**Ứng dụng quản lý tiệm bánh pizza**

1. **Cấu trúc tổng quát**

Gồm 3 tầng:

* Tầng giao diện UI – Presentation Layer (GUI)
* Tầng nghiệp vụ - Business Login Layer (BLL)
* Tầng dữ liệu – Data Access Layer (DAL)

1. **Phân tích tổng quan phần mềm:**

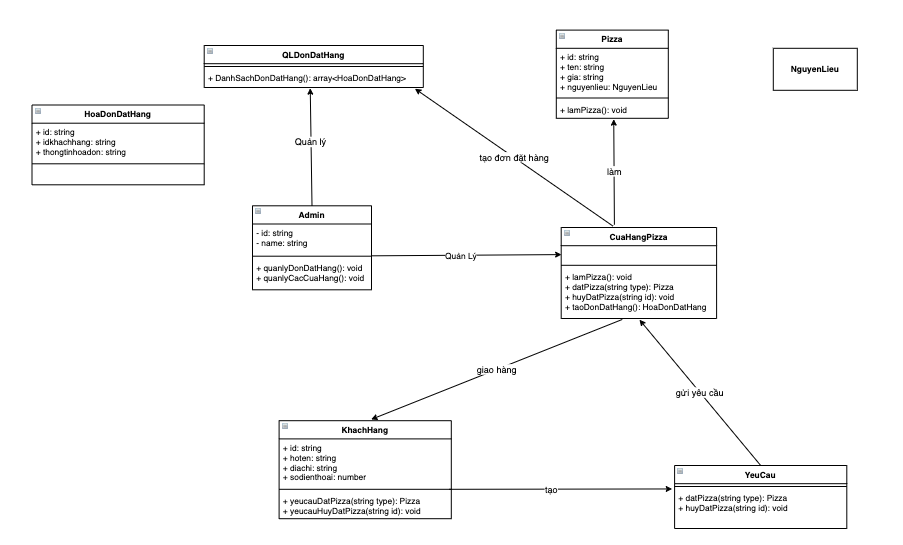
Tầng giao diện UI: có thể sử dụng form, web

Tầng nghiệp vụ (BLL) bao gồm quản lý nghiệp vụ và các chức năng bao gồm:

* Quản lý nghiệp vụ:
  + Nghiệp vụ phát sinh khi cửa hàng tạo bánh pizza như: bánh pizza có thể có nhiều loại. Và có nhiều cửa hàng ở nhiều địa điểm khác nhau như Đà Lạt, Hà Nội, Đà Nẵng
  + Các nghiệp vụ khác với người dùng như: đặt bánh pizza, huỷ đặt bánh pizza
  + Với cửa hàng pizza nghiệp vụ cần phải làm như là làm bánh pizza và gửi cho khách hàng, tạo đơn hàng cho admin.
  + Các nghiệp vụ với admin là: quản lý các cửa hàng bánh pizza, quản lý các đơn đặt hàng cửa người dùng
* Các chức năng:
  + Người dùng có thể đặt bánh pizza, huỷ đặt bánh pizza
  + Admin quản lý các cửa hàng bánh pizza và quản lý các đơn đặt hàng
  + Cửa hàng pizza nhận đặt hàng từ người dùng vào tạo bánh pizza và giao bánh cho khách hàng

Tầng dữ liệu (DAL): mô hình phát triển sử dụng CSDL như mysql, sqlserver

Sơ đồ tổng quan ứng dụng ở tầng nghiệp vụ:



1. Phân tích chi tiết phần mềm để áp dụng pattern

Ta có vấn đề với nghiệp vụ bánh pizza có thể có nhiều loại, nghĩa là sẽ có nhiều cửa hàng pizza khác cạnh tranh và ta muốn thu hút khách thì phải sẽ thêm nhiều loại bánh pizza để có thể cạnh tranh. Nếu chỉ có lớp Pizza thông thường thì ứng dụng sẽ sử dụng rất nhiều điều kiện để tạo bánh Pizza.

Ví dụ: if (type = ‘Cheese’) thì sẽ tạo bánh Cheese Pizza

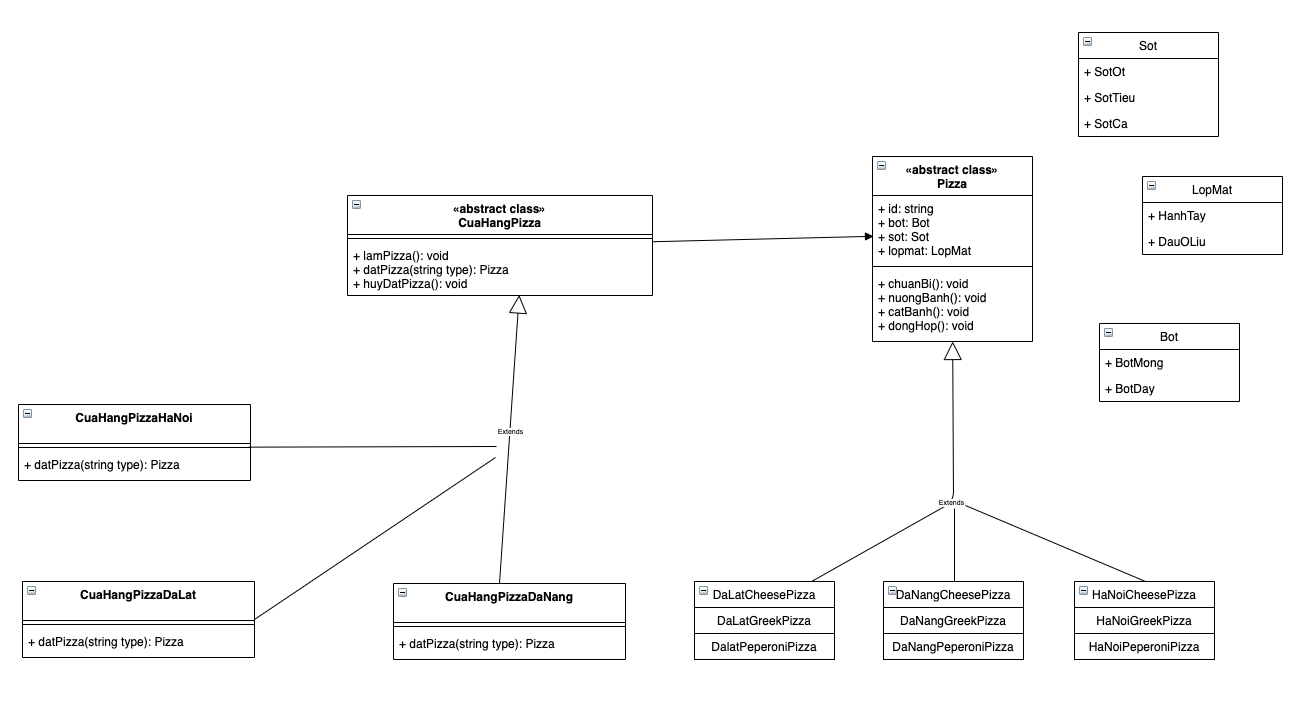
Sử dụng nhiều điều kiện trong lớp “CuaHangPizza” sẽ làm cho khó bảo trì và cập nhật thêm loại bánh Pizza.

Thêm một vấn đề nữa là có nhiều Cửa Hàng ở nhiều địa điểm khác nhau và mỗi điạ điểm lại có hương vị khác nhau như Đà Lạt, Đà Nẵng, HCM, ..

Để giải quyết vấn đề trên ta sử dụng Factory Pattern.

Áp dụng Factory Pattern cho lớp CuaHangPizza, Pizza, NguyenLieu

Sơ đồ chi tiết giải quyết vấn đề trên sử dụng Factory Pattern:



Giải thích chi tiết sơ đồ:

CuaHangPizza: là một abstract class như là một Factory

* nó có nhiệm vụ là làm bánh pizza sử dụng các phương thức bên Pizza như (chuanBi(), nuongBanh(), catBanh(), dongHop()),
* và có một abstract method để các concreateFactory bắt buộc phải thực hiện là datPizza(string type): Pizza.
* Phương thức datPizza(string type): Pizza là một factory method nó sẽ trả về đối tượng bánh pizza cụ thể

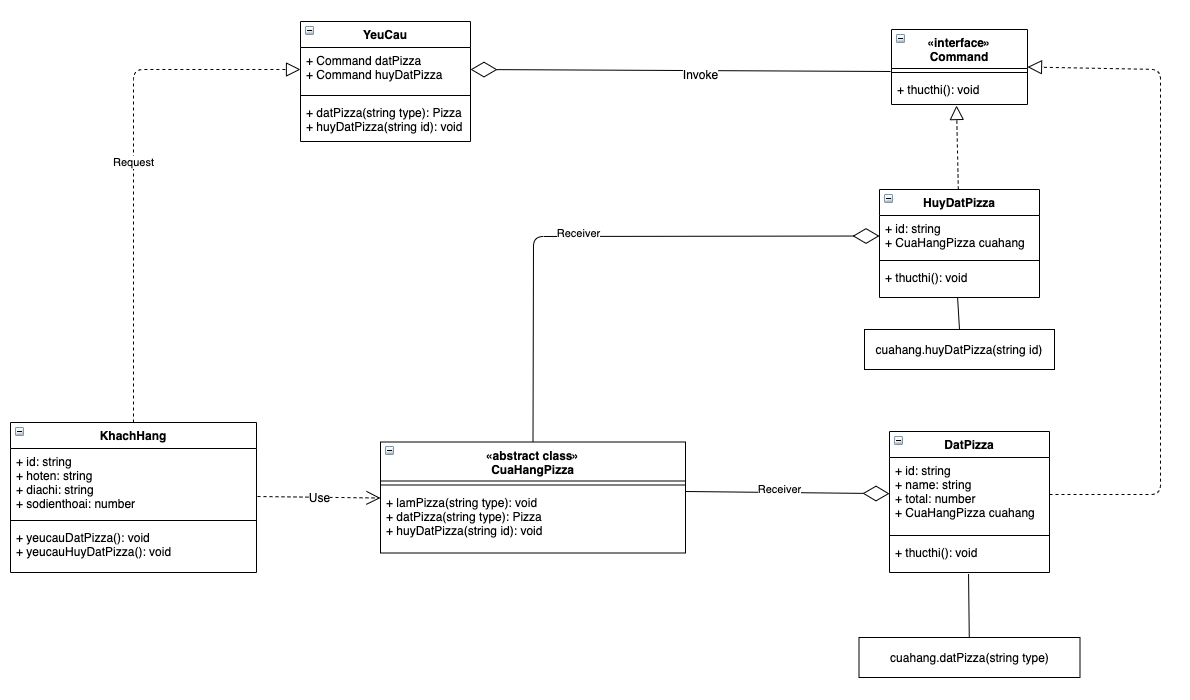
CuaHangPizzaDaLat, CuaHangPizzaHaNoi, CuaHangPizzaDaNang là các concreteFactory là các đối tượng cụ thể kế thừa từ Cửa hàng Pizza.

Pizza: như là Product trong sơ đồ của Factory Method, nó là một lớp mà các đối tượng cụ thể kế thừa.

DalatCheesePizza, DalatGreekPizza, DalatPeporoniPizza, HaNoiCheesePizza, .. là product con và là đối tượng cụ thể được tạo khi sử dụng Factory Method.

Một vấn đề nữa là ta có 2 tác vụ với khách hàng là đặt bánh pizza và huỷ đặt bánh pizza. Để giúp dễ dàng quản lý tác vụ ta sử dụng Command Pattern. Nó có ưu điểm tách các tác vụ thành các đối tượng cụ thể để dễ dàng quản lý.

Sơ đồ chi tiết áp dụng Command Pattern



Giải thích sơ đồ:

Command: đây là một interface, chứa một phương thức là thực thi (). Yêu cầu Đặt Pizza và Huỷ đặt pizza sẽ được đóng gói dưới dạng Command.

DatPizza, HuyDatPizza: là các concreateCommand, implememt method và cài đặt method từ interface Command.

- DatPizza: sử dụng phương thức datPizza() bên lớp CuaHangPizza, và nó sẽ trả về một đối tượng là Pizza

- HuyDatPizza: sử dụng phương thức huyDatPizza() bên lớp CuaHangPizza

CuaHangPizza: như là một Receiver, nó thực hiện yêu cầu từ khách hàng như datPizza(), huyDatPizza()

KhachHang như là một Client trong sơ đồ của Command Pattern , có phương thức yeucauDatPizza() và yeucauHuyDatPizza()

YeuCau: tiếp nhận Đặt Pizza và Huỷ đặt Pizza từ phía Khách Hàng và gọi phương thức bên Command để sử dụng như datPizza(), huyDatPizza().

Sơ đồ chi tiết sau khi kết hợp:

