# Tài liệu Kiến trúc Hệ thống

## System\_Architecture.docx

\*\*Ngày:\*\* 20/10/2023

\*\*Tác giả:\*\* AI Assistant

# Thiết kế Kiến trúc Hệ thống Quản Lý Kho Hàng

Tài liệu này mô tả thiết kế kiến trúc cấp cao (High-Level Design - HLD) cho hệ thống quản lý kho hàng. Thiết kế này dựa trên BRD mẫu đã cung cấp và giả định về yêu cầu dự án. Để có thiết kế chính xác và hoàn chỉnh hơn, cần có tài liệu SRS đầy đủ.

## 1. Tổng quan

Hệ thống quản lý kho hàng sẽ được thiết kế theo kiến trúc Microservices, cho phép khả năng mở rộng và bảo trì cao. Mỗi microservice sẽ chịu trách nhiệm cho một chức năng cụ thể, cho phép phát triển và triển khai độc lập. Dữ liệu sẽ được lưu trữ trong cơ sở dữ liệu quan hệ (SQL) cho tính toàn vẹn dữ liệu và khả năng truy vấn phức tạp. API RESTful sẽ được sử dụng để giao tiếp giữa các microservices và các ứng dụng khách.

## 2. Thành phần chính

\* \*\*Service Discovery:\*\* Sử dụng một service registry (ví dụ: Consul, Eureka) để các microservices có thể tìm kiếm và giao tiếp với nhau.  
\* \*\*API Gateway:\*\* Chức năng như một điểm truy cập duy nhất cho tất cả các yêu cầu từ các ứng dụng khách. Thực hiện các chức năng như routing, authentication, và rate limiting.  
\* \*\*Authentication Service:\*\* Xác thực và ủy quyền người dùng.  
\* \*\*Inventory Management Service:\*\* Quản lý thông tin hàng hóa (tên, mã, số lượng, giá, nhà cung cấp).  
\* \*\*Warehouse Management Service:\*\* Quản lý nhập/xuất kho (theo dõi số lượng, thời gian, người thực hiện).  
\* \*\*Reporting Service:\*\* Tạo báo cáo hàng tồn kho, báo cáo nhập/xuất kho.  
\* \*\*Database:\*\* Cơ sở dữ liệu quan hệ (ví dụ: PostgreSQL, MySQL) để lưu trữ dữ liệu.

## 3. Sơ đồ kiến trúc (Component Diagram)

## 4. Mô tả chi tiết các thành phần

\* \*\*API Gateway:\*\* Sử dụng công nghệ như Kong, Tyk hoặc Nginx để xử lý các request từ client và routing đến các microservices tương ứng. Thực hiện authentication và authorization trước khi chuyển request đến các microservices.  
\* \*\*Microservices:\*\* Được viết bằng ngôn ngữ lập trình Go hoặc Java, sử dụng framework Spring Boot (Java) hoặc Gin (Go). Mỗi microservice có riêng database connection và logic xử lý.  
\* \*\*Database:\*\* Sử dụng PostgreSQL hoặc MySQL, tùy thuộc vào yêu cầu cụ thể của dự án. Sơ đồ database sẽ được thiết kế chi tiết trong tài liệu riêng.  
\* \*\*Hosting:\*\* Triển khai trên nền tảng Cloud (AWS, GCP hoặc Azure). Sử dụng containerization (Docker, Kubernetes) để quản lý và mở rộng các microservices.

## 5. Công nghệ dự kiến sử dụng

\* \*\*Ngôn ngữ lập trình:\*\* Go hoặc Java  
\* \*\*Framework:\*\* Spring Boot (Java), Gin (Go)  
\* \*\*Cơ sở dữ liệu:\*\* PostgreSQL hoặc MySQL  
\* \*\*API:\*\* RESTful API  
\* \*\*Hosting:\*\* AWS, GCP hoặc Azure  
\* \*\*Containerization:\*\* Docker, Kubernetes

## 6. Lưu ý

Thiết kế này là một bản thiết kế cấp cao và có thể thay đổi dựa trên thông tin chi tiết hơn từ tài liệu SRS đầy đủ. Các chi tiết về thiết kế database, thiết kế API, và triển khai sẽ được mô tả trong các tài liệu riêng biệt.

This document provides a high-level architecture design. Further details will be elaborated in subsequent design documents.

## System Architecture Diagram 1