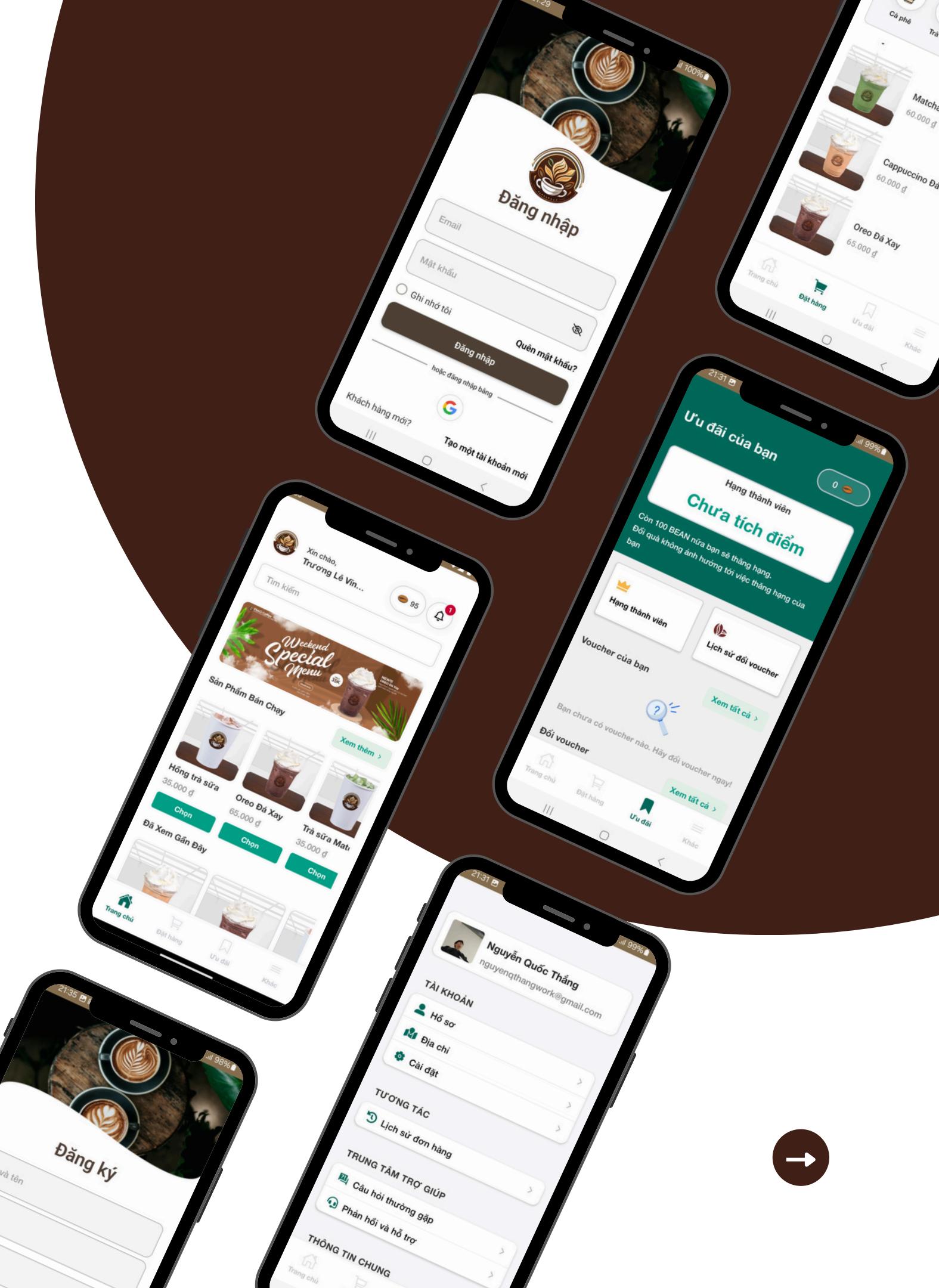




FiveD Coffee

ỨNG DỤNG QUẢN LÝ COFFEE SHOPS

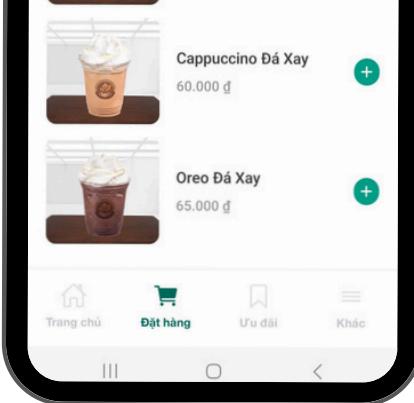
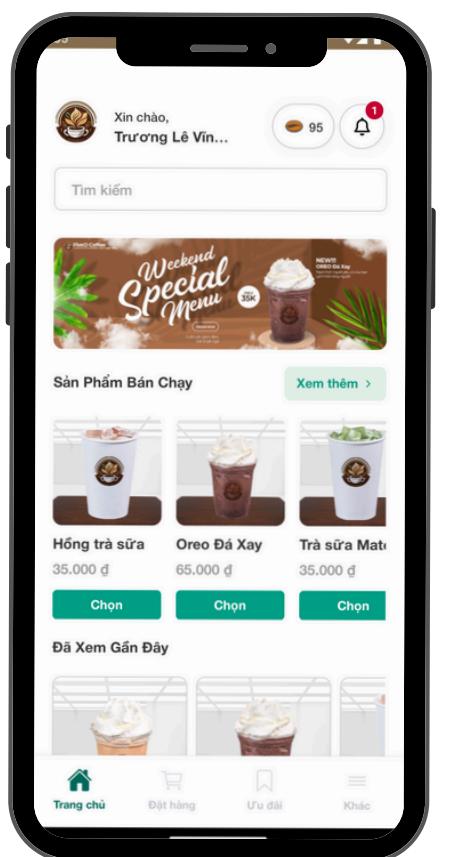


THÀNH VIÊN NHÓM

22521337 - Nguyễn Quốc Thắng
22521282 - Nguyễn Thành Tài
22520103 - Huỳnh Gia Bảo
22521314 - Trần Tuệ Tánh
22521145 - Trương Lê Vĩnh Phúc

GIÁO VIÊN HƯỚNG DẪN

GV LT: TS. Nguyễn Thị Xuân Hương
GV HDTH: ThS. Đỗ Thị Thanh Tuyền



MỤC LỤC

01

Khảo sát
bài toán

02

Đặc tả yêu cầu
bài toán

03

Phân tích
yêu cầu

04

Thiết kế
ứng dụng

05

Xây dựng
chương trình

06

Kiểm thử
chương trình

07

Hướng dẫn
cài đặt và
sử dụng

08

Kết luận

09

Giới thiệu
ứng dụng



KHẢO SÁT BÀI TOÁN

KHẢO SÁT BÀI TOÁN

Dự án xây dựng ứng dụng di động quản lý quán cà phê nhằm mục tiêu tối ưu hóa các hoạt động vận hành, nâng cao trải nghiệm khách hàng và tăng doanh thu. Thông qua quá trình khảo sát thực tế chúng em có được các thách thức và giải pháp cho mô hình kinh doanh quản lý quán cà phê.



THÁCH THỨC

QUẢN LÝ ĐƠN HÀNG VÀ THANH TOÁN

01

Dễ xảy ra sai sót
trong quá trình ghi
nhận và tính toán.

02

Thời gian xử lý
đơn hàng và
thanh toán lâu.

03

Khó kiểm soát và
theo dõi doanh
thu hàng ngày.



THÁCH THỨC

QUẢN LÝ KHO VÀ NGUYÊN LIỆU

01

Quản lý thủ công dễ gây ra thiếu hụt hoặc thừa nguyên liệu.

02

Khó khăn trong việc theo dõi tồn kho chính xác và thời gian hết hạn của nguyên liệu.



THÁCH THỨC

QUẢN LÝ KHÁCH HÀNG

01

Dễ xảy ra sai sót
trong quá trình ghi
nhận và tính toán.

02

Thiếu hệ thống
quản lý chương
trình khuyến mãi
và tích điểm cho
khách hàng.

03

Không có phương
tiện để thu thập
và phân tích phản
hồi từ khách
hàng.



THÁCH THỨC

QUẢN LÝ KHO VÀ NGUYÊN LIỆU

01

Khó khăn trong việc sắp xếp lịch làm việc phù hợp cho tất cả nhân viên.

02

Thiếu hệ thống theo dõi hiệu suất và sự hiện diện của nhân viên.





GIẢI PHÁP - XÂY DỰNG HỆ THỐNG

Tự động hóa và chính xác trong quản lý đơn hàng và thanh toán:

- Đơn hàng được ghi nhận tự động và chính xác.
- Thanh toán nhanh chóng và chính xác hơn, giảm thiểu sai sót.
- Dễ dàng theo dõi và kiểm soát doanh thu hàng ngày.

Hiệu quả trong quản lý kho và nguyên liệu:

- Đơn hàng được ghi nhận tự động và chính xác.
- Thanh toán nhanh chóng và chính xác hơn, giảm thiểu sai sót.
- Dễ dàng theo dõi và kiểm soát doanh thu hàng ngày.



GIẢI PHÁP - XÂY DỰNG HỆ THỐNG

Quản lý nhân viên hiệu quả

- Lên lịch làm việc tự động và hợp lý cho nhân viên.
- Theo dõi hiệu suất và sự hiện diện của nhân viên.

Tối ưu hóa quản lý khách hàng

- Quản lý thông tin khách hàng thân thiết chính xác và hệ thống.
- Hệ thống quản lý chương trình khuyến mãi và tích điểm tự động.
- Thu thập và phân tích phản hồi từ khách hàng để cải thiện dịch vụ.

Các chức năng chính



Đặt hàng trực tuyến
Tăng tốc độ phục vụ, tiết kiệm thời gian cho khách hàng, tăng doanh thu, thu hút khách hàng mới.



Quản lý đơn hàng
Giúp nhân viên quản lý đơn hàng dễ dàng, giảm thiểu lỗi, tăng hiệu quả hoạt động, tạo sự minh bạch cho khách hàng.



Chương trình khuyến mãi
Thu hút khách hàng, tăng doanh thu, tạo sự cạnh tranh với các quán cà phê khác.



Quản lý kho hàng
Kiểm soát tốt nguyên vật liệu, giảm thiểu lãng phí, tối ưu hóa chi phí, đảm bảo cung ứng đủ hàng hóa cho hoạt động kinh doanh.



Quản lý nhân viên
Tăng hiệu quả quản lý nhân sự, đảm bảo công bằng trong phân chia công việc, tạo sự chuyên nghiệp cho quán cà phê.



Quản lý chi nhánh
Thống kê hiệu quả hoạt động của từng chi nhánh, dễ dàng theo dõi và quản lý chuỗi quán cà phê.

ĐẶC TẢ YÊU CẦU

BÀI TOÁN

MÔ TẢ YÊU CẦU BÀI TOÁN

Một quán cà phê muốn triển khai một ứng dụng quản lý để tối ưu hóa các hoạt động hàng ngày của mình và ứng dụng cho phép khách hàng đặt hàng từ quán một cách tiện dụng. Ứng dụng này cần có khả năng quản lý các thông tin về đơn đặt hàng, nhân viên, kho nguyên liệu, doanh thu, và các thông tin khác liên quan đến hoạt động kinh doanh của quán cà phê. Đồng thời, ứng dụng cho phép khách hàng thực hiện đặt hàng, áp dụng ưu đãi, tích điểm và thanh toán đơn hàng của quán.





DANH SÁCH CÁC YÊU CẦU

- Lập danh sách các sản phẩm
- Lập danh sách khuyến mãi
- Tra cứu hóa đơn
- Lập danh sách nhân viên
- Lập danh sách nguyên vật liệu
- Lập danh sách khách hàng
- Lập danh sách ca làm việc
- Tiếp nhận đơn hàng
- Lập báo cáo kinh doanh
- Lập danh sách các chi nhánh

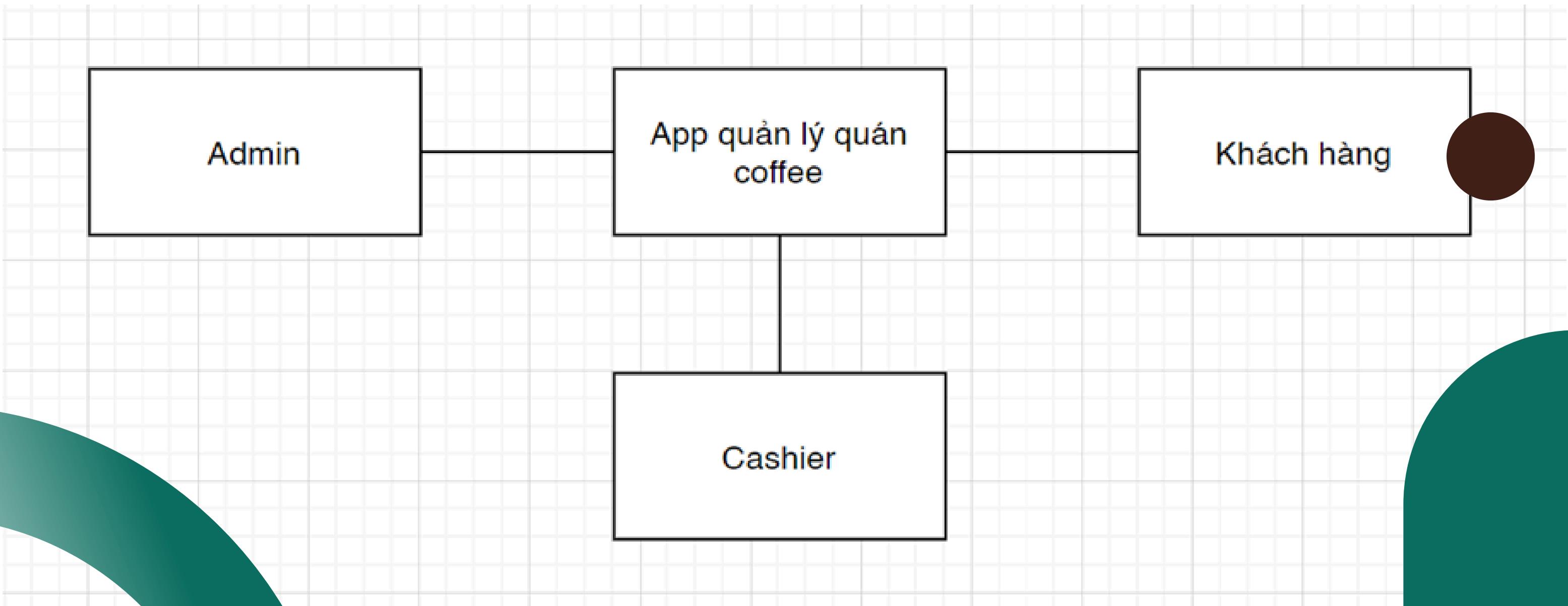
PHÂN TÍCH YÊU CẦU

CÁC TÁC NHÂN CỦA HỆ THỐNG

- User

- Admin

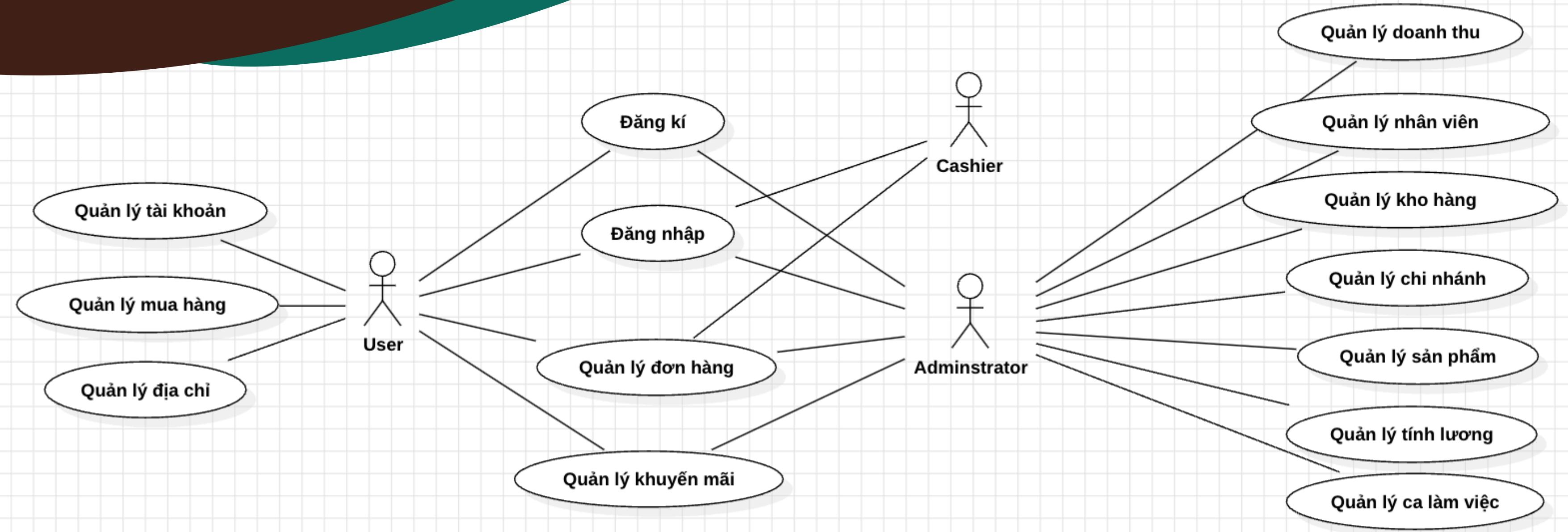
- Cashier



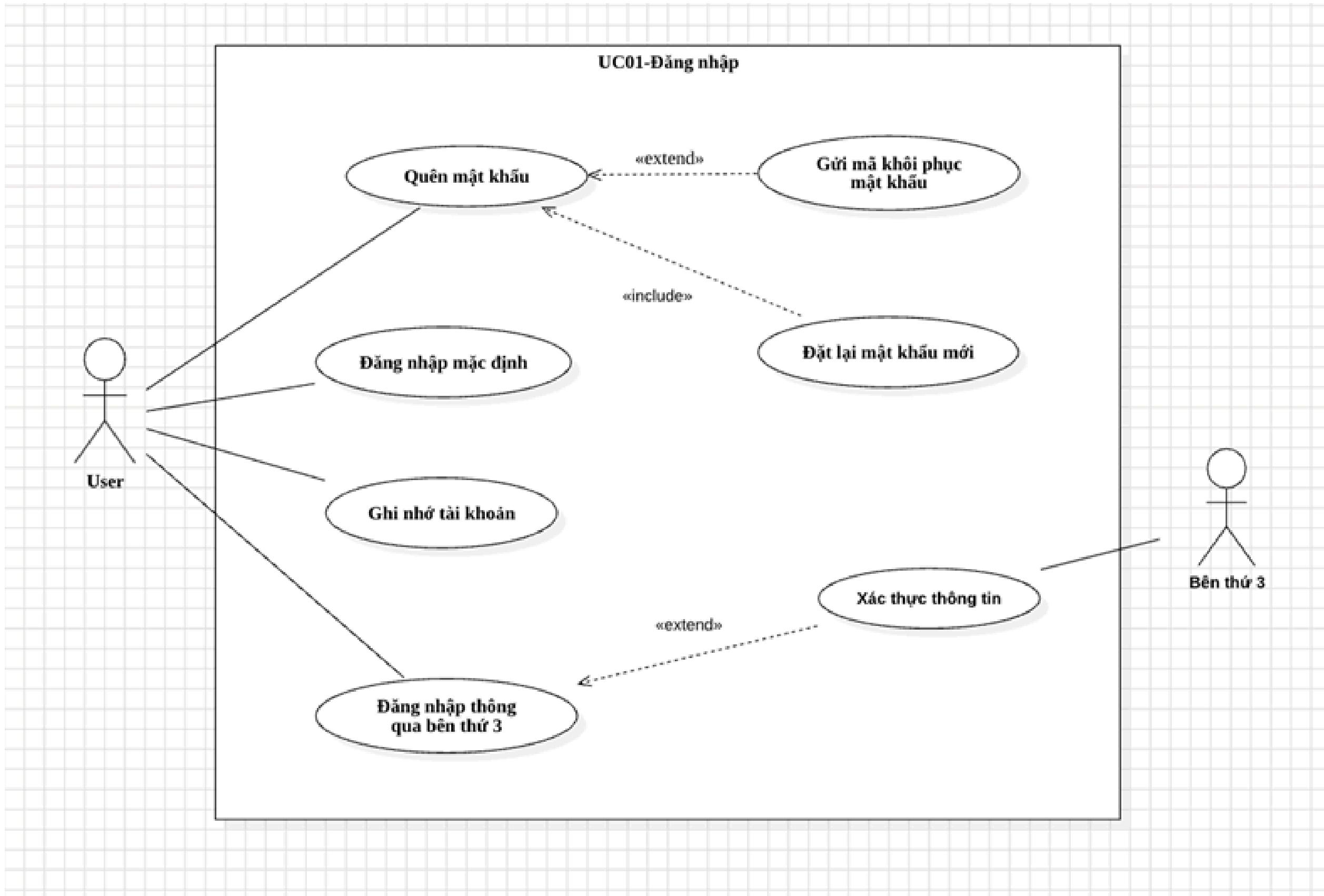
CÁC USE-CASE CỦA HỆ THỐNG

- Đăng nhập
- Đăng kí
- Quản lý tài khoản
- Quản lý đơn hàng
- Quản lý sản phẩm
- Quản lý mua hàng
- Quản lý khuyến mãi.
- Quản lý doanh thu
- Quản lý nhân viên
- Quản lý lịch biểu
- Quản lý bảng tính lương
- Quản lý kho hàng
- Quản lý chi nhánh
- Quản lý địa chỉ

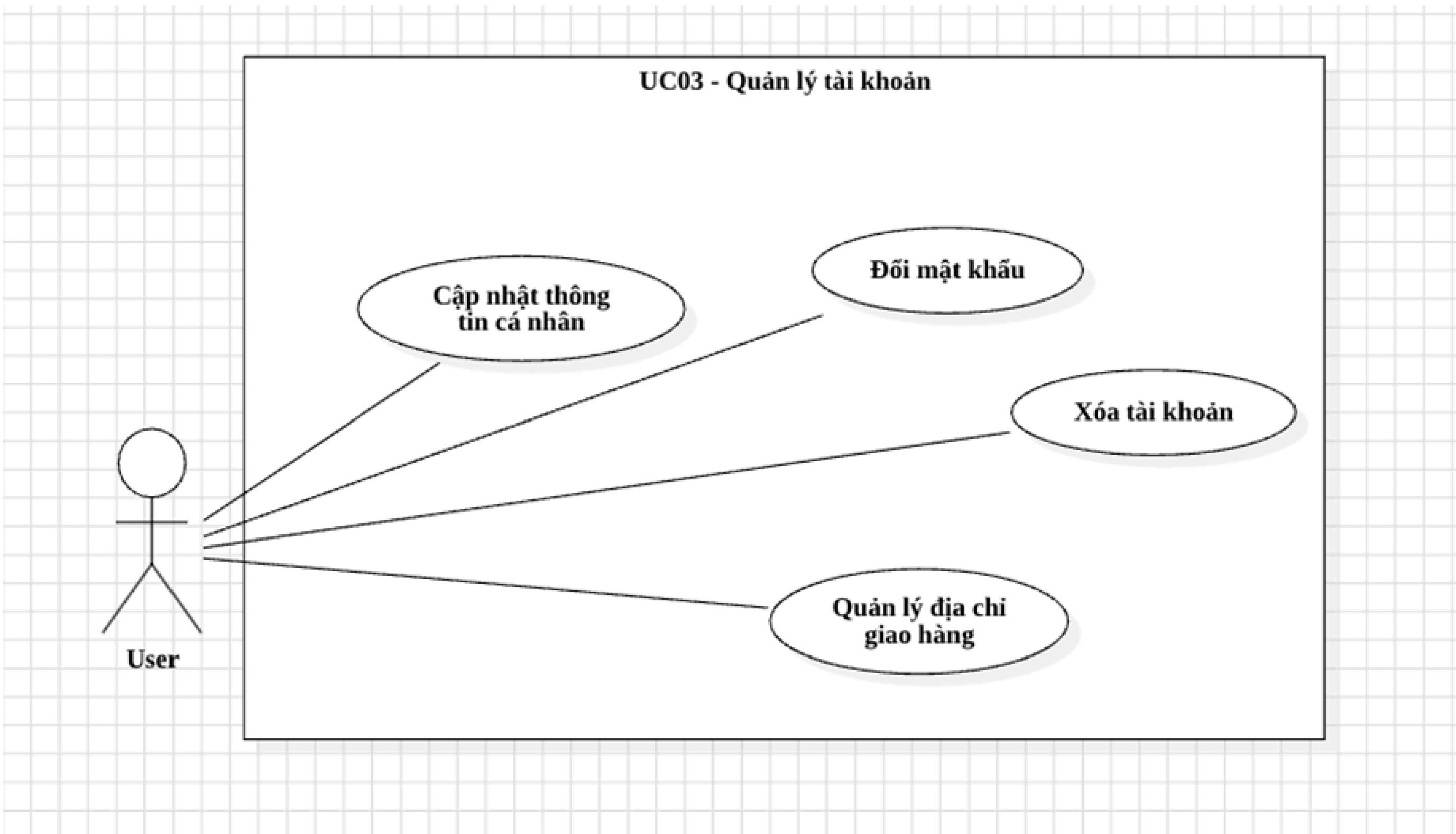




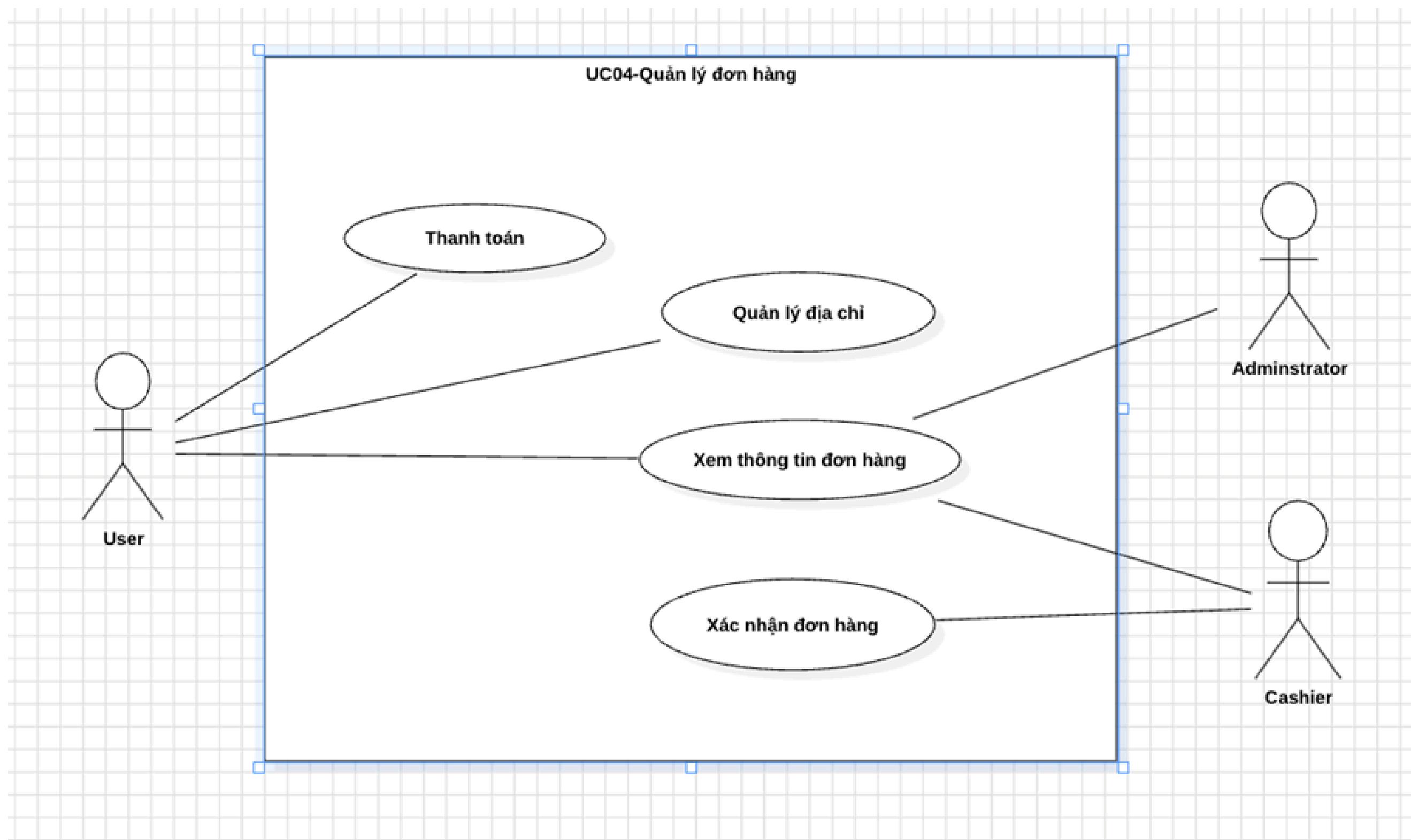
USE CASE ĐĂNG NHẬP



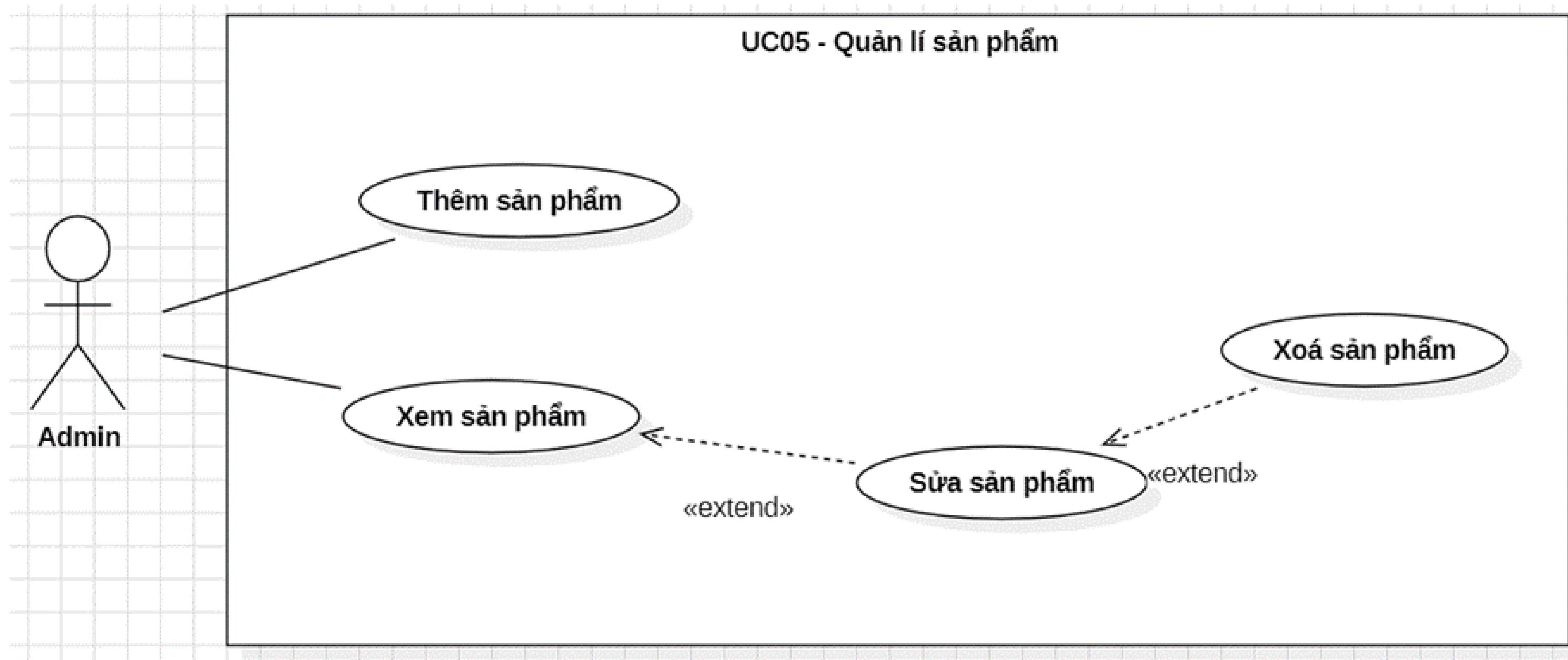
USE CASE QUẢN LÝ TÀI KHOẢN



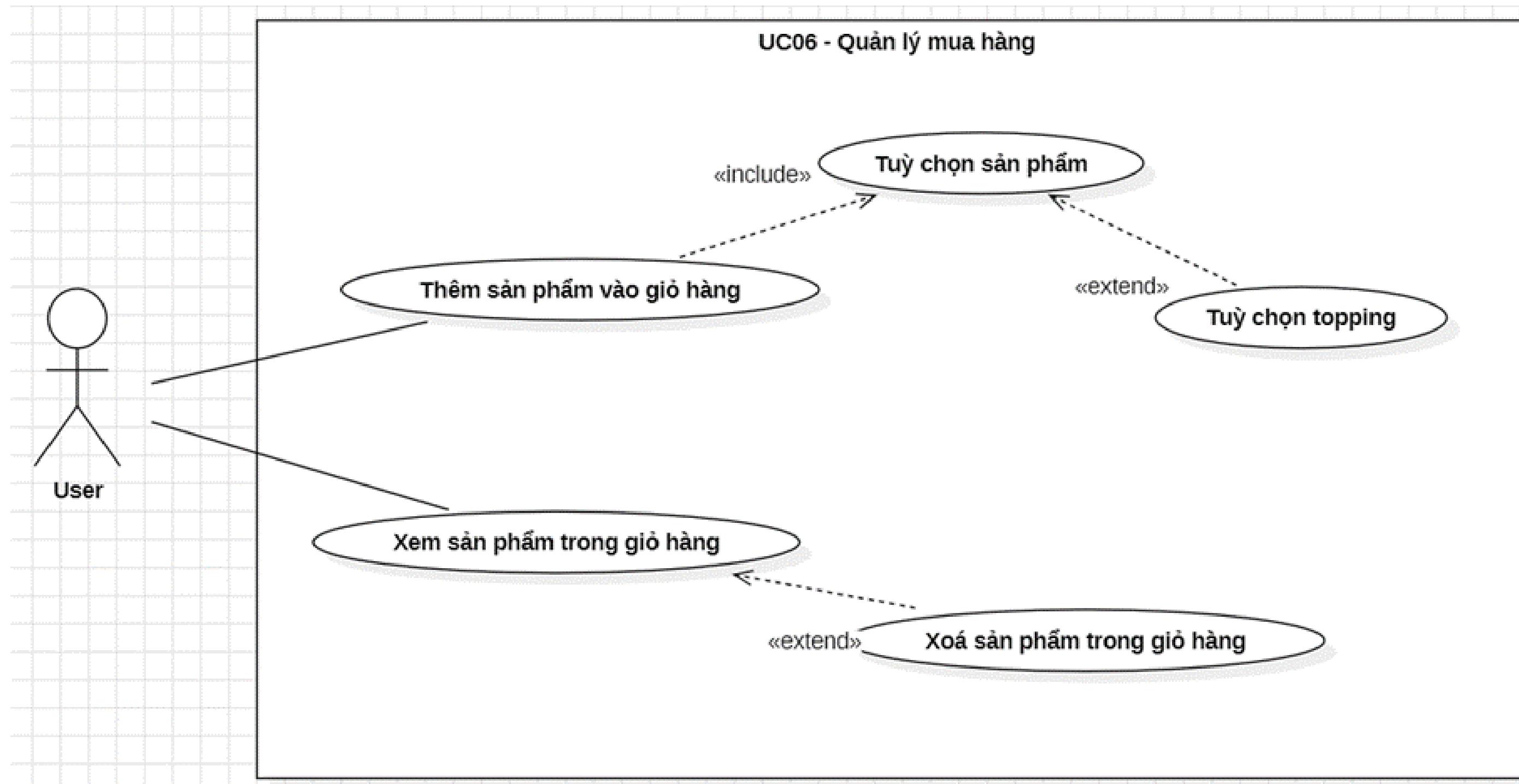
USE CASE QUẢN LÝ ĐƠN HÀNG



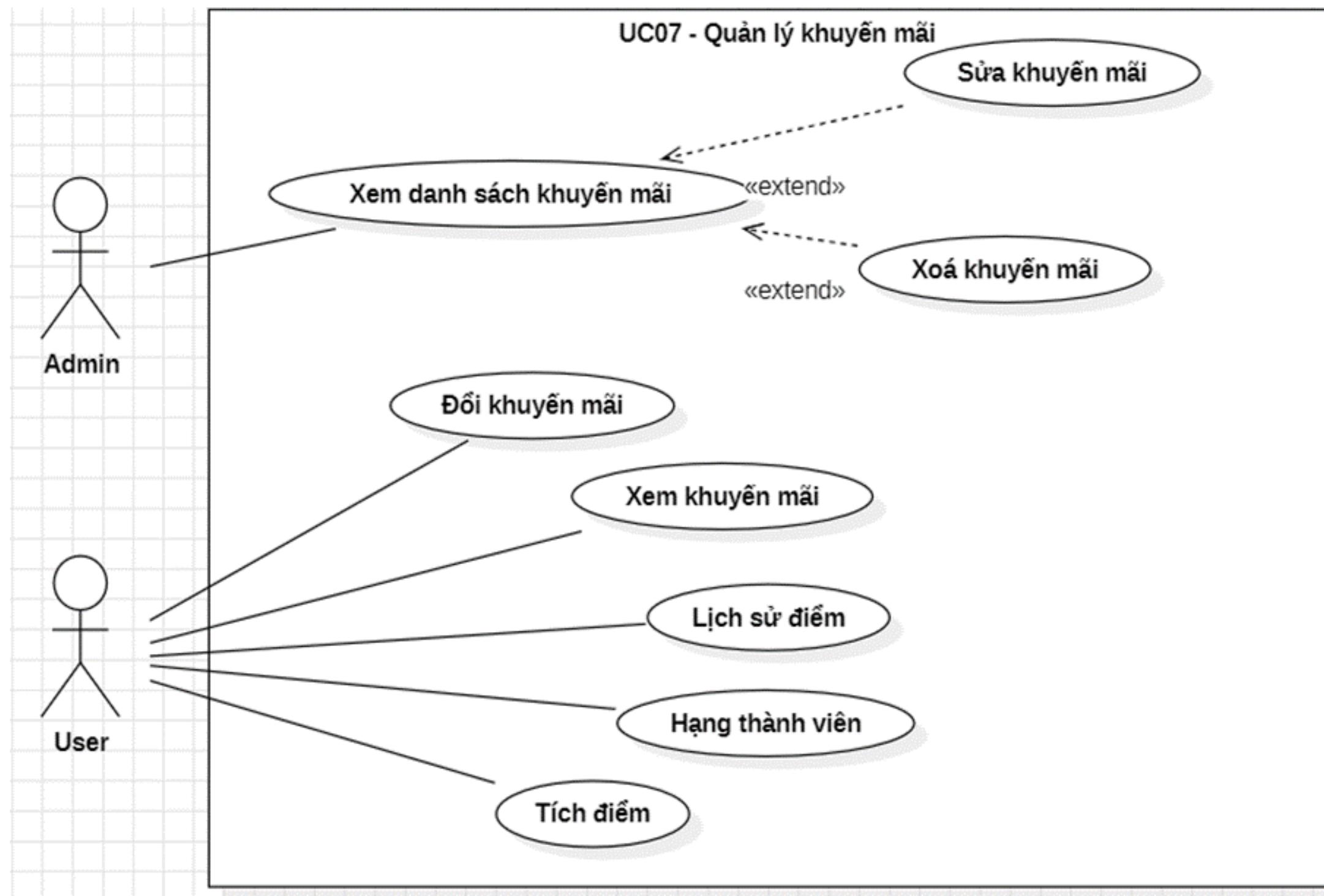
USE CASE QUẢN LÝ SẢN PHẨM



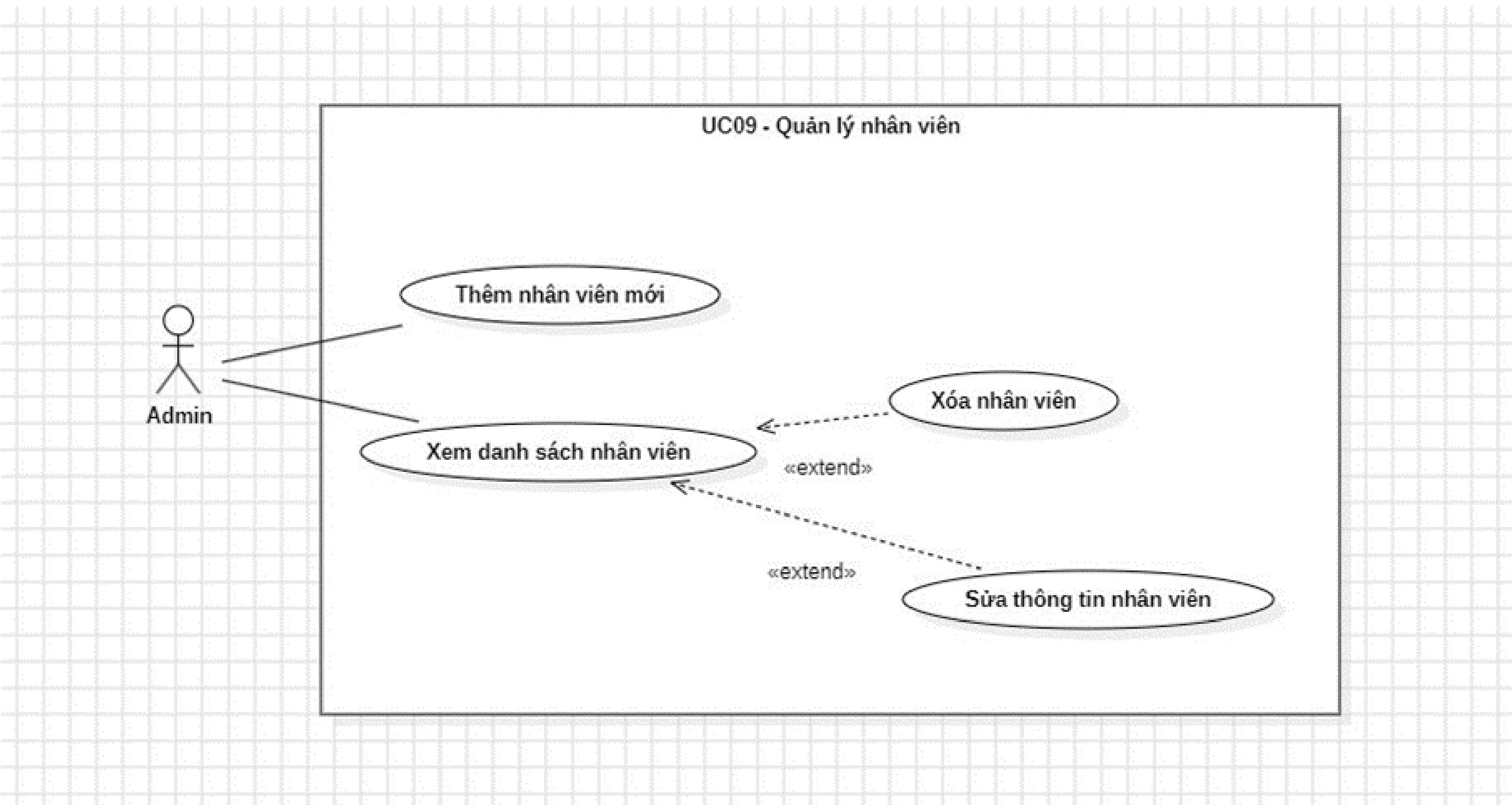
USE CASE QUẢN LÝ MUA HÀNG



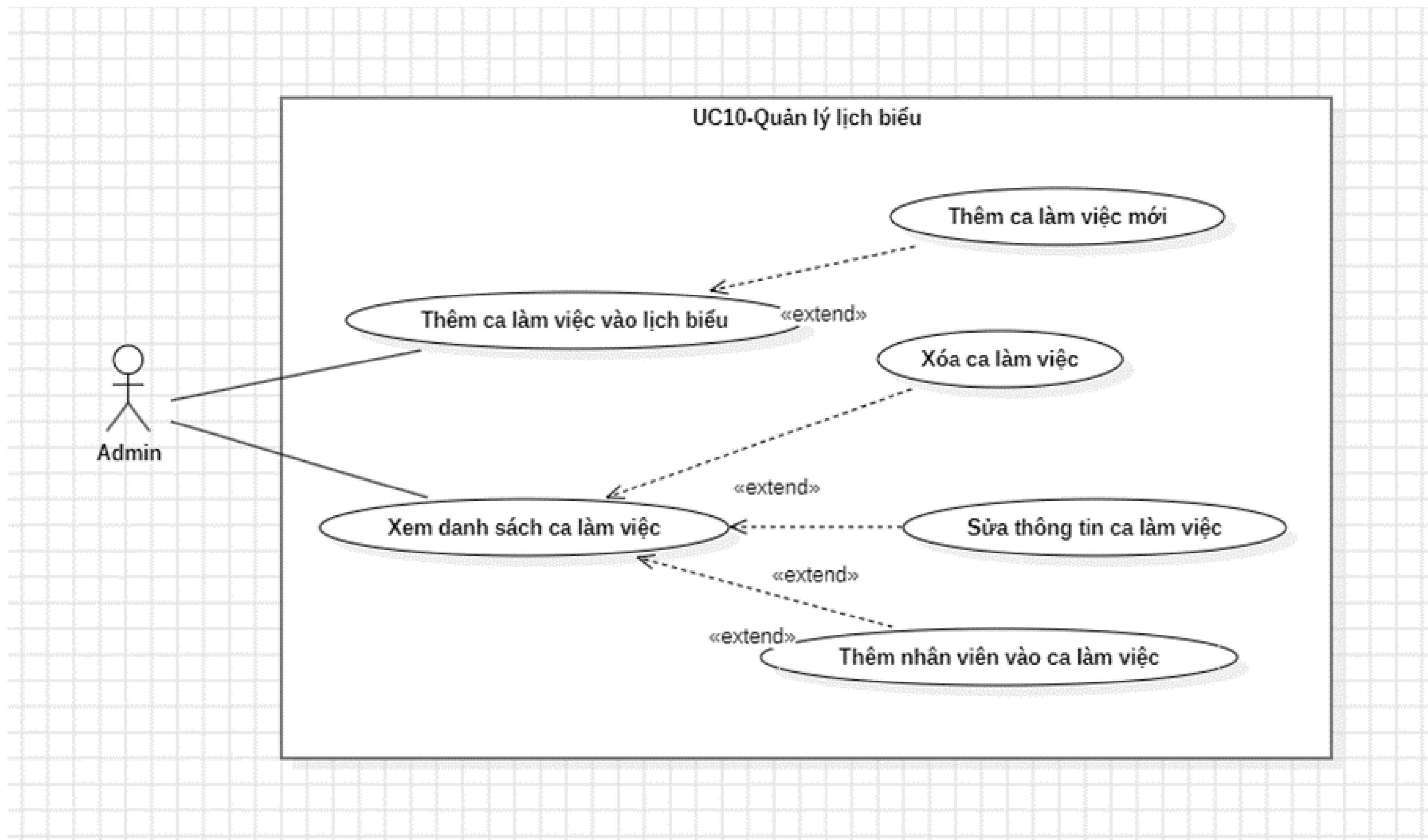
USE CASE QUẢN LÝ KHUYẾN MÃI



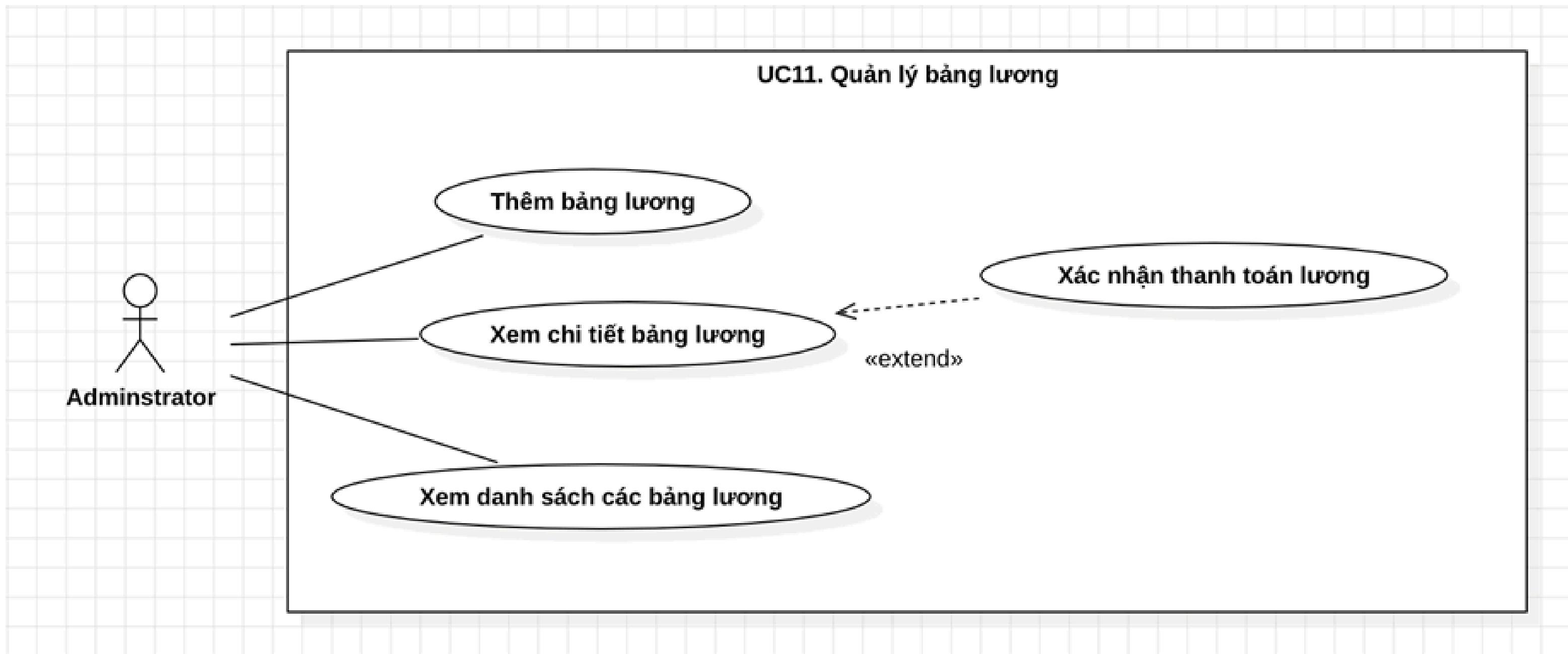
USE CASE QUẢN LÝ NHÂN VIÊN



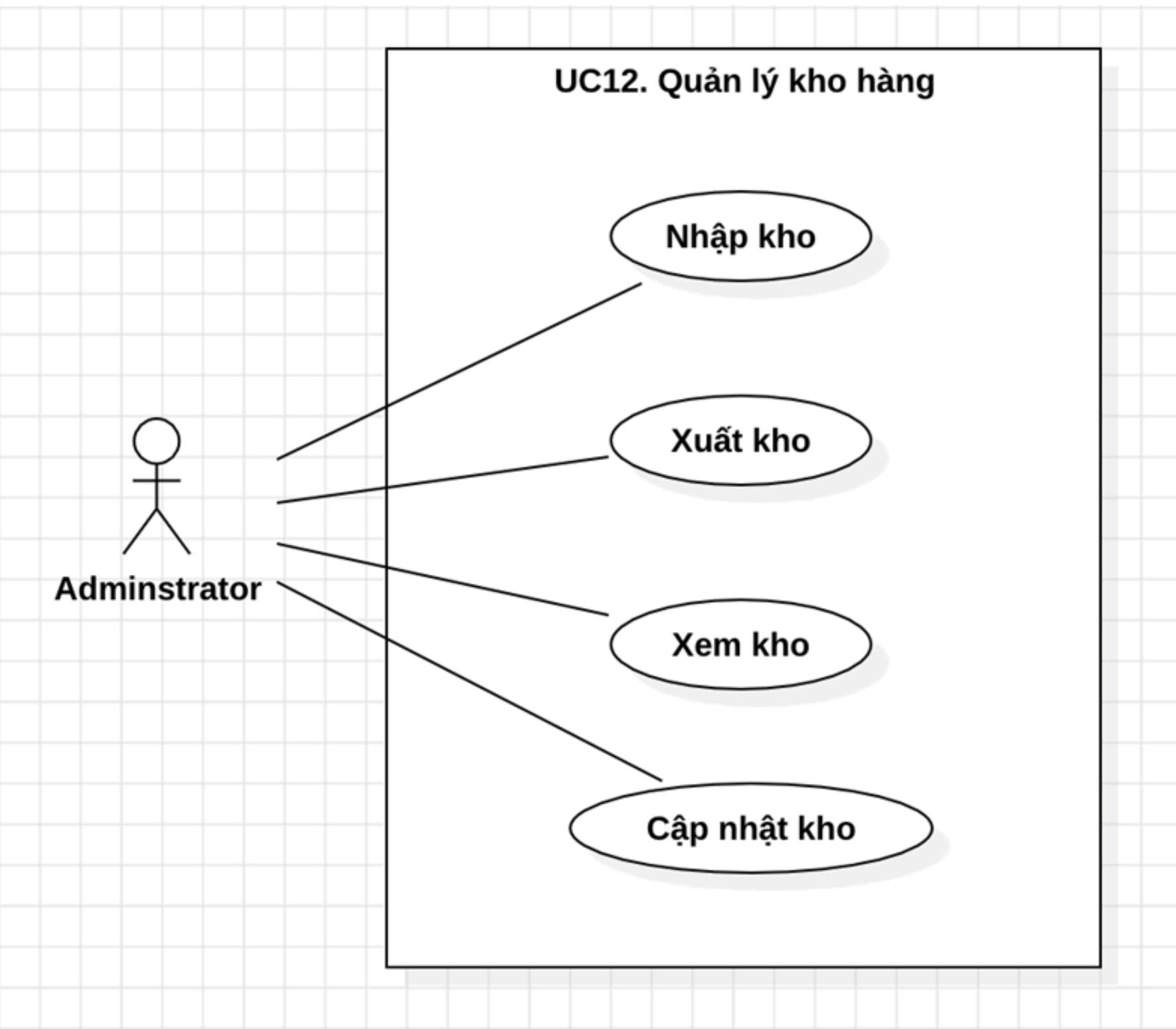
USE CASE QUẢN LÝ LỊCH BIỂU



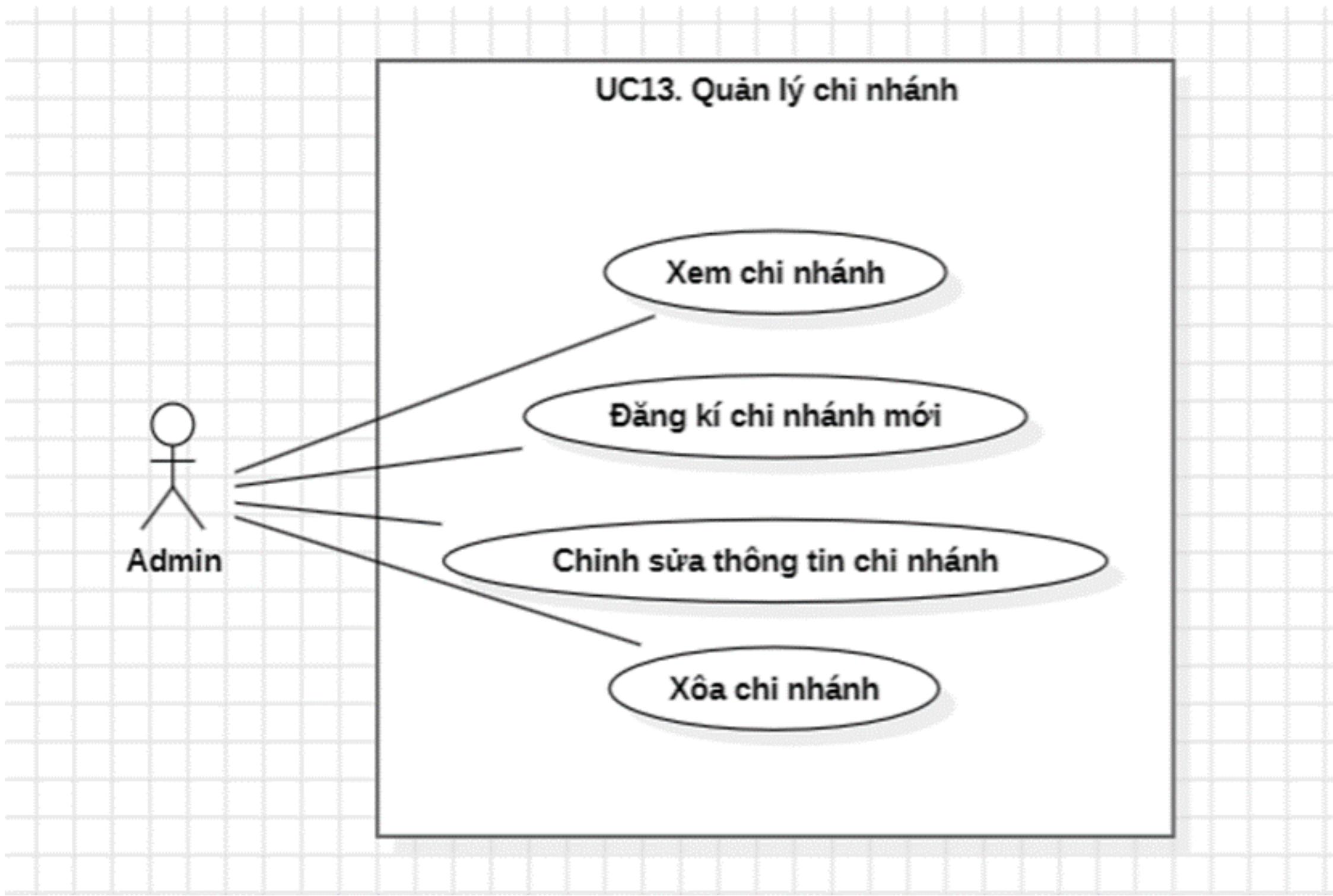
USE CASE QUẢN LÝ BẢNG LƯƠNG



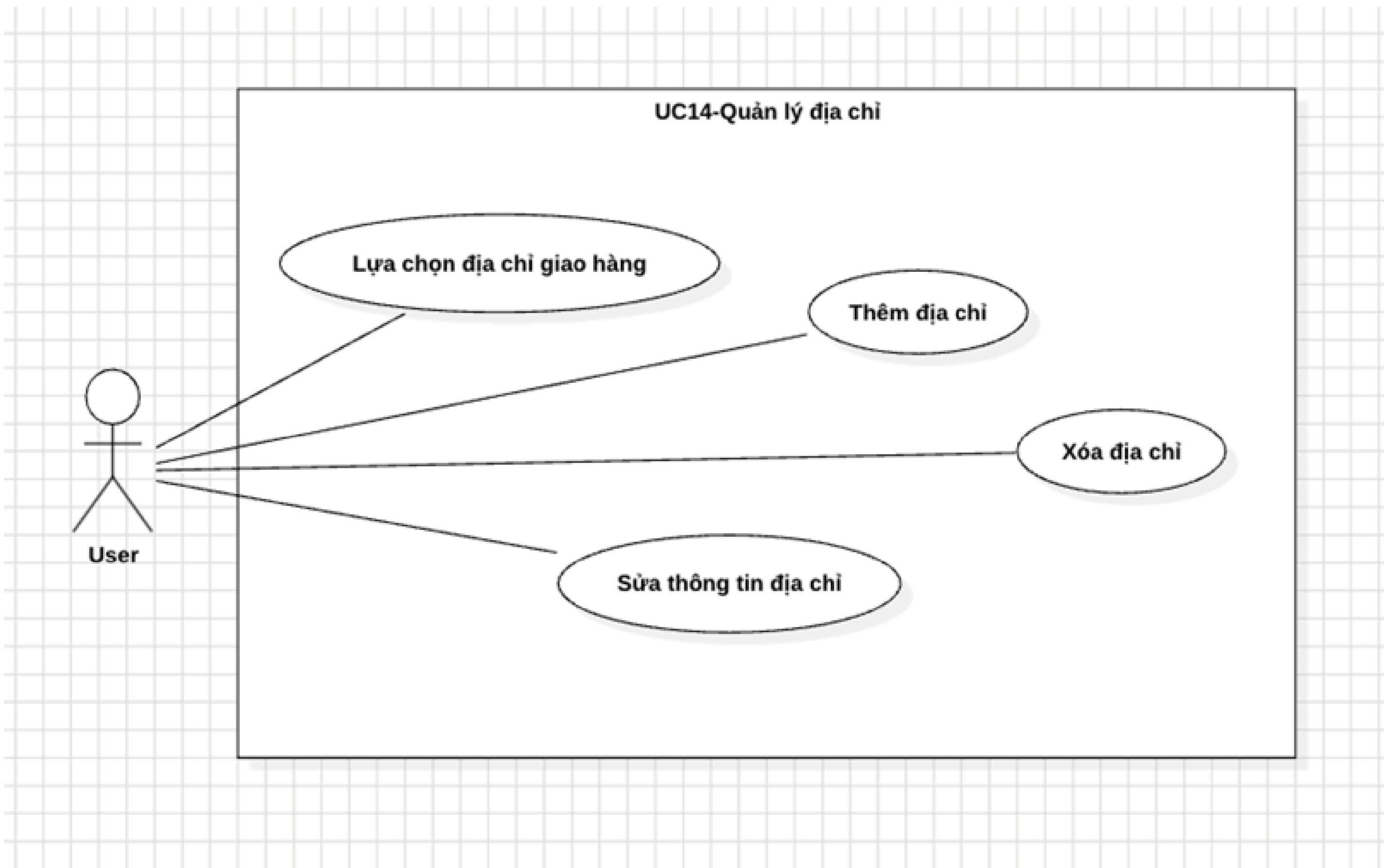
USE CASE QUẢN LÝ KHO HÀNG



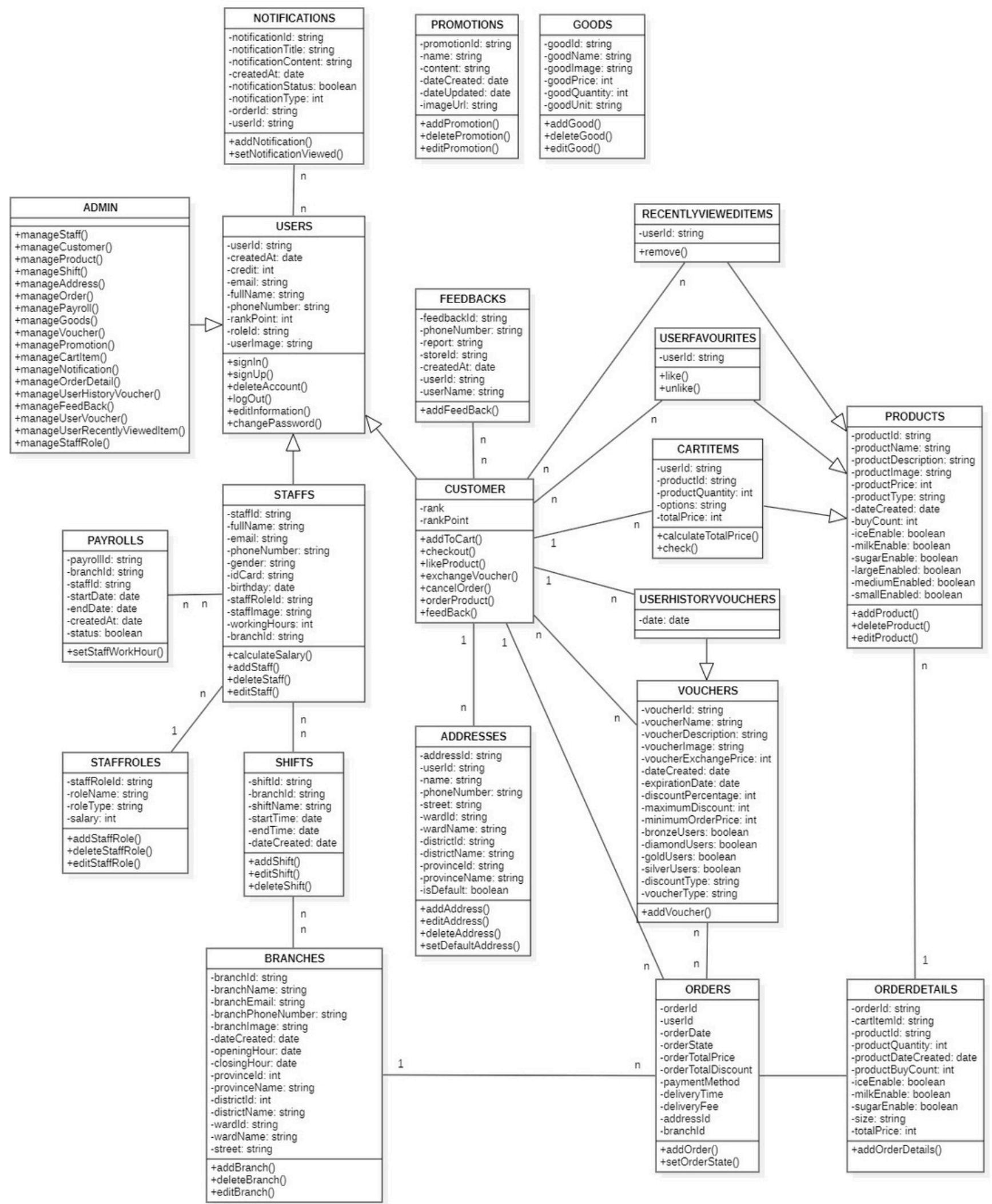
USE CASE QUẢN LÝ CHI NHÁNH



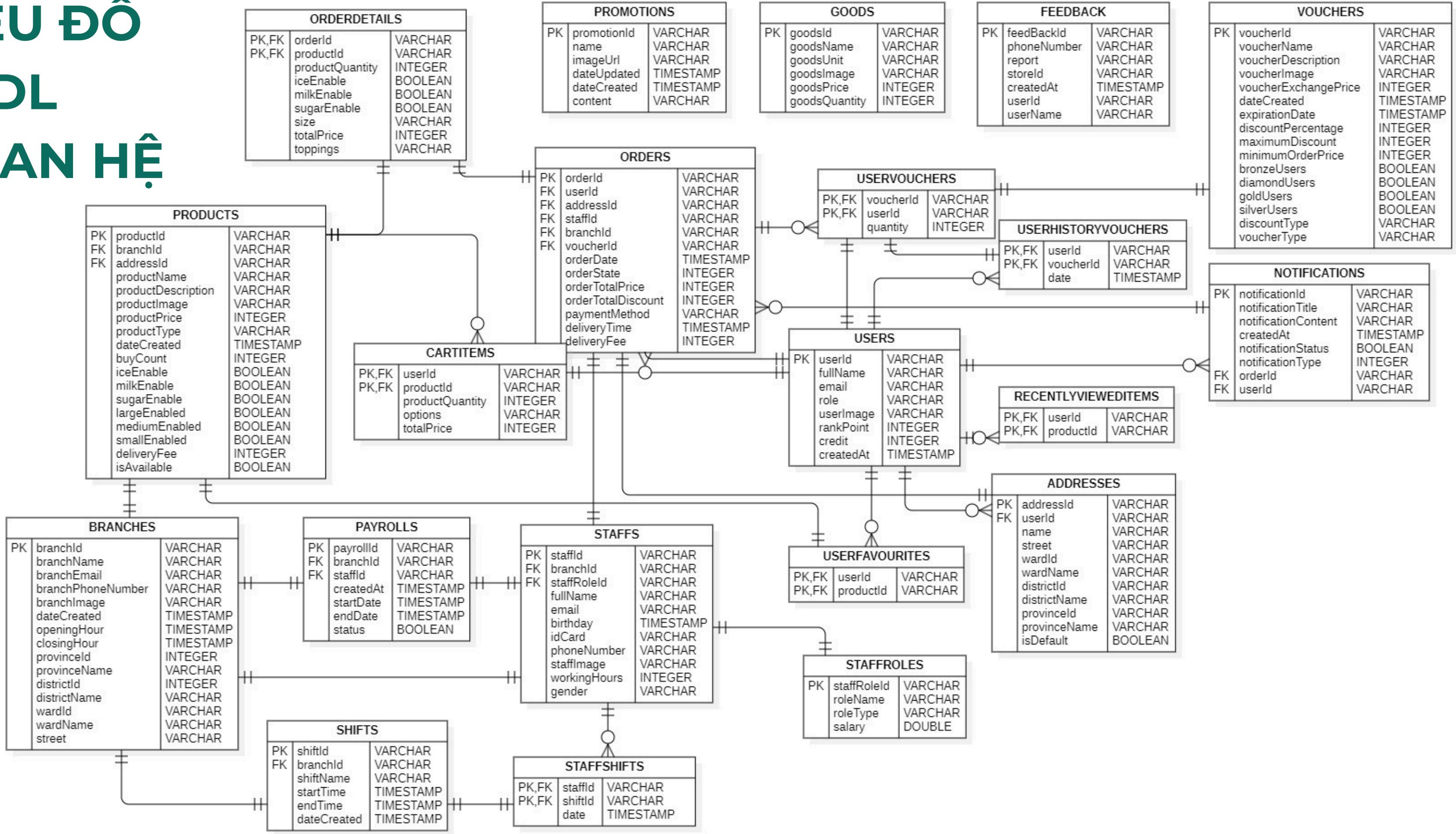
USE CASE QUẢN LÝ ĐỊA CHỈ



SƠ ĐỒ LỚP CHI TIẾT



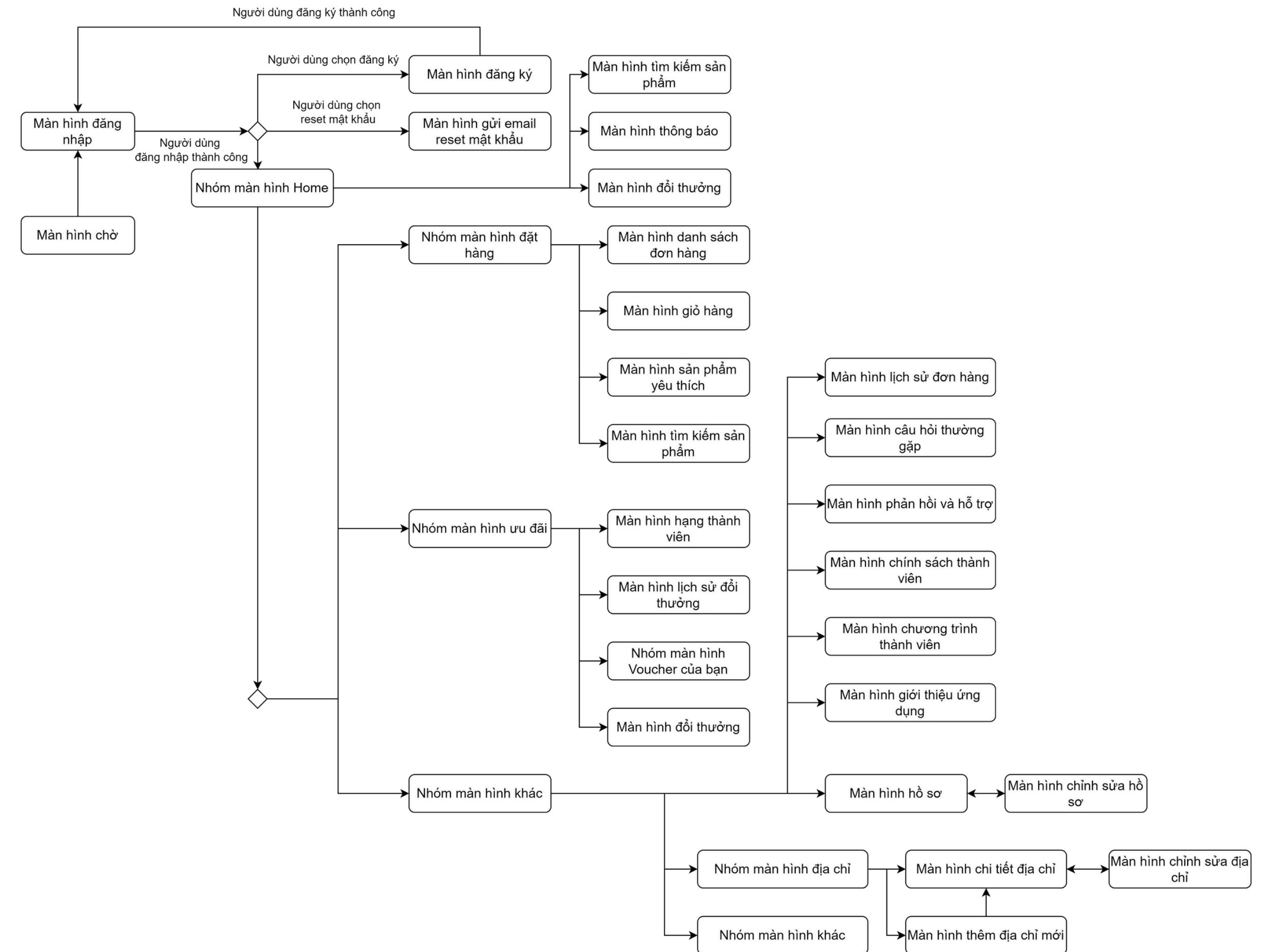
BIỂU ĐỒ CSDL QUAN HỆ



THIẾT KẾ ỨNG DỤNG

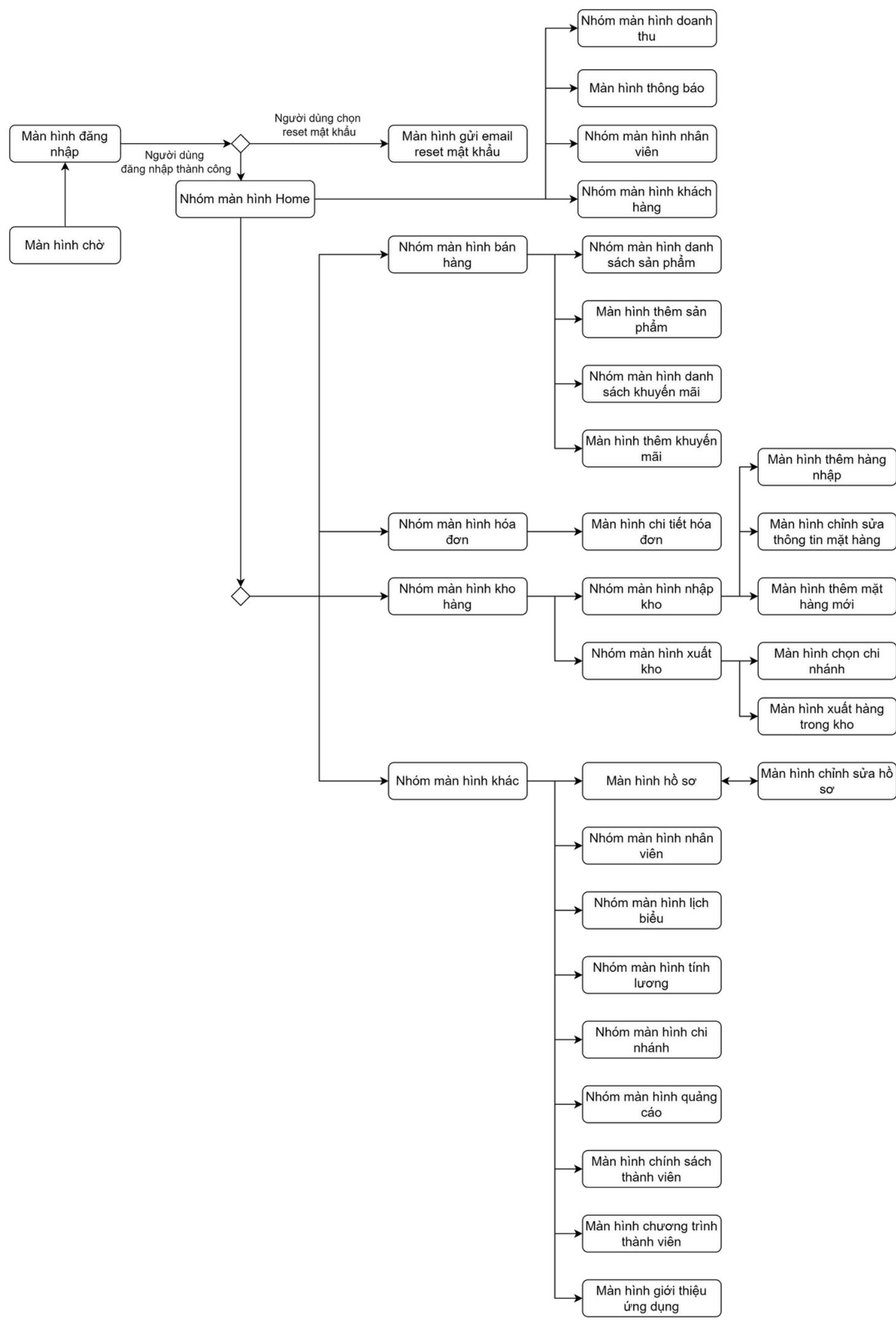
SƠ ĐỒ LIÊN KẾT MÀN HÌNH

vai trò khách hàng



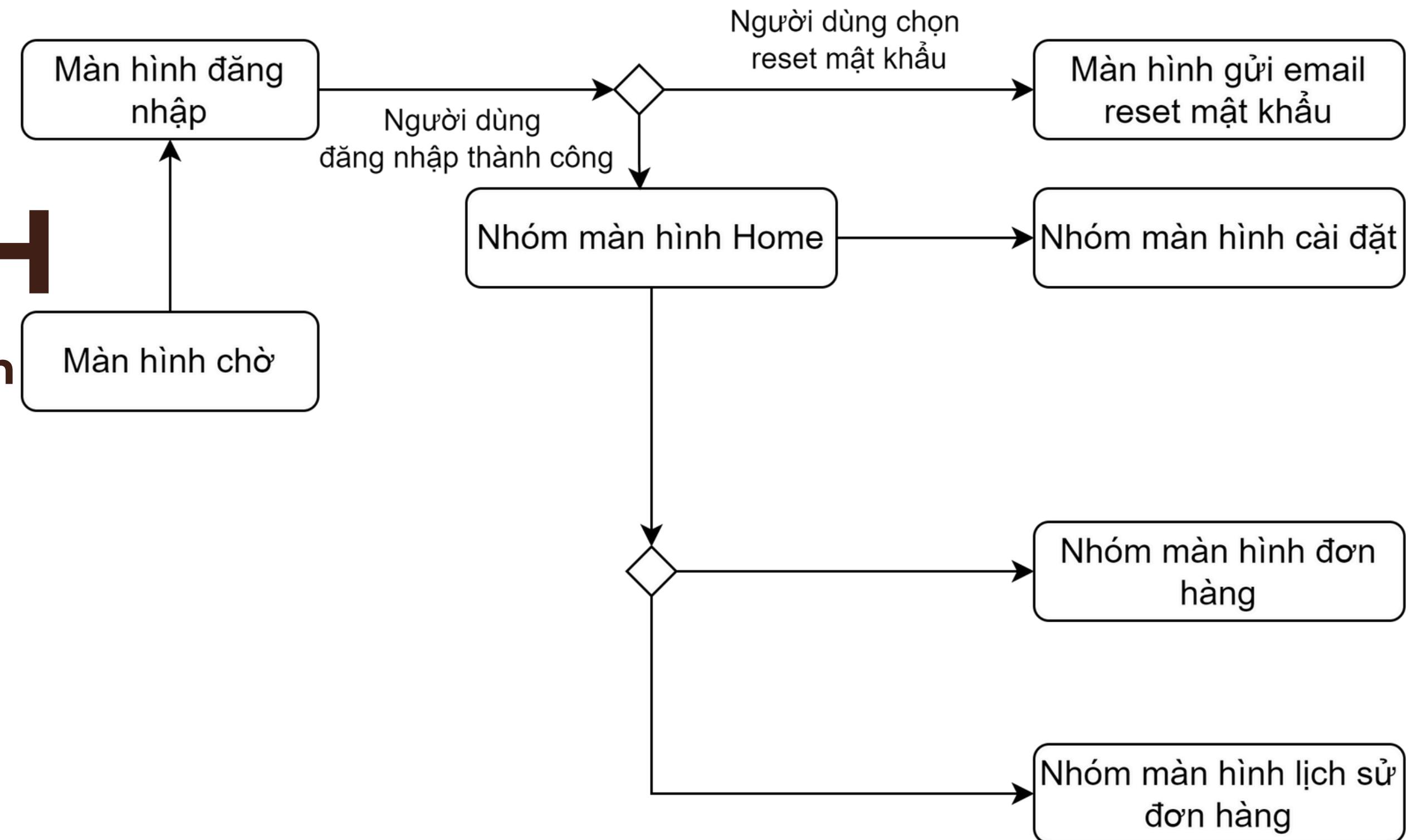
SƠ ĐỒ LIÊN KẾT MÀN HÌNH

vai trò quản trị viên

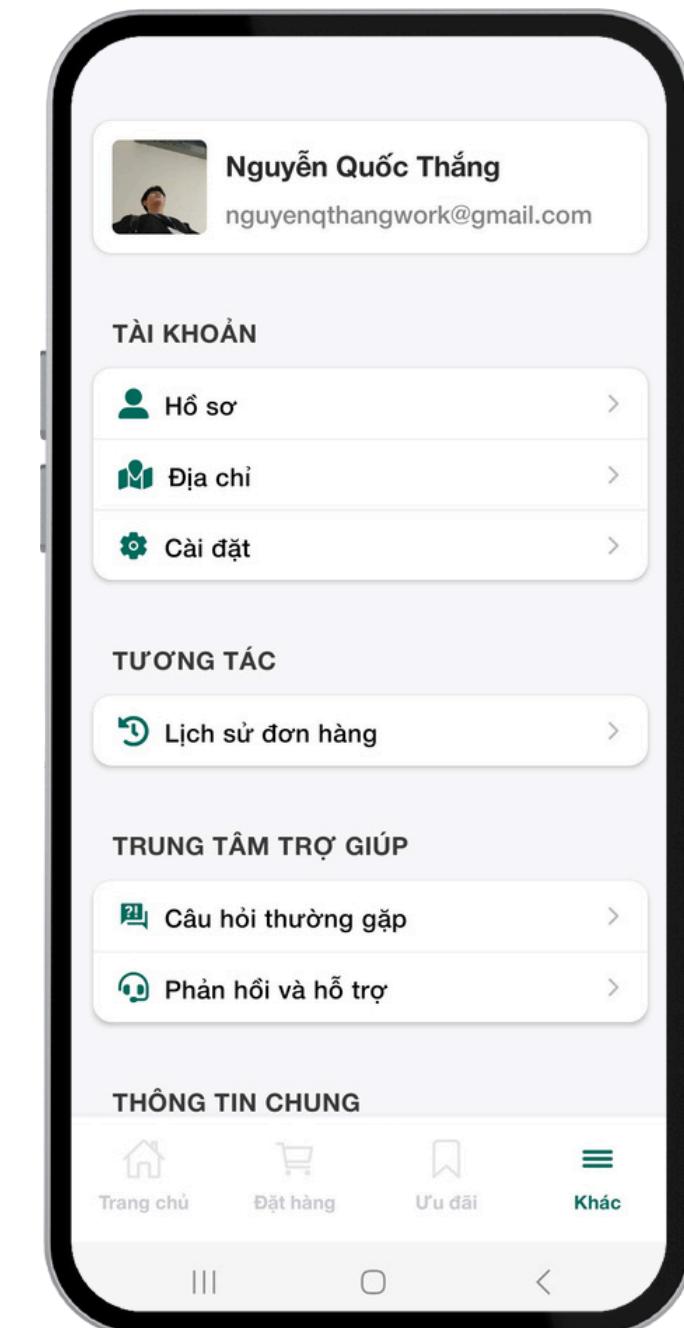
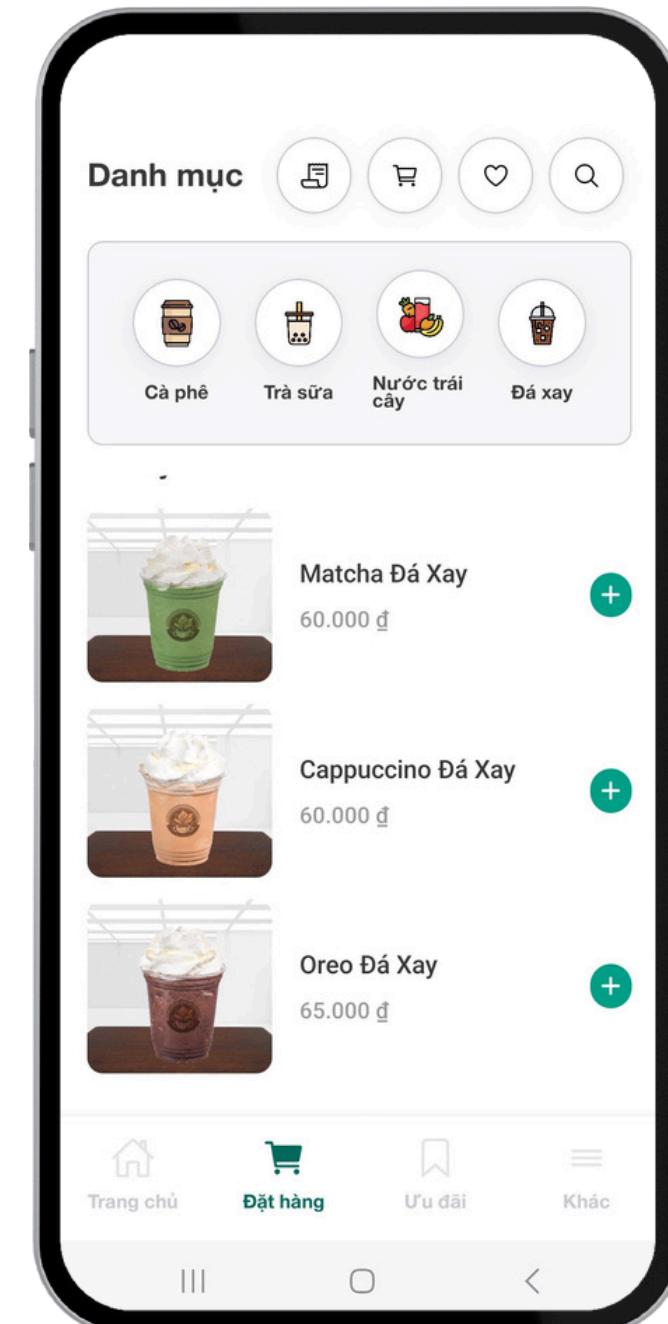
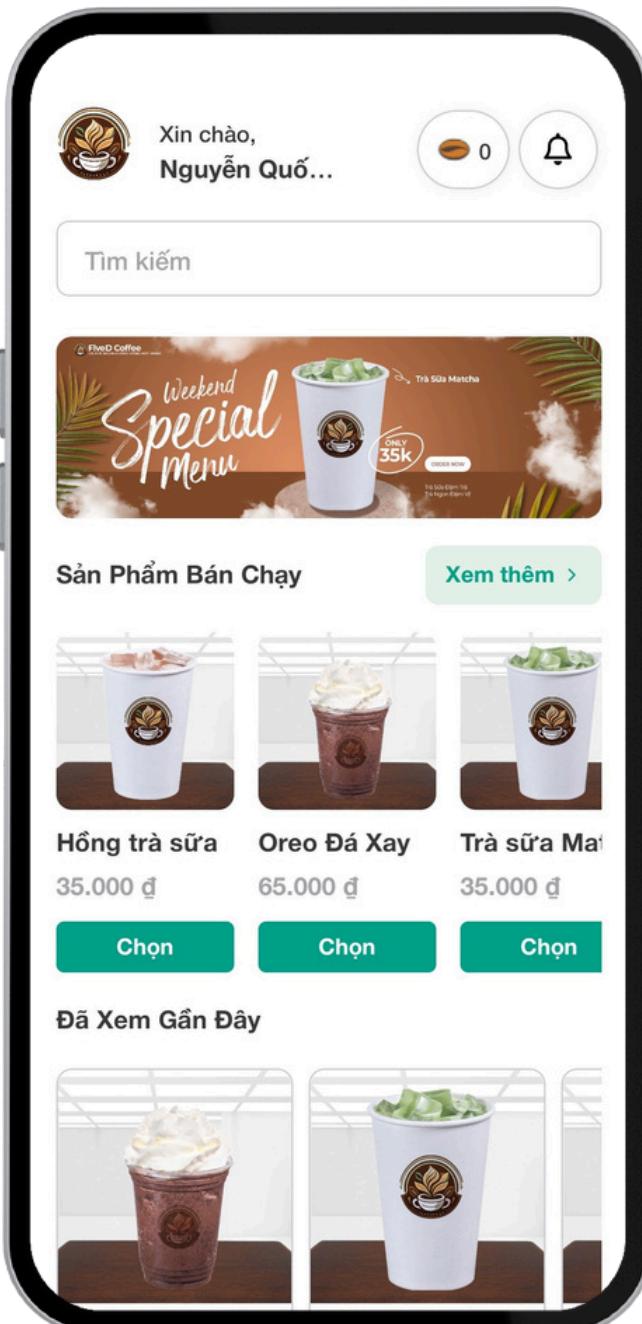
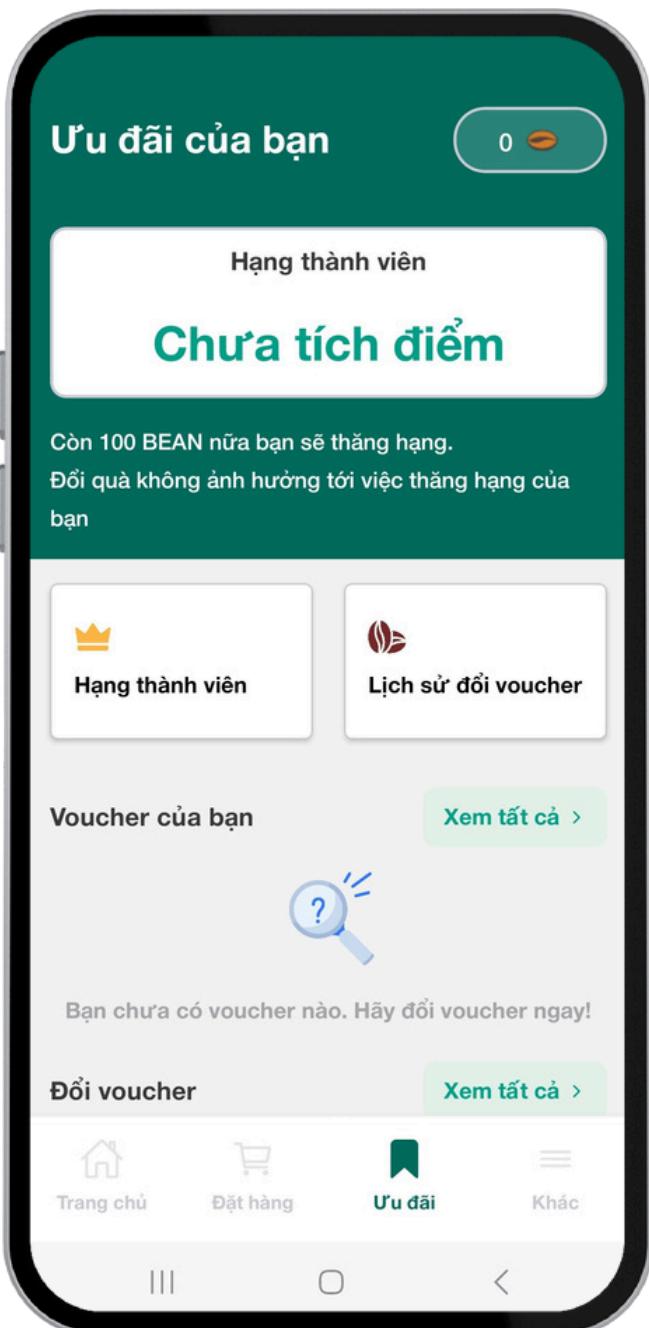


SƠ ĐỒ LIÊN KẾT MÀN HÌNH

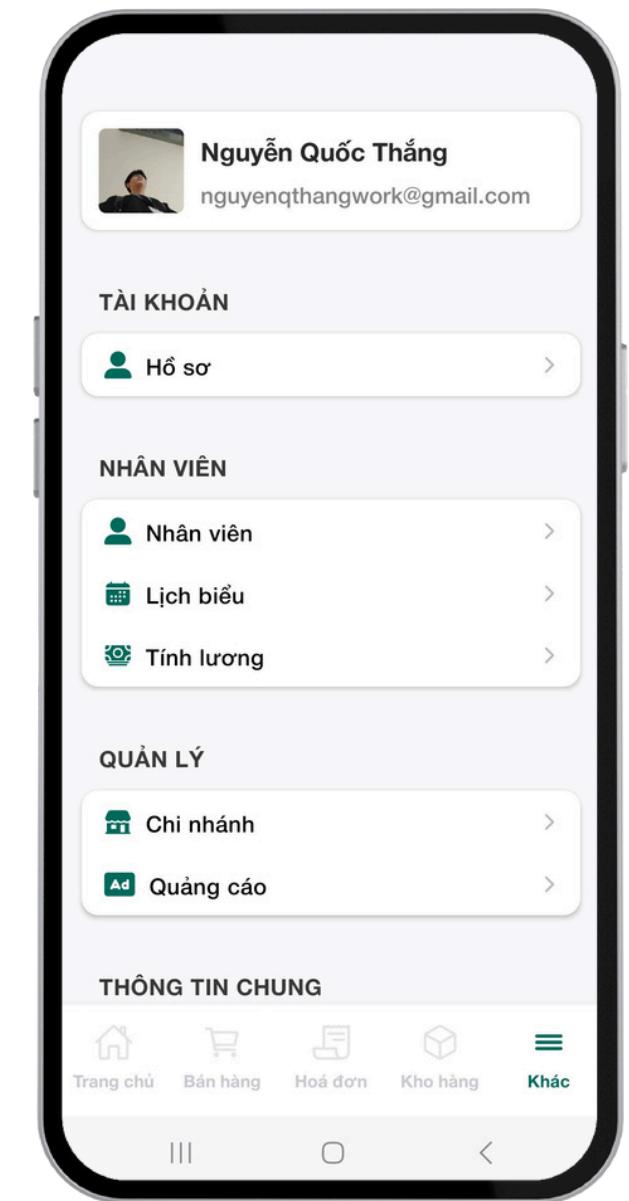
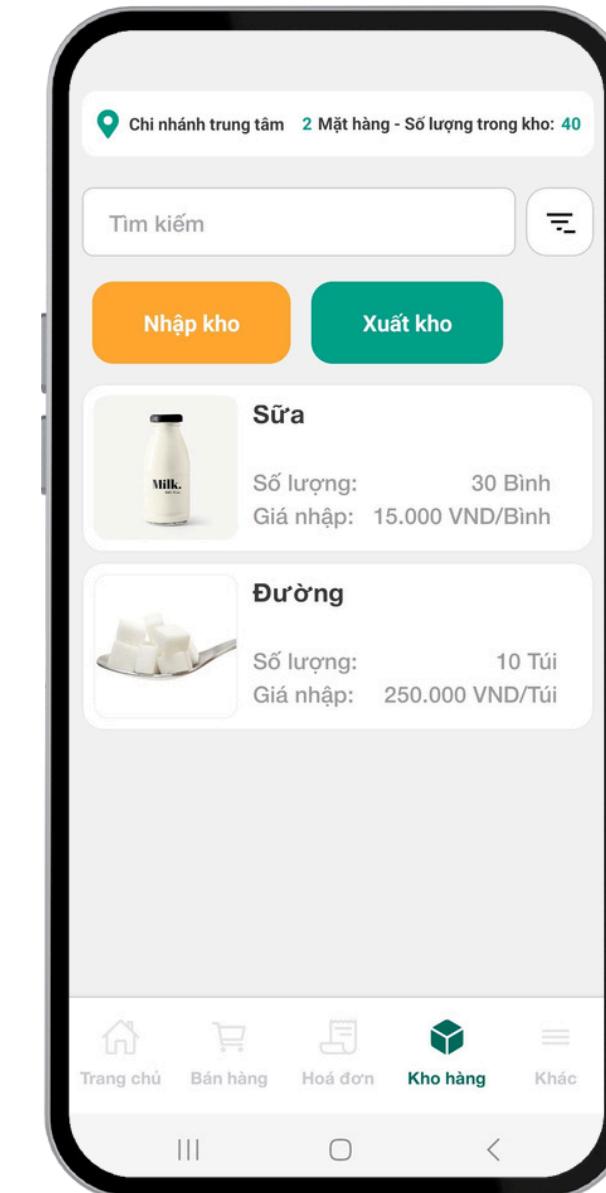
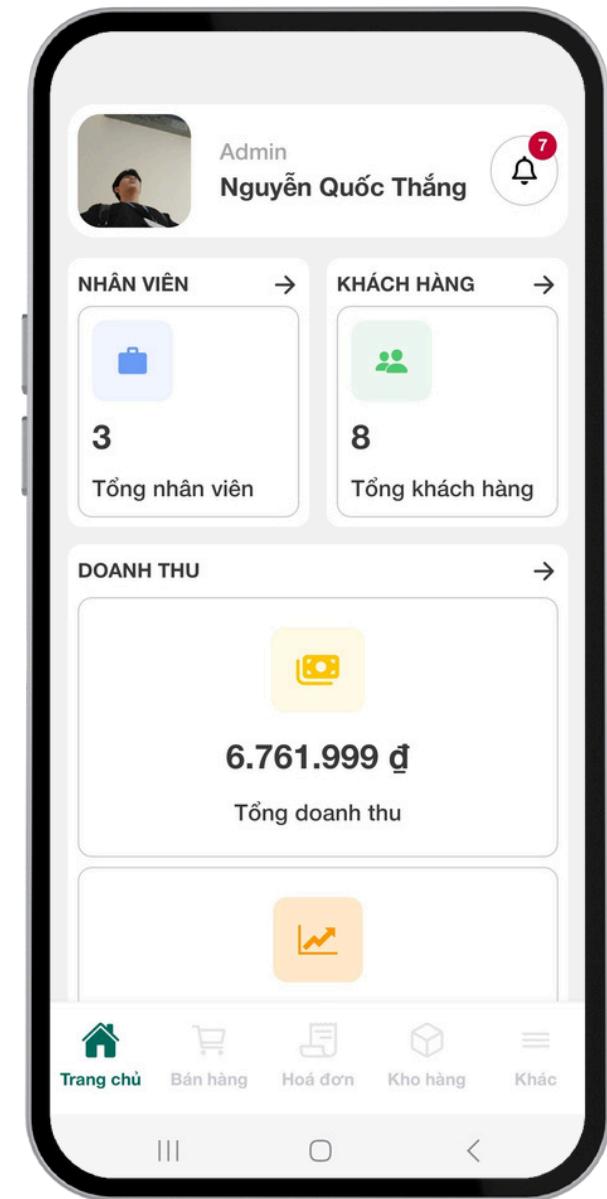
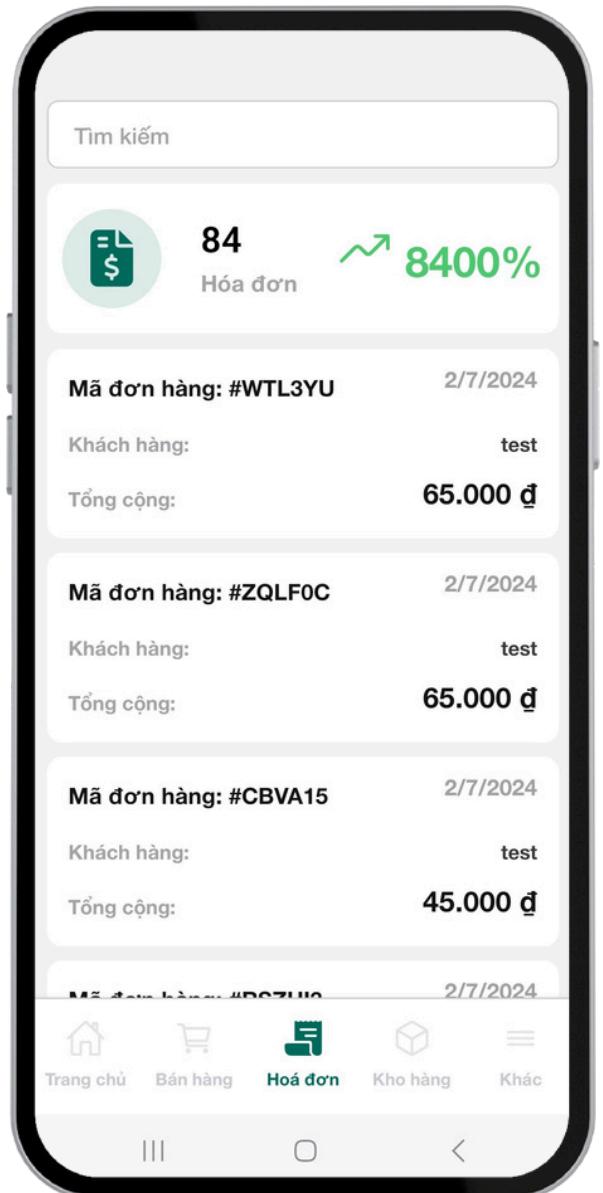
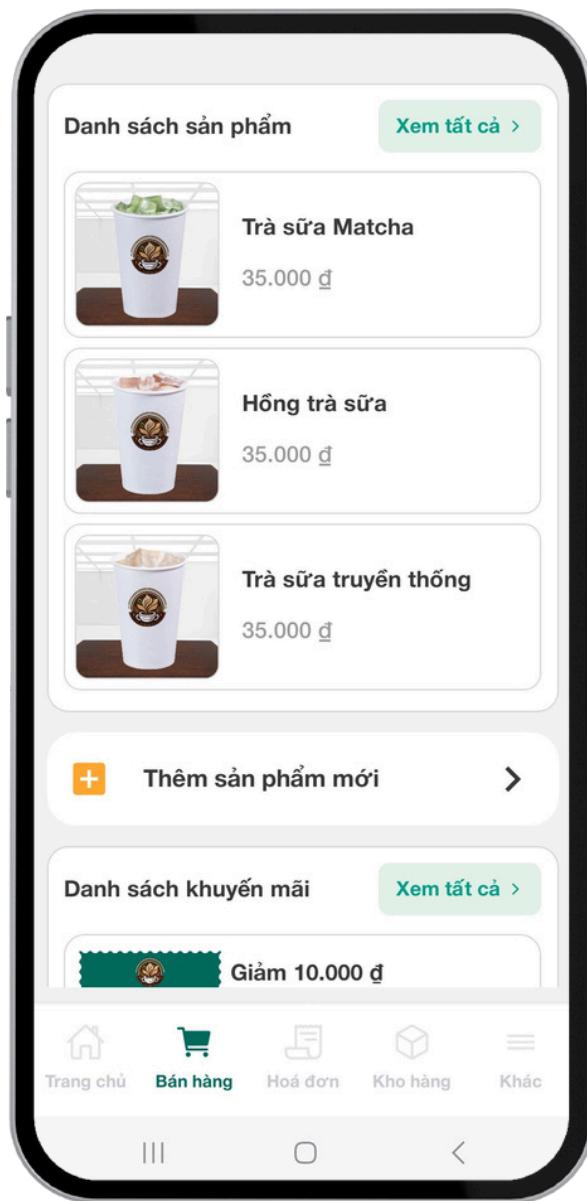
vai trò nhân viên thu ngân



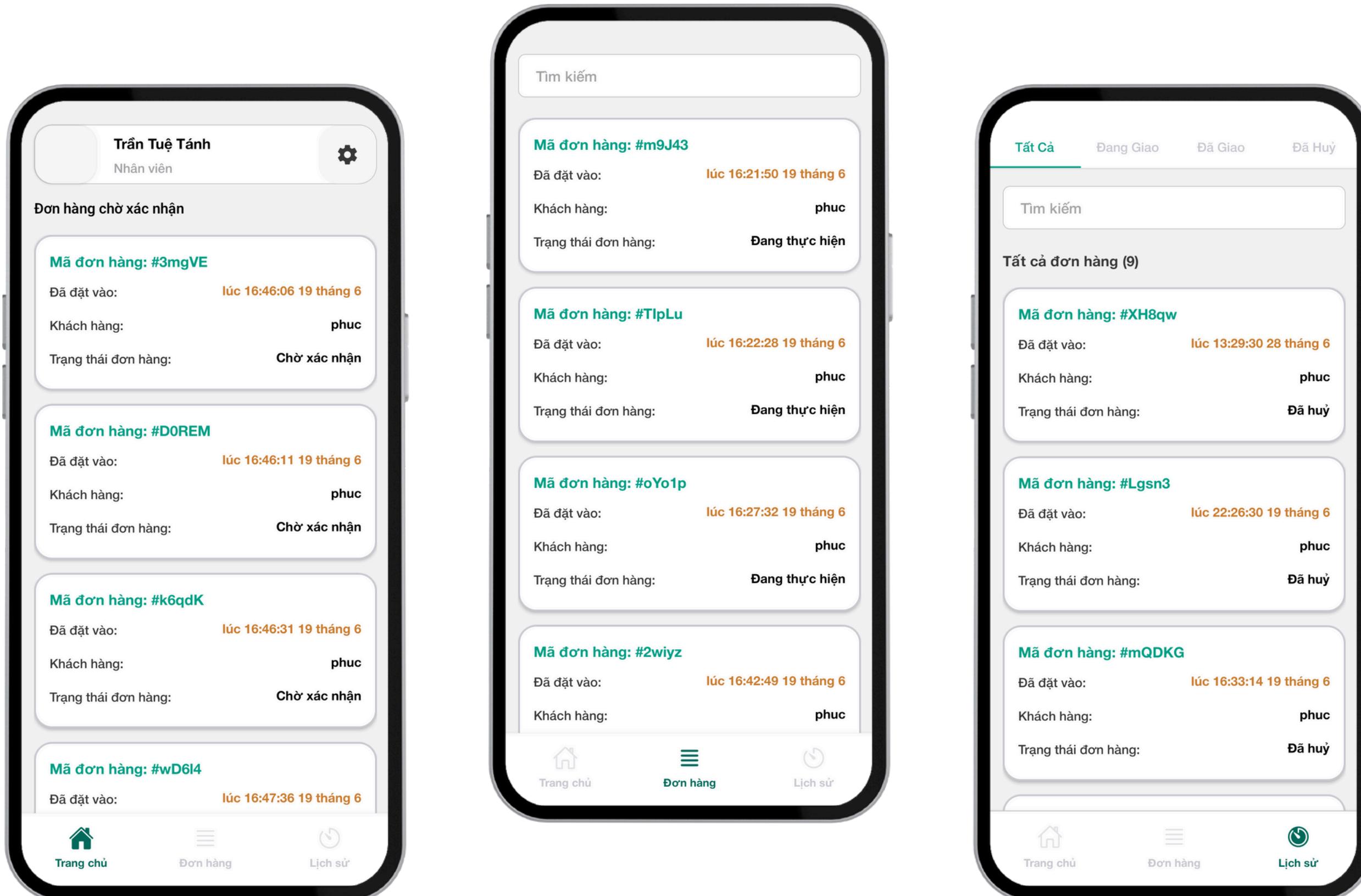
USER INTERFACE



ADMIN INTERFACE



CASHIER INTERFACE



XÂY DỰNG CHƯƠNG TRÌNH

Mô hình Client – Server

- **Client (Ứng dụng di động):**

- Được phát triển bằng React Native, chạy trên thiết bị di động của người dùng.
- Chịu trách nhiệm giao diện người dùng, xử lý các thao tác của người dùng (nhập liệu, điều hướng, tương tác với các thành phần giao diện).
- Gửi yêu cầu đến Server thông qua giao thức HTTP/HTTPS.
- Nhận và xử lý phản hồi từ Server để cập nhật giao diện và hiển thị thông tin cho người dùng.

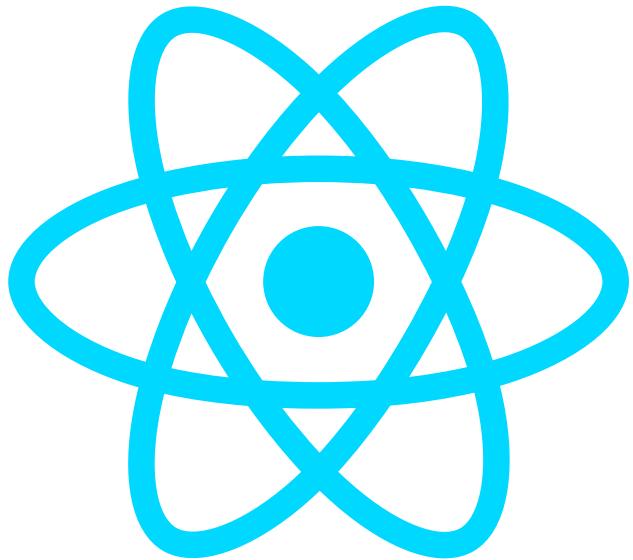
Mô hình Client – Server

- **Server (Backend):**

- Được xây dựng bằng Node.js và kết nối với FirebaseAdmin (Firebase) để quản lý dữ liệu.
- Xử lý các yêu cầu nhận được từ Client.
- Thực hiện các logic nghiệp vụ, chẳng hạn như: Tương tác với các API bên ngoài (PayOS & VietQR, FirebaseAdmin) để thực hiện các chức năng thanh toán, quản lý dữ liệu.
- Tạo phản hồi cho Client, bao gồm:
 - Dữ liệu cần hiển thị.
 - Thông báo về kết quả thao tác.
 - Thông báo lỗi.

Công nghệ chính

- **React Native:** Framework phát triển ứng dụng đa nền tảng (Android, iOS) dựa trên ReactJS.
- **Node.js:** Môi trường thực thi JavaScript bên phía máy chủ, được sử dụng để xây dựng API backend cho ứng dụng.



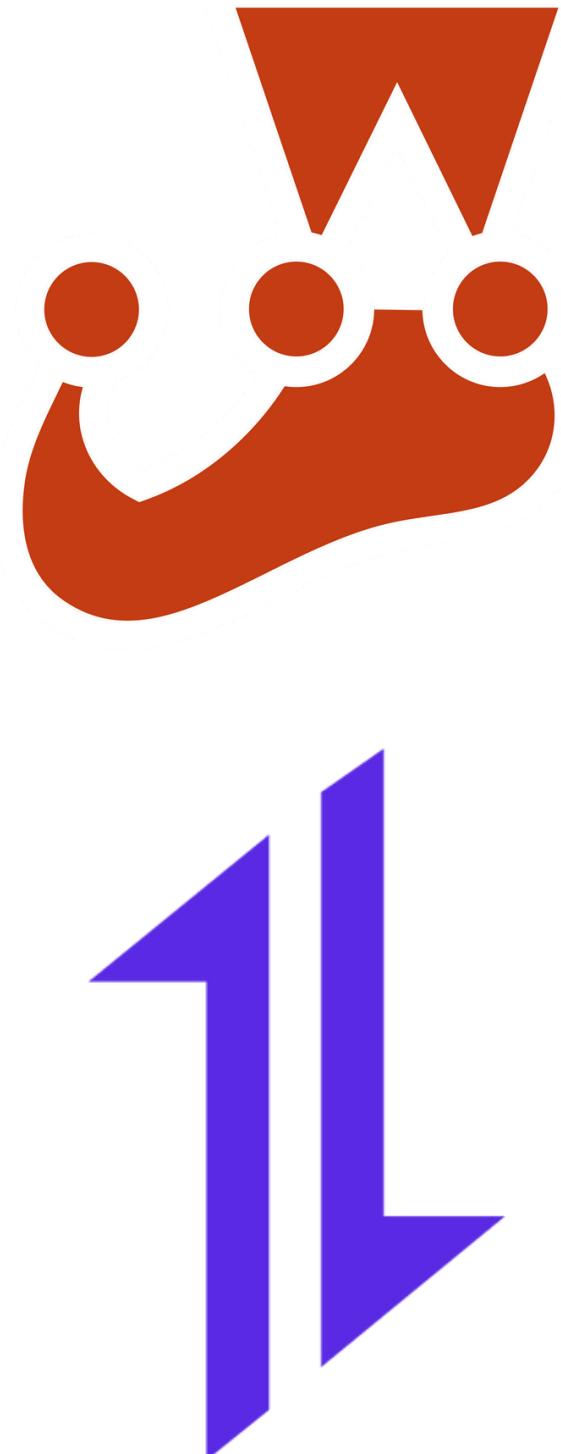
Thư viện hỗ trợ

- **PayOS:** Cổng thanh toán trực tuyến giúp tích hợp chức năng thanh toán cho ứng dụng.
- **Redux:** Thư viện quản lý trạng thái ứng dụng, giúp đơn giản hóa việc xử lý dữ liệu phức tạp.
- **Firebase:** Nền tảng đám mây cung cấp cơ sở dữ liệu, giúp hỗ trợ việc xây dựng ứng dụng một cách hiệu quả.



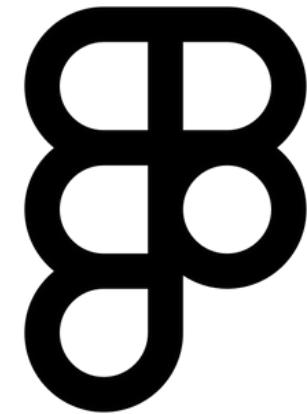
Thư viện hỗ trợ

- **Jest:** Thư viện kiểm thử JavaScript, được thiết kế để đảm bảo mã nguồn hoạt động chính xác. Nó cung cấp các công cụ mạnh mẽ để tạo và thực hiện các bài kiểm thử đơn vị, kiểm thử tích hợp, và kiểm thử hệ thống một cách dễ dàng.
- **Axios:** Thư viện JavaScript dùng để thực hiện các yêu cầu HTTP, giúp tương tác với các API một cách dễ dàng và hiệu quả. Nó hỗ trợ các phương thức HTTP phổ biến như GET, POST, PUT, DELETE, và hỗ trợ các tính năng như xử lý lỗi, hủy yêu cầu, và quản lý token.



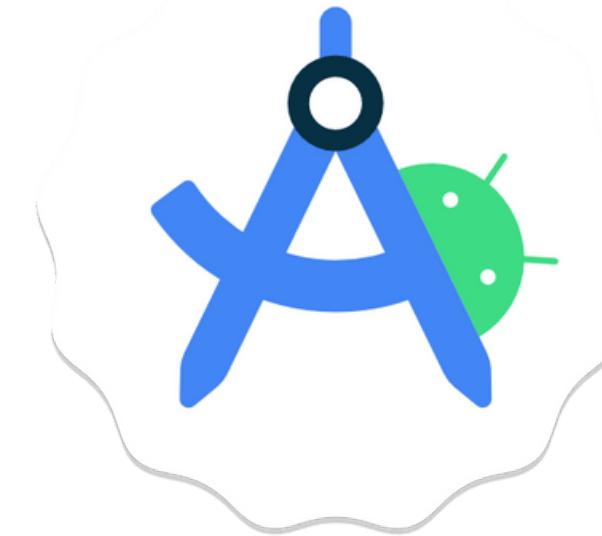
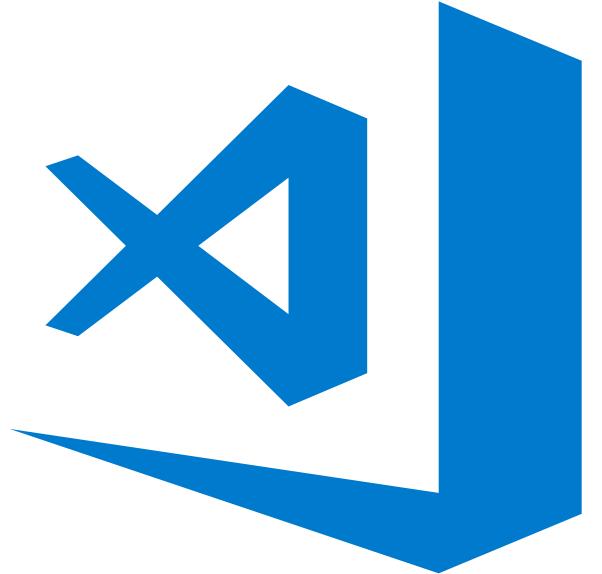
Công cụ thiết kế

- **Figma:** Công cụ thiết kế giao diện người dùng trực quan, giúp tạo prototype và giao diện cho ứng dụng.
- **StarUML:** Công cụ tạo biểu đồ UML (Unified Modeling Language) giúp mô tả kiến trúc, lớp và mối quan hệ giữa các phần tử trong ứng dụng.



Công cụ phát triển

- **Visual Studio Code:** IDE mã nguồn mở hỗ trợ lập trình đa ngôn ngữ, cung cấp nhiều tính năng hỗ trợ phát triển ứng dụng.
- **Android Studio:** IDE chính thức của Google cho phát triển ứng dụng Android, cung cấp các công cụ và tính năng mạnh mẽ để xây dựng, kiểm thử và gỡ lỗi ứng dụng Android.



KIỂM THỬ CHƯƠNG TRÌNH

Kiểm thử chương trình

- **Kiểm thử hộp đen**

Kiểm thử hộp đen là phương pháp kiểm thử tập trung vào chức năng của phần mềm dựa trên yêu cầu và giao diện bên ngoài, mà không cần quan tâm đến cấu trúc code bên trong.

Kiểm thử chương trình

- Ví dụ code Unit Test (Jest-Expo)



```
import React from "react";
import { render, fireEvent } from "@testing-library/react-native";
import testComponent from "../../testComponent";

describe("testComponent", () => {
  const onPressMock = jest.fn();

  test("renders correctly", () => {
    const { getByTestId } = render(<testComponent onPress={onPressMock} />);
    const button = getByTestId("button-test-id");
    expect(button).toBeTruthy();
  });

  test("calls onPress when button is pressed", () => {
    const { getByTestId } = render(<testComponent onPress={onPressMock} />);
    const button = getByTestId("button-test-id");
    fireEvent.press(button);
    expect(onPressMock).toHaveBeenCalled();
  });
});
```

Kiểm thử chương trình

- Ví dụ kiểm thử Button

```
● ● ●

import React from "react";
import { render, fireEvent } from "@testing-library/react-native";
import { useNavigation } from "@react-navigation/native";
import AddVoucherButton from "../../components/Admin/Button/AddVoucherButton";
jest.mock("@react-navigation/native");

describe("AddVoucherButton", () => {
  const navigation = {
    navigate: jest.fn(),
  };

  beforeEach(() => {
    useNavigation.mockReturnValue(navigation);
  });

  test("renders correctly with given props", () => {
    const { getByTestId } = render(<AddVoucherButton />);
    const touchable = getByTestId("addVoucherButton");
    expect(touchable).toBeTruthy();
  });

  test("TouchableOpacity calls goToAddVoucherScreen when pressed", () => {
    const { getByTestId } = render(<AddVoucherButton />);
    const touchable = getByTestId("addVoucherButton");
    fireEvent.press(touchable);
    expect(navigation.navigate).toHaveBeenCalledWith("AdminAddVoucher");
  });
});
```

```
● ● ●

test("Ionicons renders correctly", () => {
  const { getByTestId } = render(<AddVoucherButton />);
  const icon = getByTestId("addIcon");
  expect(icon).toBeTruthy();
});

});
```

Kiểm thử chương trình

- Ví dụ kiểm thử Card



```
import React from "react";
import { render, fireEvent } from "@testing-library/react-native";
import ApplyVoucherButton from "../../components/Client/Button/ApplyVoucherButton";

describe("ApplyVoucherButton", () => {
  const onPressMock = jest.fn();

  test("renders correctly", () => {
    const { getByTestId } = render(
      <ApplyVoucherButton onPress={onPressMock} />
    );

    const button = getByTestId("applyVoucherButton");
    const icon = getByTestId("voucherIcon");

    expect(button).toBeTruthy();
    expect(icon).toBeTruthy();
  });

  test("calls onPress when button is pressed", () => {
    const { getByTestId } = render(
      <ApplyVoucherButton onPress={onPressMock} />
    );

    const button = getByTestId("applyVoucherButton");
    fireEvent.press(button);
    expect(onPressMock).toHaveBeenCalledWith();
  });
});
```

Kiểm thử chương trình

- Kết quả kiểm thử với Jest-Expo

```
PS D:\Code Workspace\CoffeeShopManagement\CoffeeShopManagement> npm run test
> coffeeshopmanagement@1.0.0 test
> jest

PASS __tests__/_cards/CategoryIcon.test.js
PASS __tests__/_buttons/AddGoodButton.test.js
PASS __tests__/_buttons/CustomChooseButton.test.js
PASS __tests__/_cards/OrderCard.test.js
PASS __tests__/_cards/VoucherCard.test.js
PASS __tests__/_buttons/AddNewItemButton.test.js
PASS __tests__/_buttons/DeleteButton.test.js
PASS __tests__/_buttons/AddNewVoucherButton.test.js
PASS __tests__/_cards/ChooseDeliveryButton.test.js
PASS __tests__/_buttons/AddNewStaffButton.test.js
PASS __tests__/_cards/ItemCard.test.js
PASS __tests__/_cards/ApplyVoucherButton.test.js
PASS __tests__/_cards/BranchCard.test.js
PASS __tests__/_cards/BrownTextButton.test.js
PASS __tests__/_cards/ProductCard.test.js
PASS __tests__/_buttons/AddStaffButton.test.js
PASS __tests__/_cards/BrownButton.test.js
PASS __tests__/_cards/ViewProductCard.test.js
PASS __tests__/_cards/PromotionCard.test.js
PASS __tests__/_buttons/AddButtonItem.test.js
PASS __tests__/_buttons/ColorButton.test.js
PASS __tests__/_buttons/AddNewShiftButton.test.js
PASS __tests__/_buttons/ShiftListButton.test.js
PASS __tests__/_buttons/AddVoucherButton.test.js

Test Suites: 24 passed, 24 total
Tests:       47 passed, 47 total
Snapshots:   0 total
Time:        5.545 s
```

HƯỚNG DẪN CÀI ĐẶT VÀ SỬ DỤNG

Hướng dẫn cài đặt cho nhà phát triển

Bước 1: Chuẩn bị môi trường

- Đảm bảo máy tính của bạn kết nối internet ổn định.
- Cài đặt các phần mềm và công cụ cần thiết:
 - Node.js
 - Git

Bước 2: Tải mã nguồn

- Mở terminal hoặc command prompt.
- Clone repository của phần mềm từ GitHub:
`git clone https://github.com/FiveD-SE/CoffeeShopManagement`
- Di chuyển vào thư mục chứa mã nguồn:
`cd CoffeeShopManagement`

Bước 3: Cài đặt các thư viện

- Chạy lệnh sau để cài đặt các thư viện và phụ thuộc cần thiết
`npm install`

Hướng dẫn cài đặt cho nhà phát triển

Bước 4: Cấu hình môi trường

Tạo file env.js trong thư mục gốc của project và cấu hình các biến môi trường cần thiết, bao gồm: REACT_APP_FIREBASE_API_KEY, REACT_APP_FIREBASE_AUTH_DOMAIN, REACT_APP_FIREBASE_PROJECT_ID, REACT_APP_FIREBASE_STORAGE_BUCKET, REACT_APP_FIREBASE_MESSAGING_SENDER_ID, REACT_APP_FIREBASE_APP_ID, REACT_APP_FIREBASE_MEASUREMENT_ID, REACT_APP_GHN_TOKEN, REACT_APP_GHN_SHOP_ID

Bước 5: Khởi chạy ứng dụng

Chạy lệnh sau để khởi chạy ứng dụng:

npm run android

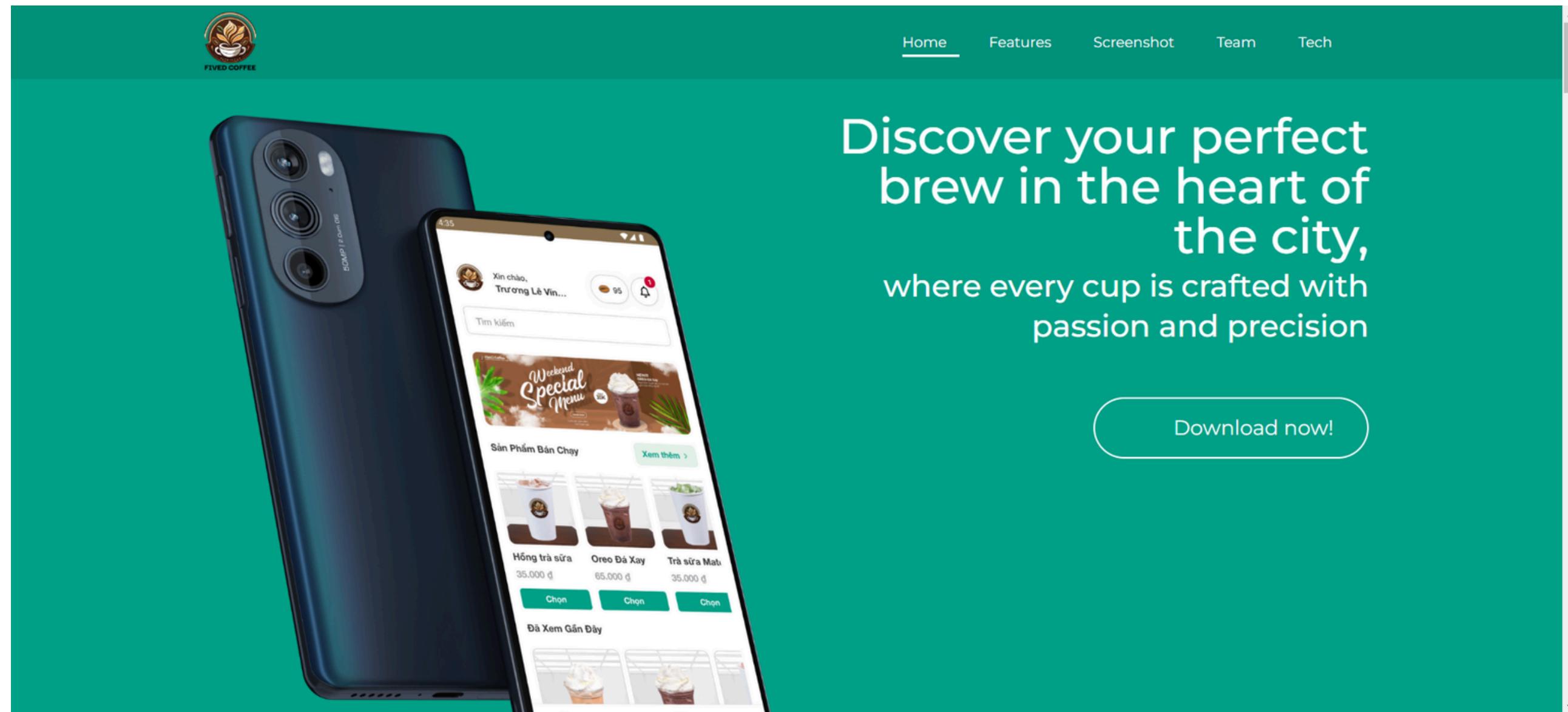
Hoặc

npm start

Hướng dẫn cài đặt cho người dùng cuối

Bước 1: Truy cập link: <https://cfbe.up.railway.app/> và tìm đến nút download

Bước 2: Tải về và tiến hành cài đặt phần mềm



Hướng dẫn sử dụng cho khách hàng

Bước 1: Mở ứng dụng sau khi khởi chạy thành công

Bước 2: Chọn Đăng nhập hoặc Đăng ký nếu bạn chưa có tài khoản.

Bước 3: Nhập thông tin đăng nhập (email và mật khẩu) và nhấn Đăng nhập.

Bước 4: Tiến hành truy cập vào các tính năng của ứng dụng

Hướng dẫn sử dụng cho admin

Bước 1: Mở ứng dụng sau khi cài đặt thành công.

Bước 2: Chọn Đăng nhập.

Bước 3: Nhập thông tin đăng nhập (email và mật khẩu) và nhấn Đăng nhập.

Bước 4: Tiến hành truy cập vào các tính năng của ứng dụng

Hướng dẫn sử dụng cho cashier

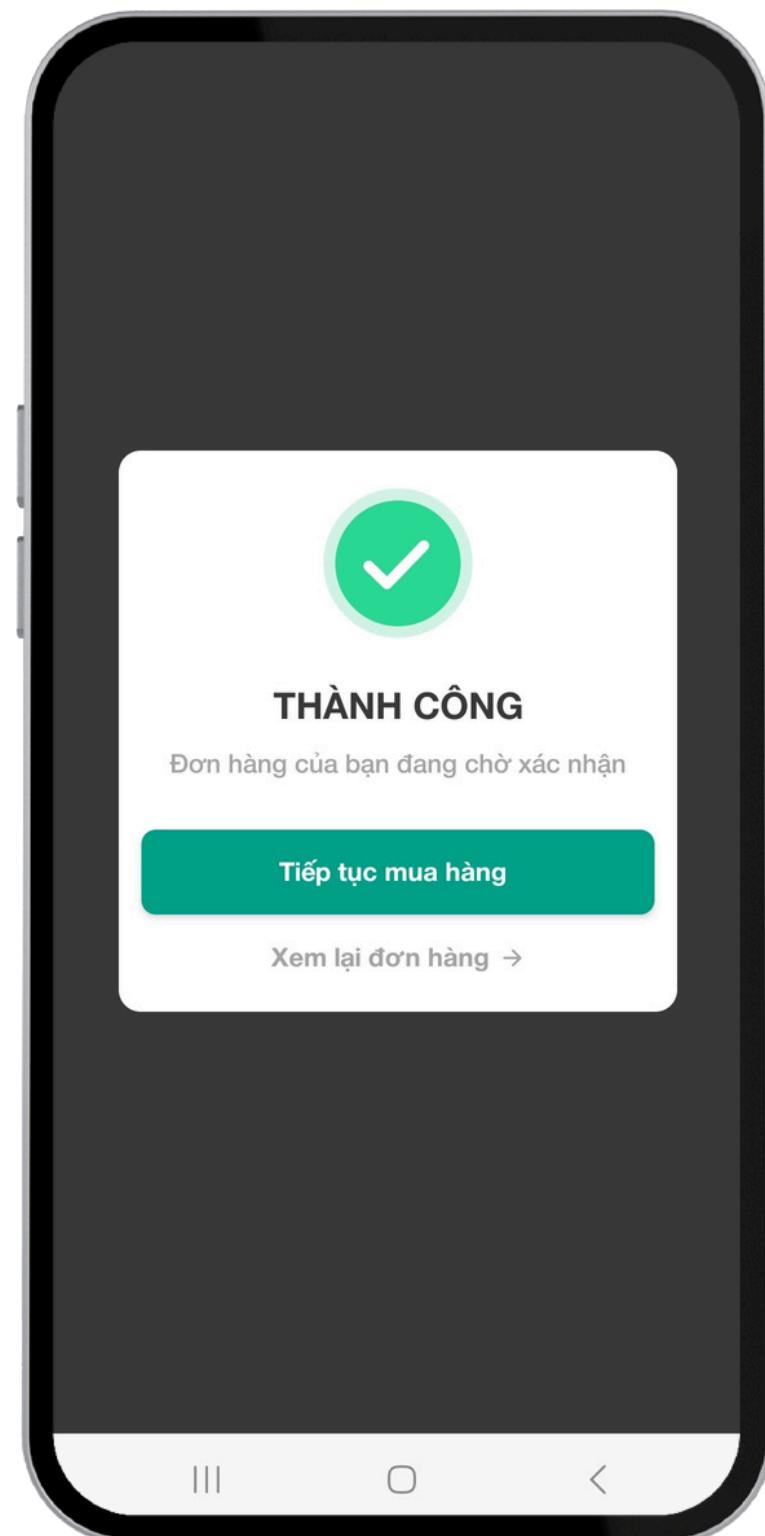
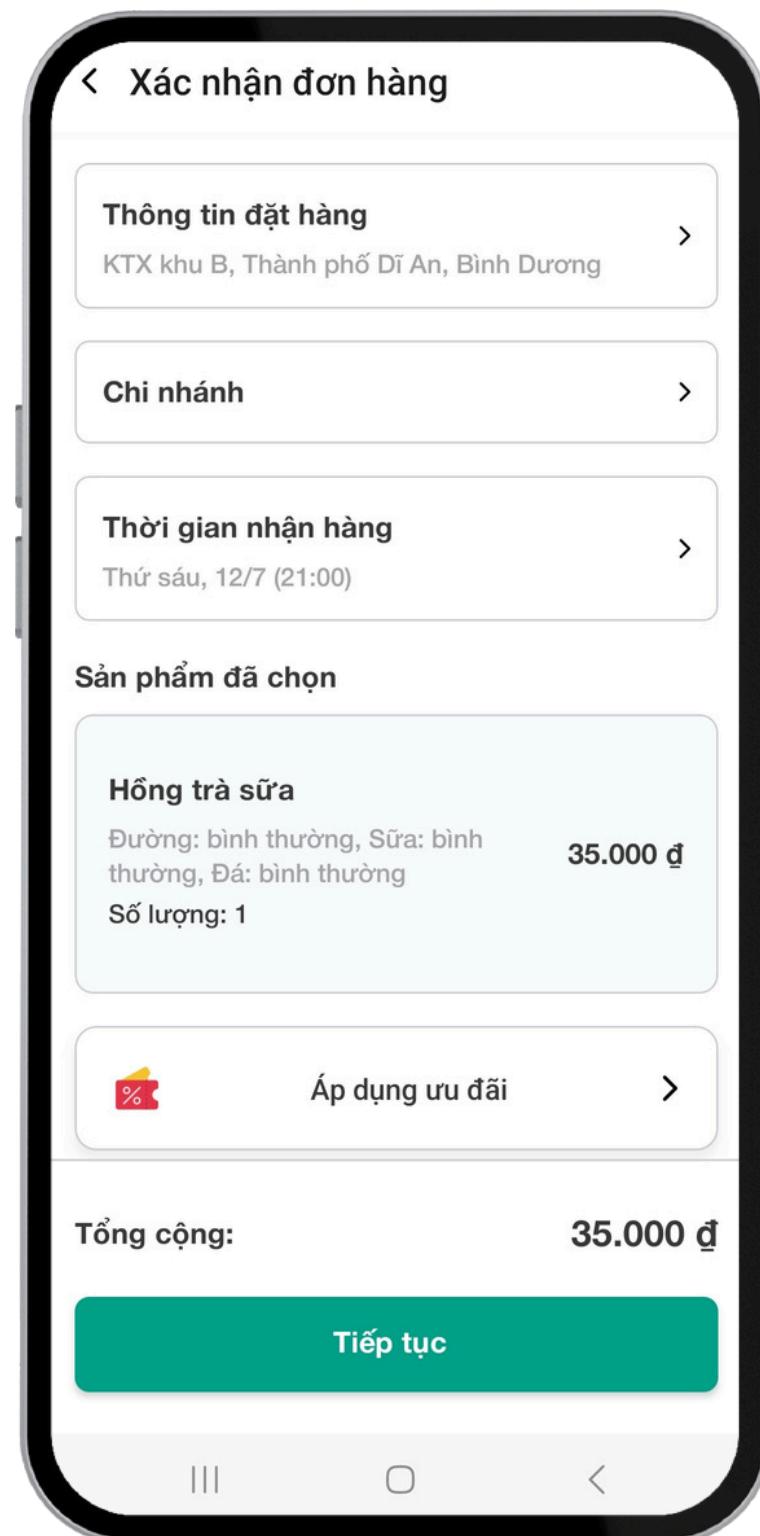
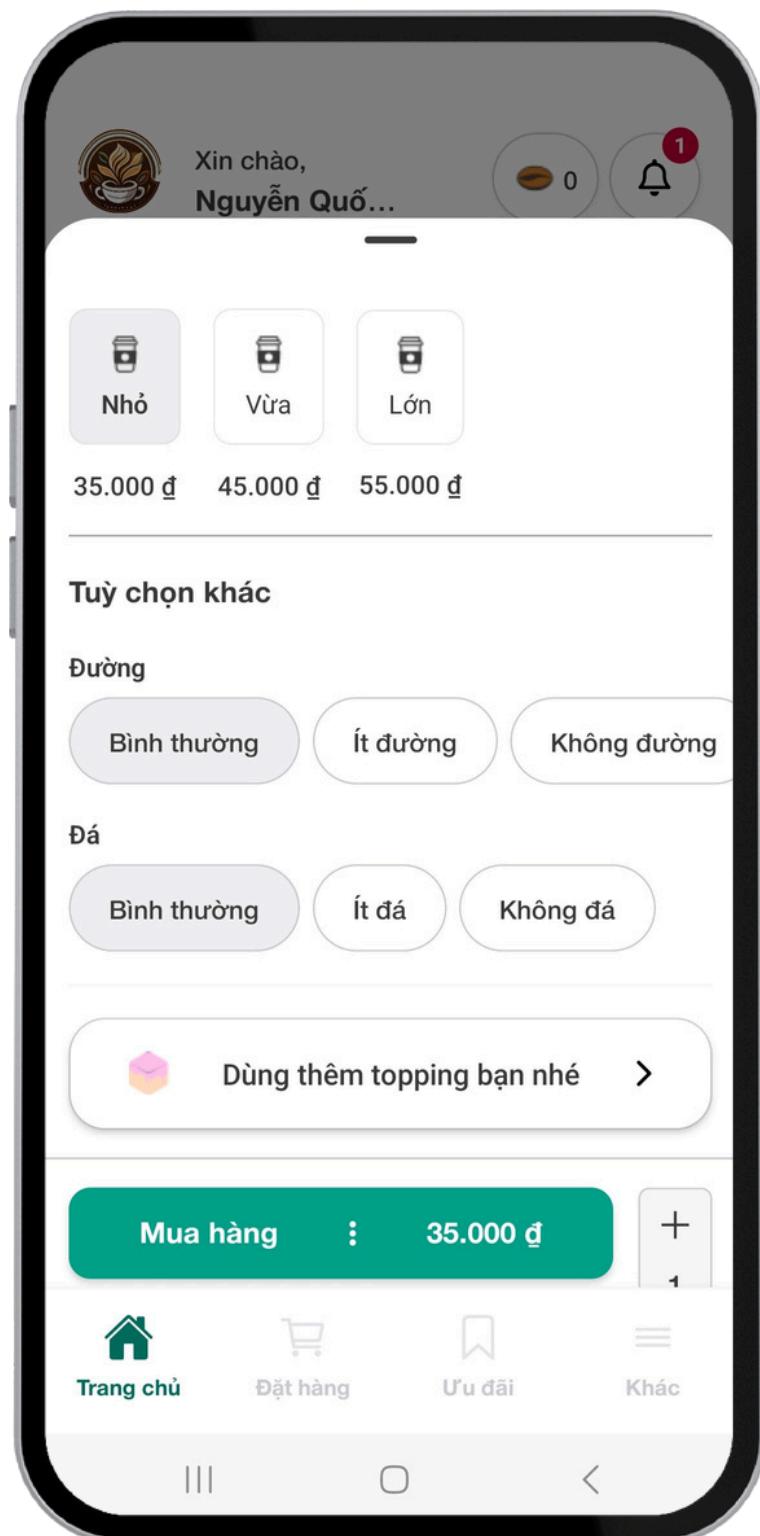
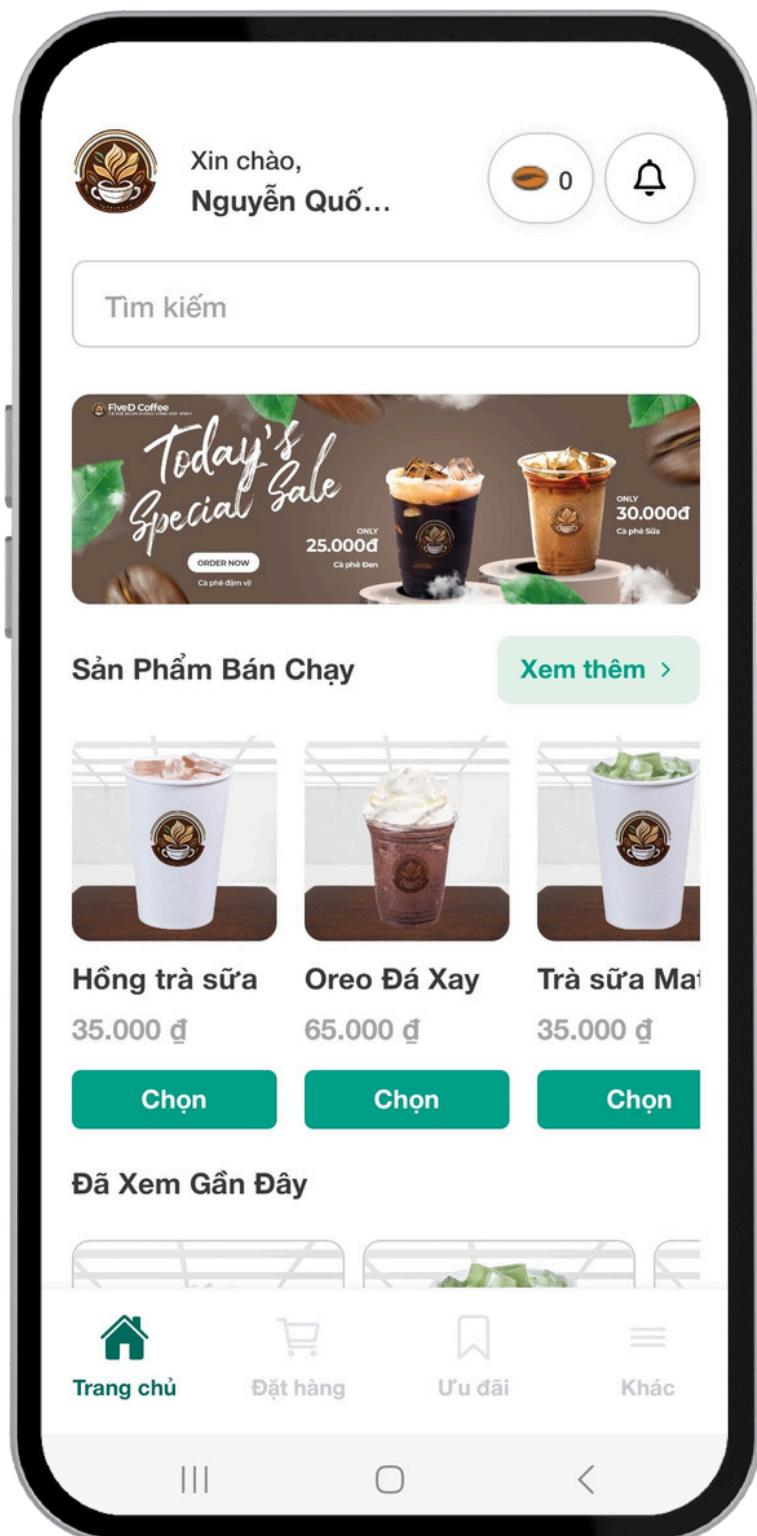
Bước 1: Mở ứng dụng sau khi cài đặt thành công.

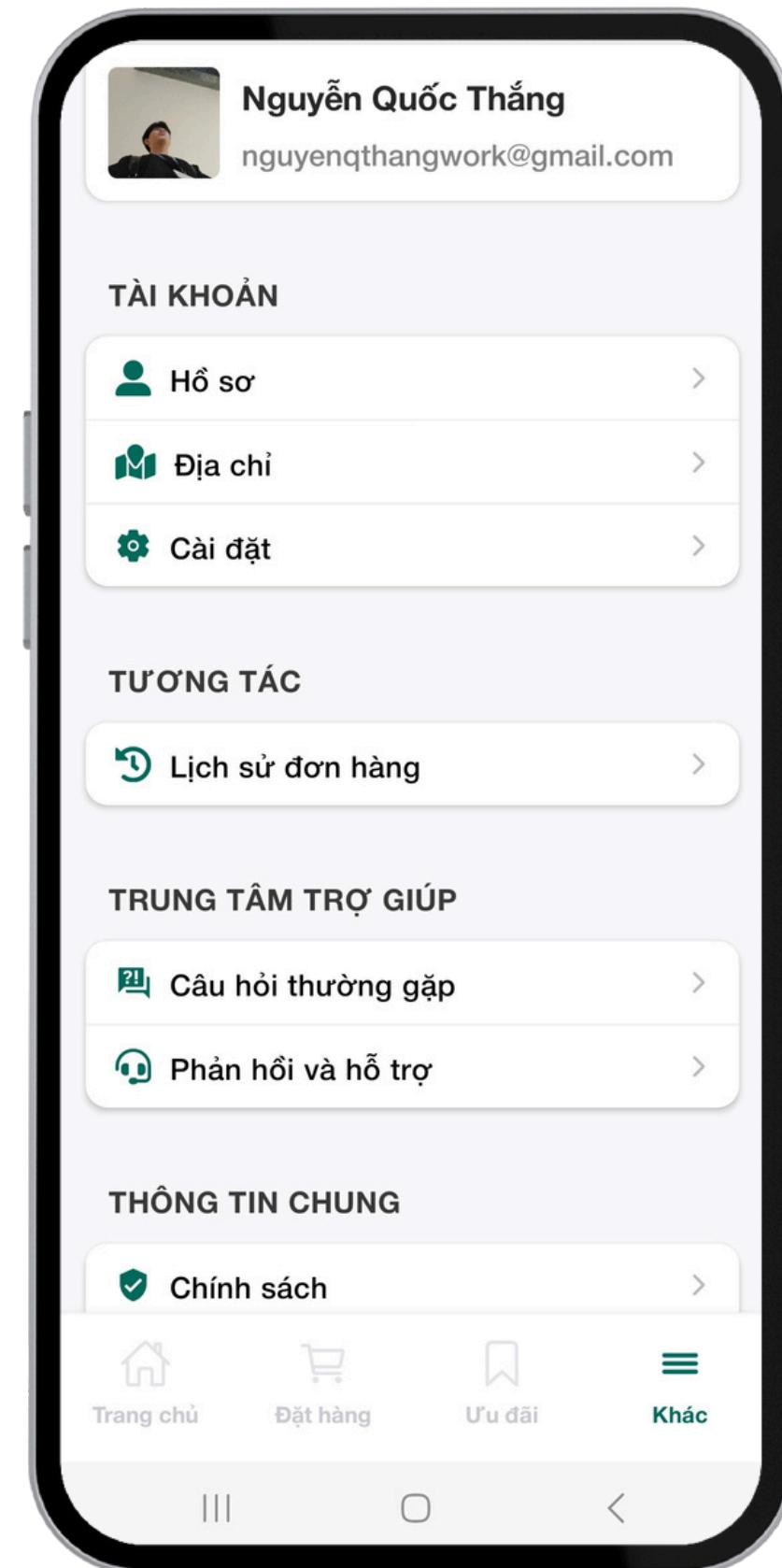
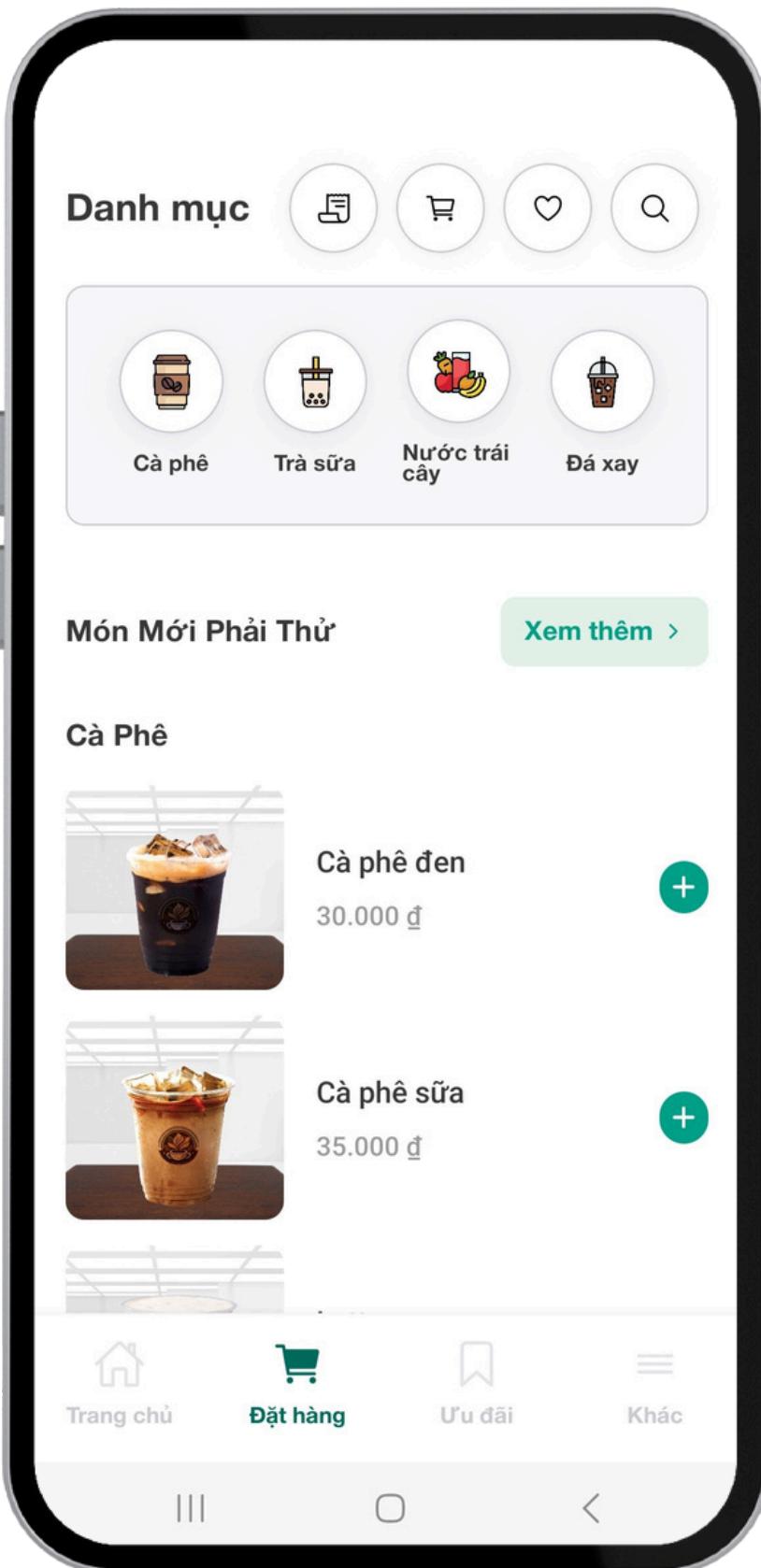
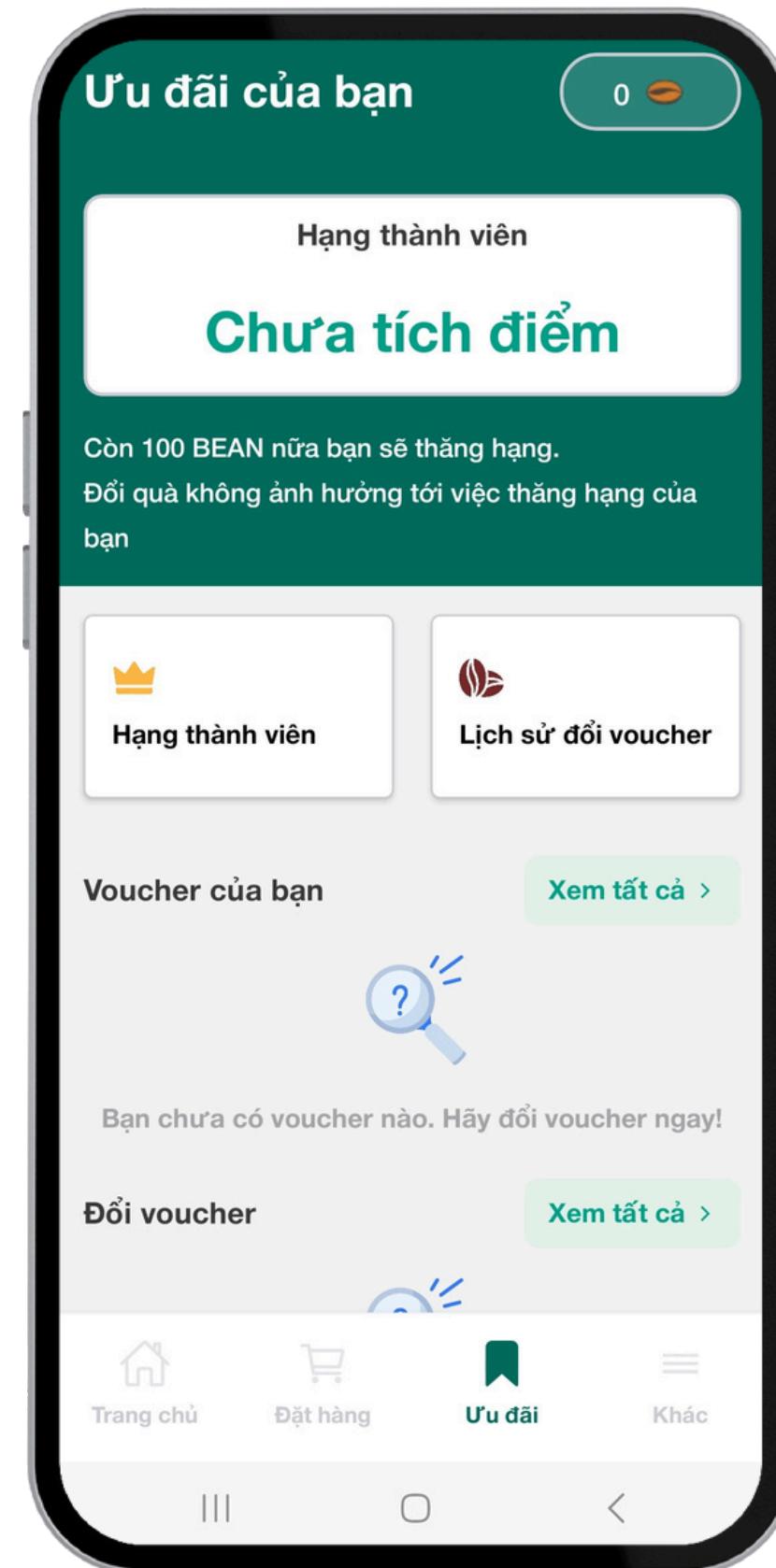
Bước 2: Chọn Đăng nhập.

Bước 3: Nhập thông tin đăng nhập (số điện thoại và mật khẩu) và nhấn Đăng nhập.

Bước 4: Tiến hành truy cập vào các tính năng của ứng dụng







KẾT LUẬN

Kết luận

Dự án “Ứng dụng quản lý Coffee Shops” đã được hoàn thành với những thành tựu đáng kể. Ứng dụng đã đáp ứng được các yêu cầu cơ bản về quản lý thông tin sản phẩm, đơn hàng, nhân viên, kho nguyên liệu, doanh thu, khuyến mãi và khách hàng. Hệ thống cung cấp giao diện trực quan, dễ sử dụng cho cả người dùng và quản trị viên, giúp đơn giản hóa các thao tác quản lý và mang lại trải nghiệm tích cực cho khách hàng.

TRÌNH BÀY TRÊN ỨNG DỤNG THỰC



**CẢM ƠN CÔ
ĐÃ LẮNG NGHE!**

