**Thông tin thiết kế database**

Bởi vì dự án tập trung vào bài viết, nên **tag**, **hastag**, phải thật sự chú trọng, tập trung vào:

* Tìm kiếm
* Thống kê
* Công cụ đăng bài phải thật mạnh, tập trung vào file Word, Pdf thì chịu rồi, Có thống kê bằng Excel: người dùng, bài viết.
* Có công cụ load ảnh thật tốt, vì sẽ đăng nhiều ảnh đấy.

Dự án của ta sẽ thuộc kiểu một blog:

* **Chia sẻ các câu chuyện**.
* Giới thiệu các dự án (ta sẽ không tuyển thẳng cộng tác viên trên đây, mà sẽ có các đường link dẫn về trang facebook, ở đó sẽ trao đổi với admin sau. Cho dù có cho người dùng nộp CV, thì họ cũng không nộp đâu, vì rủi ro bỏ mẹ đi được! Sau này có uy tín thì mới phát triển sau).
* Không có tài khoản người dùng, chỉ có tài khoản admin, công tác viên. Vì người dùng cũng không đăng ký đâu. Muốn trao đổi, thì có các kênh sau: discord, facebook, message, … ??? TL: không nên, viết blog mà không có comment thì nhảm bỏ mẹ, tạo cảm giác clb này đéo có ai quan tâm ấy! 🡪 cần có tài khoản người dùng, tạo qua google, facebook
* Tóm lại là chỉ hoạt động cho mục đích quảng bá clb thôi. (Thế thì xây dựng theo microservice mà ăn loz à!!!).
* Nên nhớ, website của ta là 1 hệ thống mở, người dùng có thể tự do đăng ký và xác nhận gmail 🡪 chính vì thế ta cần có hệ thống quản lý người dùng hiệu quả và ban nick khi cần thiết.

**Các thực thể bao gồm:**

* Super admin.
* Admin.

Gọi chung là User.

* Ctv đăng bài.
* Ngươi dùng thông thường

**User-case**:

Người dùng thông thường:

Ordinary\_user

Các thông báo bao giồm:

* Thay đổi quyền.
* Phản hồi comment.
* Nick bị ban hay xóa vĩnh viễn sẽ có email gửi về người dùng (bởi vì khi đăng ký, ta bắt buộc phải có xác nhận từ gmail) 🡪 bắt buộc người dùng phải có gmail đăng ký.

Cộng tác viên:

Có chức năng tương tự người dùng + thêm các chức năng sau:

ctv

Lưu ý:

Chức năng xóa comment, không hoạt động với:

* Comment của admin, super admin đã bị ghim lên đầu.

Chức năng thông báo còn có thêm khả năng:

* Thông báo khi có ai đó comment blog.

Chức năng tắt thông báo, không hoạt động với:

* Thông báo comment đã ghim của admin, super admin.
* Xóa bài viết của super admin.

Admin:

Có chức năng tương tự admin, nhưng:

* Cho phép duyệt các bài viết.
* Đọc các feedback người dùng.
* Thêm chức năng: thêm/thăng cộng tác viên (cộng tác viên sau 7 ngày được thêm, mới có thể đăng bài).
* Thêm chức năng: giáng quyền CTV 🡪 người dùng thông thường (hoạt động tương tự ban nick.)
* Ban nick người dùng thông thường, CTV.
* Không nên cho phép Admin có quyền ban người dùng, vì sẽ có khoảng từ 4 – 6 Admin quản lý, mà chỉ cần 1 Admin phá 🡪 Toang

TL: như thế thì chức năng của Admin lại quá tù, thiếu uy quyền với CTV, người dùng thông thường 🡪 vẫn cho phép ban từ 3 🡪 30 ngày, nhưng với các hạn chế sau:

* Không thể ban quá 3 người/1h , 10 người/ngày, 20 người/tháng.
* Mọi hoạt động ban người dùng, đều có email ngầm gửi đến admin.
* Nếu muốn ban quá 3 người/1h, 10 người/ngày, 20 người/tháng thì phải tạo thông báo đến admin và ghi rõ lý do, có bao gồm ảnh và danh sách đối tượng cần ban.
* Nếu có hơn 10 người/ngày, 20 người/ 2ngày, 30 người/ tháng bị ban nick, tính trên tất cả admin 🡪 gửi email đến super admin với tiêu đề [**Hoạt động ban nick/giáng quyền bất thường**].

Super Admin:

Chỉ có duy nhất 1 tài khoản super admin.

Có các chức năng tương tự Admin, nhưng:

* Thêm chức năng thống kê: Số lượng tài khoản, vai trò, bài viết.
* Thêm chức năng: thêm, xóa/giáng quyền Admin, Ctv.
* Ban nick, xóa vĩnh viễn người dùng.
* Đọc feadback của tất cả người dùng.

**Sơ đồ thực thể - liên kết EER**:

Người dùng và người dùng:

Quản lý

Bị quản lý

Người dùng và Blog.

Blog

Viết bài

0, N

1, 1

Comment

0, N

0, N

Người dùng x Blog x Người dùng:

Blog

Thông báo

0, N

0, N

0, N

**Mô hình dữ liệu**:

Tất cả cá khóa đều thuộc kiểu BIGINT.

Quan hệ khóa chính – khóa ngoại, sẽ được diến ra ở back-end.

Database “manage”:

Người dùng (user):

* userId (32 byte)
* loginName
* password
* roleId – ảnh hưởng đến phân quyền.
* userName – tên người dùng.
* birthDay.
* gender.
* email.
* phone – chỉ riêng super admin, mới có thể xem được.
* createUserId – cho biết ai thêm user.
* state – phần lớn bảng đều có.
* updateTime – phần lớn bảng đều có.
* createTime – phần lớn bảng đều có.

Vai trò (Role):

* roleId – Khóa chính
* roleName – Tên vai trò.
* updateTime

Role-Phân quyền(rolepermission) – Hiện tại bỏ table này đi, chỗ bảng phân quyền thay permissionId thành roleId. Bởi tao chưa cần phân quyền chi tiết như vậy!!!

* roleId
* permissionId

Phân quyền (permissionId) – đường dẫn url đặc biệt, cần có các quyền đặc thù

* pId – Khóa chính.
* baseUrl – đường dẫn.
* permissionId – id quyền 🡪 thay thành roleId
* state.
* createTime.
* updateTime.

Ban/khóa tài khoản (blockaccount):

* blockId – Khóa chính
* userId
* requestIp – Để phân tích (đối với tài khoản vừa bị ban nick, dùng máy đăng nhập tài khoản đó để tạo tài khoản mới trong thời gian 3 ngày.🡪 block tiếp)??? Vấn đề này cần nghiên cứu thêm, hình trên có lỗ hổng là nếu người dùng có 2 máy, 1 máy tạo, 1 máy vi phạm 🡪 không chống được, nhưng tạm thời cứ dùng cách này đi.
* adminId – ai đã ban nick này.
* blockTime
* startTime
* endTime
* reason – lý do bị ban.
* state – vào thời điểm kết thúc ban nick, ta sẽ set = 3 (đã xóa), hoặc chủ động đặt nó = 3 cũng được.
* createTime
* updateTime

Bảng thống kê số lượng tài khoản mà 1 admin đã block (adminblocknum):

* userId – khóa chính
* accph – Số lượng account trên giờ.
* hstartTime – thời điểm bắt đầu tính số lượng acc/p .
* accpd – Số lượng account trên ngày.
* dstartTime – thời điểm bắt đầu tính số lượng acc/d.
* accpm – Số lượng account trên tháng.
* mstartTime – thời điểm bắt đầu tính số lượng acc/m
* total – tổng số lượng account đã ban.
* updateTime.

Bảng thống kế số lượng tài khoản đã bị ban trong 1 thời gian nhất định (timeblocknum) – bảng này chỉ có 1 dòng và khóa chính là “time\_ban”:

* timeId – Khóa chính
* accph – Số lượng account trên giờ.
* hstartTime – thời điểm bắt đầu tính số lượng acc/p .
* accpd – Số lượng account trên ngày.
* dstartTime – thời điểm bắt đầu tính số lượng acc/d.
* accpm – Số lượng account trên tháng.
* mstartTime – thời điểm bắt đầu tính số lượng acc/m
* total – tổng số lượng account đã ban.
* updateTime.

Thông báo (notification):

Các loại thông báo bao gồm:

* Thay đổi roleId người dùng.
* Cập nhật bảng phân quyền.
* Thay đổi thông tin người dùng.
* Ban nick, giáng quyền.
* Admin, super admin ghim comment
* Comment.

Các cột:

* notiId – Khóa chính.
* notiType – loại thông báo.
* fromId – người thực hiện/thay đổi khiến hình thành nên thông báo.
* targetId – người chịu sự thay đổi.
* information – thông tin thay đổi, tối đa 100 ký tự, phần còn lại là dấu …, vì không nên copy nguyên nội dung comment làm gì, vì nó có độ dài 1000, hiện hết lên thông báo thì sao hiển thị hết.
* targetUrl – url đích hiển sự thay đổi.
* createTime

Feedback (feedback):

* feedbackId – Khóa chính
* userId
* feedback - tối đa 1000 từ.
* allow\_admin
* state – ta sẽ không xóa, mà chỉ ẩn đi thôi
* createTime.
* updateTime.

Bảng ghi lại các thay đổi nội dung của các bảng sau:

* Thay đổi roleId người dùng.
* Cập nhật bảng phân quyền.
* Thay đổi thông tin người dùng.
* Ban nick, giáng quyền.

Bao gồm các cột (userlogRecord).

* logId – Khóa chính.
* userId – người thực hiện.
* bizType – thao tác trên bảng nào.
* opType - thêm | xóa | sửa.
* bizNo – Khóa chính của dữ liệu thao tác.
* title – Tiêu đề.
* detail – Nội dung chi tiết.
* requestIp – địa chỉ ip
* createTime.

Ghi lại thông tin đăng nhập của người dùng (userloginlog).

* logId – Khóa chính.
* userId.
* userName – để tiện cho việc hiển thị thông tin.
* browserName – từ trình duyệt nào.
* osName – hệ điều hành nào.
* loginIp – địa chỉ Ip.
* loginType – quyền người đăng nhập.
* createTime – thời gian đăng nhập.

Thống kê số lượng blog của 1 user (userblognum):

* userId
* userName
* num
* total\_views – Tổng số views của tất cả bài viết.
* interestest – Tên bài viết được yêu thích nhất.
* key\_interestest – blogKey của bài viết đó.
* views\_interestest – Số lượng view của bài viết được yêu thích nhất.
* updateTime

Database “content”:

Bài viết (blog):

* blogId – Khóa chính.
* blogKey – để làm PathVariable, giúp truy vấn tiện lợi hơn.
* authId – Tác giả bài viết.
* authName – Tên tác giả, để cải thiện tốc độ truy vấn.
* majorImg – hình ảnh đại diện chính cho bài viết.
* thumbnail – Phần tóm tắt bên dưới hình ảnh, liệt kê danh sách cá bài viết.
* title – tiêu đề.
* content – Nội dung sẽ rất dài đấy.
* allow\_notify – Có cho phép thông báo hay không (0 – No, 1 - Yes).
* views – Số lượt xem
* interested – Thể hiện mức độ quan tâm từ 1 🡪 5, thay cho đánh giá 5 sao.
* state.
* createTime
* updateTime.

Blog-hastag (bloghastag):

* blogId
* hastagId
* sort – thể hiện mức độ ưu tiên của hastag đối với bài viết. Ảnh hưởng đến sắp xếp sau khi tìm kiếm thành công. Là 1 dãy số 1 -> 10. Mức độ ưu tiên của các hastag không thể giống nhau. Sau này cần nâng cấp lên là tự code đánh giá mức độ ưu tiên của hastag, chứ không phải người dùng??? TL: không cần, sau này ta cần có bộ đánh index riêng cho từng bài viết, phụ thuộc vào hastag và số lần lặp từ bên trong bài viết 🡪 mỗi bài viết có tối đa 10 hastag. Nếu người dùng không setup mức độ ưu tiên, mức độ ưu tiên sẽ theo thứ tự trong dãy hastag gửi về.
* updateTime

Hastag (hastag) – **Sẽ cần cải tiến thêm, để cải thiện tốc độ tìm kiếm và gợi ý dữ liệu.**

* hastagId – Khóa chính
* hastagName.
* state
* createTime.
* updateTime.

Comment:

* commentId
* userId
* blogId
* parentId – cha của nó là commentId, thể hiện mối quan hệ reply trong comment.
* Comment – độ dài khoảng 1000 từ.
* state – thực tế, ta không xóa comment mà chỉ ẩn nó đi thôi.
* createTime.
* updateTime.

Database upload:

Thông tin file lưu trữ (uploadfile):

* fileId – Khóa chính.
* fileName
* fileContentType – nội dung gửi qua internet(ví dụ: image/png, application/pdf, …)
* fileType – định dạng file file (img, docx, pdf).
* fileSuffix ~ extension file.
* dir ~ baseUrl trong back-end (thư mục con chứa file).
* fileUrl ~ fileName trong back-end (toàn bộ đường dẫn file).
* fileSize – kích thước file.
* origin – loại lưu.
* state
* createTime.