# Tổng quan phần mềm

## Bối cảnh dự án (Task 1.1)

* Mục tiêu dự án: Thiết kế một hệ thống quản lý bán hàng POS (Point of Sale) cho tiệm cà phê. Hệ thống nhằm cải thiện quy trình gọi món và cung cấp các lựa chọn thanh toán linh hoạt, từ đó nâng cao trải nghiệm của khách hàng.
* Các bên liên quan chính:
  + Khách hàng: Người đến thưởng thức đồ uống tại tiệm hoặc đặt mang đi.
  + Nhân viên: Người nhận đơn hàng, chế biến, phục vụ và thực hiện thanh toán.
  + Quản lý: Người điều hành và giám sát toàn bộ hoạt động của tiệm cà phê.
* Kỳ vọng:
  + Hệ thống cho phép đặt món qua website mà không cần đăng nhập.
  + Hỗ trợ nhiều phương thức thanh toán, bao gồm cả trực tuyến và trực tiếp.
* Phạm vi của dự án:
  + Phát triển hệ thống quản lý bán hàng POS cho tiệm cà phê.
  + Hỗ trợ đặt món nhanh thông qua website, không yêu cầu đăng nhập.
  + Hỗ trợ thanh toán trực tuyến tiện lợi.
  + Lưu trữ thông tin truy cập trong 30 ngày cho người dùng mới để hiển thị lịch sử hóa đơn.
  + Hiển thị chi tiết hóa đơn bao gồm:

∗ Mã hóa đơn

∗ Số tiền

∗ Tình trạng món

## Quy trình phát triển hệ thống web bán cà phê theo mô hình POS

### Thu thập yêu cầu

* Lên lịch cuộc họp với khách hàng: Thảo luận về yêu cầu chi tiết của hệ thống, bao gồm các tính năng cần thiết như quản lý sản phẩm, giỏ hàng, thanh toán, và quản lý đơn hàng.
* Ghi chép và phân tích yêu cầu: Xác định các chức năng quan trọng như đăng nhập, tìm kiếm sản phẩm, xem chi tiết món ăn, thêm vào giỏ hàng và thanh toán.
* Thống kê giao diện người dùng và các tính năng khác: Xác định các yêu cầu về giao diện và chức năng người dùng, bao gồm tính năng đăng nhập, quản lý giỏ hàng, xem thông tin món ăn, và xử lý thanh toán.

### Thiết kế

* Xây dựng sơ đồ kiến trúc mô hình API: Thiết kế các thành phần chính của hệ thống bao gồm:
  + Model: Quản lý dữ liệu và logic nghiệp vụ, bao gồm các lớp và đối tượng xử lý thông tin về sản phẩm, giỏ hàng, đơn hàng và thanh toán.
  + API Controller: Xử lý các yêu cầu HTTP từ người dùng (GET, POST, PUT, DELETE), điều hướng đến các logic phù hợp trong Model và trả về phản hồi dưới dạng JSON hoặc XML.
  + API Routes: Định nghĩa các tuyến đường (routes) để giao tiếp với các Controller thông qua các phương thức HTTP. Mỗi route tương ứng với một hành động cụ thể của người dùng (ví dụ: lấy danh sách sản phẩm, tạo đơn hàng, thanh toán).
* Thiết kế giao diện người dùng: Dựa trên yêu cầu của khách hàng, giao diện người dùng sẽ tương tác với API thông qua các yêu cầu HTTP. Giao diện sẽ hiển thị dữ liệu nhận được từ các API endpoint như danh sách món ăn, giỏ hàng và phương thức thanh toán.
* Xác định các thành phần UI và trải nghiệm người dùng: Giao diện người dùng sẽ sử dụng các phương thức HTTP để tương tác với API, gửi các yêu cầu như tạo đơn hàng, xem giỏ hàng, thanh toán, và nhận phản hồi từ API. Trải nghiệm người dùng sẽ được tối ưu hóa bằng cách hiển thị thông tin từ API một cách mượt mà và nhanh chóng.

### Lập trình

* Sử dụng REST API với NestJS để phát triển ứng dụng web:
  + Xây dựng các thành phần của hệ thống theo mô hình REST API. NestJS sẽ xử lý logic nghiệp vụ (Model) và các yêu cầu HTTP (Controller).
  + REST API sẽ có các endpoint để thao tác với dữ liệu như tạo đơn hàng, lấy thông tin sản phẩm, cập nhật giỏ hàng, và xử lý thanh toán.
  + React frontend sẽ gửi các yêu cầu HTTP tới các API endpoints và hiển thị dữ liệu nhận được từ backend.
* Sử dụng React và Axios để xử lý giao diện động:
  + React sẽ xử lý giao diện người dùng, tương tác với backend thông qua các yêu cầu HTTP với thư viện Axios hoặc Fetch API.
  + Các tính năng như thêm sản phẩm vào giỏ hàng, xóa sản phẩm, và cập nhật giỏ hàng sẽ được thực hiện mà không cần tải lại trang, mang lại trải nghiệm người dùng mượt mà.
* Kết nối với MySQL để quản lý dữ liệu:
  + NestJS sẽ sử dụng các thư viện như TypeORM hoặc Sequelize để kết nối và thao tác với cơ sở dữ liệu MySQL.
  + Dữ liệu sẽ được lưu trữ và truy xuất từ các bảng trong MySQL như ‘products’, ‘orders’ để xử lý các tác vụ liên quan đến đơn hàng, giỏ hàng và sản phẩm.
* Xử lý tương tác người dùng:
  + React sẽ cung cấp giao diện cho người dùng để cập nhật giỏ hàng, kiểm tra tính hợp lệ của các đơn hàng, và thực hiện các thao tác thanh toán.
  + NestJS sẽ nhận các yêu cầu từ frontend và xử lý các thao tác thanh toán qua các phương thức đã chọn như tiền mặt, thẻ tín dụng, v.v.

### Triển khai

* Thiết lập môi trường triển khai:
  + Cấu hình máy chủ để triển khai ứng dụng NestJS, thiết lập cơ sở dữ liệu MySQL và đảm bảo môi trường phù hợp với yêu cầu hệ thống.
  + Cài đặt các phần mềm cần thiết cho môi trường triển khai, bao gồm Node.js, NestJS, và MySQL.
  + Cấu hình các tệp cấu hình của hệ thống như .env cho môi trường sản phẩm.
* Tải mã nguồn lên máy chủ:
  + Đưa mã nguồn của hệ thống lên máy chủ sản phẩm, đảm bảo các tệp cấu hình phù hợp với môi trường thực tế.
  + Sử dụng công cụ như Git để tải mã nguồn và các tệp cần thiết lên máy chủ.
  + Kiểm tra cấu trúc tệp và thư mục của hệ thống sau khi tải lên, đảm bảo mã nguồn được triển khai chính xác.
* Kiểm tra hệ thống trên môi trường sản xuất:
  + Đảm bảo hệ thống hoạt động ổn định, không có lỗi nghiêm trọng trên máy chủ sản phẩm và đáp ứng các yêu cầu của người dùng.
  + Kiểm tra các chức năng chính của ứng dụng như đăng nhập, thanh toán và quản lý giỏ hàng.
  + Theo dõi hiệu suất hệ thống và các lỗi trong log, đảm bảo không có sự cố ảnh hưởng đến người dùng.

### Bảo trì

• Cập nhật và sửa lỗi:

* Thường xuyên thực hiện các bản vá bảo mật để đảm bảo an toàn hệ thống.
* Cập nhật, bổ sung hoặc nâng cấp các tính năng mới theo yêu cầu thực tế từ người dùng hoặc khách hàng.
* Kiểm tra kỹ trước và sau mỗi lần cập nhật để đảm bảo không phát sinh lỗi mới.

## Mô tả yêu cầu chức năng và phi chức năng (Task 1.2)

### Yêu cầu chức năng

* Hiển thị menu: Cung cấp giao diện để khách hàng xem danh sách đồ uống có sẵn.
* Đặt món: Cho phép khách hàng đặt món qua website mà không cần đăng nhập.
* Tính số tiền phải trả: Hệ thống tự động tính tổng số tiền cần thanh toán dựa trên các món đã đặt.
* Lựa chọn phương thức thanh toán: Khách hàng có thể chọn thanh toán trực tuyến hoặc tại quầy.

### Yêu cầu phi chức năng

* Hiệu suất: Hệ thống cần phản hồi nhanh chóng, đặc biệt trong quá trình đặt món và thanh toán.
* Tiện dụng: Giao diện thân thiện, dễ sử dụng và không yêu cầu khách hàng phải đăng nhập để gọi món.
* Bảo mật: Đảm bảo an toàn cho dữ liệu khách hàng, đặc biệt là các thông tin thanh toán và phản hồi.
* Khả năng mở rộng: Hệ thống cần dễ dàng mở rộng để đáp ứng các yêu cầu bổ sung và quy mô hoạt động trong tương lai.
* Bảo trì: Hệ thống cần được thiết kế để dễ dàng bảo trì và cập nhật, đảm bảo tính ổn định lâu dài.

## User Stories

* Là một admin, tôi muốn quản lý danh sách tài khoản để có thể cung cấp tài khoản cho nhân viên.
* Là một admin, tôi muốn chỉnh sửa menu để thay đổi phù hợp với tình hình của quán.
* Là một nhân viên, tôi muốn thay đổi trạng thái của hóa đơn để thông báo với khách hàng rằng các món trong hóa đơn đã được hoàn thành.
* Là một khách hàng, tôi muốn xem menu trực tuyến và có thêm phương thức thanh toán trực tuyến để có thể đặt món mà không cần phải tới quầy.

# Kiến trúc phần mềm

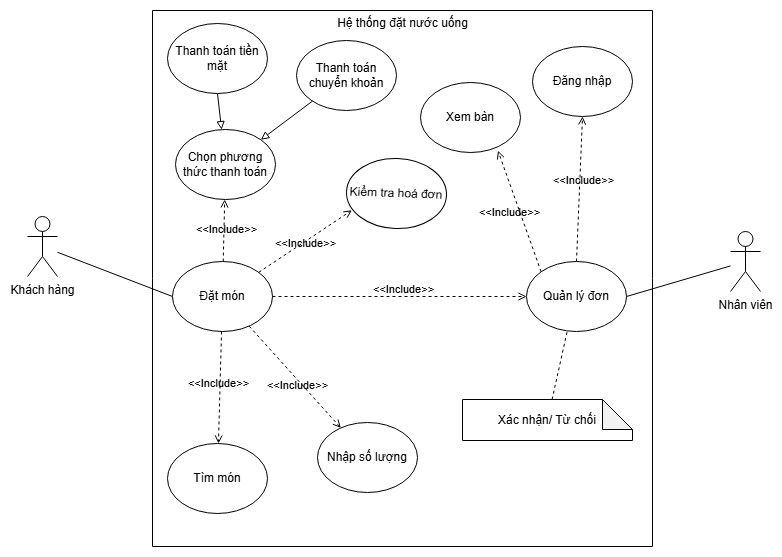
## Use-Case chính (Task 1.2)

## Use-case đặt món trước và mô tả (Task 1.3)

|  |  |
| --- | --- |
| Tên Use Case | UC01 – Đặt món nước |
| Mục tiêu | Cho phép khách hàng chọn món nước từ menu và đặt hàng thông qua hệ thống. |
| Tác nhân (Actors) | * Khách hàng * Nhân viên * Hệ thống đặt món |

|  |  |
| --- | --- |
| Mô tả tóm tắt | Khách hàng duyệt menu, chọn món nước, tuỳ chọn số lượng, chọn phương thức thanh toán, sau đó gửi đơn đặt hàng. Hệ thống tiếp nhận đơn và gửi về quầy pha chế để thực hiện. |
| Tiền điều kiện (Precondi-  tions) | * Hệ thống hoạt động ổn định * Khách hàng đã truy cập vào giao diện đặt món * Menu được hiển thị đầy đủ * Nhân viên đăng nhập vào hệ thống xác nhận đơn |
| Hậu điều kiện (Postcondi-  tions) | * Trạng thái đơn hàng được cập nhật cho khách hàng * Gửi đơn đến bộ phận pha chế |
| Luồng chính (Main Flow) | 1. Khách truy cập giao diện đặt món → Hệ thống hiển thị menu 2. Chọn món → Hiển thị tùy chọn 3. Chọn số lượng → Cập nhật giá 4. Thêm vào giỏ hàng → Giỏ hàng được cập nhật 5. Kiểm tra và xác nhận → Hiển thị thông tin đơn hàng 6. Chọn phương thức thanh toán → Hiển thị các tùy chọn 7. Hoàn tất thanh toán → Hiển thị “Chờ xác nhận” 8. Hệ thống gửi đơn đến nhân viên 9. Nhân viên kiểm tra và xác nhận đơn hàng 10. Gửi đơn đến pha chế |
| Luồng phụ (Alternative  Flows) | * Món hết: không cho đặt, báo lỗi * Thanh toán lỗi: yêu cầu chọn lại * Nhân viên từ chối đơn → Hệ thống thông báo lại cho khách |
| Ngoại lệ (Exceptions) | • Mạng không ổn định: thông báo và thử lại |

|  |  |
| --- | --- |
| Yêu cầu đặc biệt | * Giao diện thân thiện, dễ thao tác * Phản hồi nhanh |
| Tần suất sử dụng | Thường xuyên – Chức năng cốt lõi, được sử dụng nhiều lần mỗi ngày |



Hình 1: Sơ đồ use case: Đặt món nước