

ỦY BAN NHÂN DÂN THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC SÀI GÒN
KHOA CÔNG NGHỆ THÔNG TIN



ĐỒ ÁN MÔN HỌC: CÔNG NGHỆ PHẦN MỀM
ĐỀ TÀI: FE DEV (React Native + React.js) chuyên nghiệp
xử lý giao diện đặt hàng (trên Web, Mobile)

GVHD: Nguyễn Quốc Huy.

Lớp: DCT122C5.

Nhóm 4:

Trương Văn Hậu 3122411050
Nguyễn Quốc Hùng 3122411060

Thành phố Hồ Chí Minh, tháng 11 năm 2025

LỜI CAM ĐOAN

Tôi xin cam đoan nội dung được trình bày trong đồ án “**FE DEV (React Native + React.js) chuyên nghiệp xử lý giao diện đặt hàng (trên Web, Mobile)**” là do chúng tôi nghiên cứu, tìm hiểu và phát triển dưới sự dẫn dắt của GV: Nguyễn Quốc Huy. Đồ án không sao chép từ các tài liệu, công trình nghiên cứu của người khác mà không ghi rõ trong tài liệu tham khảo. Tôi xin chịu trách nhiệm về lời cam đoan này.

LỜI CẢM ƠN

Tôi xin gửi lời cảm ơn tới các thầy cô trường đại học Sài Gòn, đã tận tình giảng dạy và truyền đạt kiến thức trong suốt thời gian vừa qua. Tôi cũng xin được gửi lời cảm ơn đến các thầy cô trong bộ môn "Công Nghệ Phần Mềm" cũng như khoa Công Nghệ Thông Tin đã mang lại cho tôi những kiến thức vô cùng quý giá và bổ ích trong quá trình học tập tại trường.

Đặc biệt xin chân thành cảm ơn thầy giáo, giảng viên: Nguyễn Quốc Huy, người đã định hướng, giúp đỡ, trực tiếp hướng dẫn và tận tình chỉ bảo tôi trong suốt quá trình nghiên cứu, hoàn thiện đồ án này.

TP.Hồ Chí Minh, ngày tháng 9 năm 2025.

PRODUCT REQUIREMENTS DOCUMENT (PRD)

Xác định vấn đề

Quy trình đặt hàng phức tạp → Người dùng mất nhiều bước để hoàn tất đơn hàng từ điện thoại hoặc web

Thiếu tính minh bạch → Thời gian chuẩn bị, xác nhận đơn và giao hàng không được hiển thị rõ ràng, dễ gây mất niềm tin

Không có theo dõi thời gian thực → Khách hàng không biết trạng thái đơn hàng (đang chuẩn bị, đã giao, bị hủy...)

Xác minh giao hàng không an toàn → Không có cơ chế xác minh khi nhận hàng

Cách tiếp cận tổng thể

Nền tảng đa kênh → Web (React.js) + Mobile App (React Native/Expo)

Quy trình đặt hàng tối ưu → Menu → Giỏ hàng → Thanh toán (MoMo) trong 3 bước

Quản lý đơn hàng realtime → Cập nhật trạng thái: Chờ xử lý → Chuẩn bị → Giao hàng → Hoàn tất

Xác minh giao hàng an toàn → OTP hoặc mã QR khi giao nhận

Admin Dashboard → Quản lý menu, đơn hàng, drone, khách hàng

Kịch bản trải nghiệm

- Khách hàng mở ứng dụng FastFood (Web/Mobile)
- Đăng nhập/Đăng ký tài khoản
- Duyệt menu theo danh mục (Cơm, Mì, Đồ uống...)
- Chọn món ăn → xem thông tin, giá cả, hình ảnh
- Thêm vào giỏ hàng với số lượng tùy chỉnh
- Xem giỏ hàng → điều chỉnh đơn hàng
- Nhập thông tin giao hàng → Địa chỉ, SĐT, ghi chú
- Chọn phương thức thanh toán → MoMo
- Thanh toán online → Chuyển hướng sang ứng dụng MoMo
- Xác nhận thanh toán thành công → Quay về ứng dụng
- Xem trạng thái đơn hàng realtime (Tính năng sắp có)

Chờ xử lý (Admin xác nhận)

Chuẩn bị (Nhà bếp làm món)

Sẵn sàng giao nhận

Đơn hàng hoàn tất → Thông báo

Mục tiêu

Mục tiêu	Chỉ số	Tầm thời gian
Đặt món nhanh	≤ 3 bước	Ngay từ đầu
Thanh toán an toàn	100% transaction success	Ngay từ đầu
Giao hàng an toàn	100% xác minh OTP/QR	v2
Giao diện thân thiện	NPS $\geq 8/10$	v3

Solution Alignment – Key Features

Layer	Công nghệ	Vai trò
Frontend	React.js (Web) + React Native (Mobile)	UI/UX, quản lý giỏ hàng
Backend	Node.js	API, xử lý đơn hàng, thanh toán
Database	Supabase (PostgreSQL)	Lưu user, products, orders
Payment	MoMo API	Xử lý thanh toán online

Auth	Supabase Auth	Đăng nhập, phân quyền (User/Admin)
------	---------------	---------------------------------------

Hệ thống hiện tại đã có:

1. Nền tảng đặt món trực tuyến

Duyệt menu động

Thêm/sửa/xóa items trong giỏ hàng

Lưu giỏ hàng, tính tổng tiền tự động

2. Thanh toán MoMo

Tích hợp MoMo Sandbox API

Xử lý callback thanh toán

3. Supabase database

4. Admin Dashboard

Quản lý menu (thêm/sửa/xóa món, giá cả, hình ảnh)

Quản lý đơn hàng (xem, cập nhật status, hủy)

Quản lý khách hàng (xem profile, lịch sử đơn)

Báo cáo doanh thu, thống kê

Cần phát triển thêm:

- Real-time order status (WebSocket)
- OTP/QR verification
- Multi-language support

- Push notifications

Technical Stack

Frontend:

Web: React.js + Vite + TailwindCSS

Mobile: React Native (Expo)

Backend:

Server: Node.js + Express

Database: Supabase (PostgreSQL)

Real-time: Supabase Realtime

Payment: MoMo API

DevOps:

Hosting: Vercel (Web) / EAS (Mobile)

CI/CD: GitHub Actions

Monitoring: Sentry / LogRocket

Key Flows

Operational Flow:

Customer → Duyệt nhà hàng → Chọn món → Giỏ hàng → Thanh toán
→ Theo dõi trạng thái đơn.

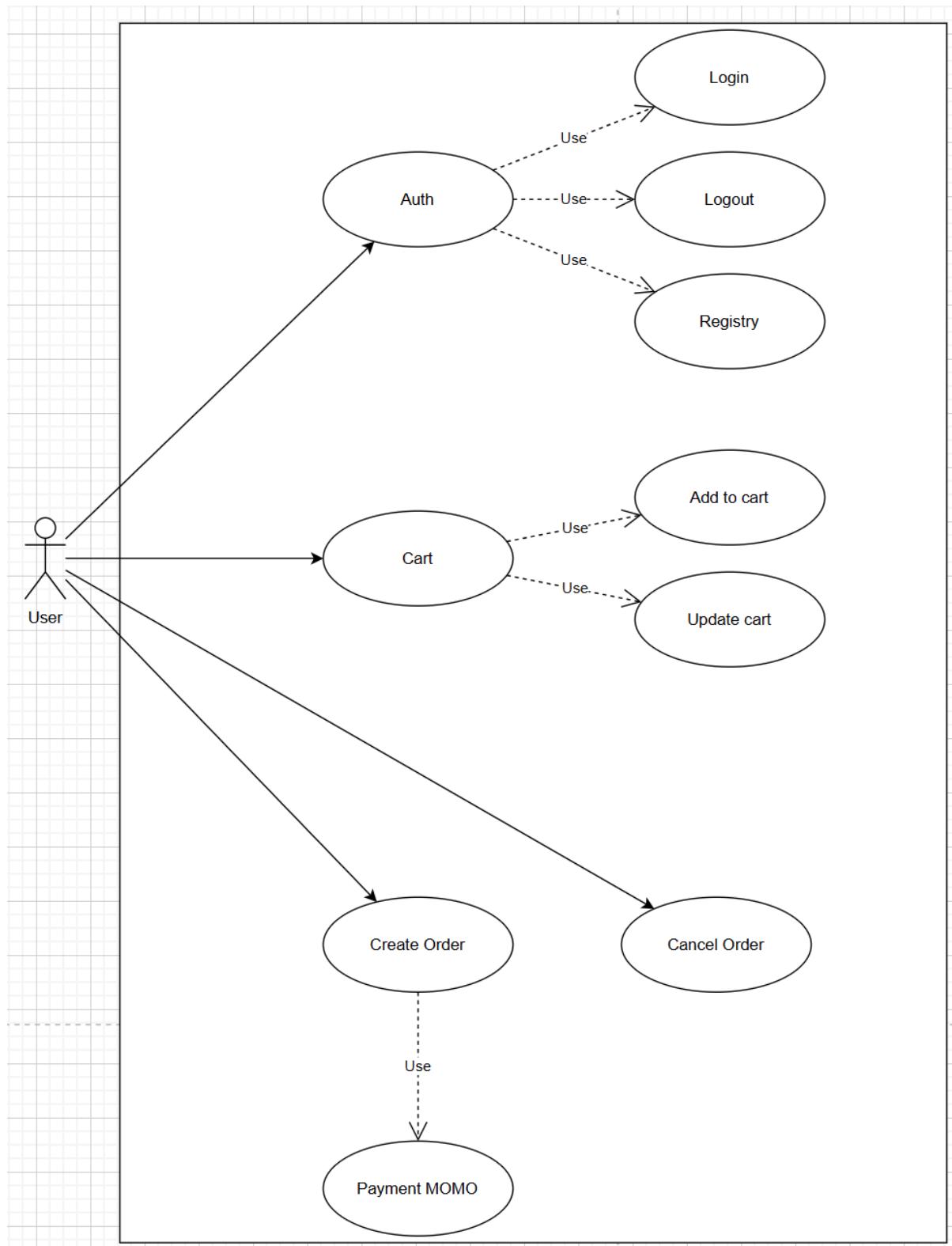
Merchant Flow:

Merchant → Nhận đơn → Xác nhận → Chuẩn bị → Giao hàng → Cập nhật trạng thái.

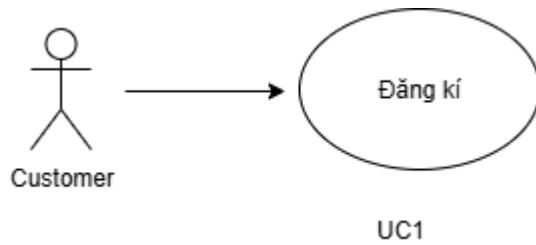
Payment Flow:

Customer → MoMo payment → Thanh toán → Hệ thống phản hồi kết quả (Success / Fail) → Cập nhật đơn hàng.

Use case processing flow



UC #001: Đăng ký

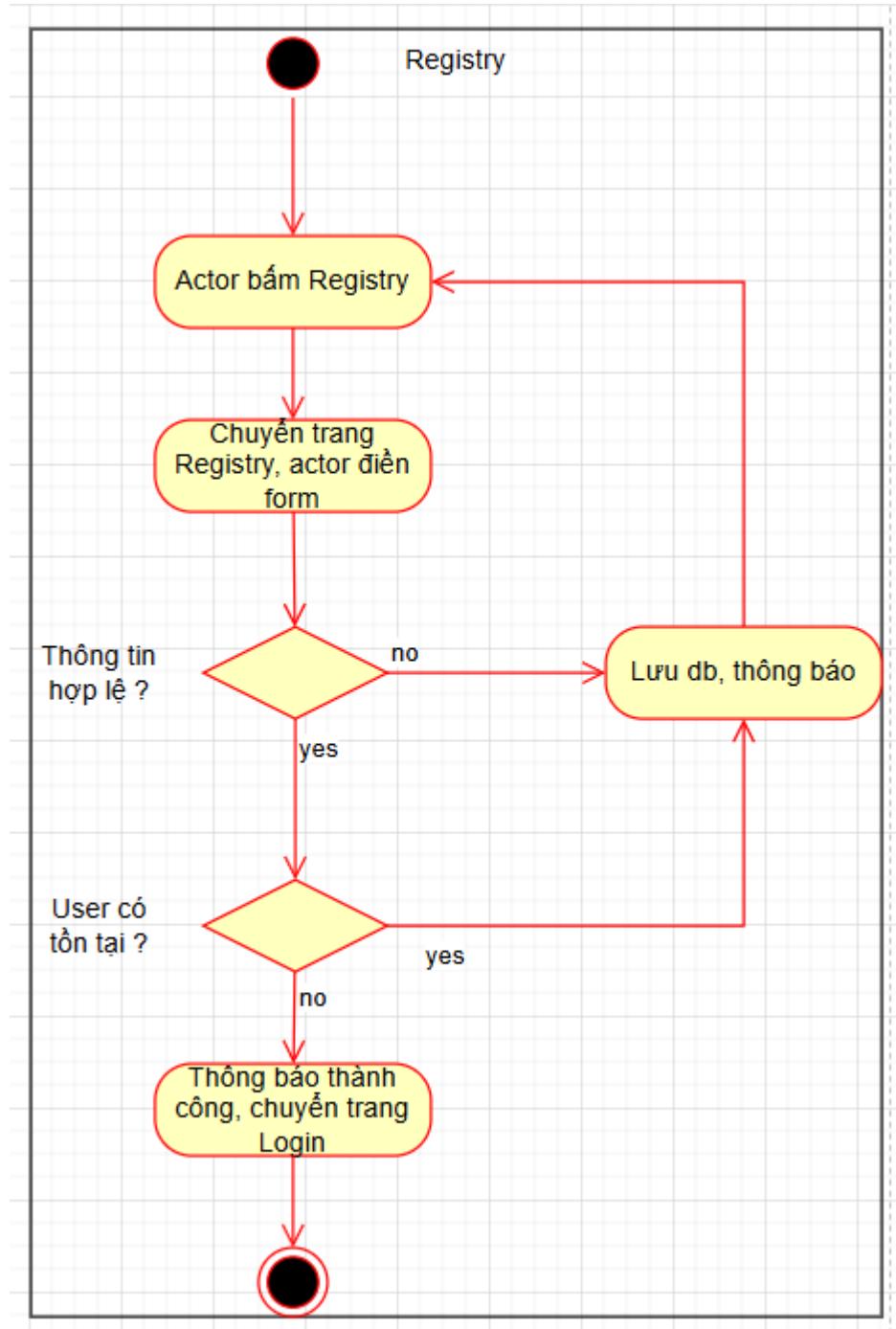


Use Case Number:	UC01	
Use Case Name:	Đăng ký.	
Actor (s):	Customer.	
Maturity:	Focused.	
Summary:	Customer tạo tài khoản mới để sử dụng dịch vụ đặt món.	
Basic Course of Events:	Actor Action	System Response
	1. Customer mở web và chọn “Đăng ký”.	
		2. Hệ thống hiển thị form đăng ký với các trường bắt buộc: (Họ tên, Số điện thoại (bắt đầu là số 0 và 10 ký tự số), Email, Giới tính, Ngày sinh, Mật khẩu, Xác nhận mật khẩu).
	3. Customer nhập thông tin đăng ký. A1, E1, E2, E3.	
	4. Customer nhấn “Tiếp tục”.	

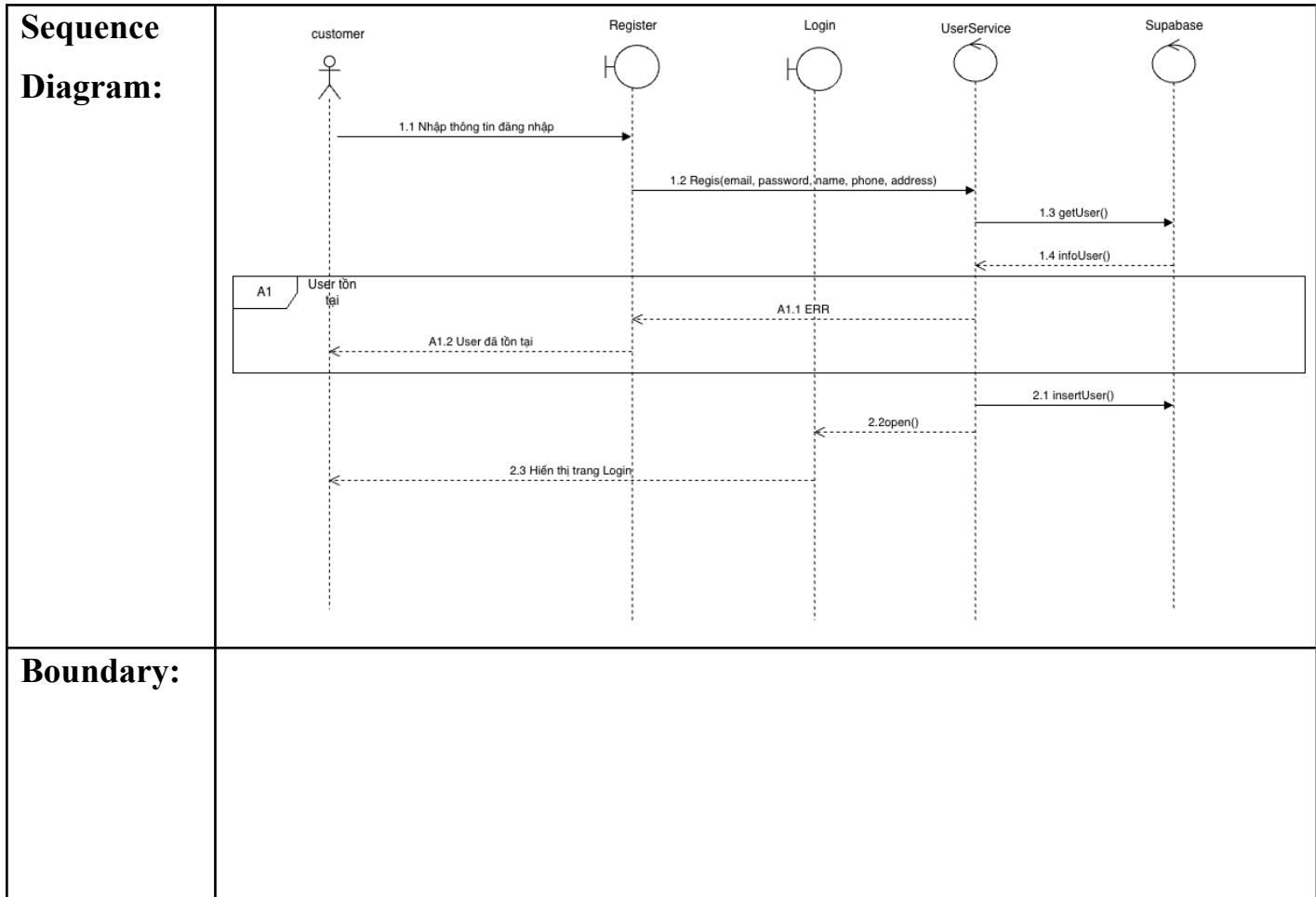
		5. Hệ thống kiểm tra (Email/SĐT chưa tồn tại). E4
		9. Hệ thống hiển thị thông báo “Đăng ký thành công” và chuyển actor đến màn hình đăng nhập.
		10. Usecase kết thúc tại đây
Alternative Path:	A1. Customer hủy đăng ký.	
	Actor Action	System Response
	A1.1. Customer hủy bỏ đăng ký	
		A1.2. Hệ thống quay về trang chủ
Exception Path:	E1. Thông tin bị bỏ trống → thông báo “Vui lòng điền đầy đủ thông tin” và yêu cầu nhập.	
Extension Points:	UC02 – Đăng nhập	
Triggers:	Actor muốn sử dụng Web và chưa có tài khoản.	
Assumptions:	-	
Preconditions:	Customer chưa có tài khoản	
Post Conditions:	Customer có tài khoản mới Customer có thể đăng nhập và sử dụng các tính năng đặt món.	
Reference: Business Rules	Một số điện thoại/Email chỉ được đăng ký một tài khoản.	
Author (s):		
Date:		

Activity

Diagram:

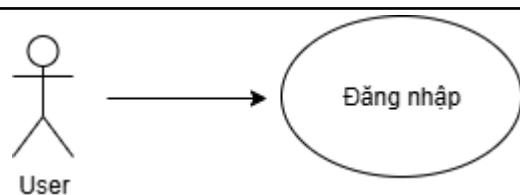


Sequence Diagram:



Boundary:

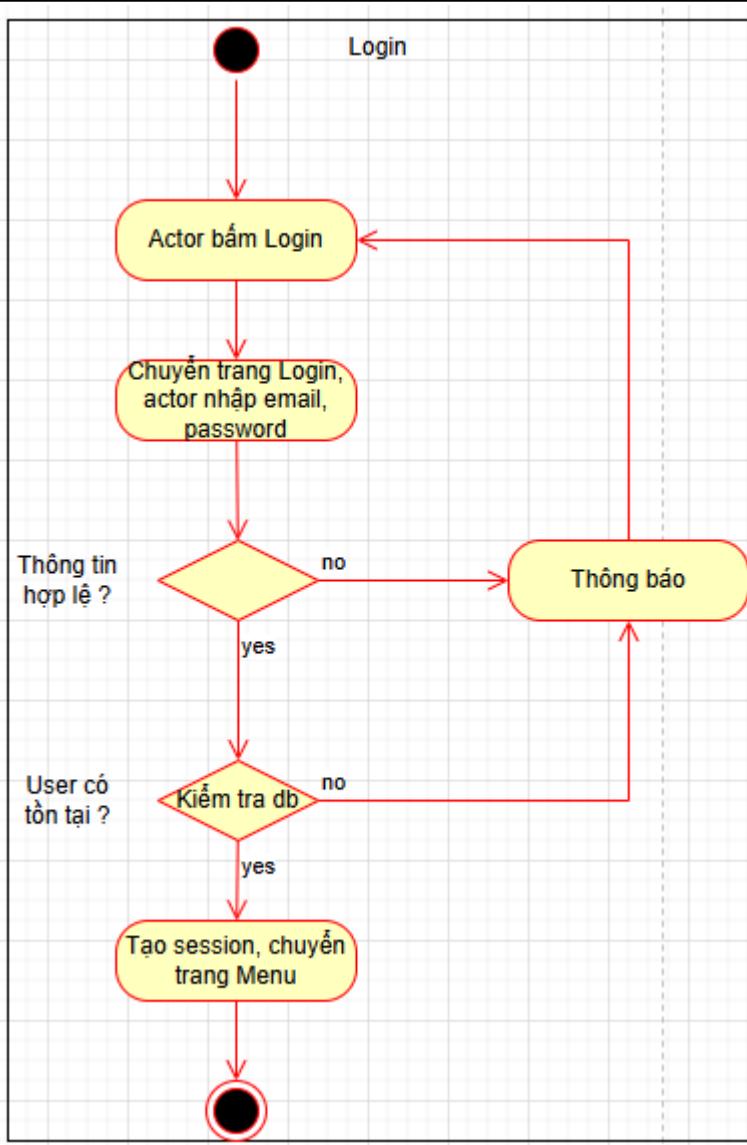
UC #002: Đăng nhập



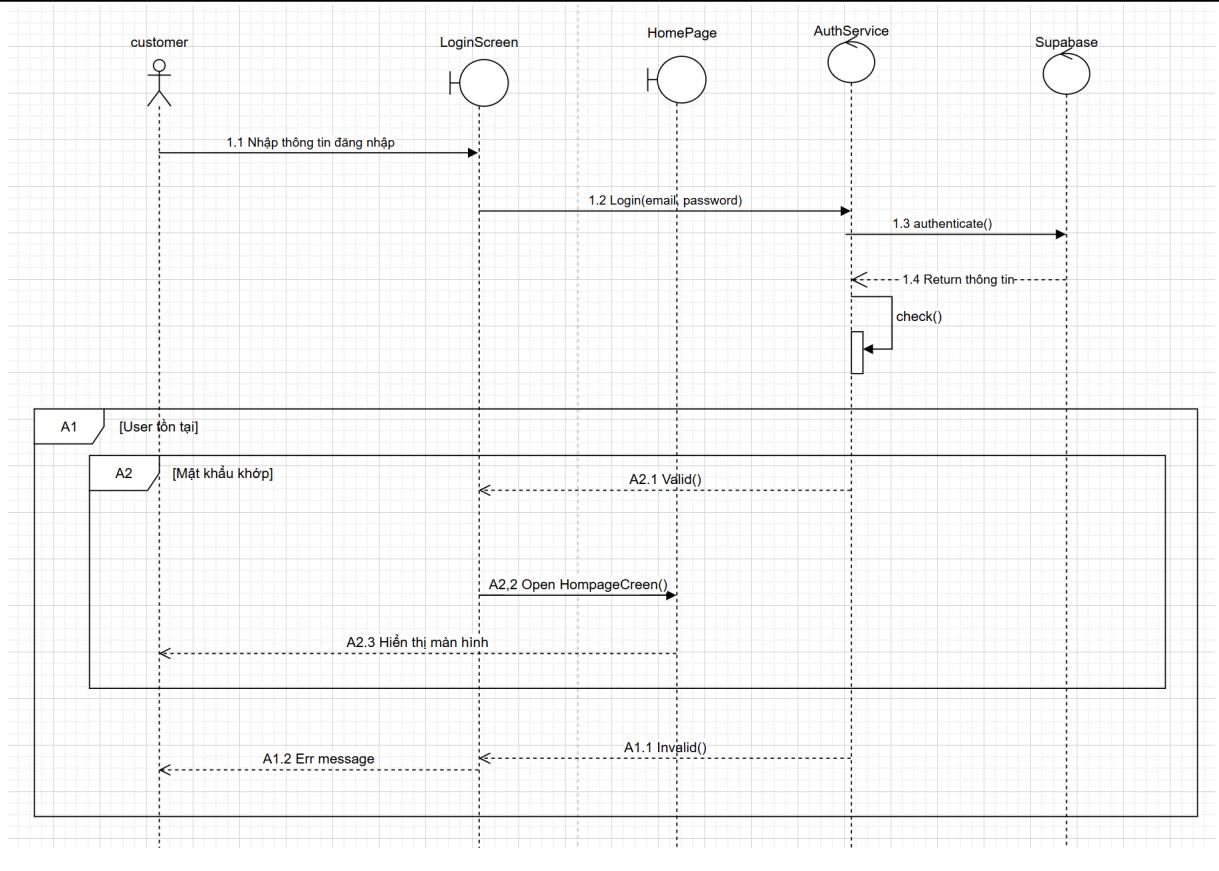
Use Case Number:	UC#002	
Use Case Name:	Đăng nhập	
Actor (s):	Customer	
Maturity:	Focused.	
Summary:	Actor đăng nhập vào hệ thống	
Basic Course of Events:	Actor Action	System Response
	1. Actor nhập Email và mật khẩu.	
		2. Hệ thống kiểm tra thông tin đăng nhập. E1, E2.
		3. Hệ thống chuyển hướng actor đến trang chủ
Alternative Path:	None	
Exception Path:	<p>E1. Nếu Email không có trong cơ sở dữ liệu:</p> <ul style="list-style-type: none"> • Hệ thống thông báo lỗi “Email không tồn tại” và đưa actor trở lại màn hình đăng nhập. <p>E2. Nếu mật khẩu không khớp:</p> <ul style="list-style-type: none"> • Hệ thống thông báo lỗi “Mật khẩu không chính xác” và đưa actor trở lại màn hình đăng nhập. 	
Extension Points:	None	
Triggers:	Actor truy cập vào hệ thống mà không có phiên làm việc (session). Phiên làm việc của actor đã hết hạn hoặc actor đã đăng xuất	
Assumptions:	Actor đã có Email và mật khẩu hợp lệ để đăng nhập vào hệ thống.	

	Hệ thống đang hoạt động bình thường và có thể truy cập cơ sở dữ liệu để xác thực thông tin.
Precondition s:	Actor chưa có phiên làm việc và đang ở trang Đăng nhập.
Post Conditions:	Actor có một phiên làm việc đang hoạt động và được chuyển đến trang chủ với các tính năng phù hợp.
Reference: Business Rules	None
Author (s):	
Date:	

**Activity
Diagram:**



Sequence

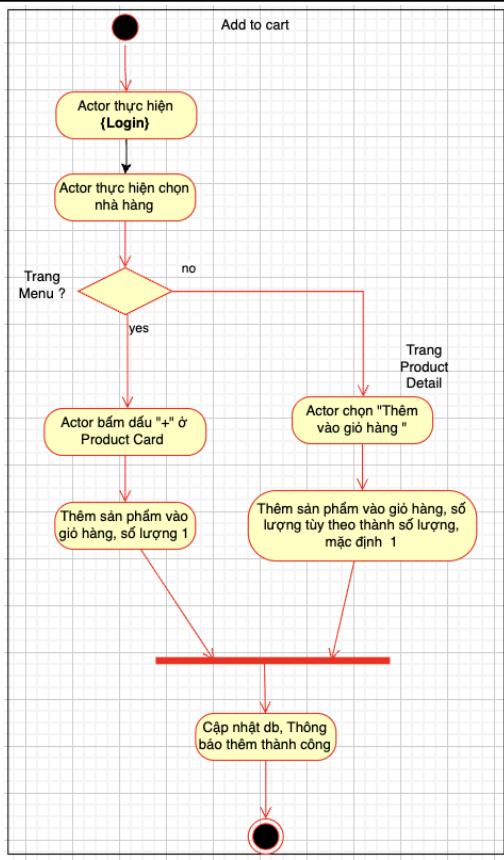


UC #003 : Giỏ hàng

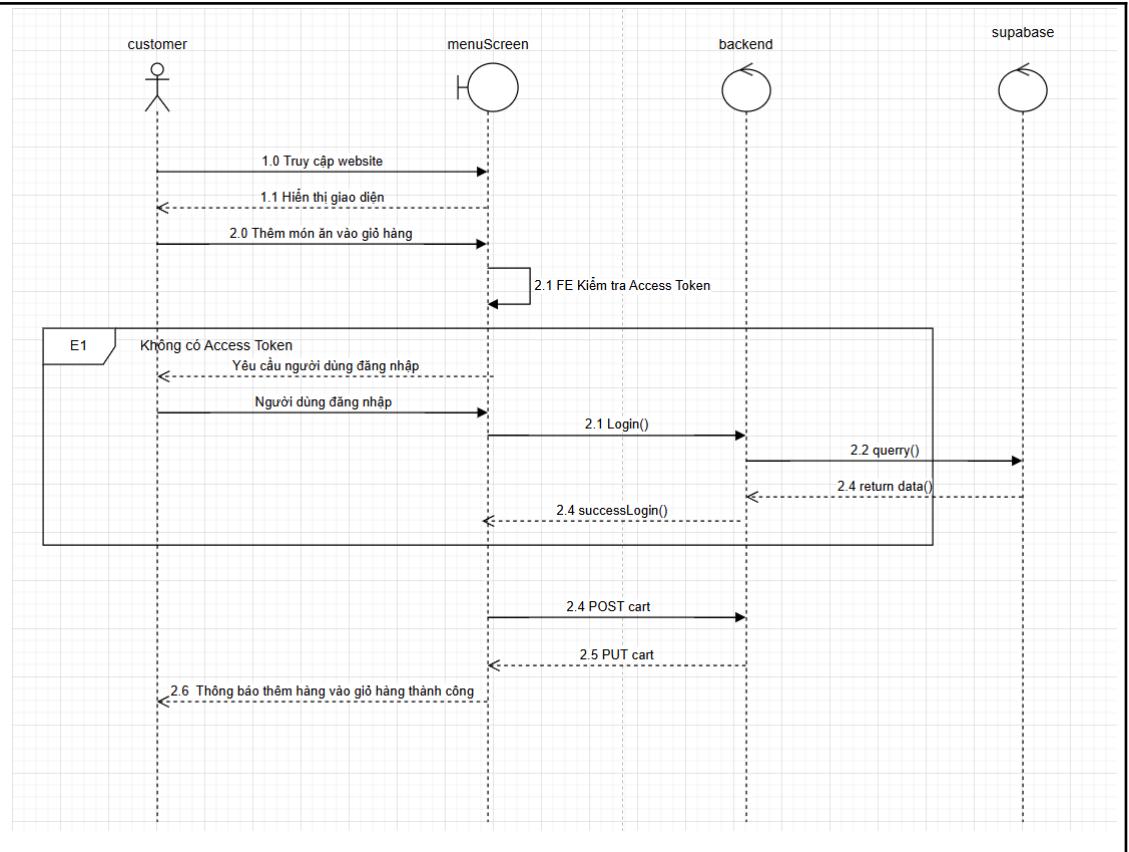
<pre> useCaseDiagram participant Customer participant GiỎ_hÀNG Customer --> GiỎ_hÀNG </pre>	
Use Case Number:	UC #003
Use Case Name:	Giỏ hàng
Actor (s):	Customer
Maturity:	Focused

Summary:	Use case này mô tả toàn bộ quy trình quản lý giỏ hàng của Customer, bao gồm: thêm sản phẩm, cập nhật/xóa sản phẩm	
Basic Course of Events:	Actor Action	System Response
	1. Buyer nhập vào sản phẩm trên trang danh mục sản phẩm. A1	
		2. Hệ thống hiển thị thông tin chi tiết sản phẩm.
	3. Buyer lựa chọn số lượng và “Thêm vào giỏ hàng”, từ trang chi tiết sản phẩm.	
	8. Buyer truy cập giỏ hàng.	
		9. Hệ thống hiển thị danh sách sản phẩm trong giỏ tổng chi phí
	10. Buyer cập nhật số lượng sản phẩm trong giỏ(tăng, giảm sản phẩm). A2	
		11. Hệ thống cập nhật giỏ và làm mới bảng thông tin tóm tắt.
	12. Buyer tiến hành checkout.	
		13. Hệ thống bắt đầu quy trình thanh toán. UseCase kết thúc tại đây.
Alternative Path:	A1. Buyer thêm sản phẩm từ Catalog	
	Actor Action	System Response
	A1.1. Buyer chọn “Thêm vào giỏ hàng”	
		A1.2. Hệ thống quay lại Bước 4

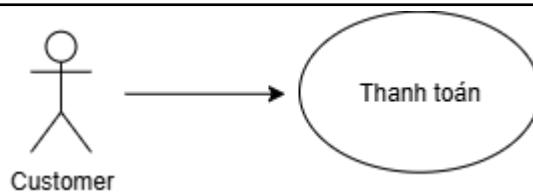
	A2. Buyer xóa một sản phẩm khỏi giỏ.						
	<table border="1"> <thead> <tr> <th>Actor Action</th><th>System Response</th></tr> </thead> <tbody> <tr> <td>A2.1. Buyer chọn “Xóa” một sản phẩm trong giỏ.</td><td></td></tr> <tr> <td></td><td>A2.2. Hệ thống quay lại bước 9</td></tr> </tbody> </table>	Actor Action	System Response	A2.1. Buyer chọn “Xóa” một sản phẩm trong giỏ.			A2.2. Hệ thống quay lại bước 9
Actor Action	System Response						
A2.1. Buyer chọn “Xóa” một sản phẩm trong giỏ.							
	A2.2. Hệ thống quay lại bước 9						
Exception Path:	None						
Extension Points:	None						
Triggers:	Buyer muốn mua sản phẩm và tiến hành đặt hàng.						
Assumptions:	Buyer đã đăng nhập vào hệ thống. Hệ thống hoạt động bình thường và kết nối được với server backend.						
Preconditions:	Buyer có tài khoản và đăng nhập thành công. Danh mục sản phẩm hiển thị đúng dữ liệu.						
Post Conditions:	None						
Reference:	Một sản phẩm có thể được thêm nhiều lần vào giỏ hàng						
Business Rules							
Author (s):	XXXXXX						
Date:							
Activity Diagram:							



**Sequence
Diagram:**



UC #004: Thanh toán

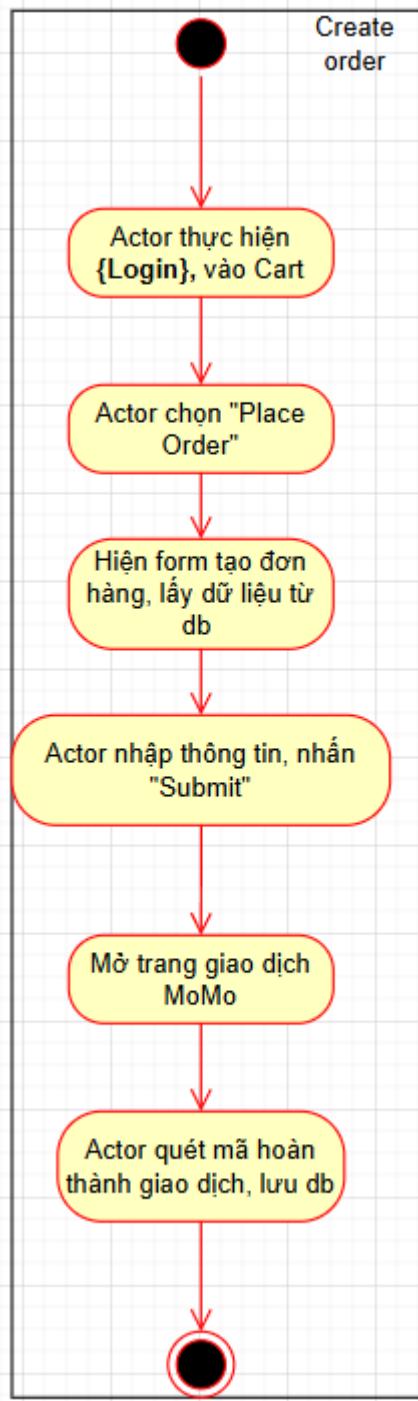


Use Case Number:	UC #004
Use Case Name:	Thanh toán

Actor (s):	Customer	
Maturity:	Focused	
Summary:	Use case này mô tả quy trình thanh toán đơn hàng sau khi Customer đã xác nhận giỏ hàng, bao gồm nhập thông tin giao hàng, lựa chọn phương thức thanh toán, xác nhận đơn hàng và xử lý kết quả thanh toán.	
Basic Course of Events:	Actor Action	System Response
	1. Customer nhấn nút “Thanh toán” từ giỏ hàng.	
		3. Hệ thống hiển thị form thông tin thanh toán: thông tin sản phẩm. hiển thị tổng số tiền phải trả.
	5. Customer chọn thanh toán	
		hệ thống nhận, xử lý backend với momo, mở URL MoMo
	Người dùng trả tiền cho MoMo A1 A2	MoMo xử lý tiền, trả về ipn cho backend xử lý so sánh

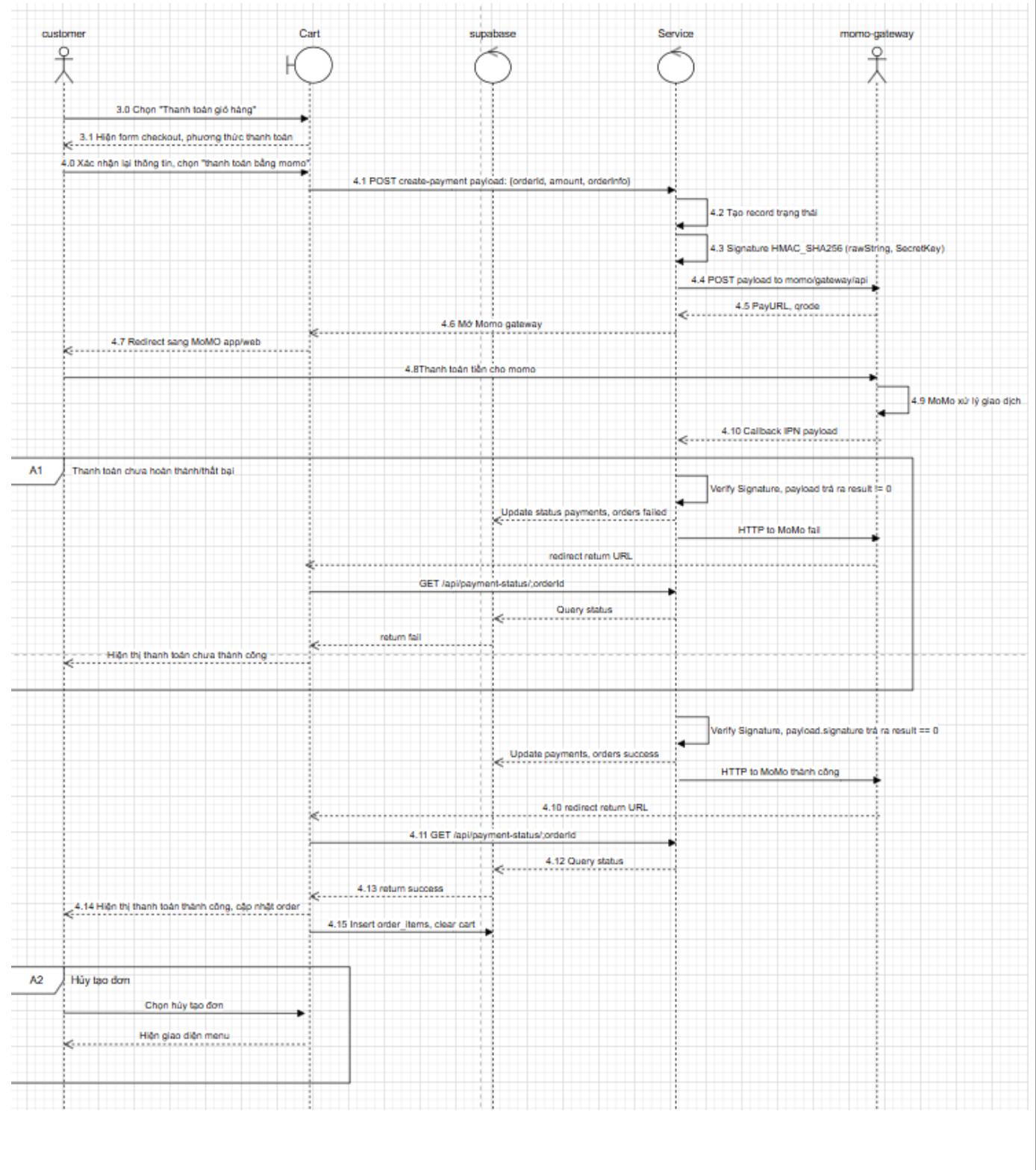
		Hệ thống hoàn thành xử lý, thông báo User thành công thanh toán đơn hàng
	người dùng xác nhận Usecase kết thúc tại đây.	
Alternative Paths:	A1. Thanh toán bị hoãn bởi người dùng	
	Actor Action	System Response
	A1.1 User cancel/không thanh toán tiền cho momo	
		A1.2 MoMo xử lý tiền, trả ipn cho backend so sánh xử lý
		A1.3 Hệ thống hoàn thành xử lý, thông báo đơn hàng bị hoãn chưa thanh toán thành công
	người dùng xác nhận Usecase kết thúc tại đây	
	A2 Huỷ thanh toán	
	Actor Action	System Response
	A2.1. Customer huỷ thanh toán	
		A2.2. Hệ thống quay lại bước 6 của Basic
Exception Paths:	None	

Extension Points:	
Triggers:	“Khách hàng nhấn ‘Đặt hàng/Thanh toán’ từ giỏ hàng để bắt đầu quy trình.”
Assumptions:	Nếu Online Payment: công thanh toán hoạt động, callback thành công.
Preconditions:	Khách hàng đã nhập địa chỉ giao hàng.
Post Conditions:	
Reference: Business Rules	Online Payment cần callback từ công thanh toán để xác nhận kết quả.
Author(s):	
Date:	

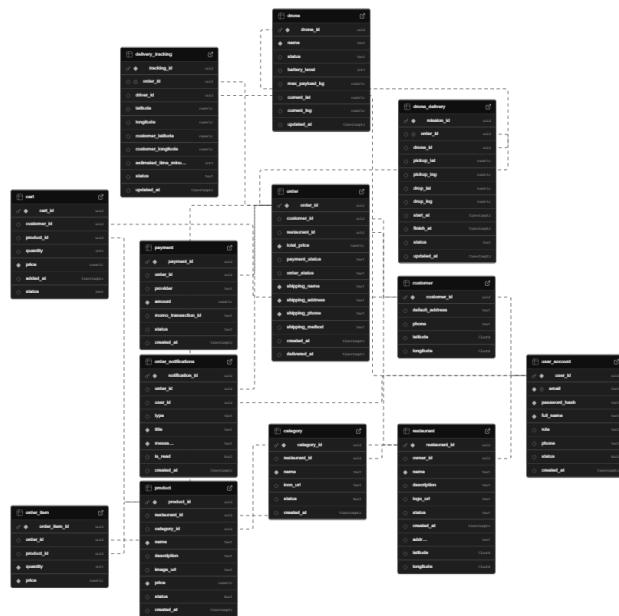


Activity Diagram:

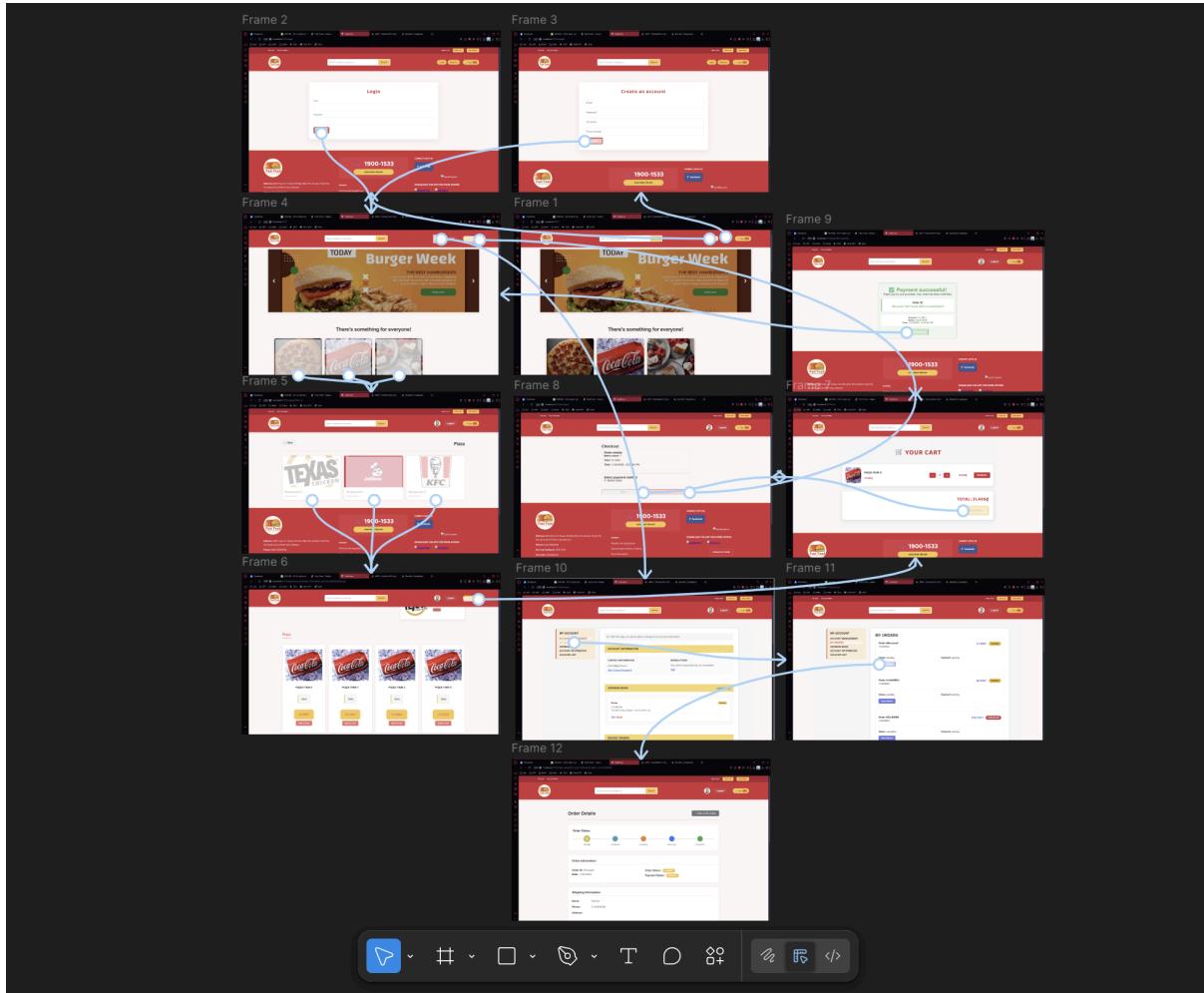
Sequence Diagram:

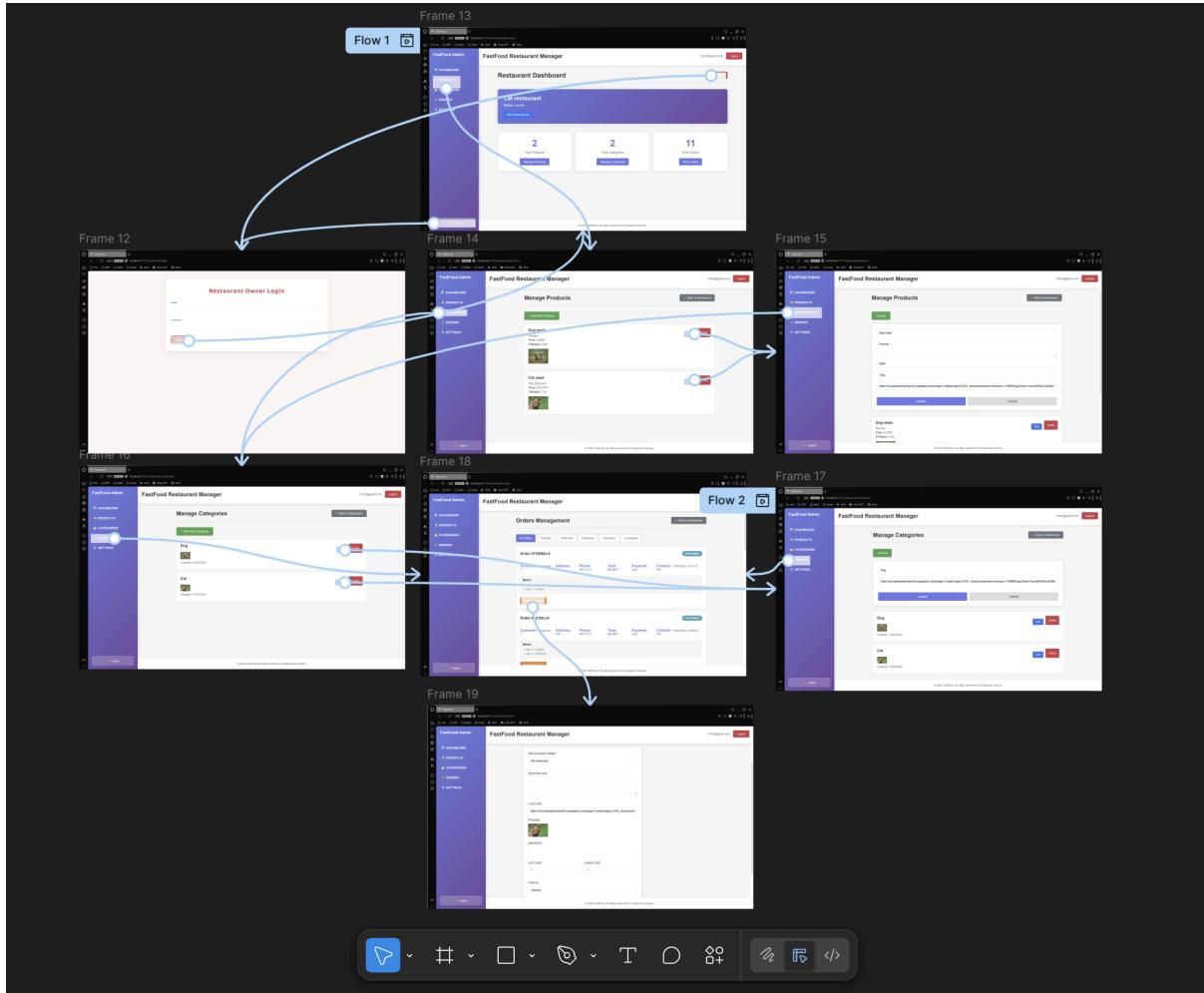


Cơ sở dữ liệu

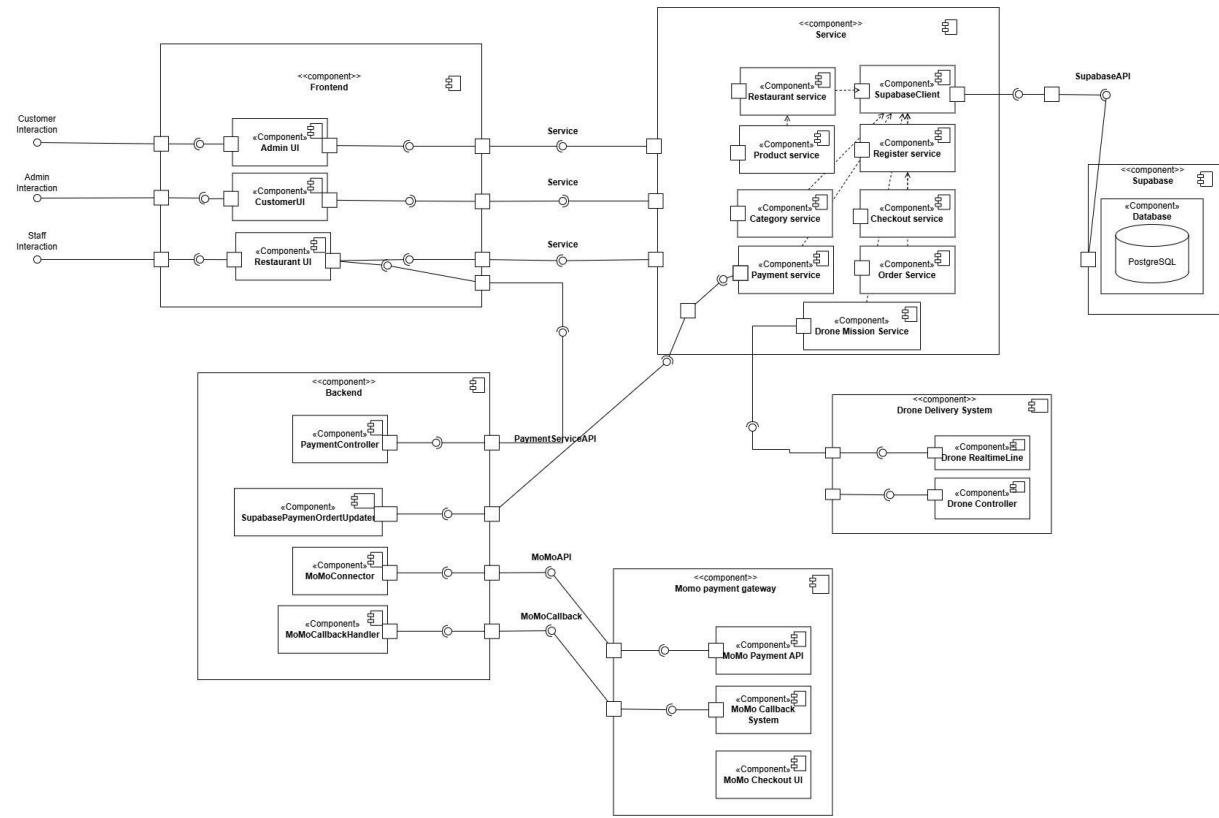


Giao diện

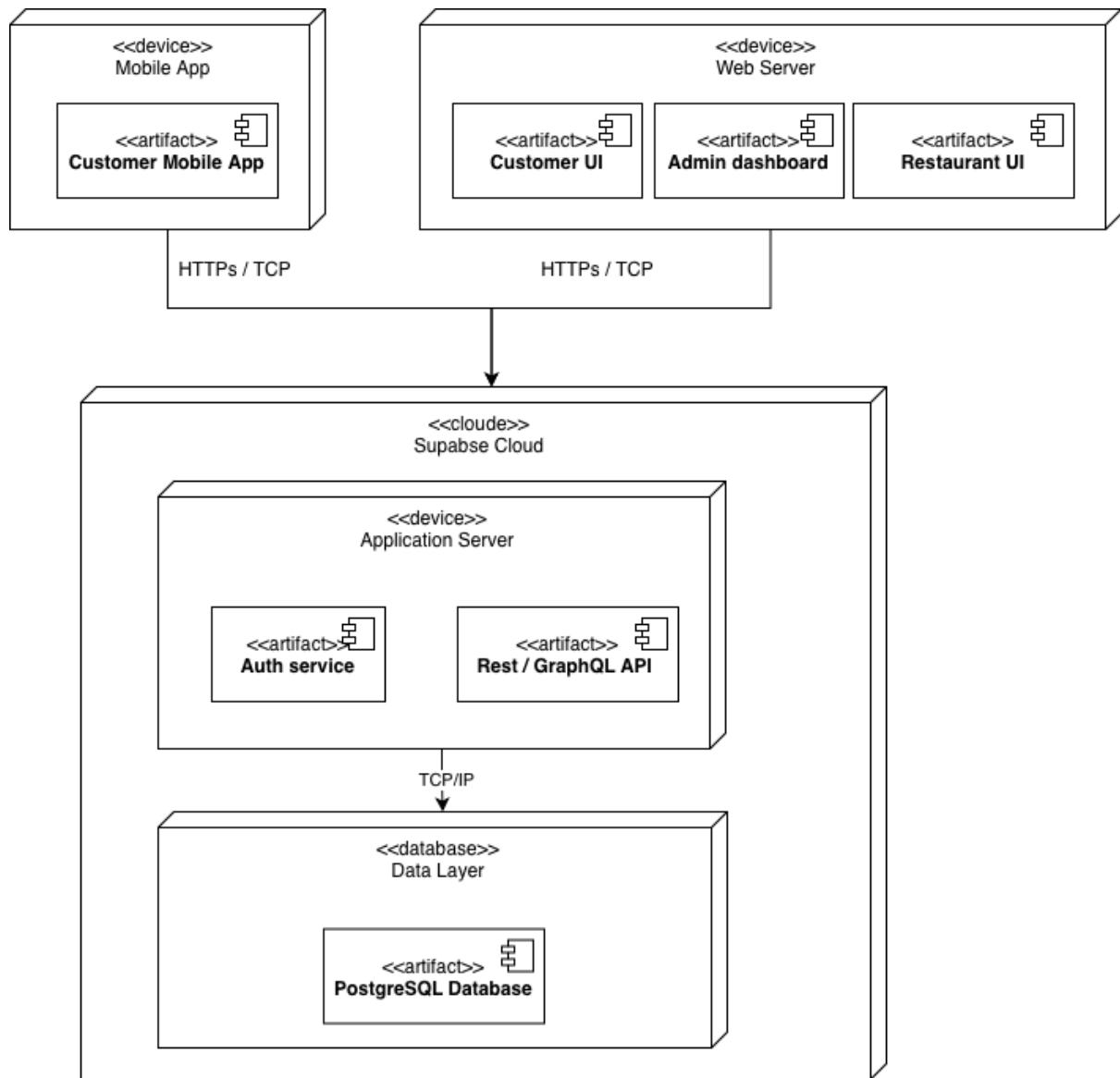




Component



Deployment



ERD

