

Giới thiệu ○ ○ ○	Giải pháp ○ ○ ○ ○ ○○○○○	Kiến trúc hệ thống ○ ○○○○ ○○○ ○○ ○○ ○○	Triển khai và Thực nghiệm ○ ○○ ○○○○	Kết luận ○○○
---------------------------	--	--	--	-----------------

# Hệ thống thực hiện A/B Test trong Quảng Cáo

## Đồ án tốt nghiệp

Lê Tiến Chiến

Ngành: Khoa học máy tính

Ngày 9 tháng 8 năm 2022

Giới thiệu	Giải pháp	Kiến trúc hệ thống	Triển khai và Thực nghiệm	Kết luận
○ ○ ○	○ ○ ○ ○ ○○○○○	○ ○○○○ ○○○ ○○ ○○ ○○	○ ○○ ○○○○	○○○

# Nội dung

Giới thiệu

Giải pháp

Kiến trúc hệ thống

Triển khai và Thực nghiệm

Kết luận

Giới thiệu	Giải pháp	Kiến trúc hệ thống	Triển khai và Thực nghiệm	Kết luận
● ○ ○	○ ○ ○ ○ ○○○○○	○ ○○○○ ○○○ ○○ ○○ ○○	○ ○○ ○○○○	○○○

# Nội dung

Giới thiệu

Vấn đề

Mục tiêu

Giải pháp

Kiến trúc hệ thống

Triển khai và Thực nghiệm

Kết luận

Giới thiệu ○ ● ○	Giải pháp ○ ○ ○ ○ ○○○○○	Kiến trúc hệ thống ○ ○○○○ ○○○ ○○ ○○ ○○	Triển khai và Thực nghiệm ○ ○○ ○○○○	Kết luận ○○○
---------------------------	--	--	--	-----------------

## Vấn đề

- ▶ Vấn đề xảy ra khi muốn thử nghiệm các giải pháp mới cho một bài toán trong nghiệp vụ
- ▶ Cách truyền thống đó là triển khai các giải pháp liên tiếp nhau và so sánh kết quả
- ▶ Giải pháp không hiệu quả vì:
  - ▶ Thời gian không đủ tin cậy
  - ▶ Số liệu không trực quan
  - ▶ Không thể so sánh nhiều giải pháp đồng thời

Giới thiệu ○ ○ ●	Giải pháp ○ ○ ○ ○ ○○○○○	Kiến trúc hệ thống ○ ○○○○ ○○○ ○○ ○○ ○○	Triển khai và Thực nghiệm ○ ○○ ○○○○	Kết luận ○○○
---------------------------	--	--	--	-----------------

## Mục tiêu

Chúng ta cần xây dựng một hệ thống có thể giải quyết được vấn đề trên

- ▶ Thử nghiệm nhiều giải pháp đồng thời
- ▶ Kết quả trực quan, đáng tin cậy
- ▶ Quản lý thời gian dễ dàng
- ▶ Thiết lập thử nghiệm dễ dàng, hiệu quả

Từ đó, em xin giới thiệu hệ thống A/B Test được xây dựng dựa trên

- ▶ Sử dụng Golang cho Backend
- ▶ Lưu trữ dữ liệu ở Redis
- ▶ Xây dựng Website với ReactJS

Giới thiệu ○ ○ ○	Giải pháp ● ○ ○ ○ ○○○○○	Kiến trúc hệ thống ○ ○○○○ ○○○ ○○ ○○ ○○	Triển khai và Thực nghiệm ○ ○○ ○○○○	Kết luận ○○○
---------------------------	--	--	--	-----------------

# Nội dung

## Giới thiệu

## Giải pháp

A/B Test là gì  
Cách hoạt động của A/B Test  
Ứng dụng của A/B Test  
Cấu trúc A/B Test

## Kiến trúc hệ thống

## Triển khai và Thực nghiệm

Giới thiệu ○ ○ ○	Giải pháp ○ ● ○ ○ ○○○○○	Kiến trúc hệ thống ○ ○○○○ ○○○ ○○ ○○ ○○	Triển khai và Thực nghiệm ○ ○○ ○○○○	Kết luận ○○○
---------------------------	--	--	--	-----------------

## A/B Test là gì

**A/B Testing** (hay còn được gọi là split testing hay bucket testing) là một phương pháp để so sánh giữa các biến thể của một thuật toán hoặc ứng dụng nào đó, từ đó tìm ra được biến thể nào có hiệu quả tốt hơn.

Giới thiệu ○ ○ ○	Giải pháp ○ ○ ● ○ ○○○○○	Kiến trúc hệ thống ○ ○○○○ ○○○ ○○ ○○ ○○	Triển khai và Thực nghiệm ○ ○○ ○○○○	Kết luận ○○○
---------------------------	--	--	--	-----------------

## Cách hoạt động của A/B Test

**A/B Testing** về cơ bản là một cuộc thử nghiệm mà trong đó, hai hoặc nhiều biến thể của trang được hiển thị cho người dùng một cách ngẫu nhiên. Và kết quả của hành động đó sẽ được thể hiện qua conversion rate (tỉ lệ chuyển đổi), từ đó phân tích thống kê được sử dụng để xác định biến thể nào hoạt động tốt hơn cho mục tiêu chuyển đổi nhất định.



Giới thiệu ○ ○ ○	Giải pháp ○ ○ ○ ● ○○○○○	Kiến trúc hệ thống ○ ○○○○ ○○○ ○○ ○○ ○○	Triển khai và Thực nghiệm ○ ○○ ○○○○	Kết luận ○○○
---------------------------	--	--	--	-----------------

## Ứng dụng của A/B Test

**A/B Testing** có các ứng dụng sau đây

- ▶ Xây dựng Website
- ▶ Email Marketing
- ▶ Quảng Cáo và Bán Hàng
- ▶ Thiết kế Ứng dụng di động

Giới thiệu ○ ○ ○	Giải pháp ○ ○ ○ ○ ●○○○○○	Kiến trúc hệ thống ○ ○○○○ ○○○ ○○ ○○ ○○	Triển khai và Thực nghiệm ○ ○○ ○○○○	Kết luận ○○○
---------------------------	---	--	--	-----------------

## Cấu trúc A/B Test

# Một A/B Test có cấu trúc như sau

**Product** Phân tách nghiệp vụ lớn

**Layer** Phân tách nghiệp vụ nhỏ

**Experiment** Đại diện cho một thí nghiệm

**Test Group** Đại diện cho một biến thể của một thí nghiệm

**Parameter** Đại diện cho các đặc tính của một biến thể

Giới thiệu ○ ○ ○	Giải pháp ○ ○ ○ ○ ○●○○○○	Kiến trúc hệ thống ○ ○○○○ ○○○ ○○ ○○ ○○	Triển khai và Thực nghiệm ○ ○○ ○○○○	Kết luận ○○○
---------------------------	---	--	--	-----------------

## Cấu trúc A/B Test

# Product

## Định nghĩa

- ▶ Product dùng để phân tách giữa nghiệp vụ lớn
- ▶ Chỉ có thể sử dụng một Product cho một đối tượng

## Ví dụ

MobileApp, Advertisement, Website, etc..

Giới thiệu ○ ○ ○	Giải pháp ○ ○ ○ ○ ○●○○○	Kiến trúc hệ thống ○ ○○○○ ○○○ ○○ ○○ ○○	Triển khai và Thực nghiệm ○ ○○ ○○○○	Kết luận ○○○
---------------------------	--	--	--	-----------------

## Cấu trúc A/B Test

# Layer

## Định nghĩa

- ▶ Layer dùng để phân tách các nghiệp vụ nhỏ trong một Product
- ▶ Có thể sử dụng nhiều Layer đồng thời cho một đối tượng
- ▶ Sở hữu 100% lượng truy cập
- ▶ Có 2 loại hash strategy để quyết định Experiment
  - ▶ Dùng UserId
  - ▶ Dùng SessionId

## Ví dụ

Giao diện, Recall, Rerank, etc..

Giới thiệu ○ ○ ○	Giải pháp ○ ○ ○ ○ ○○○●○○	Kiến trúc hệ thống ○ ○○○○ ○○○ ○○ ○○ ○○	Triển khai và Thực nghiệm ○ ○○ ○○○○	Kết luận ○○○
---------------------------	---	--	--	-----------------

Cấu trúc A/B Test

# Experiment

## Định nghĩa

- ▶ Experiment là đại diện cho một thí nghiệm, thuật toán hay giải pháp
- ▶ Có thể tạo nhiều Experiment trong một Layer
- ▶ Experiment quản lý trạng thái, thời gian và dung lượng truy cập
- ▶ Chỉ có thể sử dụng dung lượng truy cập còn lại của Layer

## Ví dụ

Thử nghiệm về màu sắc, thuật toán, etc..

Giới thiệu ○ ○ ○	Giải pháp ○ ○ ○ ○ ○○○○●○	Kiến trúc hệ thống ○ ○○○○ ○○○ ○○ ○○ ○○	Triển khai và Thực nghiệm ○ ○○ ○○○○	Kết luận ○○○
---------------------------	---	--	--	-----------------

Cấu trúc A/B Test

## Test Group

### Định nghĩa

- ▶ Test Group đại diện cho một biến thể trong một Experiment
- ▶ Có thể tạo nhiều Test Group trong một Experiment

### Ví dụ

Màu đỏ, màu trắng, màu đen, etc..

# Parameter

## Định nghĩa

- ▶ Parameter đại diện cho các đặc tính của một Test Group
- ▶ Có thể tạo nhiều Parameter trong một Test Group
- ▶ Là một cặp key-value mà đối tượng có thể đọc và sử dụng
- ▶ Chỉ có thể sử dụng key từ một danh sách tạo trước đó do đối tượng khởi tạo

## Ví dụ

color=back, color=red, enabled=true, enabled=false, etc..

Giới thiệu ○ ○ ○	Giải pháp ○ ○ ○ ○ ○○○○○	Kiến trúc hệ thống ● ○○○○ ○○○ ○○ ○○ ○○	Triển khai và Thực nghiệm ○ ○○ ○○○○	Kết luận ○○○
---------------------------	--	--	--	-----------------

# Nội dung

Giới thiệu

Giải pháp

Kiến trúc hệ thống

Domain Driven Design

Onion

Hệ thống Backend

Biểu đồ tuần tự

Triển khai và Thực nghiệm



Giới thiệu ○ ○ ○	Giải pháp ○ ○ ○ ○ ○○○○○	Kiến trúc hệ thống ○ ●○○○ ○○○ ○○ ○○ ○○	Triển khai và Thực nghiệm ○ ○○ ○○○○	Kết luận ○○○
---------------------------	--	--	--	-----------------

## Domain Driven Design là gì

Domain-Driven Design là một phương pháp hay cách thức tiếp cận trong việc phân tích và phát triển phần mềm trong khi giải quyết những vấn đề nghiệp vụ phức tạp. Ý tưởng cơ bản của cách thức này là việc xây dựng sự kết nối chặt chẽ giữa thiết kế phần mềm và mô hình nghiệp vụ trong suốt vòng đời phát triển sản phẩm.

Giới thiệu ○ ○ ○	Giải pháp ○ ○ ○ ○ ○○○○○	Kiến trúc hệ thống ○ ○○●○○ ○○○ ○○ ○○ ○○	Triển khai và Thực nghiệm ○ ○○ ○○○○	Kết luận ○○○
---------------------------	--	---	--	-----------------

## Ba nguyên tắc

- ▶ Trọng tâm của dự án là những nguyên tắc và logic nghiệp vụ
- ▶ Các thiết kế phức tạp dựa trên các mô hình nghiệp vụ.
- ▶ Sự hợp tác giữa các chuyên gia kỹ thuật và miền là rất quan trọng để tạo ra một mô hình ứng dụng giải quyết các vấn đề cụ thể trong mô hình nghiệp vụ.

Giới thiệu ○ ○ ○	Giải pháp ○ ○ ○ ○ ○○○○○	Kiến trúc hệ thống ○ ○○●○ ○○○ ○○ ○○ ○○	Triển khai và Thực nghiệm ○ ○○ ○○○○	Kết luận ○○○
---------------------------	--	--	--	-----------------

# Đa tầng

**Tầng Application** Không chứa logic nghiệp vụ, dẫn dắt người dùng bởi các giao diện

**Tầng Domain** Chứa khái niệm về nghiệp vụ, chỉ nằm ở trung tâm, tách biệt với các tầng khác

**Tầng Infrastructure** Làm việc trực tiếp với hạ tầng và cơ sở dữ liệu



Giới thiệu	Giải pháp	Kiến trúc hệ thống	Triển khai và Thực nghiệm	Kết luận
○	○	○	○	○○○
○	○	○○○○	○○	
○	○	●○○	○○○○	
	○	○○		
	○○○○○○	○○		

Onion

## Kiến trúc Onion

Kiến trúc Onion được xây dựng trên mô hình nghiệp vụ trong đó các lớp được kết nối thông qua các interface. Ý tưởng là giữ các phụ thuộc bên ngoài càng xa càng tốt, nơi các thực thể nghiệp vụ và các quy tắc nghiệp vụ tạo thành phần cốt lõi của kiến trúc.

Giới thiệu ○ ○ ○	Giải pháp ○ ○ ○ ○ ○○○○○	Kiến trúc hệ thống ○ ○○○○ ○●○ ○○ ○○ ○○	Triển khai và Thực nghiệm ○ ○○ ○○○○	Kết luận ○○○
---------------------------	--	--	--	-----------------

Onion

## Ưu điểm

- ▶ Nó cung cấp kiến trúc linh hoạt, bền vững và dễ mở rộng
- ▶ Các lớp được liên kết chặt chẽ và có sự phân tách rõ ràng
- ▶ Nó cung cấp khả năng bảo trì tốt hơn vì tất cả mã code phụ thuộc vào các lớp sâu hơn hoặc lớp trung tâm
- ▶ Cải thiện khả năng kiểm tra vì các kiểm tra có thể được tạo cho các lớp riêng biệt mà không ảnh hưởng đến các lớp khác
- ▶ Các công nghệ có thể dễ dàng thay đổi mà không ảnh hưởng đến nghiệp vụ lõi

## Nguyên lý

**Sự phụ thuộc** Các lớp bên ngoài phụ thuộc vào các lớp bên trong và các lớp bên trong hoàn toàn không nhận biết được các lớp bên ngoài

**Bao bọc dữ liệu** Mỗi lớp bao bọc hoặc ẩn các chi tiết triển khai bên trong và cung cấp API cho lớp bên ngoài

**Tách mối quan tâm** Ứng dụng được chia thành các lớp trong đó mỗi lớp có một tập hợp các trách nhiệm và giải quyết các mối quan tâm riêng biệt

**Ràng buộc** Ràng buộc thấp trong đó một module tương tác với một module khác và không cần quan tâm đến các phần bên trong của module đó

Giới thiệu ○ ○ ○	Giải pháp ○ ○ ○ ○ ○○○○○	Kiến trúc hệ thống ○ ○○○○ ○○○ ●○ ○○	Triển khai và Thực nghiệm ○ ○○ ○○○○	Kết luận ○○○
---------------------------	--	--	--	-----------------

## Thành phần hệ thống Backend

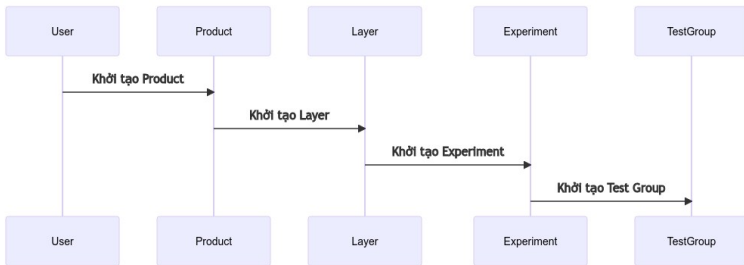
- Service** Là thành phần đảm nhiệm hệ thống HTTP, xử lý các truy cập từ Frontend
- Storage** Là thành phần tổng hợp dữ liệu, thực hiện A/B Test cho tầng Service
- Database** Là thành phần quản lý dữ liệu cho tầng Storage
- Types** Tổng hợp các thực thể ở tầng Backend



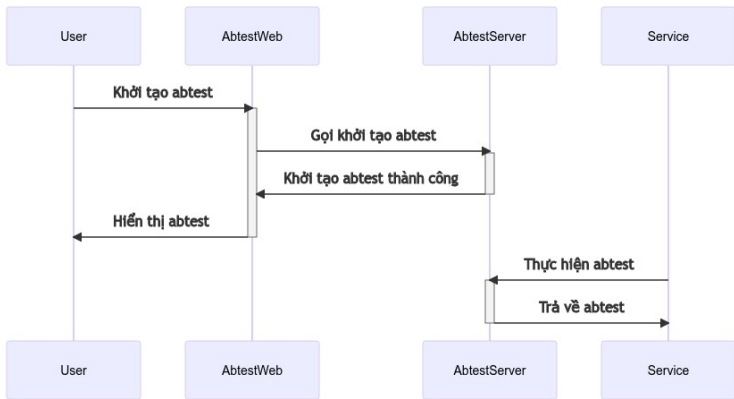
## Hệ thống API

Đường dẫn	Chú thích
/summary	Gọi toàn bộ thông tin của hệ thống A/B Test
/abtest	Thực hiện A/B Test
/product/create	Khởi tạo Product mới
/product/update	Cập nhật Product
/layer/create	Khởi tạo Layer mới
/layer/update	Cập nhật Layer
/exp/create	Khởi tạo Experiment mới
/exp/update	Cập nhật Experiment
/group/create	Khởi tạo Test Group mới
/group/update	Cập nhật Test Group mới

## Khởi tạo A/B Test



# Thực hiện A/B Test



Giới thiệu ○ ○ ○	Giải pháp ○ ○ ○ ○ ○○○○○	Kiến trúc hệ thống ○ ○○○○ ○○○ ○○ ○○ ○○	Triển khai và Thực nghiệm ● ○○ ○○○○	Kết luận ○○○
---------------------------	--	--	--	-----------------

# Nội dung

Giới thiệu

Giải pháp

Kiến trúc hệ thống

Triển khai và Thực nghiệm

Ứng dụng DDD

Thực nghiệm

Kết luận



## Hệ thống Backend

Thực thể	Khuôn mẫu	Giải thích
Service	Services	Service chịu trách nhiệm điều phối hoạt động giữa các tầng thông qua phương thức HTTP
Storage		Storage chịu trách nhiệm tổng hợp các Entity từ tầng Database
Database	Repository	Database là tầng trung gian giữa các tầng khác và tầng cơ sở dữ liệu (Redis)
Types	Aggregate	Types là nơi tổng hợp các Entities và các Value Object

# Hiển thị tất cả Product

## Product list

ID	Name	No. of Layer
1	<a href="#">SearchAds</a>	13

[Create new product](#)

# Hiển thị thông tin của một Product

SearchAds

Layers

13

Experiments

25

Layer list

ID	Name	Hash Strategy	No. of Exp
1	<a href="#">Release Layer</a>	Hash by UserID	25
2	<a href="#">Overwrite Layer</a>	Hash by UserID	0



ID	Name	Traffic	Status	Start Time	End Time	No. of Groups
1	<a href="#">new-rerank-svc</a>	100%	Start	03/03/2022, 14:26:06	05/03/2022, 23:59:59	2
2	<a href="#">pvr-boost</a>	100%	Start	08/03/2022, 16:39:10	11/03/2022, 23:59:59	1
3	<a href="#">new-rerank-svc - v2</a>	100%	Start	10/03/2022, 09:11:39	11/03/2022, 22:59:59	2

## Hiển thị thông tin của một Experiment

new-rerank-svc

## Traffic

100%

## Test Groups

2

Start Time

03/03/2022, 14:26:06

End Time

05/03/2022, 23:59:59

### Test Group list

ID	Name	Parameter Name	Parameter Value
1	<a href="#">base</a>	rerankServiceEnable	false
2	<a href="#">open</a>	rerankServiceEnable	true

[Create new test group](#)

Giới thiệu	Giải pháp	Kiến trúc hệ thống	Triển khai và Thực nghiệm	Kết luận
○ ○ ○	○ ○ ○ ○ ○○○○○	○ ○○○○ ○○○ ○○ ○○ ○○	○ ○ ○○ ○○○○	●○○

# Nội dung

Giới thiệu

Giải pháp

Kiến trúc hệ thống

Triển khai và Thực nghiệm

Kết luận

## Kết luận

Trong khoá luận này, em đã giới thiệu hệ thống thực hiện A/B Test có những chức năng sau:

- ▶ Thiết lập thử nghiệm A/B Test
- ▶ Thực hiện A/B Test

## Hướng phát triển

Trong tương lai, em cần xây dựng thêm một số chức năng để hệ thống trở nên hoàn thiện:

- ▶ Thu thập hiệu suất của A/B Test
- ▶ Hiển thị kết quả của A/B Test