>Sửa</td><td>Xóa</td></tr> <tr> <td>SN02</td><td>Snack ngô</td><td>Ăn vặt</td><td>250 gr</td><td>Ngon</td><td>5000 Vnd</td><td>50 gói</td><td>AV01</td><td>01/05/2023</td><td>01/11/2023</td><td>Xem</td><td>Sửa</td><td>Xóa</td></tr> </tbody> </table> <p>9. Quản lý click sửa mặt hàng Snack bí đỏ (dòng 1).</p> <p>10. GD sửa mặt hàng hiện ra:</p> <ul style="list-style-type: none"> • Mã MH: SN01 • Tên MH: Snack bí đỏ 	Mã	Tên	Loại	Kích thước	Mô tả	Giá	Tồn kho	Lô sản xuất	NSX	HSD	Xem	Sửa	Xóa	SN01	Snack bí đỏ	Ăn vặt	250 gr	Ngon	5000 Vnd	50 gói	AV01	01/05/2023	01/11/2023	Xem	Sửa	Xóa	SN02	Snack ngô	Ăn vặt	250 gr	Ngon	5000 Vnd	50 gói	AV01	01/05/2023	01/11/2023	Xem	Sửa	Xóa
Mã	Tên	Loại	Kích thước	Mô tả	Giá	Tồn kho	Lô sản xuất	NSX	HSD	Xem	Sửa	Xóa																												
SN01	Snack bí đỏ	Ăn vặt	250 gr	Ngon	5000 Vnd	50 gói	AV01	01/05/2023	01/11/2023	Xem	Sửa	Xóa																												
SN02	Snack ngô	Ăn vặt	250 gr	Ngon	5000 Vnd	50 gói	AV01	01/05/2023	01/11/2023	Xem	Sửa	Xóa																												

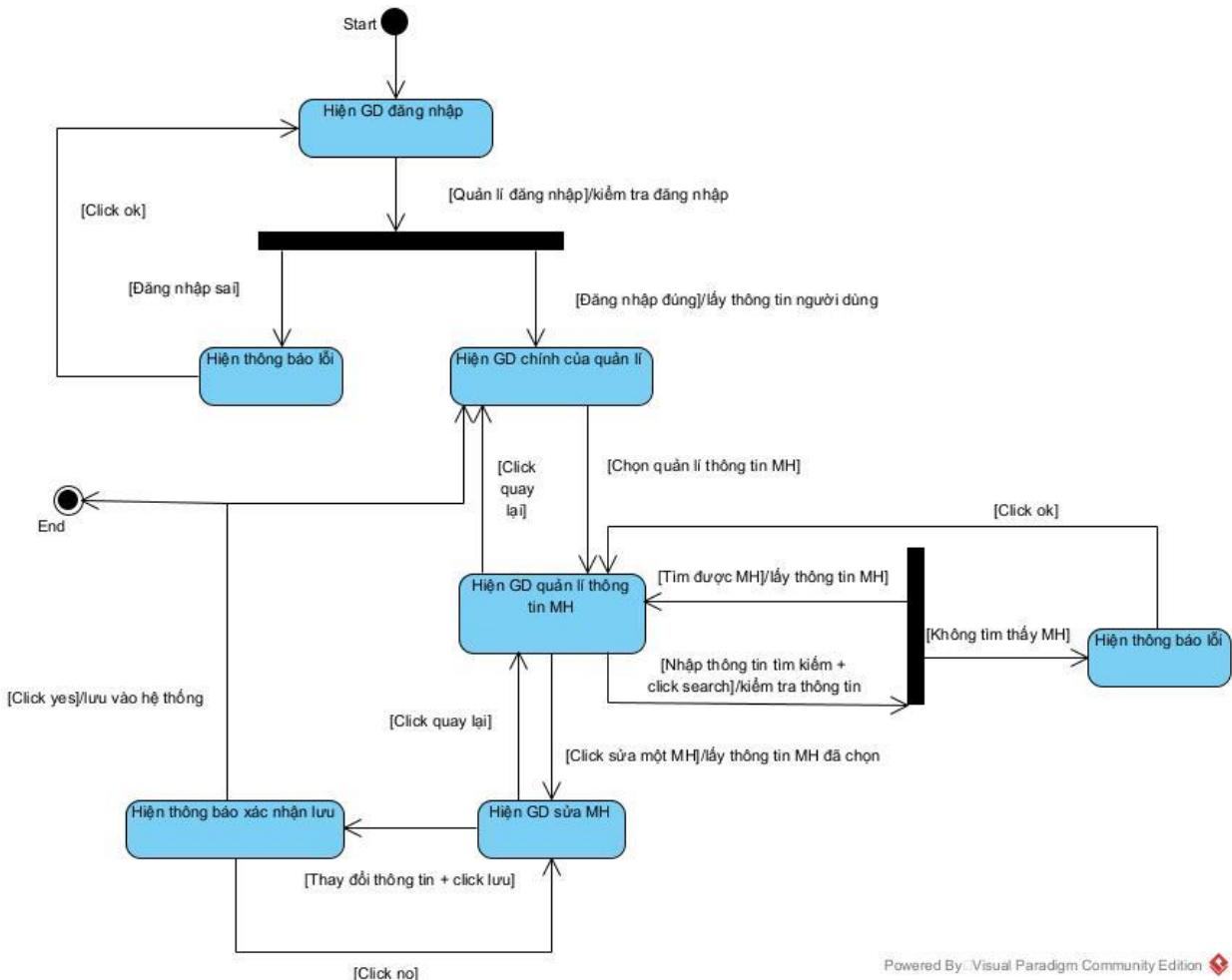
	<ul style="list-style-type: none"> • Loại MH: Ăn vặt • Kích thước: 250gr • Mô tả: Ngon • Giá bán: 5000 Vnd • Số lượng tồn: 50 gói • Lô sản xuất: AV01 • Ngày sản xuất: 01/05/2023 • Hạn sử dụng: 01/11/2023 • Nút/link để lưu/quay lại. <p>11. Quản lý sửa giá bán thành 10000 Vnd + click lưu.</p> <p>12. Hệ thống hiển thị thông báo xác nhận lưu hay không.</p> <p>13. Quản lý click xác nhận.</p> <p>14. Hệ thống lưu vào DB và quay về GD chính của quản lý (bước 4).</p>
Exception	<p>3. Quản lý nhập sai thông tin đăng nhập + click login.</p> <p>3.1. Hệ thống hiển thị thông báo “Sai thông tin đăng nhập”.</p> <p>3.2. Quản lý click ok của thông báo.</p> <p>3.3. Hệ thống quay lại GD đăng nhập (bước 2).</p> <p>7. Quản lý nhập vào ô tìm kiếm từ khóa “Chào tân” + click tìm kiếm.</p> <p>7.1. Hệ thống hiển thị thông báo “Mặt hàng không tồn tại”.</p> <p>7.2. Quản lý click ok của thông báo.</p> <p>7.3. Hệ thống vẫn ở nguyên GD quản lý thông tin mặt hàng (bước 6).</p>

b, Biểu đồ trạng thái

- Mô tả hoạt động:

- Từ giao diện đăng nhập, nếu đăng nhập sai thì hiện thông báo lỗi, nếu đăng nhập đúng thì chuyển sang giao diện chính của quản lý.
- Tại giao diện chính của quản lý, khi chọn quản lý thông tin, chuyển sang giao diện quản lý thông tin MH.
- Tại giao diện quản lý thông tin MH, khi nhập từ khóa + click tìm kiếm, nếu có mặt hàng chứa từ khóa thì cập nhật lại giao diện quản lý thông tin MH, nếu không có mặt hàng chứa từ khóa thì hiện thông báo lỗi.
- Từ giao diện quản lý thông tin MH, khi click sửa một MH, hệ thống chuyển sang giao diện sửa MH.
- Tại giao diện sửa MH, khi sửa thông tin + click lưu, hệ thống chuyển sang giao diện xác nhận lưu.
- Tại giao diện xác nhận lưu, nếu click no thì quay lại giao diện sửa MH, nếu click yes thì hệ thống lưu và kết thúc.

- Tại tất cả các giao diện, khi click quay lại đều quay lại giao diện trước đây.



c, Biểu đồ lớp thực thể

Bước 1: Thay thế bằng Scenario ở phần a.

Bước 2+3: Trích danh từ và đánh giá

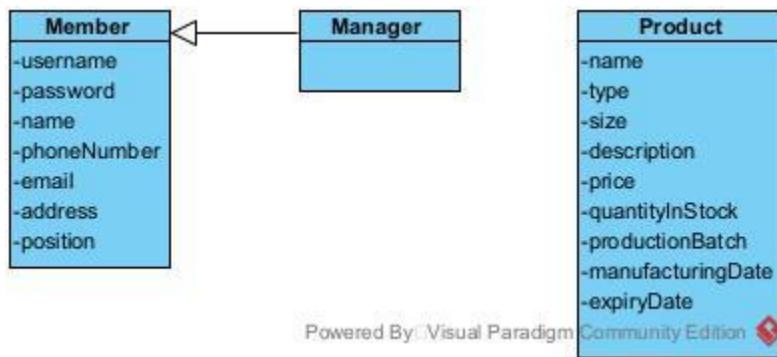
- Hệ thống (loại bỏ)
- Thông tin (loại bỏ)
- Thành viên (làm lớp thực thể) => class Member
- Siêu thị (loại bỏ)
- Mặt hàng (làm lớp thực thể) => class Product
- Quản lí (làm lớp thực thể) => class Manager
- Username (làm thuộc tính)
- Password (làm thuộc tính)
- Giao diện (loại bỏ)

- Mã MH, tên MH, loại MH, kích thước, giá bán, lô sản xuất, ngày sản xuất (làm thuộc tính)
- Thông báo (loại bỏ)
- Từ khóa (loại bỏ)

Bước 4: Xét quan hệ số lượng giữa các lớp

Bước 5: Bổ sung quan hệ đối tượng

- Quản lí kê thừa thành viên.



d, Biểu đồ lớp đầy đủ

- Quản lí đăng nhập vào hệ thống -> Giao diện đăng nhập hiện ra -> đề xuất lớp LoginView:

- Input cho username -> inpUsername
- Input cho password -> inpPassword
- Nút submit để đăng nhập -> subLogin

- Quản lí nhập username/password và click login -> hệ thống phải kiểm tra thông tin đăng nhập đúng không -> cần phương thức:

- Name: checkLogin()
- Input: username, password (Member)
- Output: Member
- Owner: Member

- Khi đăng nhập thành công -> hiển thị giao diện chính của quản lí -> đề xuất lớp ManagerHomeView:

- Một nút quản lí thông tin mặt hàng -> subManageProduct
- Một nút xem báo cáo thống kê -> subSeeStatistical

- Một nút đăng xuất -> subSignOut

- Quản lý chọn nút quản lý thông tin mặt hàng -> hiển thị giao diện quản lý thông tin mặt hàng -> để xuất lớp ManageProductView:

- Input cho tên mặt hàng -> inpSearchProductName
- Một nút để tìm mặt hàng -> subSearch
- Một nút để thêm mặt hàng -> subAddProduct
- Một nút để quay lại -> subBack
- Một danh sách mặt hàng -> outsubListProduct

- Khi nhân viên nhập từ khóa và search -> hệ thống cần kiểm tra tất cả các mặt hàng có chứa từ khóa -> cần phương thức:

- Name: searchProduct()
- Input: từ khóa
- Output: List<Product>
- Owner: Product

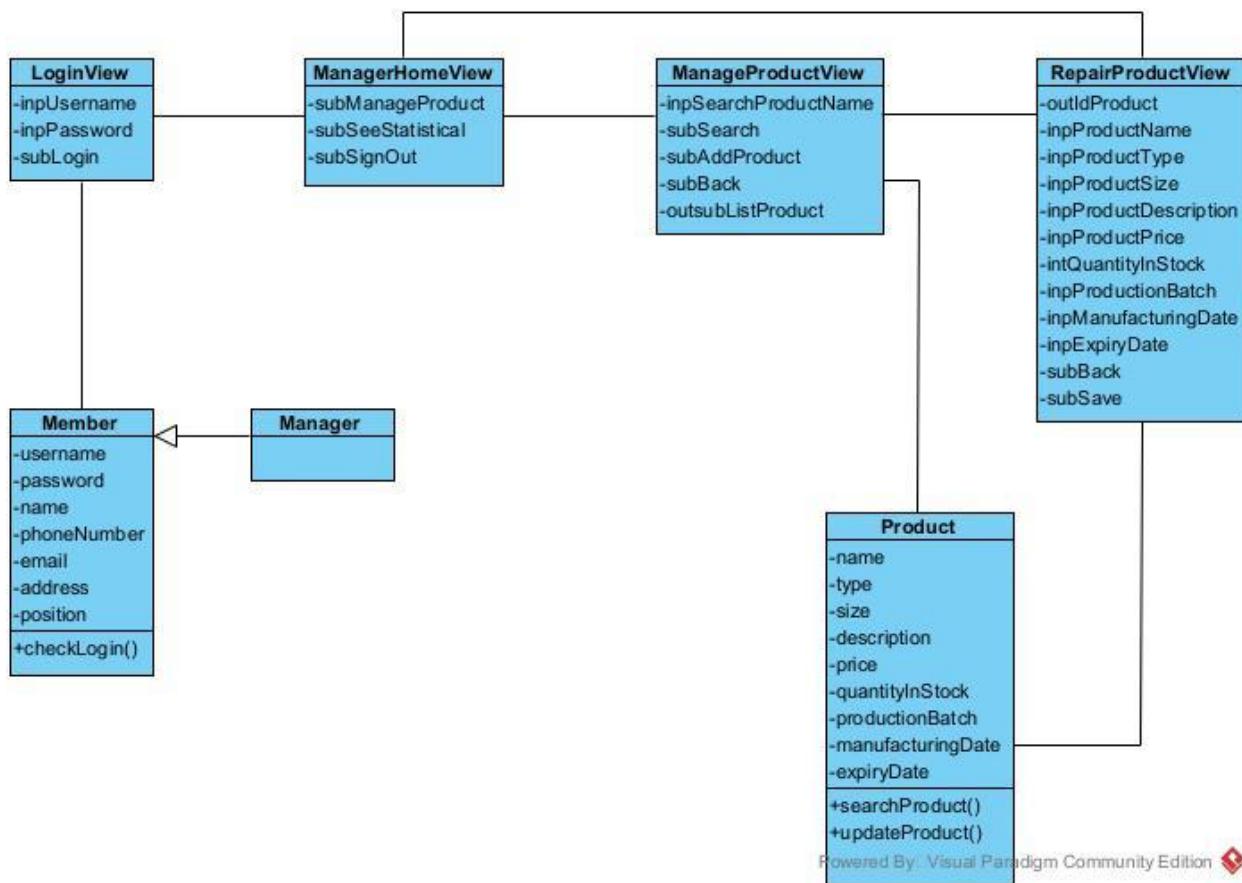
- Kết quả được trả về và hiển thị trên ManageProductView.

- Quản lý click nút sửa của mặt hàng cần sửa thông tin -> hiển thị giao diện sửa mặt hàng -> để xuất lớp RepairProductView:

- Hiển thị mã mặt hàng -> outIdProduct
- Input cho tên mặt hàng -> inpProductName
- Input cho loại mặt hàng -> inpProductType
- Input cho kích thước -> inpProductSize
- Input cho mô tả -> inpProductDescription
- Input cho giá cả -> inpProductPrice
- Input cho số lượng tồn -> inpQuanlityInStock
- Input cho lô sản xuất -> inpProductionBatch
- Input cho ngày sản xuất -> inpManufacturingDate
- Input cho hạn sử dụng -> inpExpiryDate
- Nút để quay lại -> subBack
- Nút để lưu -> subSave

- Quản lý sửa thông tin cần sửa và click lưu -> hệ thống cập nhật thông tin mặt hàng -> cần phương thức:

- Name: updateProduct()
 - Input: một mặt hàng (Product)
 - Output: Boolean
 - Owner: Product
- Sau khi lưu vào DB, hệ thống quay về giao diện chính của quản lí (ManagerHomeView)

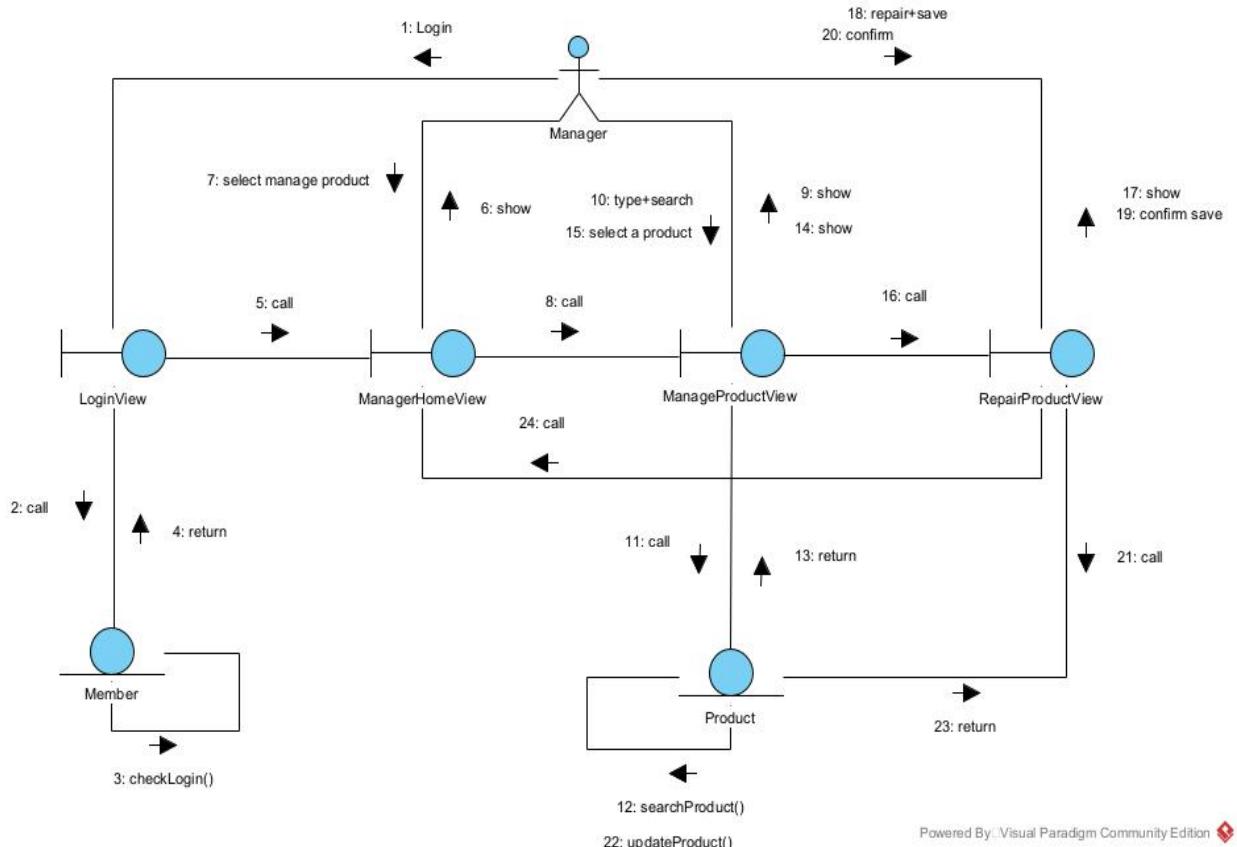


e, Biểu đồ cộng tác

- Kịch bản v.2:

1. Tại lớp LoginView, quản lí đăng nhập.
2. Lớp LoginView gọi lớp Member để kiểm tra.
3. Lớp Member kiểm tra thông tin tài khoản.
4. Lớp Member trả kết quả cho lớp LoginView.
5. Lớp LoginView gọi lớp ManagerHomeView.
6. Lớp ManagerHomeView hiển thị cho quản lí.
7. Quản lí chọn chức năng quản lí thông tin mặt hàng.

8. Lớp ManagerHomeView gọi lớp ManageProductView.
9. Lớp ManageProductView hiển thị cho quản lí.
10. Quản lí nhập từ khóa và tìm kiếm.
11. Lớp ManageProductView gọi lớp Product để lấy thông tin mặt hàng.
12. Lớp Product tìm kiếm và lấy thông tin mặt hàng.
13. Lớp Product trả kết quả cho lớp ManageProductView.
14. Lớp ManageProductView hiển thị cho quản lí
15. Quản lí chọn 1 mặt hàng để sửa.
16. Lớp ManageProductView gọi lớp RepairProductView.
17. Lớp RepairProductView hiển thị cho quản lí.
18. Quản lí sửa thông tin và lưu.
19. Lớp RepairProductView hiển thị thông báo xác nhận lưu cho quản lí.
20. Quản lí chọn xác nhận.
21. Lớp RepairProductView gọi lớp Product để cập nhật thông tin.
22. Lớp Product cập nhật thông tin mặt hàng
23. Lớp Product trả kết quả cho lớp RepairProductView.
24. Lớp RepairProductView gọi lớp ManagerHomeView.



2. Thanh toán cho khách hàng tại quầy

a, Scenario

Scenario	Thanh toán cho khách hàng tại quầy
Actor	Thu ngân, khách hàng
Pre-condition	Thu ngân có tài khoản cho thu ngân. Khách hàng có thẻ thành viên, có phiếu giảm giá và muốn thanh toán 2 mặt hàng Snack bí đỏ và Lay's vị bò bắp tiền mặt
Post-condition	Thanh toán thành công cho khách hàng
Main events	<p>1. Thu ngân B vào hệ thống để thanh toán cho khách hàng.</p> <p>2. GD đăng nhập hiện ra:</p> <ul style="list-style-type: none"> • Ô nhập username • Ô nhập password • Nút login <p>3. Thu ngân nhập username = CashierB, password = CB001, click login.</p> <p>4. GD chính của thu ngân hiện ra:</p> <ul style="list-style-type: none"> • Nút/link thanh toán. • Một nút đăng xuất. <p>5. Thu ngân chọn chức năng thanh toán.</p> <p>6. GD thanh toán hiện ra:</p> <ul style="list-style-type: none"> • Thông tin hóa đơn đang tạo: <ul style="list-style-type: none"> ○ Mã HD: HD03 ○ Tên siêu thị: Hà Đông ○ Địa chỉ siêu thị: Hà Đông, Hà Nội ○ Mã thu ngân: TN01 ○ Tên thu ngân: Luân ○ Thời gian thanh toán: 21/08/2023 • Một nút/link dùng thẻ thành viên. • Một nút/link dùng phiếu giảm giá. • Một ô hiển thị mã thẻ thành viên (mã khách hàng). • Một ô hiển thị mã phiếu giảm giá. • Bảng để điền thông tin cho mặt hàng tiếp theo: <ul style="list-style-type: none"> ○ Thông tin mặt hàng: mã MH, tên MH, kích thước, hạn sử dụng, số lượng, đơn giá, thành tiền. ○ Nút/link để lưu/xóa. • Bảng để hiển thị danh sách các mặt hàng khách hàng mua gồm các thông tin: mã MH, tên MH, đơn giá, số lượng, số tiền.

- Một dòng hiển thị số mặt hàng, tổng tiền trước khi giảm giá.
- Một dòng hiển thị số tiền giảm giá.
- Một dòng hiển thị số tiền phải thanh toán.
- Một dòng để nhập số tiền khách trả.
- Một dòng để hiển thị số tiền trả khách.
- Một nút/link thanh toán.
- Một nút/link hủy.

7. Thu ngân nhập mã MH = "SN01" và số lượng = 10 + click lưu.

8. Hệ thống giữ nguyên giao diện thanh toán với các thông tin như ở bước 6 ngoại trừ việc thêm MH vào danh sách mặt hàng, cập nhật số lượng MH, tổng tiền, số tiền phải thanh toán.

STT	Mã MH	Tên MH	Đơn giá	Số lượng	Thành tiền
1	SN01	Snack bí đỏ	10000 Vnd	10	100000 Vnd

- Số lượng MH: 10
- Tổng tiền: 100000 Vnd
- Số tiền phải thanh toán: 100000 Vnd

9. Thu ngân xóa mã MH cũ ở bảng thông tin mặt hàng và nhập mã MH tiếp theo = "LS01", số lượng = 4 + click lưu.

10. Hệ thống giữ nguyên giao diện thanh toán với các thông tin như ở bước 8 ngoại trừ việc thêm MH vào danh sách mặt hàng, cập nhật số lượng MH, tổng tiền, số tiền phải thanh toán.

STT	Mã MH	Tên MH	Đơn giá	Số lượng	Thành tiền
1	SN01	Snack bí đỏ	10000 Vnd	10	100000 Vnd
2	LS01	Lay's vị bò	20000 Vnd	4	80000 Vnd

- Số lượng MH: 14
- Tổng tiền: 180000 Vnd
- Số tiền phải thanh toán: 180000 Vnd

11. Sau khi nhập xong hết mặt hàng, thu ngân click dùng thẻ thành viên.

12. GD dùng thẻ thành viên hiện ra:

- Một ô nhập mã thẻ
- Một nút/link kiểm tra
- Thông tin khách hàng:

	<ul style="list-style-type: none"> <input type="radio"/> Mã KH <input type="radio"/> Tên KH <input type="radio"/> Số điện thoại <input type="radio"/> Email <input type="radio"/> Địa chỉ <ul style="list-style-type: none"> • Một nút/link xác nhận/quay lại. <p>13. Thu ngân nhập mã thẻ = “KH04” + click kiểm tra.</p> <p>14. Hệ thống cập nhật thông tin trên GD dùng thẻ thành viên (bước 12):</p> <ul style="list-style-type: none"> • Mã KH: KH04 • Tên KH: Kiều • Số điện thoại: 0123456789 • Email: k@gmail.com • Địa chỉ: ABC <p>15. Thu ngân click xác nhận của GD dùng thẻ thành viên.</p> <p>16. Hệ thống quay lại GD thanh toán với thông tin như bước 10 ngoại trừ việc cập nhật mã thẻ thành viên:</p> <ul style="list-style-type: none"> • Mã thẻ thành viên: KH04 <p>17. Thu ngân thông báo số tiền phải trả cho khách</p> <p>18. Khách hàng trả tiền.</p> <p>19. Thu ngân nhập tiền khách đưa = 200000 Vnd.</p> <p>20. Hệ thống giữ nguyên GD thanh toán ngoại trừ việc cập nhật:</p> <ul style="list-style-type: none"> • Tiền khách đưa: 200000 Vnd • Tiền thối lại: 40000 Vnd <p>21. Thu ngân click thanh toán của GD thanh toán.</p> <p>22. Hệ thống hiển thị thông báo xác nhận thanh toán.</p> <p>23. Thu ngân click xác nhận của thông báo.</p> <p>24. Hệ thống lưu hóa đơn vào DB và hiển thị thông báo thanh toán thành công.</p> <p>25. Thu ngân trả tiền thừa + click ok của thông báo.</p> <p>26. Hệ thống quay về GD chính của thu ngân (bước 4).</p>
Exception	<p>3. Thu ngân nhập sai thông tin đăng nhập + click login.</p> <p>3.1. Hệ thống thông báo “Sai thông tin đăng nhập”.</p> <p>3.2. Thu ngân click ok của thông báo.</p>

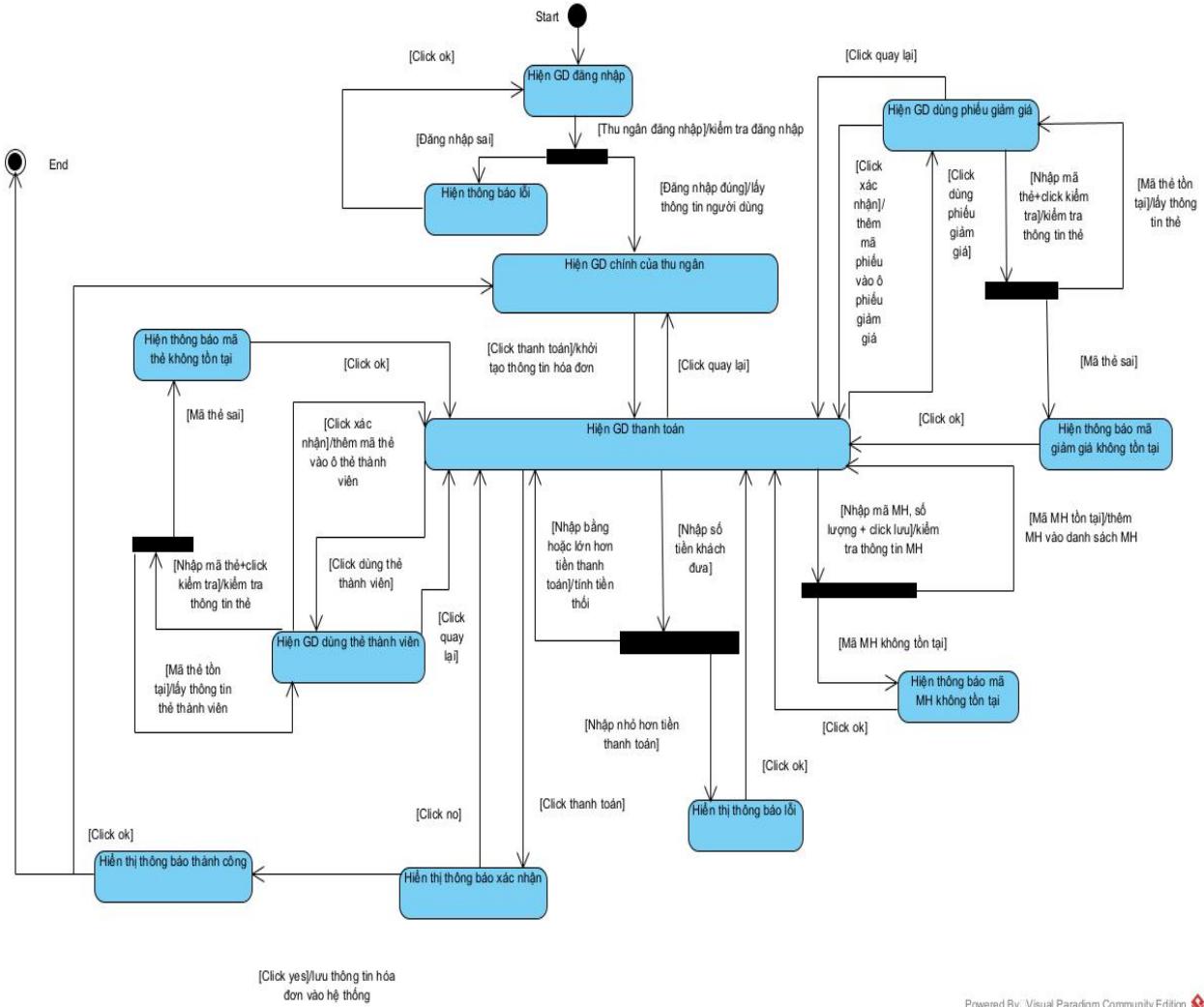
- 3.3. Hệ thống quay lại GD đăng nhập (bước 2).
7. Thu ngân nhập mã MH = “NotExist”, số lượng = 5 + click lưu.
- 7.1. Hệ thống thông báo “Mặt hàng chưa tồn tại”.
- 7.2. Thu ngân click ok của thông báo.
- 7.3. Hệ thống quay lại GD thanh toán với các thông tin cũ như bước 6.
9. Thu ngân nhập mã MH = “NotExist”, số lượng = 5 + click lưu.
- 9.1. Hệ thống thông báo “Mặt hàng chưa tồn tại”.
- 9.2. Thu ngân click ok của thông báo.
- 9.3. Hệ thống quay lại GD thanh toán với thông tin như bước 8.
13. Thu ngân nhập mã thẻ = “NotExist” + click kiểm tra.
- 13.1. Hệ thống thông báo “Mã thẻ không tồn tại hoặc bị khóa”.
- 13.2. Thu ngân click ok của thông báo.
- 13.3. Hệ thống quay lại GD thanh toán với các thông tin như bước 10.
17. Nếu KH có phiếu giảm giá, thu ngân click dùng phiếu giảm giá.
- 17.1. GD dùng phiếu giảm giá hiện ra:
- Một ô nhập mã phiếu giảm giá
 - Nút/link kiểm tra
 - Thông tin phiếu giảm giá:
 - Mã phiếu:
 - Giá trị:
 - Ngày phát hành:
 - Ngày hết hạn:
 - Nút/link xác nhận/quay lại.
- 17.2. Thu ngân nhập mã phiếu giảm giá = “NotExist” + click kiểm tra.
- 17.3. Hệ thống thông báo “Mã giảm giá không tồn tại hoặc đã hết hạn”.
- 17.4. Thu ngân click ok của thông báo.
- 17.5. Hệ thống quay lại GD thanh toán với các thông tin như bước 16.

	<p>17.6. Thu ngân click dùng phiếu giảm giá</p> <p>17.7. Thu ngân nhập mã phiếu giảm giá = “AV20” + click kiểm tra.</p> <p>17.8. Hệ thống cập nhật thông tin phiếu giảm giá trên GD dùng phiếu giảm giá (bước 18):</p> <ul style="list-style-type: none"> • Mã phiếu: AV20 • Giá trị: 20000 Vnd • Ngày phát hành: 01/01/2023 • Ngày hết hạn: 31/12/2024 <p>17.9. Thu ngân click xác nhận của GD dùng phiếu giảm giá.</p> <p>17.10. Hệ thống quay lại GD thanh toán với thông tin như bước 16 ngoại trừ việc cập nhật thông tin mã phiếu giảm giá, số tiền giảm, số tiền phải thanh toán:</p> <ul style="list-style-type: none"> • Mã phiếu giảm giá: AV20 • Số tiền giảm: 20000 Vnd • Số tiền phải thanh toán: 160000 Vnd <p>19. Thu ngân nhập tiền khách đưa = 20000 Vnd.</p> <p>19.1. Hệ thống thông báo “Số tiền vừa nhập không đủ thanh toán”.</p> <p>19.2. Thu ngân click ok của thông báo.</p> <p>19.3. Hệ thống quay lại GD thanh toán với thông tin như bước 17.</p>
--	---

b, Biểu đồ trạng thái

- Từ giao diện đăng nhập, nếu đăng nhập sai thì hiện thông báo lỗi, nếu đăng nhập đúng thì chuyển sang giao diện chính của thu ngân.
- Tại giao diện chính của thu ngân, click thanh toán, chuyển sang giao diện thanh toán.
- Tại giao diện thanh toán, nhập mã MH, số lượng + click lưu, nếu mã MH không tồn tại thì hiện thông báo lỗi, nếu mã MH tồn tại thì lấy thông tin và cập nhật lại giao diện thanh toán.
- Tại giao diện thanh toán, chọn dùng thẻ thành viên, chuyển sang giao diện dùng thẻ thành viên.
- Tại giao diện dùng thẻ thành viên, nhập mã thẻ + click kiểm tra, nếu mã thẻ sai thì hiển thị thông báo lỗi, nếu mã thẻ đúng thì lấy thông tin và cập nhật lại giao diện dùng thẻ thành viên, sau đó click xác nhận thì chuyển sang giao diện thanh toán.

- Tại giao diện thanh toán, chọn dùng phiếu giảm giá, chuyển sang giao diện dùng phiếu giảm giá.
- Tại giao diện dùng phiếu giảm giá, nhập mã phiếu + click kiểm tra, nếu mã phiếu sai thì hiển thị thông báo lỗi, nếu mã phiếu đúng thì lấy thông tin và cập nhật lại giao diện dùng phiếu giảm giá, sau đó click xác nhận thì chuyển sang giao diện thanh toán.
- Tại giao diện thanh toán, nếu nhập nhỏ hơn số tiền cần thanh toán thì hiện thông báo lỗi, nếu nhập bằng hoặc lớn hơn số tiền thanh toán thì tính tiền thối và cập nhật lại giao diện thanh toán.
- Tại giao diện thanh toán, khi chọn thanh toán, hệ thống chuyển sang thông báo xác nhận.
- Tại giao diện thông báo xác nhận, nếu chọn no thì quay lại giao diện thanh toán bằng. Nếu chọn yes thì hệ thống lưu hóa đơn, sau đó quay lại giao diện chính của thu ngân và kết thúc.
- Tại tất cả các giao diện, nếu chọn quay lại đều quay về giao diện trước đây.



c, Biểu đồ lớp thực thể

Bước 1: Thay thế bằng Scenario ở phần a

Bước 2+3: Trích danh từ và đánh giá

- Khách hàng (làm lớp thực thể) => class Customer
- Thu ngân (làm lớp thực thể) => class Cashier
- Thành viên (làm lớp thực thể) => class Member
- Quầy (loại bỏ)
- Thẻ thành viên (làm lớp thực thể) => class MemberCard
- Phiếu giảm giá (làm lớp thực thể) => class Coupon
- Mặt hàng (làm lớp thực thể) => class Product
- Hệ thống (loại bỏ)
- Giao diện (loại bỏ)

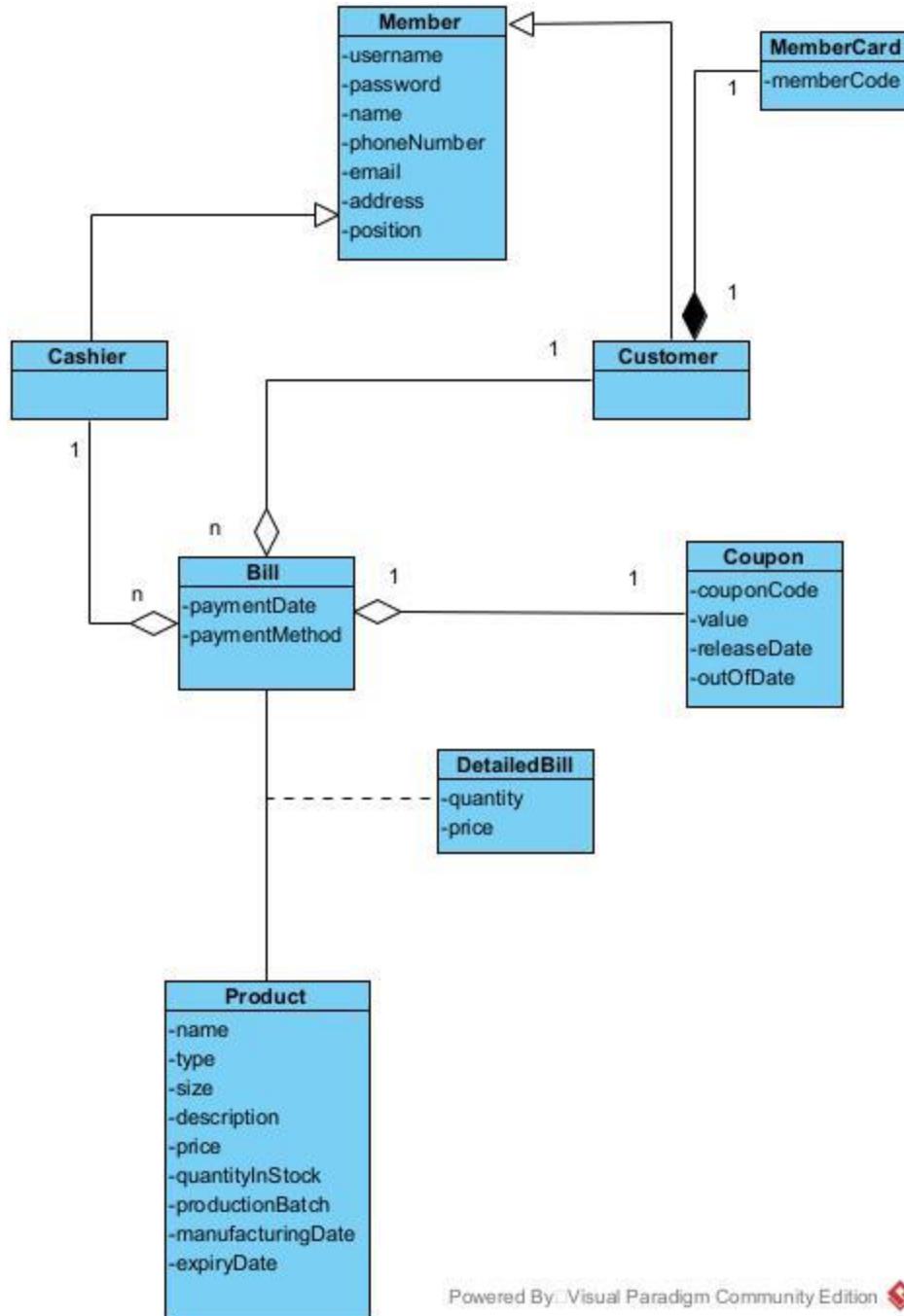
- Username (làm thuộc tính)
- Password (làm thuộc tính)
- Siêu thị (loại bỏ)
- Mã MH, tên MH, đơn giá (làm thuộc tính)
- Mã KH, tên KH, số điện thoại, email, địa chỉ (làm thuộc tính)
- Mã phiếu, giá trị, ngày phát hành, ngày nhận, ngày hết hạn (làm thuộc tính)
- Tiền mặt (loại bỏ)
- Hóa đơn (làm lớp thực thể) => class Bill

Bước 4: Xét quan hệ số lượng giữa các lớp

- Một thu ngân thanh toán nhiều hóa đơn => quan hệ Cashier và Bill là 1-n.
- Một khách hàng mua nhiều hóa đơn => quan hệ Customer và Bill là 1-n.
- Một khách hàng có một thẻ thành viên => quan hệ Customer và MemberCard là 1-1.
- Một hóa đơn có một phiếu giảm giá, một phiếu giảm giá dùng cho nhiều hóa đơn => quan hệ Bill và Coupon là 1-1.
- Một hóa đơn có nhiều mặt hàng, một mặt hàng được bán trong nhiều hóa đơn. Đề xuất thêm lớp DetailedBill xác định duy nhất một mặt hàng trong một hóa đơn.

Bước 5: Bổ sung quan hệ đối tượng

- Thu ngân kế thừa thành viên.
- Khách hàng kế thừa thành viên.
- Thu ngân là thành phần của hóa đơn.
- Khách hàng là thành phần của hóa đơn.
- Khách hàng là thành phần của thẻ thành viên.
- Phiếu giảm giá là thành phần của hóa đơn.
- Mặt hàng và hóa đơn liên kết tạo ra hóa đơn chi tiết duy nhất.



d, Biểu đồ lớp đầy đủ

- Thu ngân vào hệ thống -> hiển thị giao diện đăng nhập -> để xuất lớp LoginView:

- Input cho username: inpUsername
- Input cho password: inpPassword
- Nút submit để đăng nhập: subLogin

- Thu ngân nhập username và password + click đăng nhập -> hệ thống cần kiểm tra thông tin tài khoản -> cần phương thức:

- Name: checkLogin()
- Input: username, password (Member)
- Output: Member
- Owner: Member

- Khi đăng nhập thành công -> hiển thị giao diện chính của thu ngân -> để xuất lớp CashierHomeView:

- Một nút/link thanh toán -> subPayment
- Một nút đăng xuất -> subSignOut

- Thu ngân click thanh toán -> hiển thị giao diện thanh toán -> để xuất lớp PaymentView:

- Hiển thị mã HD -> outIdBill
- Hiển thị tên siêu thị -> outSupermarketName
- Hiển thị địa chỉ siêu thị -> outSupermarketAddress
- Hiển thị mã TN -> outIdCashier
- Hiển thị tên TN -> outCashierName
- Hiển thị thời gian thanh toán -> outPaymentDate
- Hiển thị mã thẻ thành viên -> outMemberCard
- Hiển thị phiếu giảm giá -> outCoupon
- Nút dùng thẻ thành viên -> subUseMemberCard
- Nút dùng phiếu giảm giá -> subUseCoupon
- Bảng để điền thông tin cho mặt hàng tiếp theo
 - Input cho mã MH -> inpIdProduct
 - Output cho tên MH -> outProductName
 - Output cho kích thước MH -> outProductSize
 - Output cho hạn sử dụng -> outProductExpiryDate
 - Input cho số lượng -> inpProductQuantity
 - Hiển thị giá bán -> outProductPrice
 - Hiển thị tổng tiền của mặt hàng -> outAmountMoney
 - Nút/link để lưu mặt hàng khỏi bảng danh sách muốn mua -> subSave
 - Nút/link để xóa mặt hàng khỏi bảng danh sách muốn mua -> subDelete
- Bảng để hiển thị danh sách mặt hàng khách mua -> outsubProductOfBill

- Tổng tiền trước giảm giá -> outTotalBeforeDiscount
- Số lượng mặt hàng -> outNumberOfProducts
- Số tiền được giảm -> outReducesAmount
- Số tiền phải thanh toán -> outAmountToBePaid
- Số tiền khách trả -> inpCustomerPaid
- Số tiền trả khách -> outReturnedCustomer
- Nút/link thanh toán bằng tiền mặt -> subPaymentCash
- Nút/link hủy thanh toán -> subCancel

- Thu ngân nhập mã MH và số lượng + click lưu -> hệ thống cần kiểm tra mặt hàng đã tồn tại trong hệ thống chưa -> cần phương thức:

- Name: checkProductById()
- Input: id mặt hàng (Product)
- Output: Product
- Owner: Product

- Kết quả được trả về trên PaymentView, nhân viên click lưu -> hệ thống cập nhật lại PaymentView với thông tin cũ ngoại trừ mặt hàng vừa thêm vào danh sách muốn mua.

- Thu ngân click dùng thẻ thành viên -> hệ thống hiển thị giao diện dùng thẻ thành viên -> để xuất lớp MemberCardView:

- Input cho mã thẻ thành viên -> inpIdMemberCard
- Một nút kiểm tra -> subCheck
- Mã KH -> outIdCustomer
- Tên KH -> outCustomerName
- Số điện thoại KH -> outCustomerPhone
- Email KH -> outCustomerEmail
- Địa chỉ KH -> outCustomerAddress
- Nút để quay lại -> subBack
- Nút để xác nhận -> subConfirm

- Thu ngân nhập mã thẻ + click kiểm tra -> hệ thống cần kiểm tra thông tin thẻ thành viên đã tồn tại chưa -> cần phương thức:

- Name: checkMemberCard()
- Input: mã thẻ thành viên (MemberCard)

- Output: MemberCard
- Owner: MemberCard

- Kết quả được trả về trên MemberCardView, nhân viên click xác nhận -> hệ thống quay lại giao diện PaymentView.

- Thu ngân click dùng phiếu giảm giá -> hiển thị giao diện dùng phiếu giảm giá -> để xuất lớp CouponView:

- Input cho mã phiếu giảm giá -> inpIdCoupon
- Nút kiểm tra mã phiếu -> subCheck
- Mã phiếu -> outIdCoupon
- Giá trị -> outValue
- Ngày phát hành -> outReleaseDate
- Ngày nhận -> outReceiveDate
- Ngày hết hạn -> outOutOfDate
- Nút quay lại -> subBack
- Nút xác nhận -> subConfirm

- Thu ngân nhập mã phiếu giảm giá + click kiểm tra -> hệ thống cần kiểm tra thông tin phiếu giảm giá đã tồn tại chưa -> cần phương thức:

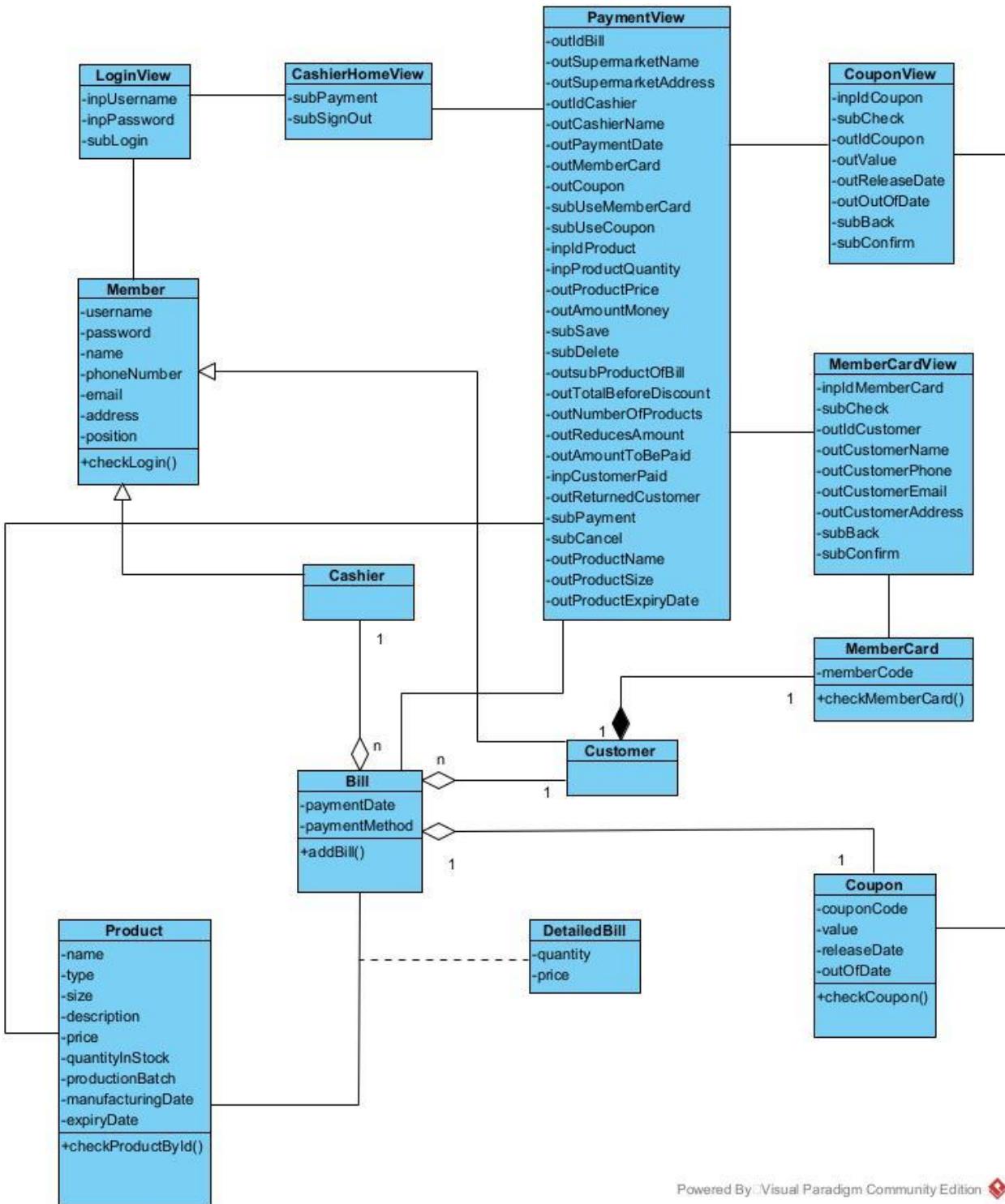
- Name: checkCoupon()
- Input: mã của phiếu giảm giá (Coupon)
- Output: Coupon
- Owner: Coupon

- Kết quả được trả về trên CouponView, nhân viên click xác nhận -> hệ thống quay lại giao diện PaymentView.

- Thu ngân nhập tiền khách đưa + click thanh toán -> hệ thống cần lưu hóa đơn vào DB -> cần phương thức:

- Name: addBill()
- Input: một hóa đơn (Bill)
- Output: Boolean
- Owner: Bill

- Sau khi lưu vào DB, hệ thống quay về giao diện chính của thu ngân (CashierHomeView).



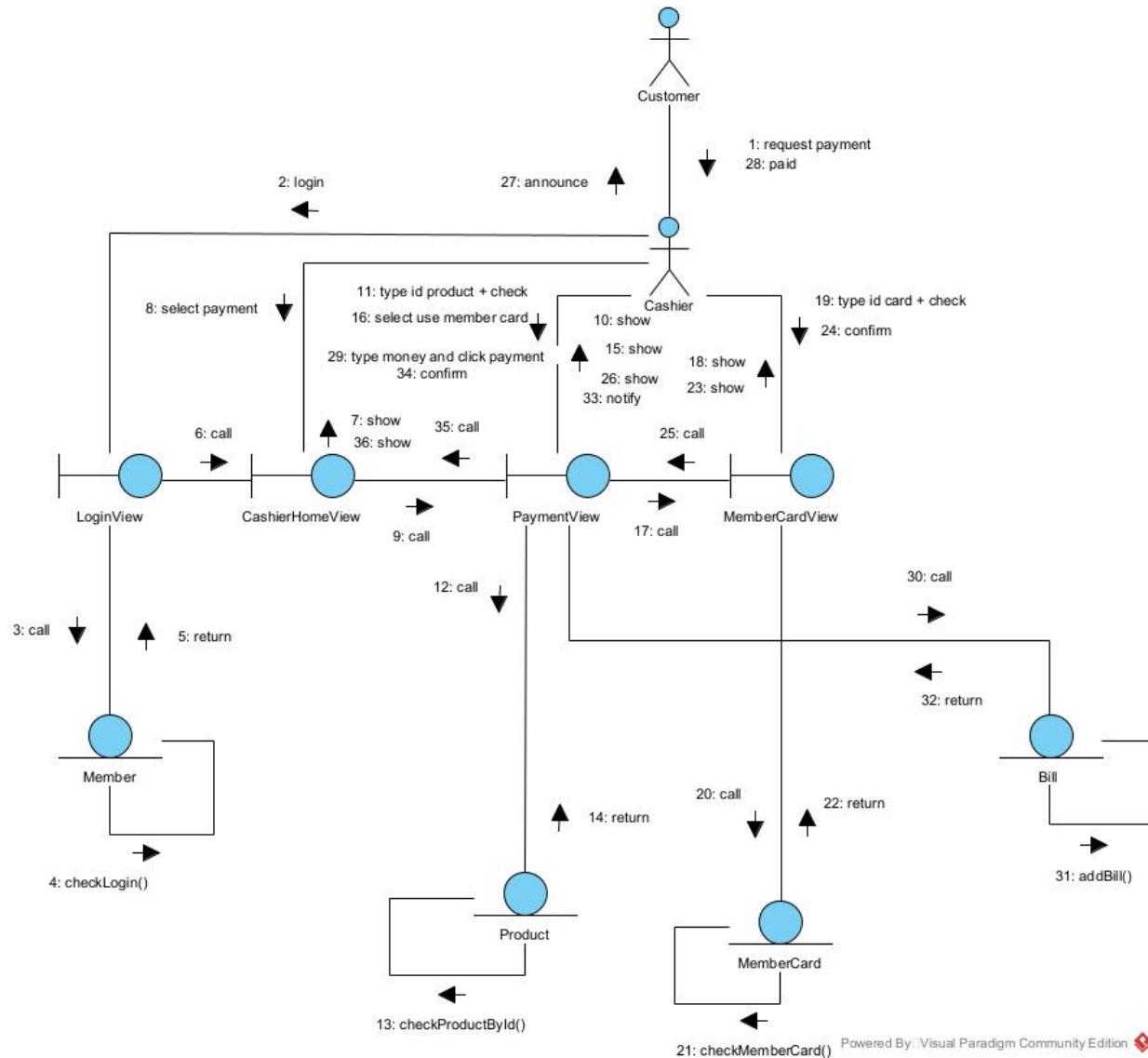
Powered By: Visual Paradigm Community Edition

e, Biểu đồ cộng tác

- Kịch bản v.2:

1. Khách hàng yêu cầu thanh toán.

2. Tại lớp LoginView, thu ngân đăng nhập.
3. Lớp LoginView gọi lớp Member để kiểm tra.
4. Lớp Member kiểm tra thông tin đăng nhập.
5. Lớp Member trả kết quả cho lớp LoginView.
6. Lớp LoginView gọi lớp CashierHomeView.
7. Lớp CashierHomeView hiển thị cho thu ngân.
8. Thu ngân chọn thanh toán.
9. Lớp CashierHomeView gọi lớp PaymentView.
10. Lớp PaymentView hiển thị cho thu ngân.
11. Thu ngân nhập mã mặt hàng, số lượng và lưu.
12. Lớp PaymentView gọi lớp Product để kiểm tra.
13. Lớp Product kiểm tra thông tin mặt hàng.
14. Lớp Product trả về cho lớp PaymentView.
15. Lớp PaymentView hiển thị cho thu ngân. (lặp từ bước 11 đến 15 cho đến khi hết mặt hàng cần thanh toán)
16. Thu ngân chọn dùng thẻ thành viên.
17. Lớp PaymentView gọi lớp MemberCardView.
18. Lớp MemberCardView hiển thị cho thu ngân.
19. Thu ngân nhập mã thẻ và chọn kiểm tra.
20. Lớp MemberCardView gọi lớp MemberCard.
21. Lớp MemberCard kiểm tra thông tin thẻ.
22. Lớp MemberCard trả kết quả cho lớp MemberCardView.
23. Lớp MemberCardView hiển thị cho thu ngân.
24. Thu ngân chọn xác nhận.
25. Lớp MemberCardView gọi lớp PaymentView.
26. Lớp PaymentView hiển thị cho thu ngân.
27. Thu ngân thông báo cho khách hàng
28. Khách hàng trả tiền.
29. Thu ngân nhập số tiền khách trả và click thanh toán.
30. Lớp PaymentView gọi lớp Bill yêu cầu lưu.
31. Lớp Bill lưu hóa đơn.
32. Lớp Bill trả kết quả cho lớp PaymentView.
33. Lớp PaymentView thông báo cho thu ngân.
34. Thu ngân xác nhận.
35. Lớp PaymentView gọi lớp CashierHomeView.
36. Lớp CashierHomeView hiển thị cho thu ngân.



3. Xem thống kê khách hàng theo doanh thu

a, Scenario

Scenario	Xem thống kê khách hàng theo doanh thu
Actor	Quản lí
Pre-condition	Quản lí có tài khoản cấp quản lí, quản lí muốn xem thống kê khách hàng theo doanh thu
Post-condition	Quản lí đã xem được thống kê khách hàng theo doanh thu
Main events	1. Quản lí A vào hệ thống để xem thống kê khách hàng theo doanh thu. 2. GD đăng nhập hiện ra: <ul style="list-style-type: none"> Ô nhập username

- Ô nhập password
- Nút login

3. Quản lí nhập username = ManagerA, password = MA001, click login.

4. GD chính của quản lí hiện ra:

- Nút/link xem báo cáo thống kê.
- Một nút đăng xuất.

5. Quản lí chọn chức năng xem báo cáo thống kê.

6. GD chọn loại báo cáo hiện ra:

- Danh sách chọn đối tượng báo cáo: khách hàng, các loại mặt hàng.
- Danh sách chọn tiêu chí báo cáo: doanh thu, tỉ lệ mua hàng.
- Nút/link quay lại/xác nhận.

7. Quản lí chọn đối tượng là khách hàng, tiêu chí là doanh thu + click xác nhận.

8. GD xem báo cáo thống kê khách hàng hiện ra:

- Hai ô chọn ngày bắt đầu và ngày kết thúc.
- Nút/link để kiểm tra/quay lại.
- Một danh sách thông tin:

STT	Mã KH	Tên KH	SĐT	Email	Địa chỉ	Tổng tiền đã mua	Số lần mua	Số MH mua

- Một ô hiển thị tổng doanh thu trong thời gian chọn báo cáo.

9. Quản lí chọn ngày bắt đầu là 01/08/2023, ngày kết thúc là 31/08/2023 + click kiểm tra.

10. Hệ thống vẫn ở nguyên GD như bước 8 ngoại trừ cập nhật thông tin ngày bắt đầu, ngày kết thúc và danh sách khách hàng sắp xếp giảm dần theo tổng tiền mua hàng:

STT	Mã KH	Tên KH	SĐT	Email	Địa chỉ	Tổng tiền đã mua	Số lần mua	Số MH mua	Hành động
1	KH04	Kiều	0123	k@gmail.com	ABC	100000 Vnd	1	8	Xem
2	KH02	Duy	1234	d@gmail.com	BAH	50000 Vnd	1	1	Xem

- Tổng doanh thu: 150000 Vnd.

11. Quản lí click vào khách hàng có mã KH là KH04 (dòng 1)

12. GD xem báo cáo thống kê khách hàng chi tiết hiện ra:

- Mã KH: KH04
- Tên KH: Kiều

- Số điện thoại: 0123
- Email: k@gmail.com
- Địa chỉ: ABC
- Số lần mua: 1
- Số lượng mặt hàng đã mua: 8
- Tổng tiền hàng: 100000 Vnd
- Danh sách hóa đơn của khách hàng:

Mã HD	Mã TN	Tên siêu thị	Ngày thanh toán	Phương thức thanh toán	Tổng tiền	Hành động
HD02	TN01	Hà Đông	20/08/2023	Thẻ	30000 Vnd	Xem
HD01	TN05	Hà Đông	20/08/2023	Tiền mặt	70000 Vnd	Xem

- Một nút/link quay lại.

13. Quản lí click vào hóa đơn có mã HD là HD01 (đòng 2)

14. GD xem hóa đơn chi tiết hiện ra:

- Mã HD: HD01
- Tên siêu thị: Hà Đông
- Địa chỉ siêu thị: Hà Đông, Hà Nội
- Số điện thoại KH: 0123456789
- Mã TN: TN05
- Tên thu ngân: Luân
- Ngày thanh toán: 20/08/2023

STT	Mã MH	Tên MH	Đơn giá	Số lượng	Số tiền
1	SN01	Snack bí đỏ	5000 Vnd	6	30000 Vnd
2	LS01	Lay's vị bò	20000 Vnd	2	40000 Vnd

- Tổng tiền trước giảm giá: 70000 Vnd
- Số lượng mặt hàng: 8
- Số tiền được giảm: 0 Vnd
- Số tiền phải thanh toán: 70000 Vnd
- Phương thức thanh toán: Tiền mặt
- Một nút/link quay lại.

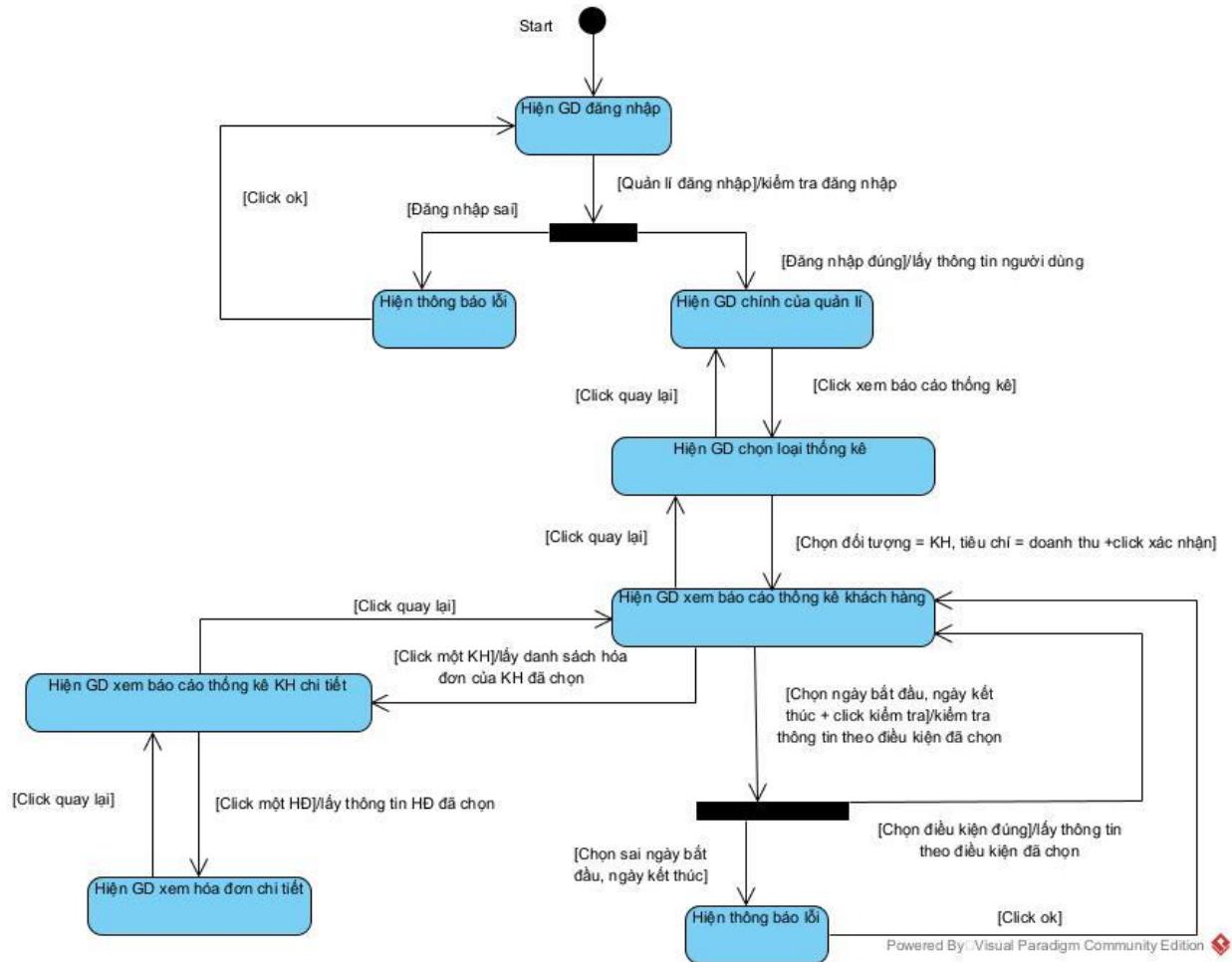
Exception	3. Quản lí nhập sai thông tin đăng nhập + click login. 3.1. Hệ thống hiển thị thông báo “Sai thông tin đăng nhập”. 3.2. Quản lí click ok của thông báo. 3.3. Hệ thống quay lại GD đăng nhập (bước 2).
-----------	--

	<p>9. Quản lý chọn ngày bắt đầu 01/08/2024, ngày kết thúc 31/08/2023 + click kiểm tra.</p> <p>9.1. Hệ thống thông báo “Ngày chưa diễn ra, vui lòng chọn ngày sớm hơn”.</p> <p>9.2. Quản lý click ok của thông báo.</p> <p>9.3. Hệ thống quay lại GD xem báo cáo thống kê khách hàng (bước 8).</p> <p>9.4. Quản lý nhập ngày bắt đầu 10/09/2023, ngày kết thúc 31/08/2023 + click kiểm tra.</p> <p>9.5. Hệ thống thông báo “Vui lòng chọn ngày bắt đầu nhỏ hơn ngày kết thúc”.</p> <p>9.6. Quản lý click ok của thông báo.</p> <p>9.7. Hệ thống quay lại GD xem báo cáo thống kê khách hàng (bước 8).</p>
--	--

b, Biểu đồ trạng thái

- Mô tả trạng thái:

- Từ giao diện đăng nhập, nếu đăng nhập sai thì hiển thị thông báo lỗi, nếu đăng nhập đúng chuyển sang giao diện chính của quản lý.
- Tại giao diện chính của quản lý, khi chọn xem báo cáo thống kê, hệ thống chuyển sang giao diện chọn loại thống kê.
- Tại giao diện chọn lại thống kê, khi chọn đối tượng = KH, tiêu chí = doanh thu + click xác nhận, hệ thống chuyển sang giao diện xem báo cáo thống kê khách hàng.
- Tại giao diện xem báo cáo thống kê khách hàng, khi chọn ngày bắt đầu và ngày kết thúc + click kiểm tra, nếu chọn ngày không hợp lệ thì hiện thông báo lỗi, nếu chọn ngày hợp lệ thì lấy thông tin khách hàng và cập nhật lại giao diện xem báo cáo thống kê khách hàng.
- Tại giao diện xem báo cáo thống kê khách hàng, khi chọn một khách hàng, hệ thống chuyển sang giao diện xem báo cáo thống kê khách hàng chi tiết.
- Tại giao diện xem báo cáo thống kê khách hàng chi tiết, khi chọn một hóa đơn, hệ thống chuyển sang giao diện xem hóa đơn chi tiết.
- Tại tất cả các giao diện, khi click quay lại đều quay lại giao diện trước đây.



c, Biểu đồ lớp thực thể

Bước 1: Thay thế bằng Scenario ở bước a

Bước 2+3: Trích danh từ và đánh giá

- Siêu thị (loại bỏ)
- Khách hàng (làm lớp thực thể) => class Customer
- Quản lí (làm lớp thực thể) => class Manager
- Thu ngân (làm lớp thực thể) => class Cashier
- Thành viên (làm lớp thực thể) => class Member
- Thẻ thành viên (làm lớp thực thể) => class MemberCard
- Phiếu giảm giá (làm lớp thực thể) => class Coupon
- Hệ thống (loại bỏ)
- Giao diện (loại bỏ)

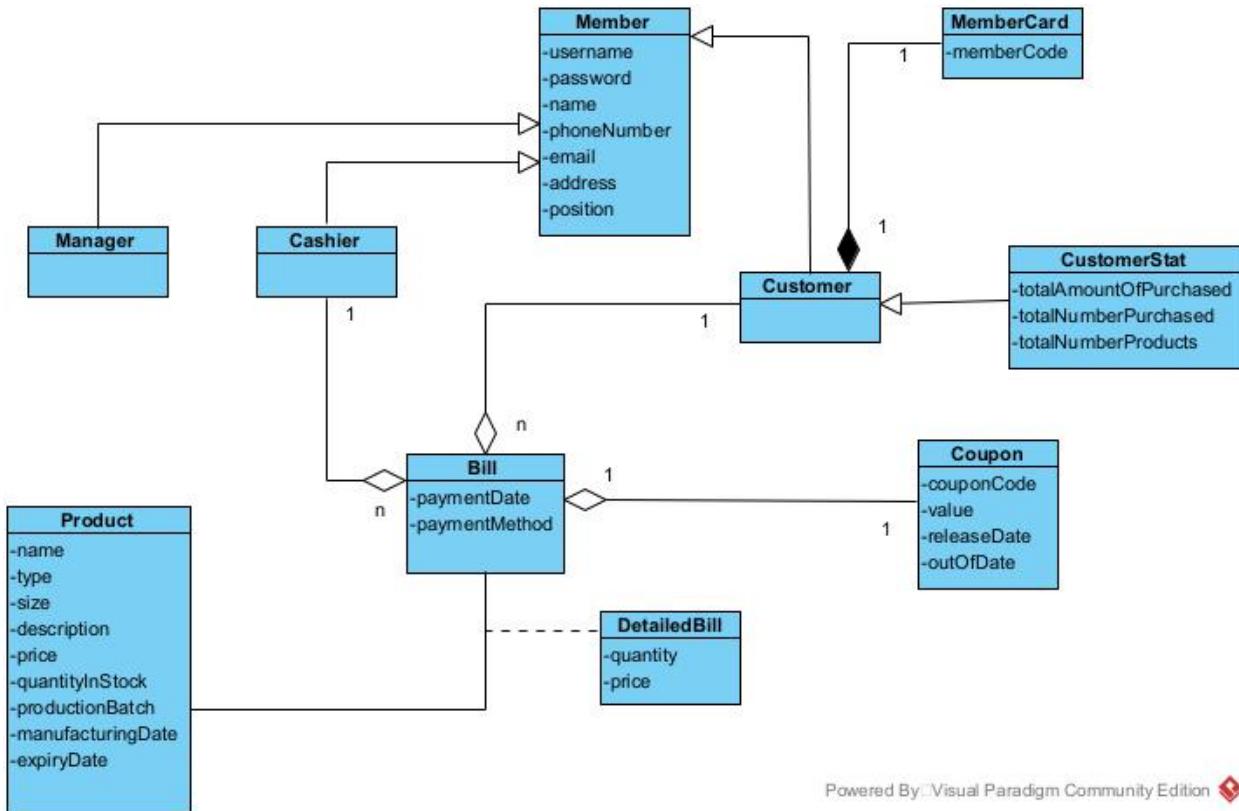
- Username (làm thuộc tính)
- Password (làm thuộc tính)
- Đối tượng (loại bỏ)
- Tiêu chí (loại bỏ)
- Ngày bắt đầu (loại bỏ)
- Ngày kết thúc (loại bỏ)
- Doanh thu (làm thuộc tính)
- Thống kê khách hàng theo doanh thu (làm lớp thực thể) => class IncomeCustomerStat
- Hóa đơn (làm lớp thực thể) => class Bill
- Mặt hàng (làm lớp thực thể) => class Product

Bước 4: Xét quan hệ số lượng giữa các lớp

- Một thu ngân thanh toán nhiều hóa đơn => quan hệ Cashier và Bill là 1-n.
- Một khách hàng mua nhiều hóa đơn => quan hệ Customer và Bill là 1-n.
- Một khách hàng có một thẻ thành viên => quan hệ Customer và MemberCard là 1-1.
- Một hóa đơn có một phiếu giảm giá, một phiếu giảm giá dùng cho nhiều hóa đơn => quan hệ Bill và Coupon là 1-1.
- Một hóa đơn có nhiều mặt hàng, một mặt hàng được bán trong nhiều hóa đơn. Đề xuất thêm lớp DetailedBill xác định duy nhất một mặt hàng trong một hóa đơn.

Bước 5: Bổ sung quan hệ đối tượng

- Quản lý, thu ngân kế thừa thành viên.
- Khách hàng kế thừa thành viên.
- Thu ngân là thành phần của hóa đơn.
- Khách hàng là thành phần của hóa đơn.
- Khách hàng là thành phần của thẻ thành viên.
- Phiếu giảm giá là thành phần của hóa đơn.
- Mặt hàng và hóa đơn liên kết tạo ra hóa đơn chi tiết duy nhất.



Powered By Visual Paradigm Community Edition

d, Biểu đồ lớp đầy đủ

- Quản lí đăng nhập vào hệ thống -> Giao diện đăng nhập hiện ra -> để xuất lớp LoginView:

- Input cho username -> inpUsername
- Input cho password -> inpPassword
- Nút submit để đăng nhập -> subLogin

- Quản lí nhập username/password và click login -> hệ thống phải kiểm tra thông tin đăng nhập đúng không -> cần phương thức:

- Name: checkLogin()
- Input: username, password (Member)
- Output: Member
- Owner: Member

- Khi đăng nhập thành công -> hiển thị giao diện chính của quản lí -> để xuất lớp ManagerHomeView:

- Một nút quản lí thông tin mặt hàng -> subManageProduct

- Một nút xem báo cáo thống kê -> subSeeStatistical
- Một nút đăng xuất -> subSignOut

- Quản lý chọn nút xem báo cáo thống kê -> hiển thị giao diện chọn loại báo cáo -> để xuất lớp SelectStatView:

- Một danh sách đối tượng báo cáo -> outsubReportObject
- Một danh sách tiêu chí báo cáo -> ousubReportCriteria
- Một nút/link quay lại -> subBack
- Một nút/link xác nhận -> subConfirm

- Quản lý chọn đối tượng = “khách hàng”, tiêu chí = “doanh thu” + click xác nhận -> hiển thị giao diện xem báo cáo thống kê khách hàng -> để xuất lớp CustomerStatView:

- Input cho ngày bắt đầu -> inpStartDate
- Input cho ngày kết thúc -> inpEndDate
- Một nút quay lại -> subBack
- Một nút kiểm tra -> subCheck
- Một danh sách thông tin khách hàng có số liệu thống kê -> outsubListCustomerStat
- Output cho tổng doanh thu trong thời gian báo cáo -> outTotalRevenue

- Quản lý chọn ngày bắt đầu và ngày kết thúc + click kiểm tra -> hệ thống cần truy vấn toàn bộ số liệu thống kê khách hàng -> cần phương thức:

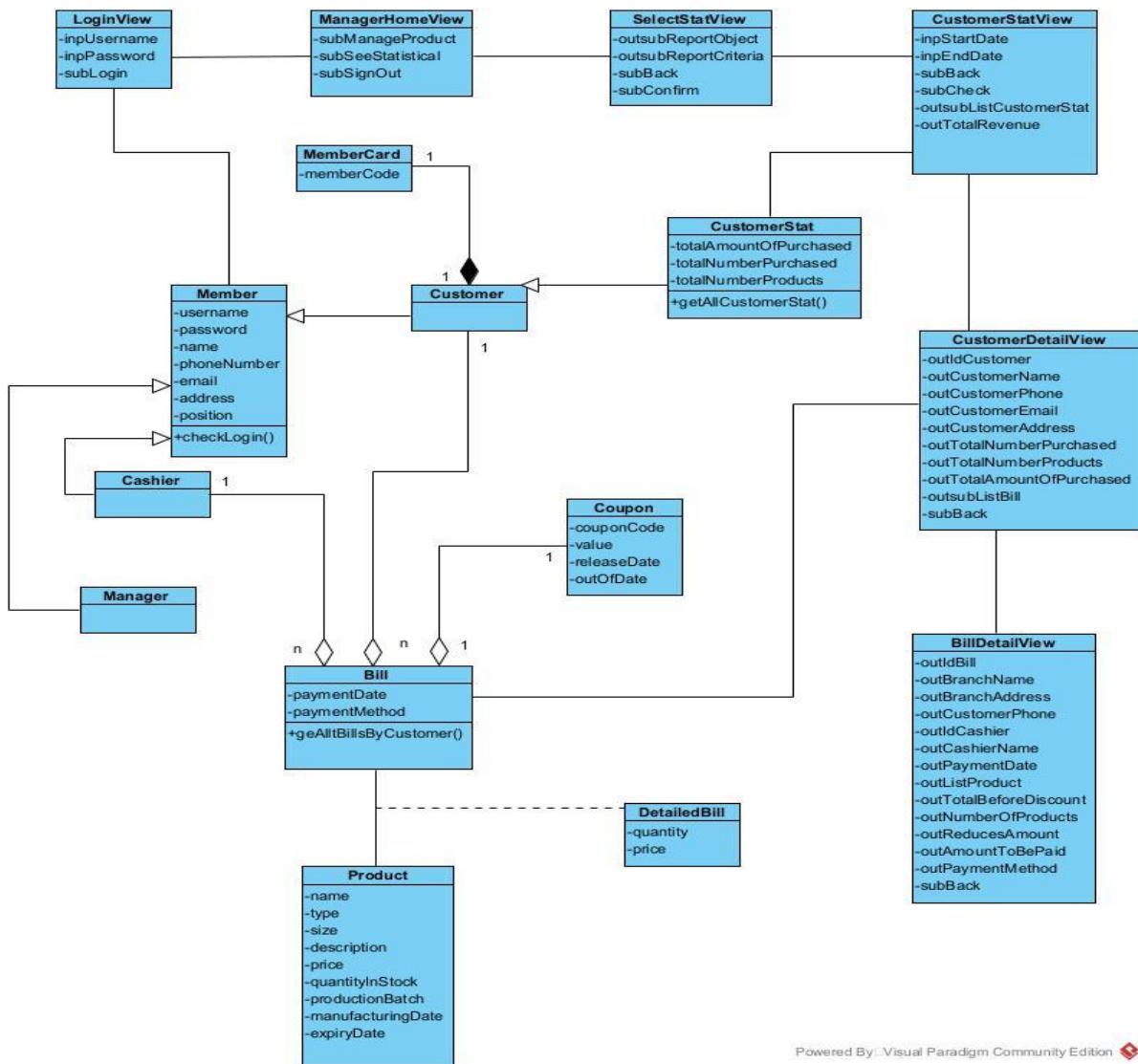
- Name: getAllCustomerStat()
- Input: ngày bắt đầu, ngày kết thúc
- Output: List<CustomerStat>
- Owner: CustomerStat

- Kết quả được trả về và hiển thị trên CustomerStatView.

- Quản lý chọn khách hàng để xem chi tiết -> hiển thị giao diện thống kê chi tiết về khách hàng -> để xuất lớp CustomerDetailView:

- Mã KH -> outIdCustomer
- Tên KH -> outCustomerName
- Số điện thoại KH -> outCustomerPhone
- Email KH -> outCustomerEmail
- Địa chỉ KH -> outCustomerAddress

- Số lần mua -> outTotalNumberPurchased
 - Số lượng MH đã mua -> outTotalNumberProducts
 - Tổng tiền hàng -> outTotalAmountOfPurchased
 - Danh sách hóa đơn liên quan đến khách hàng trong khoảng thời gian đó -> outsubListBill
 - Một nút quay lại -> subBack
- Để lấy dữ liệu khách hàng và các hóa đơn liên quan hiển thị trên CustomerDetailView -> cần phương thức:
- Name: getAllBillsByCustomer()
 - Input: ngày bắt đầu, ngày kết thúc và id khách hàng
 - Output: List<Bill>
 - Owner: Bill
- Kết quả được trả về trên CustomerDetailView.
- Quản lý chọn hóa đơn để xem chi tiết -> hiển thị giao diện chi tiết của hóa đơn -> để xuất lớp BillDetailView:
- Mã HD -> outIdBill
 - Tên chi nhánh -> outBranchName
 - Địa chỉ chi nhánh -> outBranchAddress
 - Số điện thoại KH -> outCustomerPhone
 - Mã TN -> outIdCashier
 - Tên TN -> outCashierName
 - Ngày thanh toán -> outPaymentDate
 - Danh sách các mặt hàng của hóa đơn -> outListProduct
 - Tổng tiền trước giảm giá -> outTotalBeforeDiscount
 - Số lượng mặt hàng -> outNumberOfProducts
 - Số tiền được giảm -> outReducesAmount
 - Số tiền phải thanh toán -> outAmountToBePaid
 - Phương thức thanh toán -> outPaymentMethod
 - Một nút quay lại -> subBack



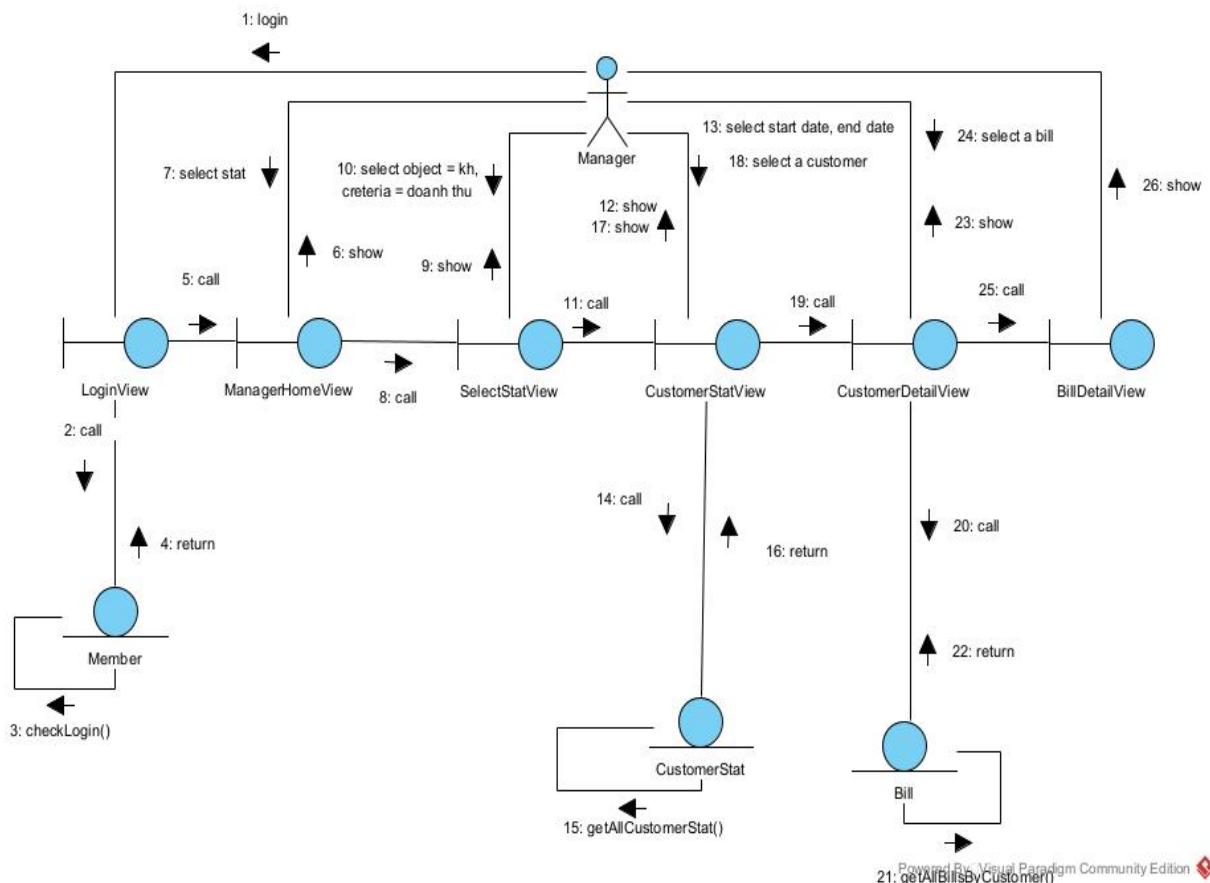
Powered By Visual Paradigm Community Edition

e, Biểu đồ giao tiếp

- Kịch bản v.2

- Tại lớp LoginView, quản lí đăng nhập.
- Lớp LoginView gọi lớp Member để kiểm tra thông tin.
- Lớp Member kiểm tra thông tin đăng nhập.
- Lớp Member trả kết quả cho lớp LoginView.
- Lớp LoginView gọi lớp ManagerHomeView.
- Lớp ManagerHomeView hiển thị cho quản lí.
- Quản lí chọn xem báo cáo thống kê.
- Lớp ManagerHomeView gọi lớp SelectStatView.
- Lớp SelectStatView hiển thị cho quản lí.

10. Quản lí chọn đối tượng = KH, tiêu chí = doanh thu và xác nhận.
11. Lớp SelectStatView gọi lớp CustomerStatView.
12. Lớp CustomerStatView hiển thị cho quản lí.
13. Quản lí chọn ngày bắt đầu và kết thúc.
14. Lớp CustomerStatView gọi lớp CustomerStat để lấy thông tin.
15. Lớp CustomerStat lấy thông tin khách hàng theo điều kiện.
16. Lớp CustomerStat trả kết quả cho lớp CustomerStatView.
17. Lớp CustomerStatView hiển thị cho quản lí.
18. Quản lí chọn một khách hàng.
19. Lớp CustomerStatView gọi lớp CustomerDetailView.
20. Lớp CustomerDetailView gọi lớp Bill để lấy thông tin hóa đơn.
21. Lớp Bill lấy thông tin hóa đơn của khách hàng.
22. Lớp Bill trả kết quả cho lớp CustomerDetailView.
23. Lớp CustomerDetailView hiển thị cho quản lí.
24. Quản lí chọn một hóa đơn.
25. Lớp CustomerDetailView gọi lớp BillDetailView.
26. Lớp BillDetailView hiển thị cho quản lí.



III. Design

1. Thiết kế lớp thực thể

Bước 1: Hoàn thiện các lớp

*) Member (không kế thừa lớp nào, bổ sung id)

- id: int
- username: String
- password: String
- name: String
- phoneNumber: String
- email: String
- address: String
- position: String

*) Manager (kế thừa Member)

*) Cashier (kế thừa Member)

*) Customer (kế thừa Member)

*) MemberCard (không kế thừa lớp nào, bổ sung id)

- id: int
- memberCode: String

*) CustomerStat (kế thừa Customer)

- totalAmountOfPurchased: float
- totalNumberPurchased: int
- totalNumberProducts: int

*) Coupon (không kế thừa lớp nào, bổ sung id)

- id: int
- couponCode: String
- value: float
- releaseDate: Date
- outOfDate: Date

*) Bill (không kế thừa lớp nào, bổ sung id)

- id: int
- paymentDate: Date
- paymentMethod: String

*) DetailedBill (không kế thừa lớp nào, bỏ sung id)

- id: int
- quantity: int
- price: float

*) Product (không kế thừa lớp nào, bỏ sung id)

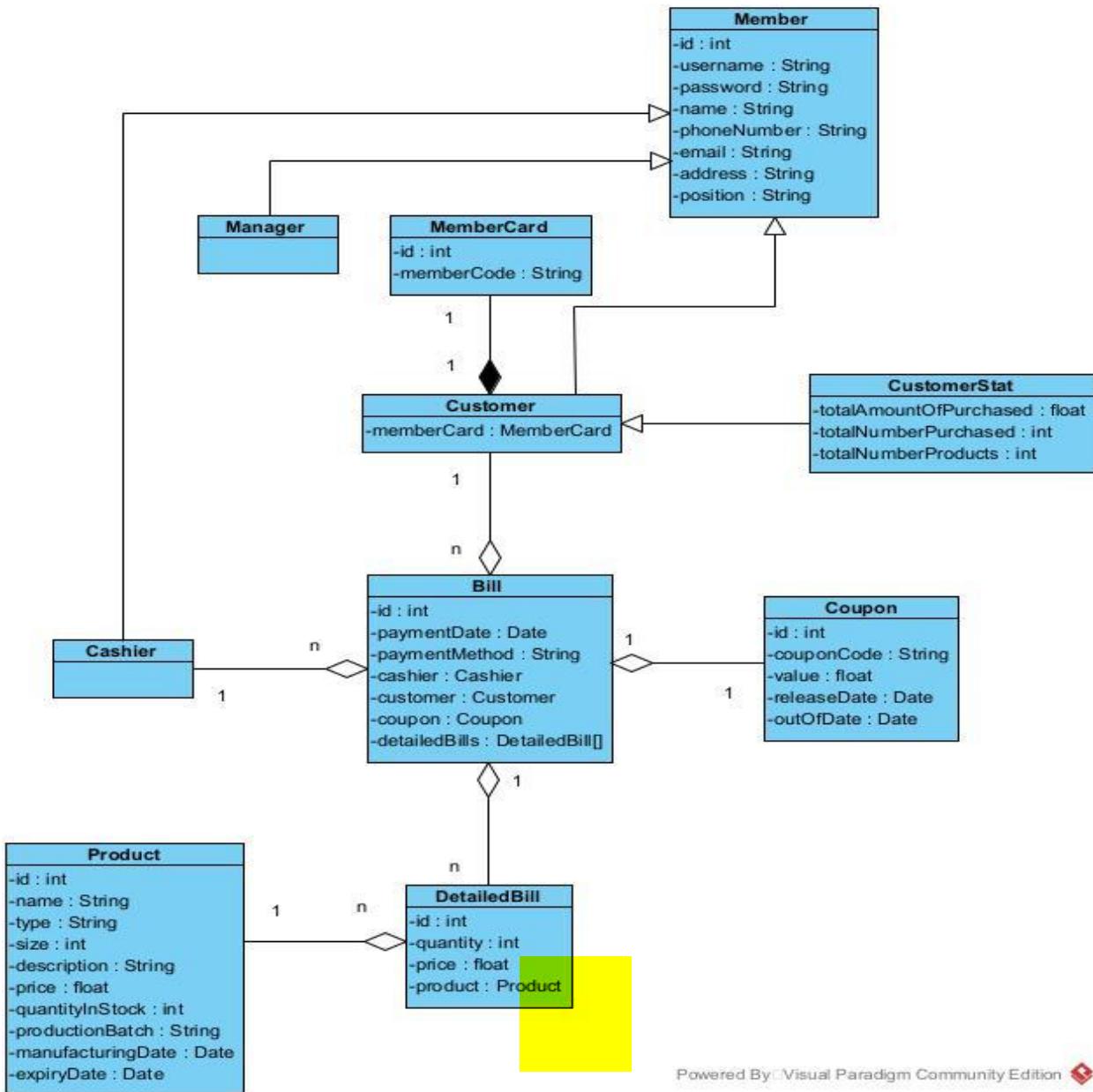
- id: int
- name: String
- type: String
- size: int
- description: String
- price: float
- quantityInStock: int
- productionBatch: String
- manufacturingDate: Date
- expiryDate: Date

Bước 2: Chuyển đổi tất cả các quan hệ association thành aggregation/composition

*) Bill + Product -> DetailedBill được chuyển thành: Product là thành phần của DetailedBill, DetailedBill là thành phần của Bill.

Bước 3: Thêm các thuộc tính đối tượng tương ứng với các mối quan hệ aggregation/composition

- MemberCard là một thành phần của User, kiểu 1-1 -> User có một MemberCard.
- User là một thành phần của Bill, kiểu 1-n -> Bill có một User.
- Cashier là một thành phần của Bill, kiểu 1-n -> Bill có một Cashier.
- Coupon là một thành phần của Bill, kiểu 1-1 -> Bill có một Coupon.
- DetailedBill là một thành phần của Bill, kiểu n-1 -> Bill có một danh sách DetailedBill.
- Product là một thành phần của DetailedBill, kiểu 1-n -> DetailedBill có một Product.



2. Thiết kế cơ sở dữ liệu

Bước 1: Mô hình lớp thực thể để xuất bảng tương ứng

- Lớp Member -> bảng tblMember.
- Lớp Manager -> bảng tblManager.
- Lớp Cashier -> bảng tblCashier.
- Lớp Customer -> bảng tblCustomer.
- Lớp MemberCard -> bảng tblMemberCard.
- Lớp Bill -> bảng tblBill.

- Lớp Coupon -> bảng tblCoupon.
- Lớp Product -> bảng tblProduct.
- Lớp DetailedBill -> bảng tblDetailedBill.

Bước 2: Các thuộc tính không đối tượng của lớp nào chuyển thành thuộc tính của bảng tương ứng

- Bảng tblMember: id, username, password, name, phoneNumber, email, address, position.
- Bảng tblManager
- Bảng tblCashier
- Bảng tblCustomer
- Bảng tblMemberCard: id, memberCode.
- Bảng tblBill: id, paymentDate, paymentMethod.
- Bảng tblCoupon: id, couponCode, value, releaseDate, outOfDate.
- Bảng tblProduct: id, name, type, size, description, price, quantityInStock, productionBatch, manufacturingDate, expiryDate.
- Bảng tblDetailedBill: id, quantity, price.

Bước 3: Chuyển quan hệ số lượng giữa các lớp thực thể thành quan hệ số lượng giữa các bảng

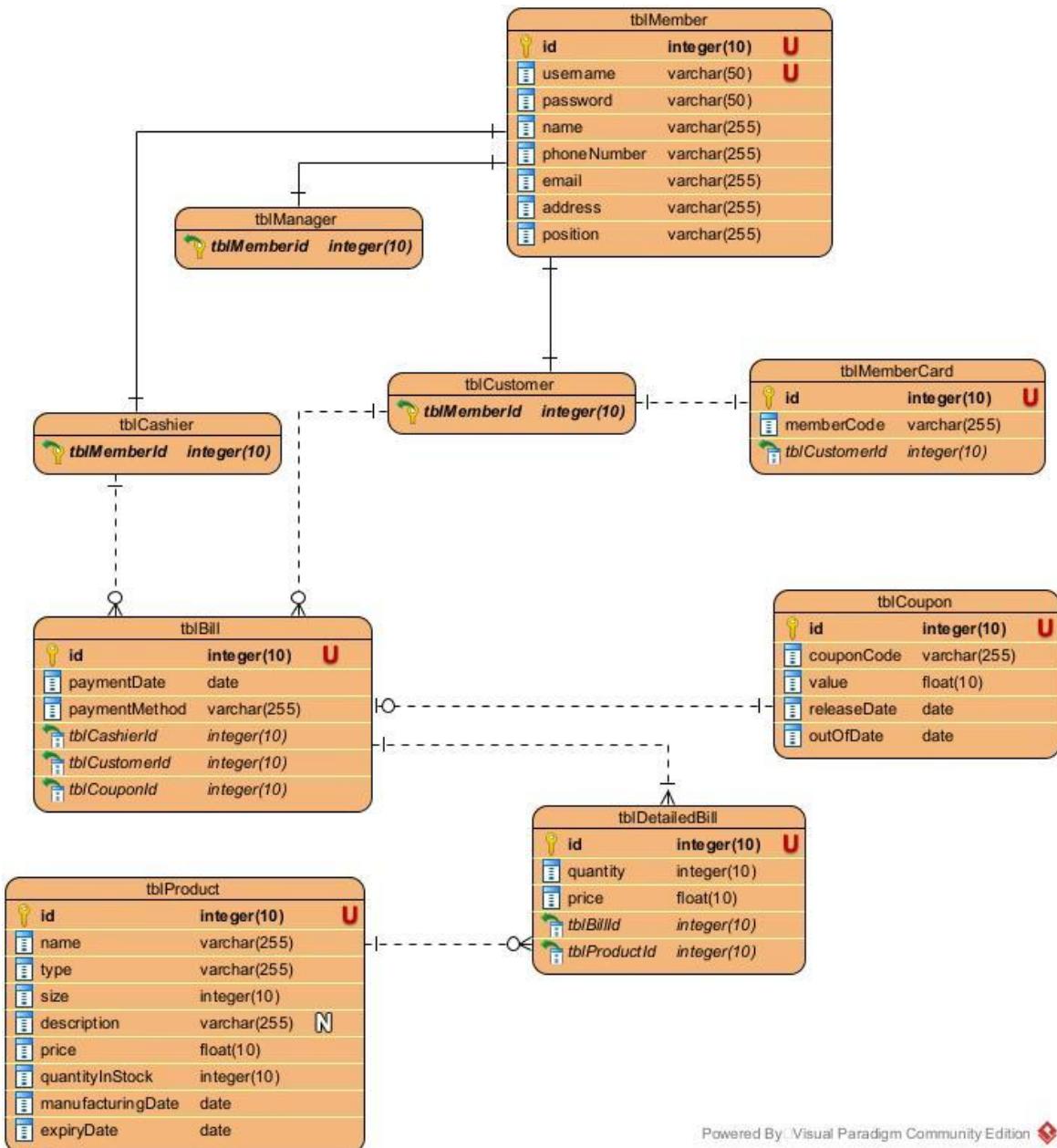
- 1 tblCustomer – 1 tblMemberCard.
- 1 tblCustomer – n tblBill.
- 1 tblCashier – n tblBill.
- 1 tblCoupon – 1 tblBill.
- 1 tblBill – n tblDetailedBill.
- 1 tblProduct – n tblDetailedBill.

Bước 4: Bổ sung các thuộc tính khóa chính

- Khóa chính được thiết lập với thuộc tính id của các bảng tương ứng, trừ các bảng tblManager, tblCashier, tblCustomer.
- Khóa ngoại được thiết lập cho các bảng:
 - 1 tblCustomer – 1 tblMember -> tblCustomer có khóa ngoại tblMemberId.
 - 1 tblManager – 1 tblMember -> tblManager có khóa ngoại tblMember.
 - 1 tblCashier – 1 tblMember -> tblCashier có khóa ngoại tblMember.

- o 1 tblCustomer – 1 tblMemberCard -> tblMemberCard có khóa ngoại tblCustomerId.
- o 1 tblCustomer – n tblBill -> tblBill có khóa ngoại tblCustomerId.
- o 1 tblCashier – n tblBill -> tblBill có khóa ngoại tblCashierId.
- o 1 tblCoupon – 1 tblBill -> tblBill có khóa ngoại tblCouponId.
- o 1 tblBill – n tblDetailedBill -> tblDetailedBill có khóa ngoại tblBillId.
- o 1 tblProduct – n tblDetailedBill -> tblDetailedBill có khóa ngoại tblProductId.

Bước 5: Loại bỏ các thuộc tính dư thừa: Không có thuộc tính dư thừa



3. Quản lý thông tin mặt hàng

a, Thiết kế biểu đồ lớp dày đủ

Bước 1: Thiết kế giao diện

*) Giao diện đăng nhập

Đăng Nhập

Tên người dùng:

Mật khẩu:

Đăng Nhập

*) Giao diện trang chủ của quản lý

Manager Home

Quản lý sản phẩm Xem báo cáo Đăng xuất

*) Giao diện quản lý mặt hàng

Manage Product

Search product											Tìm kiếm	Thêm sản phẩm	Quay lại
Mã	Tên	Loại	Kích thước	Mô tả	Giá	Tồn kho	Lô sản xuất	NSX	HSD	Xem	Sửa	Xóa	
1	Snack	ăn vặt	250gr	Ngon	5000vnđ	50 gói	AV1	01/05/2023	01/05/2023	Xem	Sửa	Xóa	
1	Snack	ăn vặt	250gr	Ngon	5000vnđ	50 gói	AV1	01/05/2023	01/05/2023	Xem	Sửa	Xóa	

*) Giao diện sửa thông tin mặt hàng

Repair Product

Mã:	<input type="text" value="Mã mặt hàng cũ ở đây"/>	Tên:	<input type="text" value="Tên mặt hàng cũ ở đây"/>
Loại:	<input type="text" value="Loại mặt hàng cũ ở đây"/>	Kích thước:	<input type="text" value="Kích thước cũ ở đây"/>
Mô tả:	<input type="text" value="Mô tả cũ ở đây"/> 	Giá:	<input type="text"/>
Số lượng tồn:	<input type="text"/>	Lô sản xuất:	<input type="text" value="Lô sản xuất cũ ở đây"/>
Ngày sản xuất:	<input type="text" value="dd/mm/yyyy"/>	Hạn sử dụng:	<input type="text" value="dd/mm/yyyy"/>
<input style="background-color: #007bff; color: white; border: none; padding: 5px 10px; margin-right: 5px;" type="button" value="Lưu"/> <input style="background-color: #dc3545; color: white; border: none; padding: 5px 10px;" type="button" value="Quay lại"/>			

Bước 2: Đề xuất lớp giao diện tương ứng

- Trang đăng nhập -> đề xuất lớp giao diện login.html
 - form:
 - action: “/login”
 - inpUsername: text
 - inpPassword: password
 - btnLogin: submit
- Trang chủ của quản lí -> đề xuất lớp giao diện manager_home.html
 - btnManageProduct: link
 - btnSeeStatistical: link
 - btnSignout: submit
 - manager: Manager
- Trang quản lí mặt hàng -> đề xuất lớp giao diện manage_product.html
 - inpSearchProductName: text
 - btnSearch: submit
 - btnAddProduct: link
 - btnBack: link
 - tblProduct: table
 - btnSee: link
 - btnRepair: link
 - btnDelete: link
 - manager: Manager
 - listProduct: Product[]

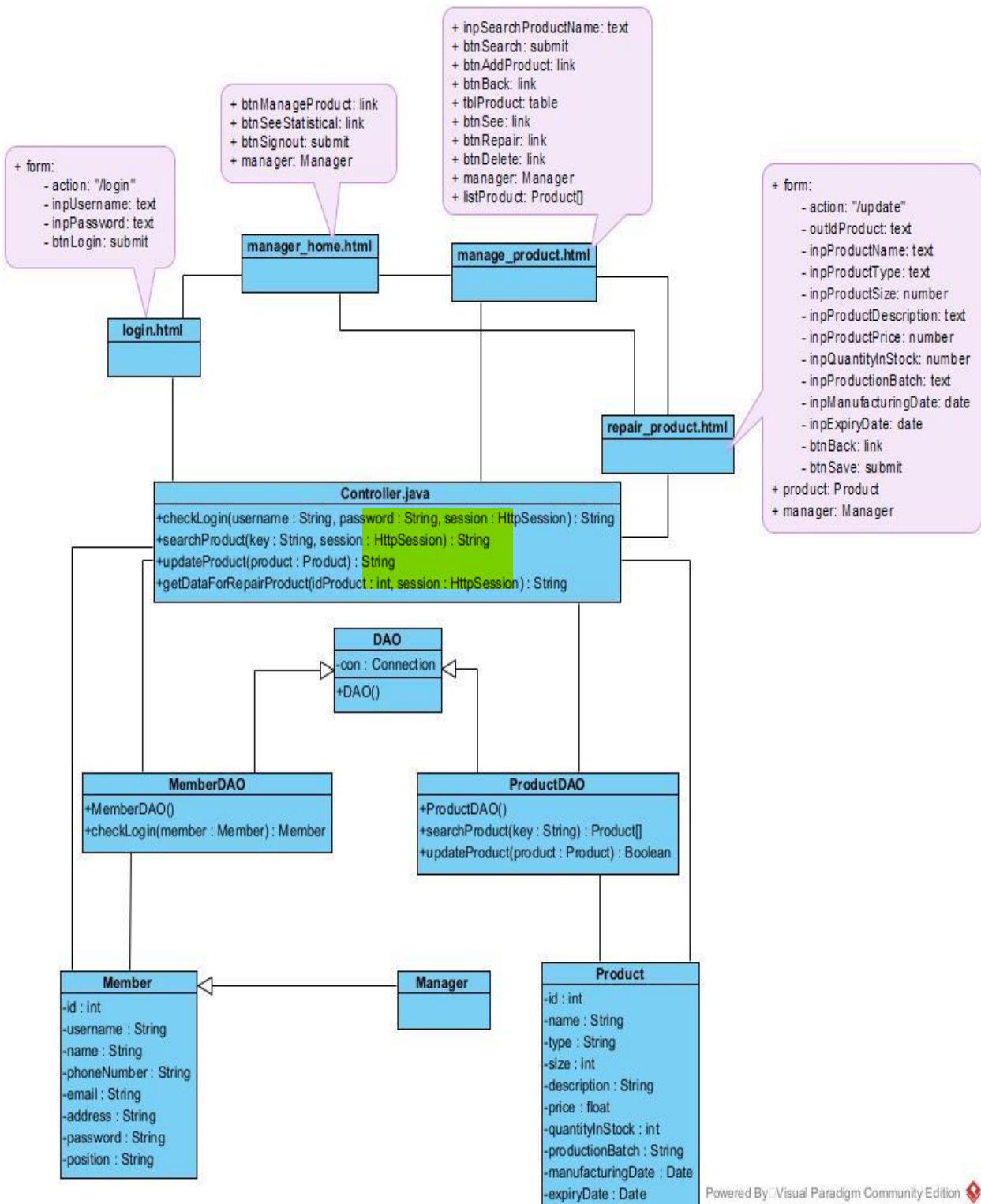
- Trang sửa mặt hàng -> để xuất lớp giao diện repair_product.html
 - form:
 - action: “/update”
 - outIdProduct: text
 - inpProductName: text
 - inpProductType: text
 - inpProductSize: number
 - inpProductDescription: text
 - inpProductPrice: number
 - inpQuantityInStock: number
 - inpProductionBatch: text
 - inpManufacturingDate: date
 - inpExpiryDate: date
 - btnBack: link
 - btnSave: submit
 - product: Product
 - manage: Manager

Bước 3: Để xuất lớp truy xuất dữ liệu

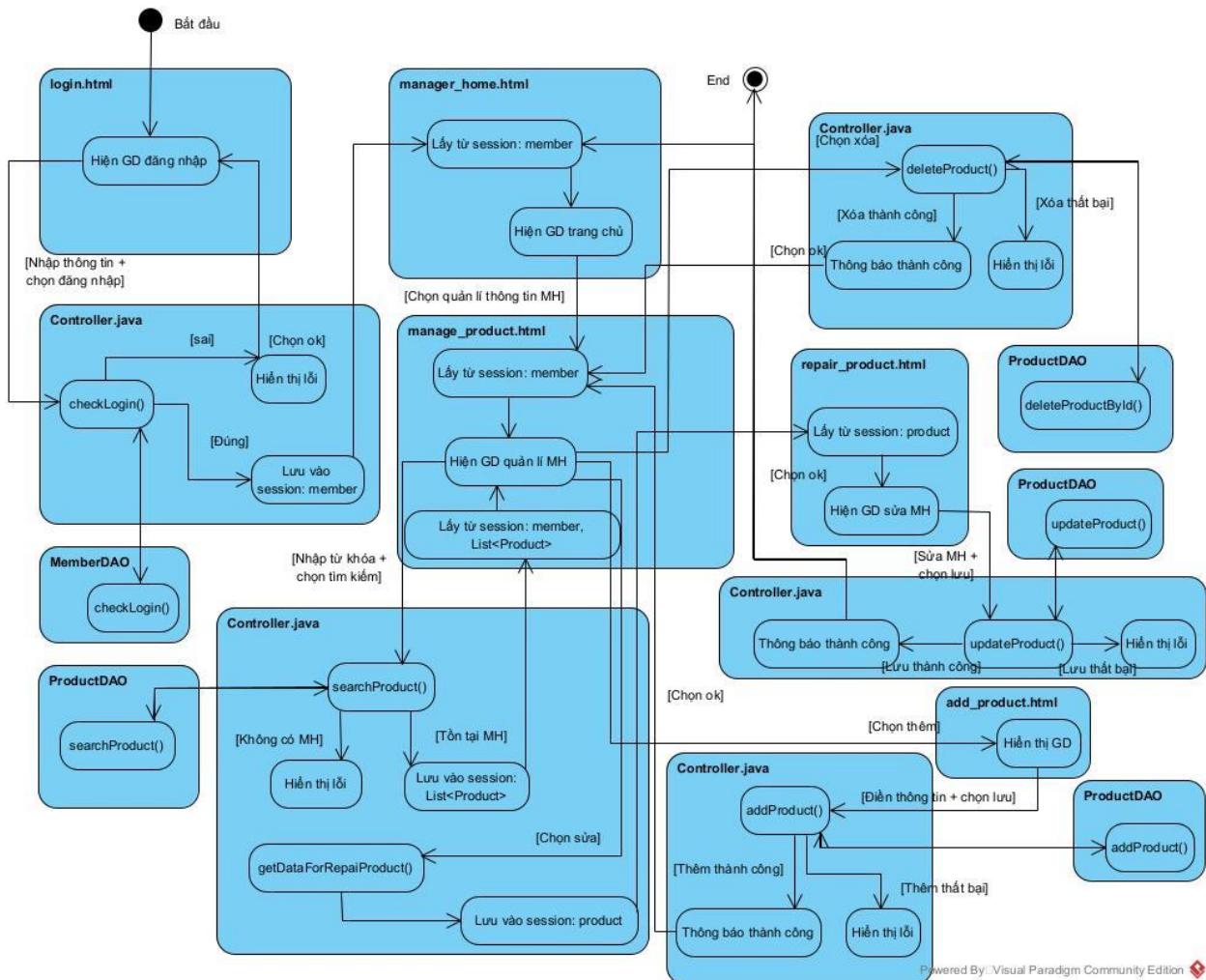
- Lớp điều khiển:
 - Controller.java:
 - checkLogin(username: String, password: String, session: HttpSession): String
 - Lấy từ request: username và password
 - Gọi lớp Member để đóng gói
 - Gọi MemberDAO để kiểm tra
 - Nếu đúng:
 - Lưu member vào session
 - Chuyển sang “manager_home.html”
 - Nếu sai: hiện thông báo sai tài khoản
 - searchProduct(key: String, session: HttpSession): String
 - Lấy từ request: key
 - Gọi ProductDAO để tìm kiếm
 - Nếu tồn tại mặt hàng:
 - Lưu danh sách mặt hàng vào session
 - Hiển thị trên “manage_product.html”

- Nếu không có mặt hàng: hiển thị thông báo mặt hàng không tồn tại
- `getDataForRepairProduct(idProduct: int, session: HttpSession): String`
 - Lấy từ request: idProduct
 - Lấy từ session: listProduct
 - Gọi lớp Product đóng gói dữ liệu cho product có id là idProduct
 - Lưu product vào session
 - Chuyển sang và hiển thị trên “repair_product.html”
- `updateProduct(product: Product): String`
 - Lấy từ request: product
 - Gọi ProductDAO để lưu
 - Nếu cập nhật thành công: quay lại “manager_home.html”
 - Nếu cập nhật thất bại: hiển thị thông báo thất bại
- Lớp giao tiếp với cơ sở dữ liệu
 - DAO là một lớp chung của DAO. Nó chỉ có cấu trúc để kết nối với DB và cung cấp kết nối chung cho tất cả các lớp DAO kế thừa trong hệ thống.
 - MemberDAO có phương thức:
 - `checkLogin(member: Member): Member` – dùng để kiểm tra và lấy thông tin của thành viên khi đăng nhập.
 - Input: Member
 - Output: Member
 - ProductDAO có hai phương thức:
 - `searchProduct(key: String): Product[]` – dùng để tìm kiếm mặt hàng có từ khóa.
 - Input: key
 - Output: Product[]
 - `updateProduct(product: Product): Boolean` – cập nhật thông tin mặt hàng vào cơ sở dữ liệu.
 - Input: Product
 - Output: Boolean

Bước 4: Bao gồm tất cả các thực thể liên quan



b, Thiết kế biểu đồ hoạt động



c, Thiết kế biểu đồ tuần tự

- Tại trang `login.html`, quản lí nhập username, password và chọn đăng nhập.
- Trang `login.html` gọi lớp `Controller.java`.
- Lớp `Controller.java` gọi hàm `checkLogin()`.
- Hàm `checkLogin()` của lớp `Controller.java` gọi lớp `Member` đóng gói dữ liệu.
- Lớp `Member` khởi tạo, gọi các phương thức `setter()`.
- Lớp `Member` trả kết quả cho hàm `checkLogin()` của lớp `Controller.java`.
- Hàm `checkLogin()` của lớp `Controller.java` gọi lớp `MemberDAO` để kiểm tra.
- Lớp `MemberDAO` gọi hàm `checkLogin()`.
- Hàm `checkLogin()` của lớp `MemberDAO` gọi lớp `Member` đóng gói lại dữ liệu.
- Lớp `Member` gọi các phương thức `setter()`.

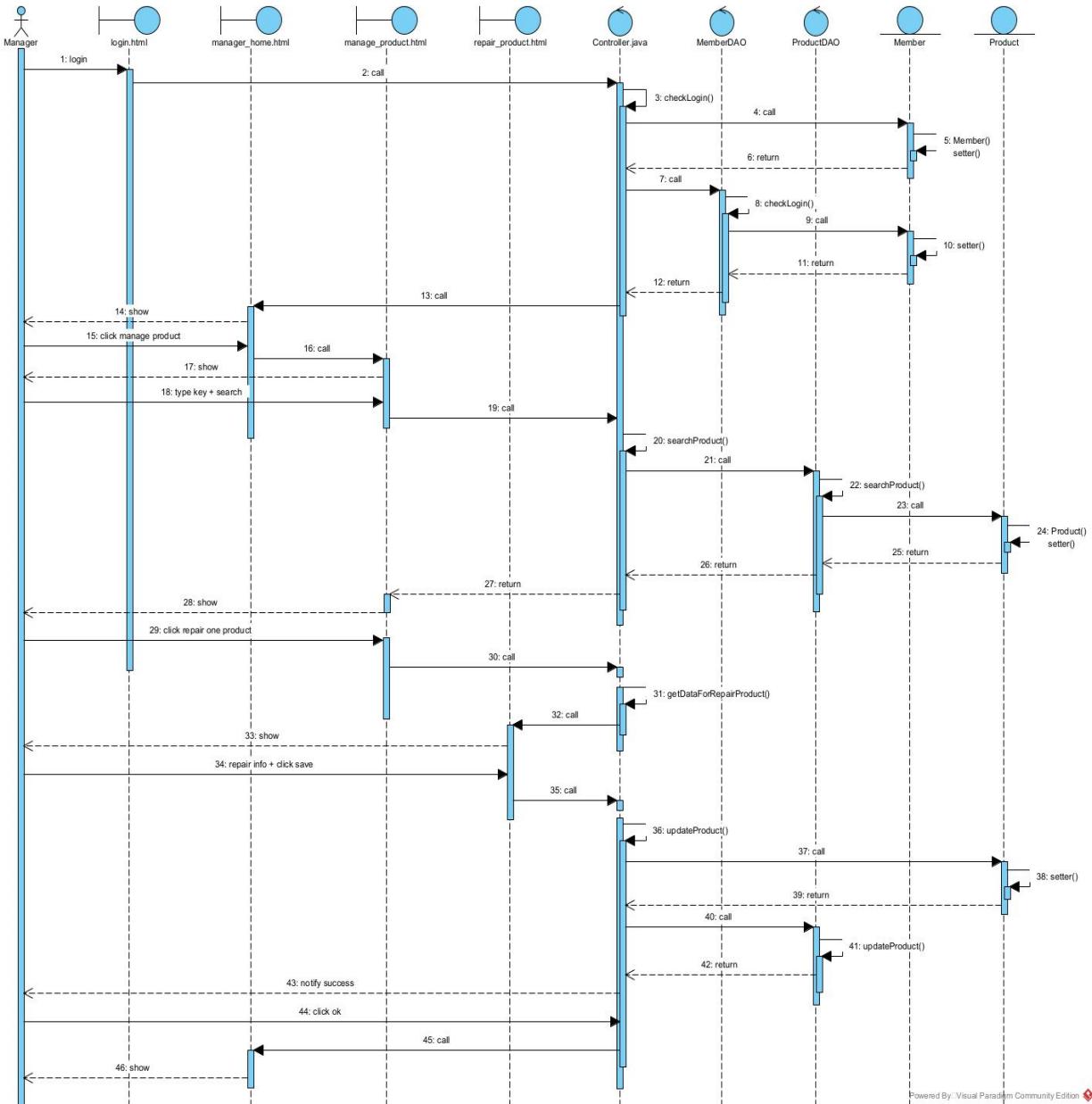
11. Lớp Member trả kết quả cho hàm checkLogin() của lớp MemberDAO.
12. Hàm checkLogin() của lớp MemberDAO trả kết quả cho hàm checkLogin() của lớp Controller.java.
13. Hàm checkLogin() của lớp Controller.java gọi trang manager_home.html.
14. Trang manager_home.html hiển thị cho quản lí.
15. Quản lí chọn chức năng quản lí thông tin mặt hàng.
16. Trang manager_home.html gọi trang manage_product.html.
17. Trang manage_product.html hiển thị cho quản lí.
18. Quản lí nhập từ khóa và chọn tìm kiếm.
19. Trang manage_product.html gọi lớp Controller.java.
20. Lớp Controller.java gọi hàm searchProduct().
21. Hàm searchProduct() của lớp Controller.java gọi lớp ProductDAO.
22. Lớp ProductDAO gọi hàm searchProduct().
23. Hàm searchProduct() của lớp ProductDAO gọi lớp Product đóng gói kết quả.
24. Lớp Product khởi tạo, gọi các phương thức setter().
25. Lớp Product trả kết quả cho hàm searchProduct() của lớp ProductDAO.
26. Hàm searchProduct() của lớp ProductDAO trả kết quả cho hàm searchProduct() của lớp Controller.java.
27. Hàm searchProduct() của lớp Controller.java trả kết quả cho trang manage_product.html.
28. Trang manage_product.html hiển thị cho quản lí.
29. Quản lí chọn một mặt hàng cần sửa.
30. Trang manage_product.html gọi lớp Controller.java.
31. Lớp Controller.java gọi hàm getDataForRepairProduct().
32. Hàm getDataForRepairProduct() gọi trang repair_product.html.
33. Trang repair_product.html hiển thị cho quản lí.
34. Quản lí sửa thông tin và chọn lưu.
35. Trang repair_product.html gọi lớp Controller.java.
36. Lớp Controller.java gọi hàm updateProduct().
37. Hàm updateProduct() của lớp Controller.java gọi lớp Product đóng gói dữ liệu lại.
38. Lớp Product gọi các phương thức setter() tương ứng.
39. Lớp Product trả kết quả cho hàm updateProduct() của lớp Controller.java.
40. Hàm updateProduct() của lớp Controller.java gọi lớp ProductDAO.
41. Lớp ProductDAO gọi hàm updateProduct().
42. Hàm updateProduct() của lớp ProductDAO trả kết quả cho hàm updateProduct() của lớp Controller.java.

43. Hàm updateProduct() của lớp Controller.java thông báo thành công cho quản lí.

44. Quản lí chọn ok.

45. Hàm updateProduct() của lớp Controller.java gọi trang manager_home.html.

46. Trang manager_home.html hiển thị cho quản lí.



4. Thanh toán cho khách tại quầy

a, Thiết kế biểu đồ lớp đầy đủ

Bước 1: Thiết kế giao diện

*) Giao diện đăng nhập

Đăng Nhập

Tên người dùng:

Mật khẩu:

Đăng Nhập

*) Giao diện trang chủ của thu ngân

Cashier Home

Thanh toán**Đăng xuất**

*) Giao diện thanh toán

Payment

Ngày thanh toán: dd/mm/yyyy

Tên siêu thị:	<input type="text"/>	Địa chỉ siêu thị:	<input type="text"/>		
Mã TN:	<input type="text"/>	Tên TN:	<input type="text"/>		
Thẻ thành viên:	<input type="text"/>	Dùng	Phiếu giảm giá:	<input type="text"/>	Dùng

Danh sách mặt hàng:

STT	Mã MH	Tên MH	Đơn giá	Số lượng	Thành tiền	Sửa
1	1	Snack	5000vnđ	1	5000vnđ	Sửa
1	1	Snack	5000vnđ	1	5000vnđ	Sửa

Tổng tiền trước giảm:

Số lượng mặt hàng:

Số tiền được giảm:

Số tiền phải thanh toán:

Số tiền khách trả:

Số tiền trả khách:

Lưu **Xóa**

Thông tin sản phẩm:

Mã sản phẩm: **Kiểm tra**

Tên sản phẩm:

Kích thước:

Hạn sử dụng: dd/mm/yyyy

Đơn giá:

Số lượng:

Thành tiền:

Thanh toán **Hủy**

*) Giao diện phiếu giảm giá

The screenshot shows a form titled "Coupon". It contains four input fields: "Mã giảm giá:" with a placeholder "Mã giảm giá", "Giá trị:" with a placeholder "Giá trị phiếu giảm giá", "Ngày phát hành:" with a placeholder "dd/mm/yyyy", and "Ngày hết hạn:" with a placeholder "dd/mm/yyyy". Below the inputs are two buttons: a blue "Kiểm tra" button and a red "Quay lại" button.

*) Giao diện thẻ thành viên

The screenshot shows a form titled "Member card". It contains six input fields: "Mã thẻ thành viên:" with a placeholder "Mã thẻ thành viên", "Mã khách hàng:" with a placeholder "Mã khách hàng", "Tên khách hàng:" with a placeholder "Tên Khách hàng", "Số điện thoại:" with a placeholder "Số điện thoại", "Email:" with a placeholder "Email", and "Địa chỉ:" with a placeholder "Địa chỉ". Below the inputs are two buttons: a blue "Kiểm tra" button and a red "Quay lại" button.

Bước 2: Đề xuất lớp giao diện tương ứng

- Trang đăng nhập -> đề xuất lớp giao diện login.html
 - form:
 - action: “/login”
 - inpUsername: text
 - inpPassword: password
 - btnLogin: submit
- Trang chủ của thu ngân -> đề xuất lớp giao diện cashier_home.html
 - btnPayment: link
 - btnSignout: submit
 - cashier: Cashier
- Trang thanh toán -> đề xuất lớp giao diện payment.html

- outIdBill: number
 - outSupermarketName: text
 - outSupermarketAddress: text
 - outIdCashier: number
 - outCashierName: text
 - outPaymentDate: date
 - outMemberCard: text
 - btnUseMemberCard: link
 - outCoupon: text
 - btnUseCoupon: link
 - inpIdProduct: number
 - inpProductQuantity: number
 - outProductName: text
 - outProductSize: number
 - outProductExpiryDate: date
 - outProductPrice: number
 - outAmountMoney: number
 - btnSave: link
 - btnDelete: link
 - tblProduct: table
 - btnRepair: link
 - outTotalBeforeDiscount: number
 - outNumberOfProducts: number
 - outReducesAmount: number
 - outAmountToBePaid: number
 - inpCustomerPaid: number
 - outReturnedCustomer: number
 - btnPayment: submit
 - btnCancel: link
 - bill: Bill
- Trang thẻ thành viên -> đê xuất lớp giao diện member_card.html
 - inpMemberCardCode: text
 - btnCheck: link
 - outIdCustomer: number
 - outCustomerName: text
 - outCustomerPhone: text
 - outCustomerEmail: text

- outCustomerAddress: text
 - btnBack: link
 - btnConfirm: submit
 - bill: Bill
- Trang phiếu giảm giá -> để xuất lớp giao diện coupon.html
 - inpCouponCode: text
 - btnCheck: link
 - outIdCoupon: number
 - outValue: number
 - outReleaseDate: date
 - outOutOfDate: date
 - btnBack: link
 - btnConfirm: submit
 - bill: Bill

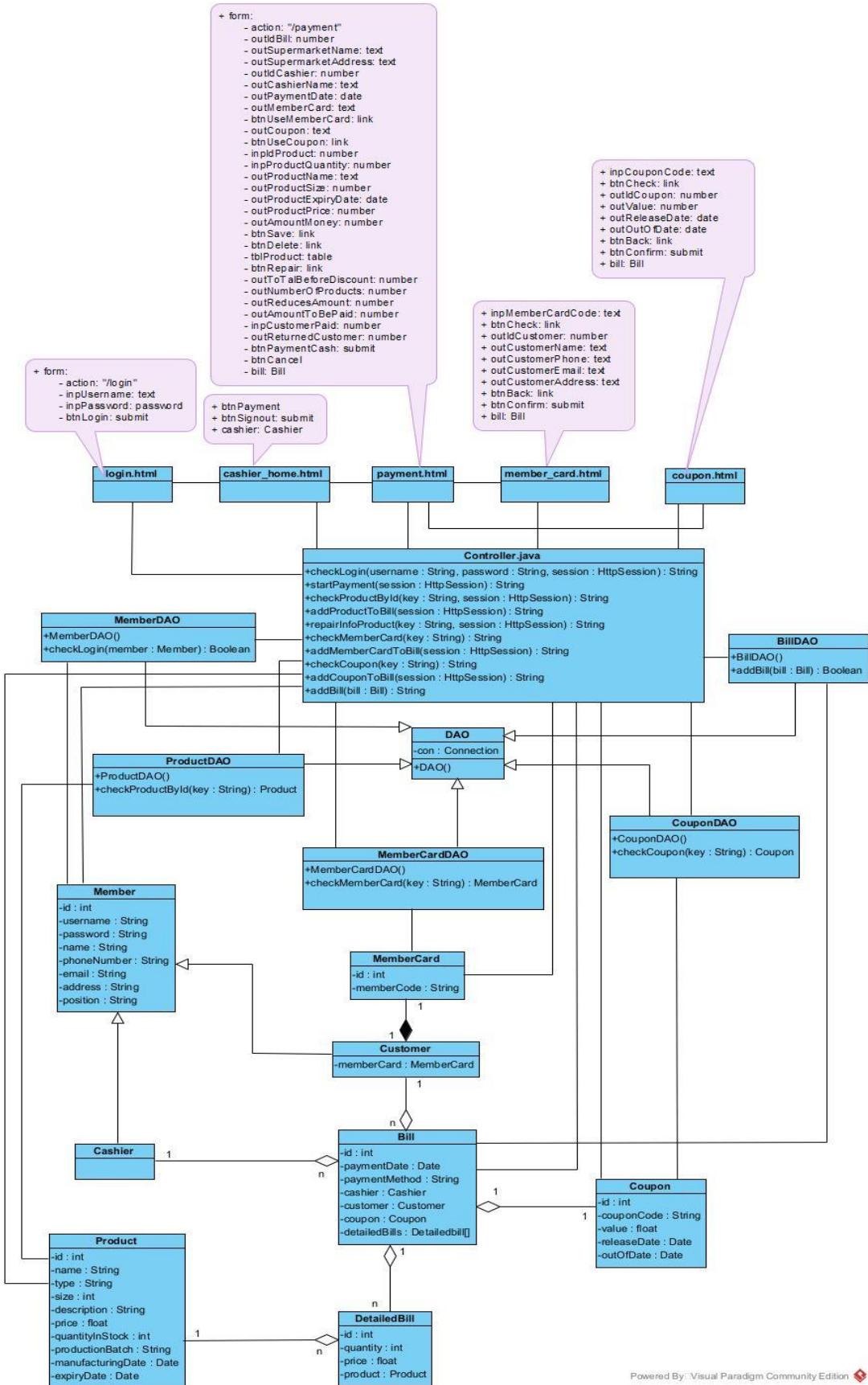
Bước 3: Để xuất lớp truy xuất dữ liệu

- Front End:
 - Controller.java:
 - checkLogin(username: String, password: String, session: HttpSession): String
 - Lấy từ request: username và password
 - Gọi lớp Member để đóng gói
 - Gọi lớp MemberDAO để kiểm tra
 - Nếu đúng:
 - Lưu member vào session
 - Chuyển sang “cashier_home.html”
 - Nếu sai: hiện thông báo sai tài khoản
 - startPayment(session: HttpSession): String
 - Lấy từ session: member
 - Tạo một đối tượng bill và setCashier là member
 - Lưu bill vào session và chuyển sang “payment.html”
 - checkProductById(key: String, session: HttpSession): String
 - Lấy từ request: key
 - Gọi lớp ProductDAO để kiểm tra
 - Nếu tồn tại mặt hàng:
 - Lấy thông tin mặt hàng và lưu vào session

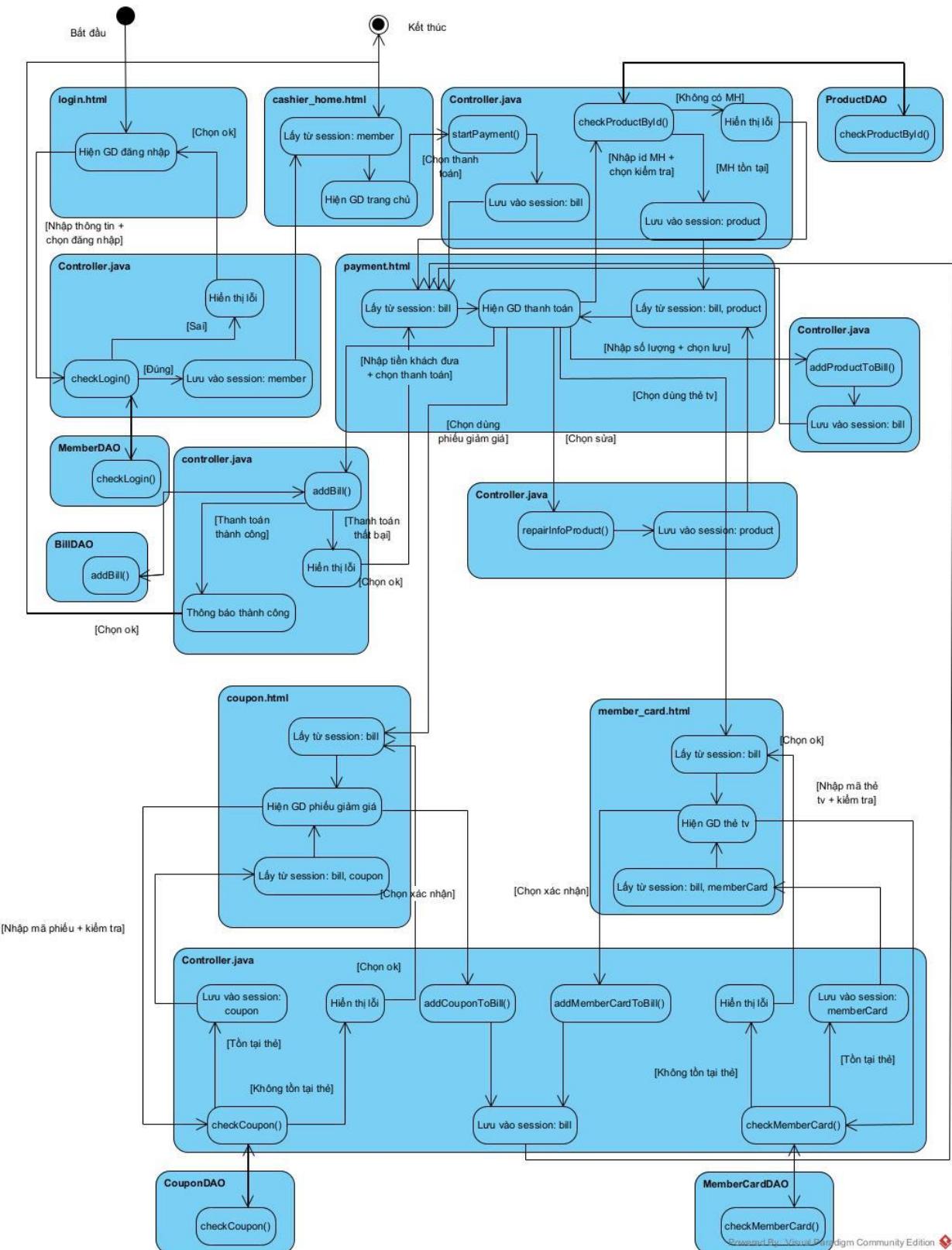
- Hiển thị trên “payment.html”
 - Nếu không có mặt hàng: hiển thị thông báo mặt hàng không tồn tại
- addProductToBill(session: HttpSession): String
 - Lấy từ session: product, bill
 - Thêm product vào bill và lưu lại bill vào session
 - Hiển thị trên “payment.html”
- repairInfoProduct(key: String, session: HttpSession): String
 - Lấy từ request: idProduct
 - Lấy từ session: bill
 - Gọi lớp Product đóng gói mặt hàng có id là idProduct
 - Lưu product vào session
- checkMemberCard(key: String): String
 - Lấy từ request: key
 - Gọi lớp MemberCardDAO để kiểm tra
 - Nếu tồn tại MemberCard:
 - Lấy thông tin thẻ và lưu vào session
 - Hiển thị trên “member_card.html”
 - Nếu không có thẻ: hiển thị thông báo thẻ thành viên không tồn tại
- addMemberCardToBill(session: HttpSession): String
 - Lấy từ session: bill, memberCard
 - Thêm memberCard vào bill và lưu lại bill vào session
 - Quay lại và hiển thị trên “payment.html”
- checkCoupon(key: String): String
 - Lấy từ request: key
 - Gọi lớp MemberCardDAO để kiểm tra
 - Nếu tồn tại Coupon:
 - Lấy thông tin phiếu giảm giá và lưu vào session
 - Hiển thị trên “coupon.html”
 - Nếu không có phiếu giảm giá: hiển thị phiếu giảm giá không tồn tại
- addCouponToBill(session: HttpSession): String
 - Lấy từ session: bill, coupon
 - Thêm coupon vào bill và lưu lại bill vào session

- Quay lại và hiển thị trên “payment.html”
- addBill(bill: Bill): String
 - Lấy từ request: bill
 - Gọi lớp BillDAO để lưu
 - Nếu thêm thành công: quay lại “cashier_home.html”
 - Nếu thêm thất bại: hiển thị thông báo thất bại
- Back End
 - DAO là một lớp chung của DAO. Nó chỉ có cấu trúc để kết nối với DB và cung cấp kết nối chung cho tất cả các lớp DAO kế thừa trong hệ thống.
 - MemberDAO có phương thức:
 - checkLogin(member: Member): Member – kiểm tra và lấy thông tin của thành viên khi đăng nhập.
 - Input: Member
 - Output: Member
 - ProductDAO có phương thức:
 - checkProductById(key: String): Product – kiểm tra và lấy thông tin của mặt hàng.
 - Input: id mặt hàng
 - Output: Product
 - MemberCardDAO có phương thức:
 - checkMemberCard(key: String): MemberCard – kiểm tra và lấy thông tin thẻ thành viên.
 - Input: key
 - Output: MemberCard
 - CouponDAO có phương thức:
 - checkCoupon(key: String): Coupon – kiểm tra và lấy thông tin phiếu giảm giá.
 - Input: key
 - Output: Coupon
 - BillDAO có phương thức:
 - addBill(bill: Bill): Boolean – lưu hóa đơn vào cơ sở dữ liệu.
 - Input: Bill
 - Output: Boolean

Bước 4: Bao gồm tất cả các thực thể liên quan



b, Thiết kế biểu đồ hoạt động

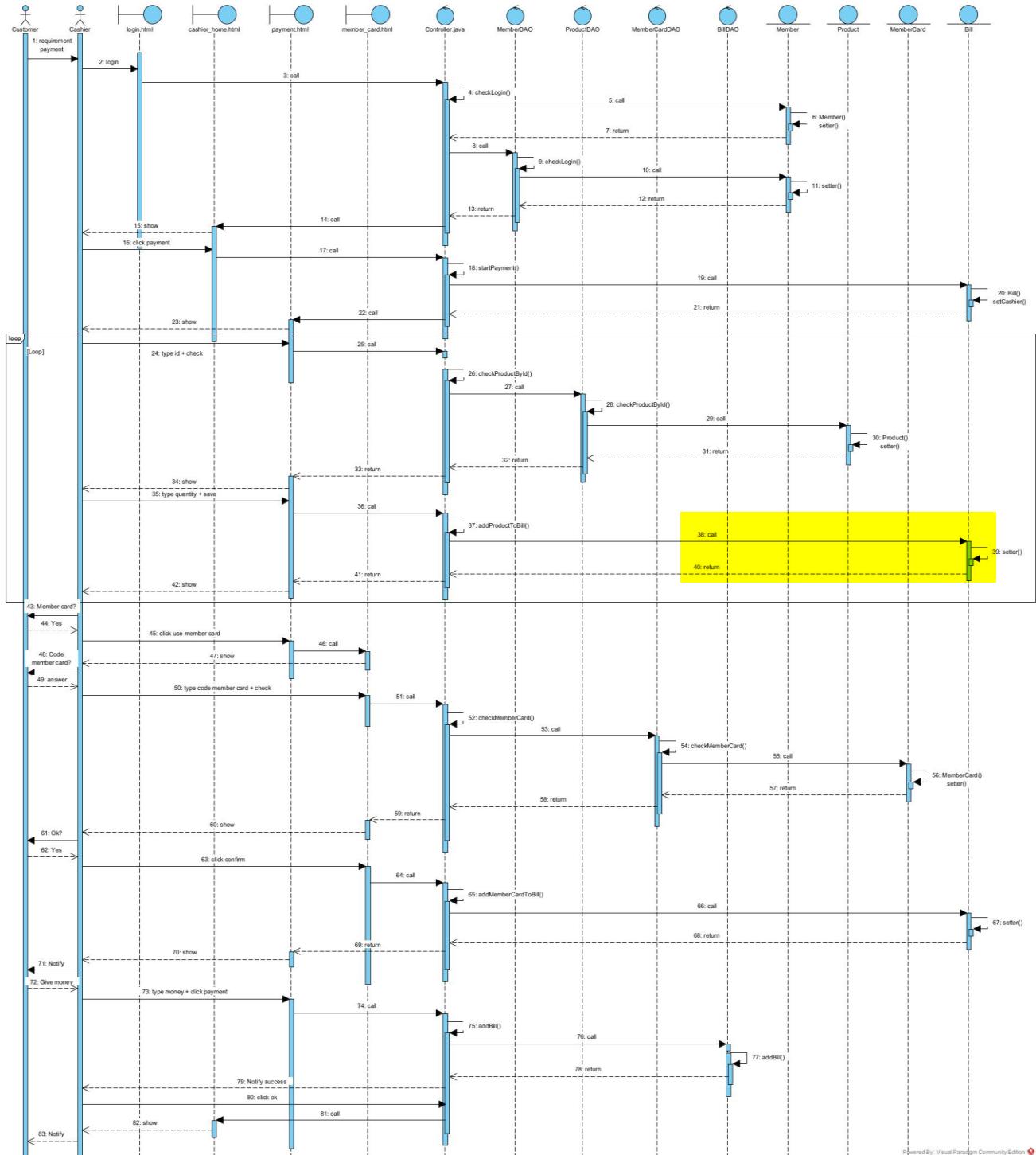


c, Thiết kế biểu đồ tuần tự

1. Khách hàng yêu cầu thanh toán.
2. Tại trang login.html, thu ngân nhập username, password và chọn đăng nhập.
3. Trang login.html gọi lớp Controller.java.
4. Lớp Controller.java gọi hàm checkLogin().
5. Hàm checkLogin() của lớp Controller.java gọi lớp Member đóng gói dữ liệu.
6. Lớp Member khởi tạo, gọi các phương thức setter().
7. Lớp Member trả kết quả cho hàm checkLogin() của lớp Controller.java.
8. Hàm checkLogin() của lớp Controller.java gọi lớp MemberDAO để kiểm tra.
9. Lớp MemberDAO gọi hàm checkLogin().
10. Hàm checkLogin() của lớp MemberDAO gọi lớp Member đóng gói dữ liệu.
11. Lớp Member gọi các phương thức setter().
12. Lớp Member trả kết quả cho hàm checkLogin() của lớp MemberDAO.
13. Hàm checkLogin() của lớp MemberDAO trả kết quả cho hàm checkLogin() của lớp Controller.java.
14. Hàm checkLogin() của lớp Controller.java gọi trang cashier_home.html.
15. Trang cashier_home.html hiển thị cho thu ngân.
16. Thu ngân chọn thanh toán.
17. Trang cashier_home.html gọi lớp Controller.java.
18. Lớp Controller.java gọi hàm startPayment().
19. Hàm startPayment() gọi lớp Bill.
20. Lớp Bill khởi tạo, gọi phương thức setCashier().
21. Lớp Bill trả kết quả cho hàm startPayment().
22. Hàm startPayment() gọi trang payment.html.
23. Trang payment.html hiển thị cho thu ngân.
24. Thu ngân nhập mã MH và chọn kiểm tra.
25. Trang payment.html gọi lớp Controller.java.
26. Lớp Controller.java gọi hàm checkProductId().
27. Hàm checkProductId() của lớp Controller.java gọi lớp ProductDAO.
28. Lớp ProductDAO gọi hàm checkProductId().
29. Hàm checkProductId() của lớp ProductDAO gọi lớp Product đóng gói dữ liệu.
30. Lớp Product khởi tạo, gọi các phương thức setter().
31. Lớp Product trả kết quả cho hàm checkProductId() của lớp ProductDAO.
32. Hàm checkProductId() của lớp ProductDAO trả kết quả cho hàm checkProductId() của lớp Controller.java.

33. Hàm checkProductById() của lớp Controller.java trả kết quả cho trang payment.html.
34. Trang payment.html hiển thị cho thu ngân.
35. Thu ngân nhập số lượng và chọn lưu.
36. Trang payment.html gọi lớp Controller.java.
37. Lớp Controller.java gọi hàm addProductToBill().
38. Hàm addProductToBill() gọi lớp Bill đóng gói dữ liệu.
39. Lớp Bill gọi phương thức setter() tương ứng.
40. Lớp Bill trả kết quả cho hàm addProductToBill().
41. Hàm addProductToBill() trả kết quả cho trang payment.html.
42. Trang payment.html hiển thị cho thu ngân.
43. Thu ngân hỏi khách hàng có thẻ thành viên không.
44. Khách hàng xác nhận có.
45. Thu ngân chọn dùng thẻ thành viên.
46. Trang payment.html gọi trang member_card.html.
47. Trang member_card.html hiển thị cho thu ngân.
48. Thu ngân hỏi mã thẻ thành viên.
49. Khách hàng trả lời.
50. Thu ngân nhập mã thẻ và chọn kiểm tra.
51. Trang member_card.html gọi lớp Controller.java.
52. Lớp Controller.java gọi hàm checkMemberCard().
53. Hàm checkMemberCard() của lớp Controller.java gọi lớp MemberCardDAO.
54. Lớp MemberCardDAO gọi hàm checkMemberCard().
55. Hàm checkMemberCard() của lớp MemberCardDAO gọi lớp MemberCard đóng gói dữ liệu.
56. Lớp MemberCard khởi tạo, gọi các phương thức setter().
57. Lớp MemberCard trả kết quả cho hàm checkMemberCard() của lớp MemberCardDAO.
58. Hàm checkMemberCard() của lớp MemberCardDAO trả kết quả về cho hàm checkMemberCard() của lớp Controller.java.
59. Hàm checkMemberCard() của lớp Controller.java trả về cho trang member_card.html.
60. Trang member_card.html hiển thị cho thu ngân.
61. Thu ngân xác nhận với khách hàng.
62. Khách hàng xác nhận.
63. Thu ngân chọn xác nhận.
64. Trang member_card.html gọi lớp Controller.java.

65. Lớp Controller.java gọi hàm addMemberCardToBill().
66. Hàm addMemberCardToBill gọi lớp Bill đóng gói dữ liệu.
67. Lớp Bill gọi phương thức setter() tương ứng.
68. Lớp Bill trả kết quả cho hàm addMemberCardToBill () .
69. Hàm addMemberCardToBill() trả kết quả cho trang payment.html.
70. Trang payment.html hiển thị cho thu ngân.
71. Thu ngân thông báo với khách hàng số tiền phải trả.
72. Khách hàng đưa tiền.
73. Thu ngân nhập tiền khách đưa và chọn thanh toán.
74. Trang payment.html gọi lớp Controller.java.
75. Lớp Controller.java gọi hàm addBill().
76. Hàm addBill() của lớp Controller.java gọi lớp BillDAO.
77. Lớp BillDAO gọi hàm addBill().
78. Hàm addBill() của lớp BillDAO trả kết quả cho hàm addBill() của lớp Controller.java.
79. Hàm addBill() của lớp Controller.java thông báo thành công cho thu ngân.
80. Thu ngân chọn ok.
81. Hàm addBill() của lớp Controller.java gọi trang cashier_home.html
82. Trang cashier_home.html hiển thị cho thu ngân.
83. Thu ngân thông báo cho khách hàng và trả tiền thừa (nếu có).



5. Xem thống kê khách hàng theo doanh thu

a, Thiết kế biểu đồ lớp đầy đủ

Bước 1: Thiết kế giao diện

*) Giao diện đăng nhập

Đăng Nhập

Tên người dùng:

Mật khẩu:

Đăng Nhập

*) Giao diện trang chủ của quản lý

Manager Home

Quản lý sản phẩm **Xem báo cáo** **Đăng xuất**

*) Giao diện chọn cách thống kê

Select stat

Chọn đối tượng báo cáo

Chọn tiêu chí báo cáo

Xác nhận **Quay lại**

*) Giao diện thống kê khách hàng theo doanh thu

Customer Stat

Ngày bắt đầu:		Ngày kết thúc:				Kiểm tra	Quay lại		
STT	Mã KH	Tên KH	SĐT	Email	Địa chỉ	Tổng tiền đã mua	Số lần mua	Số lượng đã mua	Action
1	1	Kiều	1234	k@gmail.com	Đống đa	50000	5	5	Xem
2	2	Luân	12345	l@gmail.com	Đống đa	50000	5	5	Xem

Tổng doanh thu:

*) Giao diện thống kê chi tiết khách hàng

Customer detail

Mã:	<input type="text" value="Mã Khách hàng"/>	Tên:	<input type="text" value="Tên Khách hàng"/>				
Số điện thoại:	<input type="text" value="Số điện thoại"/>	Email:	<input type="text" value="Email"/>				
Địa chỉ:	<input type="text" value="Địa chỉ"/>	Tổng tiền hàng:	<input type="text"/>				
Tổng số mặt hàng:		<input type="text"/>	Số lần mua hàng:	<input type="text"/>			
STT	Mã HD	Mã TN	Tên siêu thị	Ngày thanh toán	Phương thức thanh toán	Tổng tiền	Action
1	1	1	Hà Đông	01/05/2023	Tiền mặt	5000vnđ	Xem
1	1	1	Hà Đông	01/05/2023	Tiền mặt	5000vnđ	Xem
Quay lại							

*) Giao diện hóa đơn chi tiết

Bill detail

Mã:	<input type="text" value="Mã hóa đơn"/>				
Tên siêu thị:	<input type="text" value="Tên siêu thị"/>	Địa chỉ siêu thị:	<input type="text" value="Địa chỉ siêu thị"/>		
Mã TN:	<input type="text" value="Mã thu ngân"/>	Tên TN:	<input type="text" value="Tên thu ngân"/>		
Số điện thoại KH:	<input type="text" value="Số điện thoại"/>	Ngày thanh toán:	<input type="text" value="dd/mm/yyyy"/>		
STT	Mã MH	Tên MH	Đơn giá	Số lượng	Số tiền
1	1	Snack	5000vnđ	1	5000vnđ
1	1	Snack	5000vnđ	1	5000vnđ
Tổng tiền trước giảm: <input type="text"/>					
Số lượng mặt hàng: <input type="text"/>					
Số tiền được giảm: <input type="text"/>					
Số tiền phải thanh toán: <input type="text"/>					
Phương thức thanh toán: <input type="text"/>					
Quay lại					

Bước 2: Đề xuất lớp giao diện tương ứng

- Trang đăng nhập -> đề xuất lớp giao diện login.html
 - form:
 - action: “/login”
 - inpUsername: text
 - inpPassword: password

- btnLogin: submit
- Trang chủ của quản lí -> để xuất lớp giao diện manager_home.html
 - btnManageProduct: link
 - btnSeeStatistical: link
 - btnSignout: submit
 - manager: Manager
- Trang chọn cách thống kê -> để xuất lớp giao diện select_stat.html
 - reportObject: Select
 - reportCriteria: Select
 - btnBack: link
 - btnConfirm: submit
 - manager: Manager
- Trang thống kê khách hàng theo doanh thu -> để xuất lớp giao diện customer_stat.html
 - inpStartDate: date
 - inpEndDate: date
 - btnCheck: submit
 - btnBack: link
 - outTotalRevenue: number
 - tblCustomerStat: table
 - btnSee: link
 - manager: Manager
- Trang thống kê chi tiết khách hàng -> để xuất lớp giao diện customer_detail.html
 - outIdCustomer: text
 - outCustomerName: text
 - outCustomerPhone: text
 - outCustomerEmail: text
 - outCustomerAddress: text
 - outTotalNumberPurchased: number
 - outTotalNumberProducts: number
 - outTotalAmountOfPurchased: number
 - tblBill: table
 - btnSee: link
 - btnBack: link
 - manager: Manager
 - customer: Customer

- startDate: date
 - endDate: date
- Trang hóa đơn chi tiết -> để xuất lớp giao diện bill_detail.html
 - outIdBill: text
 - outBranchName: text
 - outBranchAddress: text
 - outCustomerPhone: text
 - outIdCashier: text
 - outCashierName: text
 - outPaymentDate: date
 - tblProduct: table
 - outTotalBeforeDiscount: number
 - outNumberOfProducts: number
 - outReducesAmount: number
 - outAmountToBePaid: number
 - outPaymentMethod: text
 - btnBack: link
 - manager: Manager
 - bill: Bill

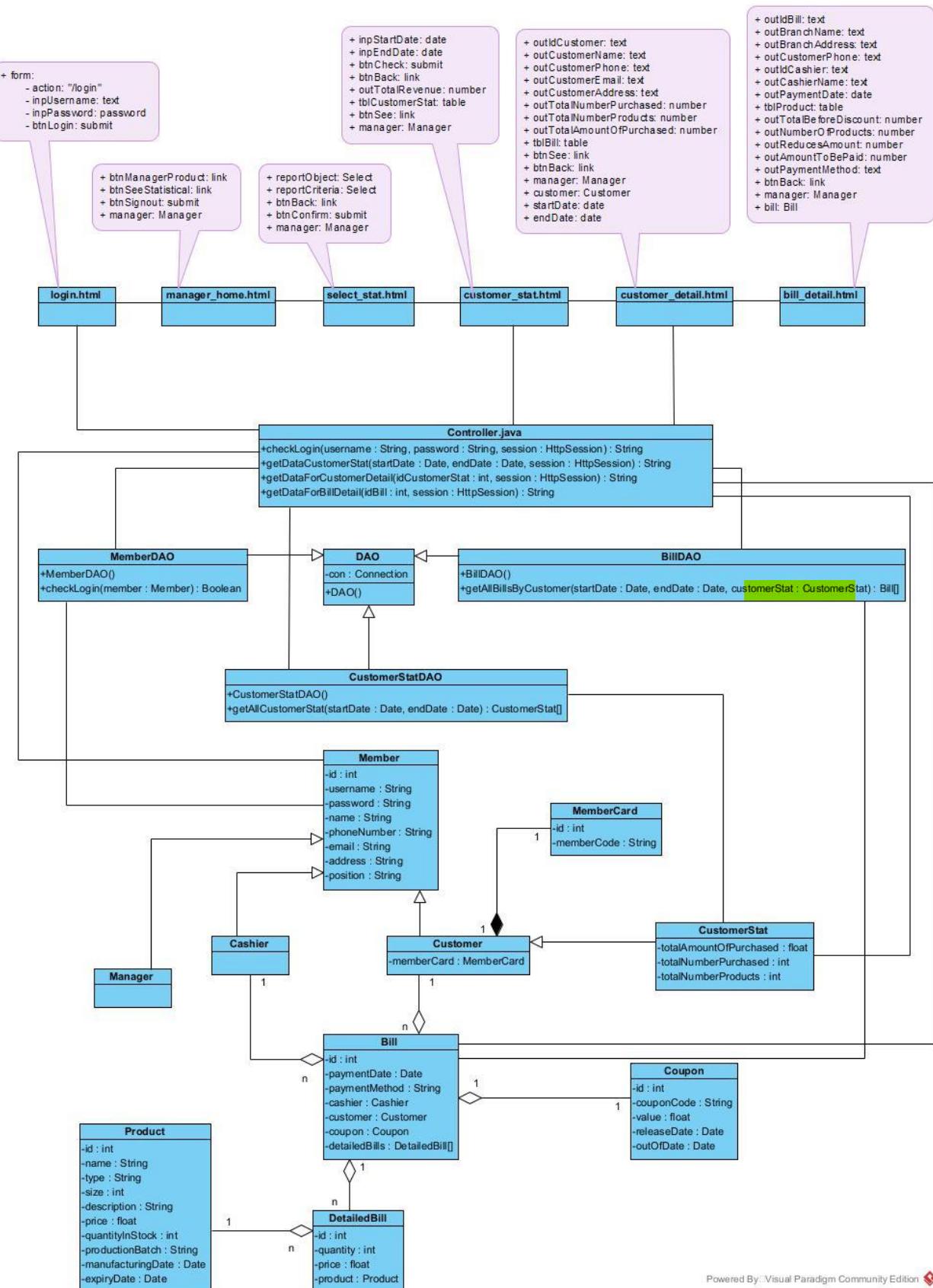
Bước 3: Để xuất lớp truy xuất dữ liệu

- Front End
 - Controller.java:
 - checkLogin(username: String, password: String, session: HttpSession): String
 - Lấy từ request: username và password
 - Gọi lớp Member để đóng gói
 - Gọi MemberDAO để kiểm tra
 - Nếu đúng:
 - Lưu member vào session
 - Chuyển sang “manager_home.html”
 - Nếu sai: hiện thông báo sai tài khoản
 - getDataForCustomerStat(startDate: Date, endDate: Date, session: HttpSession): String
 - Lấy từ request: startDate, endDate
 - Lấy từ session: member

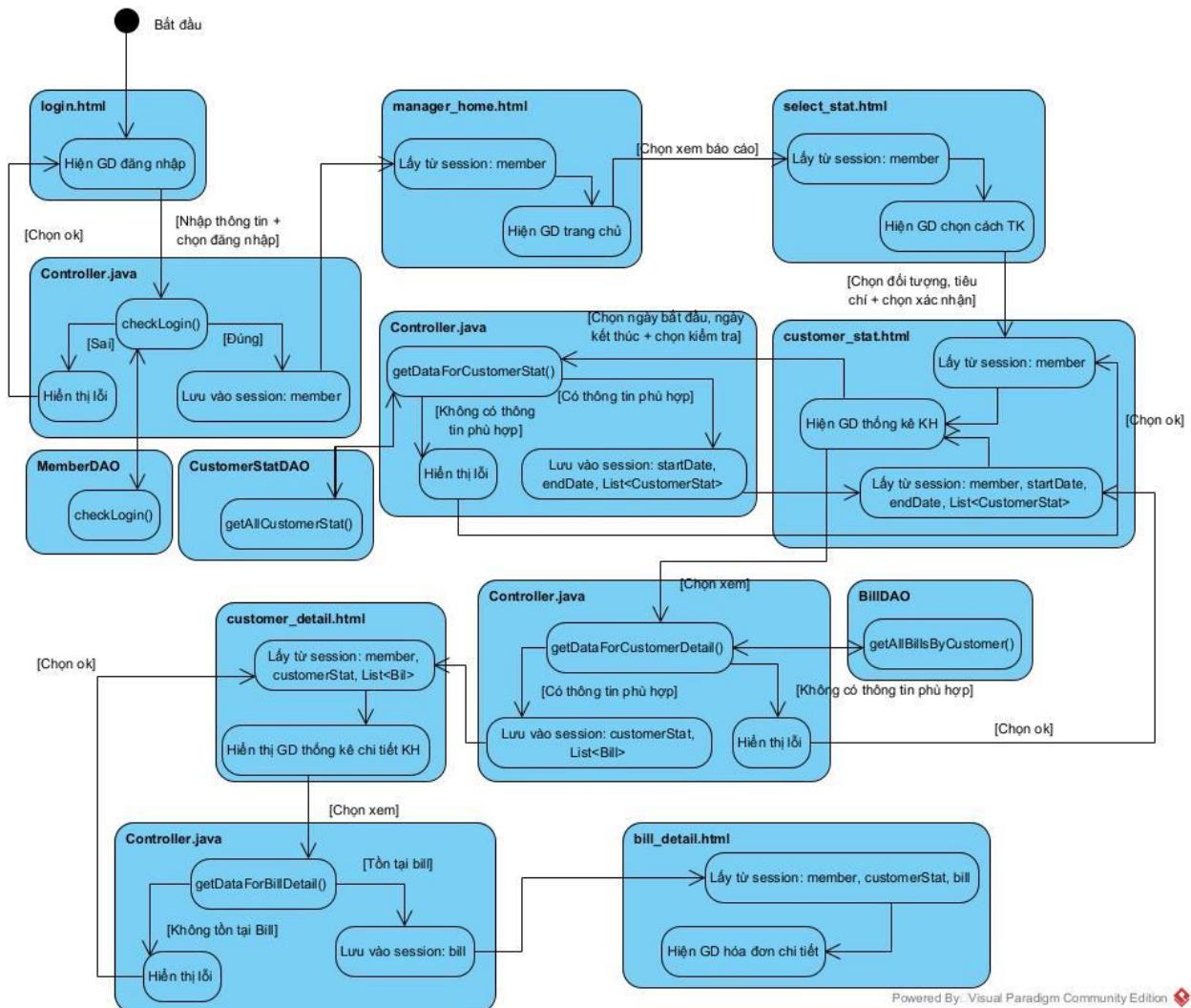
- Lưu startDate, endDate vào session
- Gọi lớp CustomerStatDAO để lấy dữ liệu bằng phương thức getAllCustomerStat()
 - Nếu tồn tại khách hàng:
 - Lưu danh sách thống kê khách hàng vào session
 - Hiển thị trên “customer_stat.html”
 - Nếu không có khách hàng: hiển thị thông báo lỗi.
- getDataForCustomerDetail(idCustomerStat: int, session: HttpSession): String
 - Lấy từ request: idCustomerStat
 - Lấy từ session: member, listCustomerStat, startDate, endDate
 - Gọi lớp CustomerStat để đóng gói dữ liệu cho customerStat có id là idCustomerStat và lưu vào session
 - Gọi lớp BillDAO để lấy dữ liệu
 - Nếu tồn tại hóa đơn:
 - Lưu danh sách hóa đơn vào session
 - Hiển thị trên “customer_detail.html”
 - Nếu không có hóa đơn: hiển thị thông báo không có.
- getDataForBillDetail(idBill: int, session: HttpSession): String
 - Lấy từ request: idBill
 - Lấy từ session: member, customerStat, listBill
 - Gọi lớp Bill để đóng gói dữ liệu cho bill có id là idBill và lưu vào session
 - Hiển thị trên “bill_detail.html”
- Back End
 - DAO là một lớp chung của DAO. Nó chỉ có cấu trúc để kết nối với DB và cung cấp kết nối chung cho tất cả các lớp DAO kế thừa trong hệ thống.
 - MemberDAO có phương thức:
 - checkLogin(member: Member): Member – kiểm tra và lấy thông tin thành viên khi đăng nhập.
 - Input: Member
 - Output: Member
 - CustomerStatDAO có phương thức:

- getAllCustomerStat(startDate: Date, endDate: Date): CustomerStat[] – lấy toàn bộ thông tin khách hàng theo điều kiện.
 - Input: startDate, endDate
 - Output: CustomerStat[]
- BillDAO có phương thức:
 - getAllBillsByCustomer(startDate: Date, endDate: Date, customerStat: CustomerStat): Bill[] – lấy toàn bộ bill theo điều kiện.
 - Input: startDate, endDate, customerStat
 - Output: Bill[]

Bước 4: Bao gồm tất cả các thực thể liên quan



b, Thiết kế biểu đồ hoạt động



c, Thiết kế biểu đồ tuần tự

1. Tại trang login.html, quản lý nhập username, password và chọn đăng nhập.
2. Trang login.html gọi lớp Controller.java.
3. Lớp Controller.java gọi hàm checkLogin().
4. Hàm checkLogin() của lớp Controller.java gọi lớp Member đóng gói dữ liệu.
5. Lớp Member khởi tạo, gọi các phương thức setter().
6. Lớp Member trả kết quả cho hàm checkLogin() của lớp Controller.java.
7. Hàm checkLogin() của lớp Controller.java gọi lớp MemberDAO để kiểm tra.
8. Lớp MemberDAO gọi hàm checkLogin().
9. Hàm checkLogin() của lớp MemberDAO gọi lớp Member đóng gói dữ liệu.
10. Lớp Member gọi các phương thức setter().
11. Lớp Member trả kết quả cho hàm checkLogin() của lớp MemberDAO.

12. Hàm checkLogin() của lớp MemberDAO trả kết quả cho hàm checkLogin() của lớp Controller.java.
13. Hàm checkLogin() của lớp Controller.java gọi trang manager_home.html.
14. Trang manager_home.html hiển thị cho quản lí.
15. Quản lí chọn chức năng xem báo cáo thống kê.
16. Trang manager_home.html gọi trang select_stat.html.
17. Trang select_stat.html hiển thị cho quản lí.
18. Quản lí chọn đối tượng, tiêu chí và chọn xác nhận.
19. Trang select_stat.html gọi trang customer_stat.html.
20. Trang customer_stat.html hiển thị cho quản lí.
21. Quản lí chọn ngày bắt đầu, ngày kết thúc và chọn kiểm tra.
22. Trang customer_stat.html gọi lớp Controller.java.
23. Lớp Controller.java gọi hàm getDataForCustomerStat().
24. Hàm getDataForCustomerStat() gọi lớp CustomerStatDAO.
25. Lớp CustomerStatDAO gọi hàm getAllCustomerStat().
26. Hàm getAllCustomerStat() gọi lớp CustomerStat đóng gói dữ liệu.
27. Lớp CustomerStat khởi tạo, gọi các phương thức setter().
28. Lớp CustomerStat trả kết quả cho hàm getAllCustomerStat().
29. Hàm getAllCustomerStat() trả kết quả cho hàm getDataForCustomerStat().
30. Hàm getDataForCustomerStat() trả kết quả cho trang customer_stat.html.
31. Trang customer_stat.html hiển thị cho quản lí.
32. Quản lí chọn xem một khách hàng.
33. Trang customer_stat.html gọi lớp Controller.java.
34. Lớp Controller.java gọi hàm getDataForCustomerDetail().
35. Hàm getDataForCustomerDetail() gọi lớp BillDAO.
36. Lớp BillDAO gọi hàm getAllBillsByCustomer().
37. Hàm getAllBillsByCustomer() gọi lớp Bill đóng gói dữ liệu.
38. Lớp Bill khởi tạo, gọi các phương thức setter().
39. Lớp Bill trả kết quả cho hàm getAllBillsByCustomer().
40. Hàm getAllBillsByCustomer() trả kết quả cho hàm getDataForCustomerDetail().
41. Hàm getDataForCustomerDetail() gọi trang customer_detail.html.
42. Trang customer_detail.html hiển thị cho quản lí.
43. Quản lí chọn xem một hóa đơn.
44. Trang customer_detail.html gọi lớp Controller.java.
45. Lớp Controller.java gọi hàm getDataForBillDetail().
46. Hàm getDataForBillDetail() gọi trang bill_detail.html.

47. Trang bill_detail.html hiển thị cho quản lí.

