**TÀI LIỆU THIẾT KẾ KIẾN TRÚC**

| **Dự án** | Thiennguyen |
| --- | --- |
| **Phiên bản** | 1.0 |
| **Ngày** | 31/05/2022 |
| **Người viết** |  |
| **Người phê duyệt** |  |

| **Thay đổi bản ghi** | | | |
| --- | --- | --- | --- |
| **Ngày** | **Người viết** | **Phiên bản** | **Nội dung sửa đổi** |
| 31/05/2022 | Trịnh Văn Hoàng | 1.0 | Thêm mới |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Nội Dung**

[1 Tổng quan 3](#_heading=h.1fob9te)

[1.1 Giới thiệu 3](#_heading=h.3znysh7)

[1.2 Mục tiêu 3](#_heading=h.2et92p0)

[1.3 Đối tượng 3](#_heading=h.tyjcwt)

[1.4 Phạm vi 3](#_heading=h.3dy6vkm)

[2 Thiết kế kiến trúc 4](#_heading=h.1t3h5sf)

[2.1 Các thành phần 4](#_heading=h.4d34og8)

[2.1.1 Tầng trình diễn (Presentation Layer) 4](#_heading=h.2s8eyo1)

[2.1.2 Tầng xử lý nghiệp vụ (Business Layer) 6](#_heading=h.17dp8vu)

[2.1.3 Tầng lưu trữ (Persistence Layer) 6](#_heading=h.3rdcrjn)

[2.1.4 Tích hợp với các hệ thống khác 6](#_heading=h.26in1rg)

[2.2 Danh sách các chức năng 7](#_heading=h.lnxbz9)

[2.2.1 Trang landing page dành cho khách hàng 7](#_heading=h.35nkun2)

[2.2.2 Trang quản trị nội dung 7](#_heading=h.1ksv4uv)

[2.2.3 App mobile 8](#_heading=h.44sinio)

[2.3 Nền tảng phát triển 9](#_heading=h.2jxsxqh)

[3 Thiết kế phi chức năng 11](#_heading=h.z337ya)

[4 Kiến trúc triển khai 12](#_heading=h.3j2qqm3)

[4.1 Môi trường Production (DC) 12](#_heading=h.1y810tw)

[5 Log và xử lý lỗi 14](#_heading=h.2xcytpi)

[5.1 Logging 14](#_heading=h.1ci93xb)

[5.2 Exception Handling 15](#_heading=h.qsh70q)

[6 Bảo mật 16](#_heading=h.3as4poj)

[6.1 Mức ứng dụng 16](#_heading=h.1pxezwc)

[7 Khả năng mở rộng 16](#_heading=h.2p2csry)

[8 Tính sẵn sàng của hệ thống 17](#_heading=h.147n2zr)

[9 Sao lưu / lưu trữ dữ liệu 18](#_heading=h.3o7alnk)

# 

## Các thành phần

## Tầng trình diễn (Presentation Layer)

Tầng trình diễn cung cấp sự tương tác giữa người dùng và hệ thống thông qua một giao diện đồ họa dạng trang web. Tầng này bao gồm các thành phần:

1. Giao diện quản lý CMS

Thành phần này cung cấp giao diện web dành cho nhân viên quản trị của hệ thống có thể tương tác điều khiển nội dung của hệ thống. Thành phần này cung cấp các tính năng chính như:

* Quản lý nạp rút
* Quản lý tài khoản.
* Quản lý lịch sử cược.

Các truy vấn dữ liệu hoặc yêu cầu nghiệp vụ thông qua giao diện sẽ được chuyển hóa thành các lời gọi API theo định dạng RESTful và gửi tới Tầng xử lý nghiệp vụ để thực hiện.

1. Giao diện website cho khách hàng

Thành phần này cung cấp giao diện web dành cho khách hàng để tương tác với hệ thống. Thông qua giao diện này, khách hàng có thể thực hiện các tính năng như:

* Xem giới thiệu, các thông tin chung.
* Chơi các game như tài xỉu, casino,v.v
* Xem thông tin cược, lịch sử cược.

Các truy vấn dữ liệu hoặc yêu cầu nghiệp vụ thông qua giao diện sẽ được chuyển hóa thành các lời gọi API theo định dạng RESTful và gửi tới Tầng xử lý nghiệp vụ để thực hiện.

1. Máy chủ phục vụ tài nguyên tĩnh – Static Server

Thành phần này cung cấp các tài nguyên tĩnh phục vụ cho việc hiển thị nội dung của CMS và Landing Page bao gồm các loại tài nguyên như: Ảnh (jpg, png, svg...), HTML, JavaScript, Font...

## Tầng xử lý nghiệp vụ (Business Layer)

Tầng xử lý nghiệp vụ tiếp nhận và xử lý các yêu cầu nghiệp vụ từ người dùng thông qua việc tương tác với Tầng trình diễn. Tầng này bao gồm các thành phần:

1. Ứng dụng xử lý nghiệp vụ - API Server:

Thành phần này tiếp nhận và thực thi các yêu cầu nghiệp vụ từ Tầng trình diễn dưới dạng các API RESTful. Các yêu cầu nghiệp vụ sẽ được kiểm tra tính hợp lệ, đáp ứng các ràng buộc, quyền hạn và được cụ thể hóa thành các thao tác tương tác với Cơ sở dữ liệu và Máy chủ lưu trữ tập tin. Ngoài ra, thành phần này cũng tương tác với bộ nhớ đệm InMemory để lưu trữ tạm thời các thông tin thường xuyên được truy cập nhằm giảm thiểu số lượng thao thác lên Cơ sở dữ liệu, tối ưu hóa khả năng vận hành của hệ thống.

1. Bộ nhớ đệm InMemory – Cache server:

Thành phần này cung cấp khả năng lưu trữ và truy vấn thông tin với tốc độ cao do dữ liệu được lưu trên bộ nhớ trong (RAM). Dữ liệu được lưu tại đây là các dữ liệu mang tính tạm thời, không bền vững và được lưu trữ/truy vấn theo dạng key-value.

## Tầng lưu trữ (Persistence Layer)

Tầng lưu trữ thực hiện lưu trữ dữ liệu đảm bảo nhất quán và bền vững. Tầng này bao gồm các thành phần*:*

1. Cơ sở dữ liệu – Database:

Thành phần này là nơi lưu trữ dữ liệu của toàn hệ thống. Nó cung cấp khả năng lưu trữ, truy vấn và cập nhật thông tin một cách nhất quán, chính xác và bền vững

1. Hệ thống lưu trữ tập tin chia sẻ – Share Storage:

Thành phần này là nơi lưu trữ các tập tin như ảnh, video, ... được chia sẻ giữa các máy chủ thông qua các giao thức phổ biến như NFS, SMB, ...

## Tích hợp với các hệ thống khác

Hệ thống Thien nguyen có tích hợp với các hệ thống sau:

1. Mail Service:

Đây là hệ thống email, thành phần này được tích hợp vào hệ thống Thien nguyen để gửi các yêu cầu liên hệ của khách hàng qua email.

## Danh sách các chức năng

## Trang landing page dành cho khách hàng

Danh sách chức năng của trang landing page dành cho khách hàng:

* Màn hình home của trang landing page
* Danh sách, xem chi tiết tin tức
* Danh sách KOL.
* Xem sao kê của từng KOL.

## Trang quản trị nội dung

Danh sách chức năng của trang quản trị nội dung:

* Đăng nhập, đăng xuất, đổi ngôn ngữ.
* Thông tin cá nhân, đổi mật khẩu.
* Quản trị viên.
* Người dùng.
* Bài đăng.
* Mục tiêu.
* Tin tức & câu chuyện.
* Lời chúc mặc định.
* Popup.
* Kol.
* Mục tiêu.
* Banner.
* Tham số hệ thống.
* Thông báo đảy.
* Ảnh nền bài đăng.
* Ngân hàng.
* Tài khoản ngân hàng.
* Câu hỏi thường gặp.
* Cấu hình nhận cảnh báo.
* Các bước gây quỹ.
* Bảng xếp hạng.
* Thêm giao dịch ngân hàng bằng tay.
* Tác vụ hệ thống.
* Log MB request.
* Log lịch sử chạy tiến trình.
* Báo cáo giao dịch tài khoản ngân hàng.
* Sử dụng hệ thống.
* Chi tiết sử dụng hệ thống.
* Mapping giao dịch với mục tiêu.
* Tổng hợp dữ liệu báo cáo sử dụng hệ thống.
* Tổng hợp dữ liệu thu chi tài khoản.
* Tổng hợp dữ liệu dashboard user.

## Nền tảng phát triển

Các công nghệ được sử dụng để phát triển hệ thống Thiện nguyện được lựa chọn dựa trên các tiêu chí:

* Tuân theo các quy định của MB về phát triển hệ thống.
* Đảm bảo tính tin cậy, bảo mật và đã được kiểm chứng thực tế.
* Hiện đại, hiệu xuất cao, được sử dụng rộng rãi và có khả năng mở rộng.

Dựa trên các tiêu chí trên, các công nghệ đã được lựa chọn để phát triển hệ thống gồm:

1. Cơ sở dữ liệu Mysql

MySQL là một sự lựa chọn thông dụng nhất dành cho các ứng dụng web, ứng dụng xử lý giao dịch trực tuyến và website thương mại điện tử. MySQL được tích hợp đầy đủ các tiện ích, dễ sử dụng, lại thêm khả năng giao dịch an toàn, chi phí thấp,... nên được xem là  cơ sở dữ liệu mã nguồn mở phổ biến nhất trên thế giới.

MySQL là một công cụ có tốc độ cao cùng với tính ổn định tốt. Công cụ này dễ sử dụng, lại còn hoạt động được ở nhiều hệ điều hành. Ngoài ra, tính bảo mật của MySQL rất mạnh, có thể sử dụng được ở trên nhiều ứng dụng...

1. JavaEE và Spring Framework

JavaEE tên đầy đủ là "Java Enterprise Edition", là một nền tảng dành cho việc xây dựng các ứng dụng ở quy mô doanh nghiệp. JavaEE tập hợp nhiều công nghệ kỹ thuật và phối hợp chúng với nhau, việc này sẽ làm giảm đáng kể chi phí và độ phức tạp cho việc xây dựng, phát triển, triển khai, quản lý các tầng làm việc cho các ứng dụng máy chủ trung tâm.

Spring Framework là một tổ hợp của nhiều framework nhỏ được thiết kế để hoạt động cùng với nhau. Nó cung cấp một cách hiệu quả các tính năng cần thiết để tạo ra các ứng dụng thương mại phức tạp, tạo ra một mô hình nhất quán và có thể áp dụng vào hầu hết các loại ứng dụng được tạo ra trên nền tảng JavaEE.

Sự kết hợp giữa JavaEE và Spring Framework mang đến cho ứng dụng sự ổn định, bảo mật cũng như khả năng mở rộng cộng với sự đơn giản trong quá trình phát triển ứng dụng, tích hợp với các hệ thống công nghệ khác cũng như giảm thiểu sai sót trong quá trình phát triển. Chính vì vậy, nó được sử dụng rộng rãi trong các hệ thống lớn và được tin dùng bởi hầu hết các lập trình viên.

Trong hệ thống Thiennguyen, JavaEE và Spring Framework được sử dụng để phát triển Ứng dụng xử lý nghiệp vụ - API Server.

1. Angular

Angular là một web framework được viết bằng TypeScript dựa theo kiến trúc hướng thành phần (component - based). Angular hỗ trợ xây dựng các ứng dụng SPA (Single Page Application) đem lại trải nghiệm người dùng tối ưu, cũng như giảm thiểu tải cho server do các mã HTML được render ở phía client. Angular cũng hỗ trợ rất tốt cho các công nghệ được sử dụng để phát triển web như LESS/SASS, JSLint, Babel, …

Trong hệ thống Thiennguyen, Angular được sử dụng để phát triển Giao diện quản lý nội dung – CMS.

# Thiết kế phi chức năng

* Tính ổn định: Hệ thống sử dụng các công nghệ đã được kiểm chứng thực tế như Mysql, JavaEE, Spring Framework đảm bảo tính ổn định cao của hệ thống.
* Tính bảo mật: Với việc tách rời giao diện người dùng (CMS, Landing Page) với xử lý nghiệp vụ (API Server) thì việc đảm bảo mật trở nên chặt chẽ và đơn giản hơn khi việc kiểm soát phân quyền về dữ liệu cũng như chức năng sẽ diễn ra ở cả hai mức.
* Tính mở rộng: Thông qua việc sử dụng các công nghệ có khả năng mở rộng cao như Mysql, JavaEE, Redis thì việc mở rộng hệ thống là đơn giản khi chỉ cần bổ sung máy chủ hoặc tăng cấu hình của máy chủ.

# Kiến trúc triển khai

## Môi trường Production (DC)

**Mô tả**

Các request được gửi đến hệ thống sẽ đi qua hạ tầng LB và được phân tải đều (round robin) đến từng máy chủ trong các cụm CMS + API, Landing Page và Static Server.

Cụm CSDL Mysql được cài đặt HA sử dụng Mysql RAC theo cơ chế Active/Active. Các kết nối đến CSDL được quản lý, điều phối, phân tải bởi Mysql RAC.

Cụm máy chủ cache Redis được cài đặt HA sử dụng Redis Sentinel với 1 master node và 2 slave node. Các kết nối đến cụm được quản lý, điều phối bởi Redis Sentinel. Khi master node ngưng hoạt động, Redis Sentinel sẽ chỉ định một slave trở thành master mới và các kết nối sẽ được chuyển đến master node mới này.

**Mô tả luồng dữ liệu qua ứng dụng CMS**



**Mô tả luồng dữ liệu qua ứng dụng Landing Page**



**Mô tả luông dữ liệu khi gọi API**

# Log và xử lý lỗi

## Logging

*Log của hệ thống được chia làm 2 loại là* ***Log hiệu năng*** *và* ***Log xử lý****.*

**Log hiệu năng**

Được tích hợp với hệ thống log theo dõi hiệu năng của ứng dụng liên tục, theo thời gian thực. Được tích hợp chặt chẽ với các thành phần của hệ thống gồm: CMS Web (Angular), API (Java), Landing Page (NodeJS/React).

**Log xử lý**

Là các bản ghi các sự kiện quan trọng xảy ra của hệ thống ứng dụng. Thông tin về log phục vụ việc xem lại các thông tin hoạt động của ứng dụng trong quá khứ với nhiều mục đích như gỡ lỗi, kiểm tra trạng thái hoạt động, xem thông tin, xem lỗi, xem cảnh báo.

Phương thức ghi log của hệ thống có các đặc điểm sau:

* Thư viện ghi log: SLF4J + Logback
* Nơi ghi log: File, Console
* Định dạng file log: Text
* Nơi lưu trữ: Tại thư mục ghi log của máy chủ phát sinh log
* Cơ chế rolling log: Theo ngày
* Cơ chế archive log: Sau 30 ngày
* Độ tăng trưởng: Thấp, phụ thuộc vào cường độ sử dụng, cho phép cấu hình điều chỉnh dung lượng file log lưu trữ.
* Định dạng bản tin log gồm các thông tin: Date time, Log Level, Process Id, Source Class, Content, Stack Trace

| Thông tin | Mô tả | Tiêu chí |
| --- | --- | --- |
| Date time | Thời gian xảy ra sự kiện | Bắt buộc |
| Log level | Phân loại mức độ log: Error, Info, Warning, Debug, Trace | Bắt buộc |
| Process Id | Id tiến trình | Bắt buộc |
| Source Class | Tên class trong mã nguồn phát sinh ra sự kiện | Bắt buộc |
| Content | Nội dung bản tin log | Bắt buộc |
| Stack trace | Thông tin truy vết lỗi | Không bắt buộc |

## Exception Handling

Khi hệ thống phát sinh ra ngoại lệ (exception) do dữ liệu người dùng gửi lên không hợp lệ hoặc do lỗi phát sinh từ hệ thống, toàn bộ các thay đổi về dữ liệu được thực hiện trong transaction sẽ được rollback để đảm bảo toàn vẹn dữ liệu. Đồng thời thông tin về ngoại lệ cũng được ghi lại trong Log của ứng dụng nhằm mục đích gỡ lỗi ứng dụng. Server cũng trả về một response tương ứng với loại lỗi đã xảy ra trên server để ứng dụng phía client có thể đưa ra các logic xử lý, hiển thị thông báo lỗi phù hợp với người dùng. Thông tin về lỗi được trả về cho client chỉ chứa các thông tin khái quát về lỗi như mã lỗi, loại lỗi nhằm tránh phát sinh các vấn đề về bảo mật.

# Bảo mật

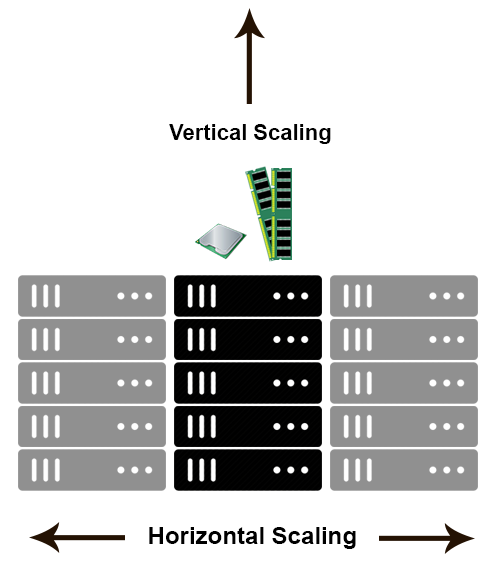
## Mức ứng dụng

Hệ thống được xây dựng hướng tới mức độ bảo mật cao nhất thông qua việc áp dụng các công nghệ tiên tiến cùng với các quy tắc lập trình an toàn để chống lại các nguy cơ bảo mật:

* XSS (Cross-site Scripting): Hệ thống sẽ escape toàn bộ dữ liệu đầu vào của người dùng và output ra dưới dạng text, ngăn ngừa các nguy cơ khai thác lỗ hổng XSS.
* CSRF/XSRF (Cross Site Request Forgery): Hệ thống thực hiện kiểm tra một header đặc biệt (X-XSRF-TOKEN) đối với các request quan trọng được gửi lên hệ thống. Theo cơ chế này, các ý đồ tấn công sử dụng lỗ hổng CSRF/XSRF sẽ không thể thực hiện được.
* SQL Injection: Tất cả các thao tác truy vấn, cập nhật CSDL của hệ thống được xây dựng trên nền tảng Java với PreparedStatement. Theo đó tránh được hoàn toàn lỗi SQL Injection.
* Session Fixation: Ngay sau khi người dùng đăng nhập thì SessionID của người dùng sẽ được thay thế bằng một ID mới. Cơ chế này khiến cho các ý đồ tấn công khai thác lỗi Session Fixation không thể thực hiện.
* Brute Force: Sử dụng cơ chế mật khẩu mạnh và yêu cầu lập trình an toàn nhằm tránh rò rỉ tên đăng nhập của người dùng.
* Sử dụng giải pháp Captcha mạnh.
* Sử dụng chứng thư bảo mật mạnh cho website.
* Thực hiện kiểm thử bảo mật website - Pentest trước khi public hệ thống ra bên ngoài(2 lần, 1 do đối tác của INCA thực hiện, 1 do đối tác và MB thực hiện)

# Khả năng mở rộng

Hệ thống Thiennguyen được xây dựng với khả năng hỗ trợ cả mở rộng theo chiều ngang (**Horizontal scaling**) và mở rộng theo chiều dọc (**Vertical scaling**).



Tại thời điểm hệ thống cần scale do số lượng người dùng gia tăng, có thể thực hiện theo 2 phương án:

**Mở rộng theo chiều dọc**: bằng cách tăng số lượng CPU/RAM/HDD của máy chủ. Phương án này rất đơn giản, dễ thực hiện nhưng lại bị giới hạn bởi số lượng phần cứng tối đa mà máy chủ hoặc hệ điều hành hỗ trợ. Ví dụ: một máy chủ IBM x3650 M4 chỉ hỗ trợ tối đa được 2 bộ vi xử lý, 16 slot RAM (~ 128GB) và 10 slot HDD. Như vậy, nếu ứng dụng cần sử dụng số tài nguyên đạt tới giới hạn này thì hệ thống sẽ không thể scale thêm được nữa. Để scale theo phương án này, ta cần cấu hình lại tham số Heap Size của Java và tăng số lượng Worker-Thread cho ứng dụng CMS+API. Đối với ứng dụng Landing page/Static, hệ thống sẽ tự động tối ưu theo tài nguyên hệ thống.

**Mở rộng theo chiều ngang**: bằng cách tăng số lượng máy chủ của hệ thống. Phương án này đem lại khả năng mở rộng cao hơn nhiều so với phương án mở rộng theo chiều dọc vì ta có thể thêm tùy ý số lượng máy chủ vào hệ thống.

# Tính sẵn sàng của hệ thống

Hệ thống được thiết kế với tư tưởng "**No Single Points of Failure**". Theo đó, để hệ thống hoạt động với mức độ sẵn sàng cao thì cần đảm bảo rằng tất cả các thành phần trong hệ thống cùng hỗ trợ hoạt động ở mức độ sẵn sàng cao với các công nghệ riêng biệt cho từng thành phần đặc trưng. Bất kỳ máy chủ đơn lẻ nào trong hệ thống ngừng hoạt động cũng sẽ không làm cả hệ thống bị gián đoạn. Tính sẵn sàng của hệ thống sẽ càng cao khi tăng số lượng máy chủ cho từng thành phần. Các công nghệ HA được sử dụng trong hệ thống gồm:

* Load-balancer: Được sử dụng để triển khai cơ chế HA cho các thành phần hoạt động theo cơ chế Active/Active như máy chủ API/ máy chủ web/ máy chủ file static.
* Redis Sentinel: Là công nghệ cluster chuyên dụng cho các máy chủ Redis.
* Mysql Cluster Manager: Là công nghệ cluster chuyên dụng cho các máy chủ CSDL Mysql.

# Sao lưu / lưu trữ dữ liệu

Quá trình sao lưu / lưu trữ dữ liệu của hệ thống bản chất là sao lưu dữ liệu được lưu trữ bởi CSDL Mysql và File Server được cụ thể hóa qua 2 công đoạn sau đây:

* Sao lưu dữ liệu trong CSDL Mysql: Quá trình này có thể được thực hiện một cách tự động bằng cách thiết lập Mysql Recovery Manager (RMAN).
* Sao lưu dữ liệu được lưu trữ bởi Shared Storage: Quá trình này có thể thực hiện tự động hoặc bán tự động phụ thuộc vào nền tảng lưu trữ được sử dụng là SAN/NAS/… mà sẽ đi kèm những công cụ sao lưu riêng của nhà sản xuất.

Đối với các máy chủ ứng dụng thì bản thân các máy chủ này không lưu trữ dữ liệu, do vậy quá trình sao lưu/lưu trữ dữ liệu là không cần thiết. Tuy nhiên, trong trường hợp bắt buộc phải thực hiện thì có thể sử dụng công cụ sao lưu/phục hồi thông qua VMWare vCenter.