1. **Mô hình phân quyền - RBAC**
2. **Tổng quan**

Cấu trúc của database với mô hình phân quyền RBAC trong project quản lý công việc bao gồm 10 collection trong MongoDB ( tương ứng với trong SQL thì gọi là 10 table ).

* Company
* User
* Role
* RoleType
* UserRole
* Department
* Privilege
* Link
* Component
* Action

1. **Mô tả chi tiết về từng collection**

* Company

Lưu trữ thông tin về các công ty trong hệ thống.

\_id: ObjectId,

name: String,

short\_name: String,

description: String

* User

Lưu trữ thông tin về tài khoản người dùng

\_id: ObjectId,

name: String,

email: String,

password: String,

company: ObjectId - liên kết với \_id tương ứng với Company

active: Boolean,

delete\_soft: Boolean,

status: Number,

reset\_password\_token: String - token xác thực đổi mật khẩu khi quên

token: String - lưu trữ token - xác thực tài khoản người dùng

* Role

Lưu thông tin các role trong hệ thống

\_id: ObjectId,

company: ObjectId,

name: String,

abstract: Array ObjectId

type: ObjectId - liên kết với \_id tương ứng với RoleType

* RoleType

Lưu thông về các loại role trong hệ thống

\_id: ObjectId,

name: String

* UserRole

Đây là 1 collection trung gian cho mối quan hệ Many to Many giữa User và Role.

\_id: ObjectId,

userId: ObjectId,

roleId: ObjectId

* Department

Lưu thông tin các đơn vị trong 1 công ty

\_id: ObjectId,

company: ObjectId - thuộc về công ty nào

name: String,

description: String,

dean: ObjectId,

vice\_dean: ObjectId,

employee: ObjectId,

parent: ObjectId

* Privilege

Bảng trung gian cho mối quan hệ Many to Many giữa Role với Link, Component, vv...

\_id: ObjectId,

resourceId: ObjectId,

resourceType: String,

roleId: ObjectId,

action: ObjectId

* Link

Lưu trữ các đường link tương ứng với các trang web cho bên frontend.

\_id: ObjectId,

company: ObjectId,

url: String,

description: String,

components: Array ObjectId

* Component

Lưu trữ các thành phần UI,UX bên phần frontend của client. Tương ứng với role nào thì thành phần UI đó sẽ được phép hiển thị bên frontend.

\_id: ObjectId,

company: ObjectId,

name: String,

description: String

* Action

Lưu trữ các hành động tương ứng được cấp cho các Component với các Role tương ứng.

\_id: ObjectId,

name: String,

see: Boolean,

open: Boolean,

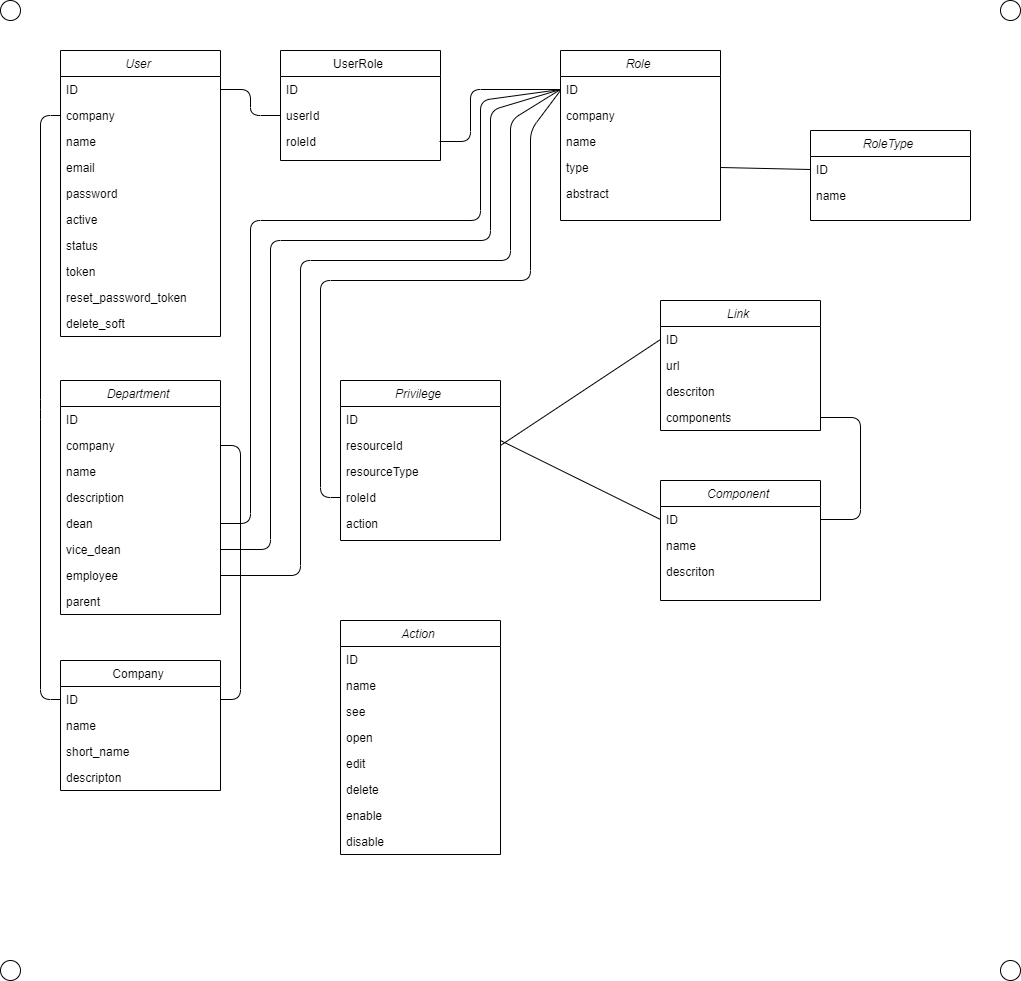
edit: Boolean,

delete: Boolean,

enable: Boolean,

disable: Boolean

1. **Sơ đồ mối quan hệ giữa các collection trong DATABASE**

****

1. **Cấu trúc thư mục**

Thư mục project bao gồm 2 phần : server và client.

1. Server

Trong thư mục server chứa mã nguồn cho phần backend của hệ thống bao gồm :

* logs: chứa các file ghi log lịch sử tương tác nhận gửi request giữa client và server
* middleware: chứa các file mã nguồn middleware làm nhiệm vụ như một trạm trung gian , xác thực thông tin từ người dùng sau đó mới cho qua xử lý để trả về dữ liệu cho client.
* models: chứa các file mã nguồn về định dạng cấu trúc cho các collection trong cơ sở dữ liệu
* seed: chứa file mã nguồn về tạo dữ liệu mẫu khi xậy dựng project.
* modules: chứa các module tương ứng với từng chức năng của hệ thống.
* package.json: chứa tên các thư viện cần thiết cần được cài đặt khi build project,..

1. Client

Cấu trúc thư mục cho phần client được xây dựng từ khung của một project ReactJS và sử dụng Redux để xử lý dữ liệu nhận về từ server. Nhìn chung phần quan trọng nhất trong thư mục client chính là **src** : chứa mã nguồn xử lý giao diện và dữ liệu nhận được từ server. Cấu trúc trong thư mục con **src** bao gồm:

* layouts: chứa layouts cho mục đích tái sử dụng ứng với các trang
* react-routes: chứa các file Router có nhiệm vụ như 1 bộ định tuyến sẽ chuyển hướng người dùng đến các trang tương ứng với các đường link. Cụ thể trong thư mục này có 3 file quan trọng:
  + AuthRoute : dùng cho những page chưa cần có sự xác thực tài khoản: VD: Login
  + PrivateRoute : dùng cho những page đòi hỏi người dùng đã được xác thực thì mới có thể truy cập vào được.
  + Routes : File tổng hợp các page của AuthRoute và PrivateRoute
* redux: chứa file *combine\_reducers.js và Store.js* là file mã nguồn định nghĩa 1 store cho project khi áp dụng với redux.
* modules: chứa các thư mục module con tương ứng với các chức năng. Về cầu trúc của 1 module bao gồm:
  + components: chứa các file mã nguồn về giao diện cho từng module
  + redux: mỗi module con sẽ được xây dựng theo một cấu trúc redux riêng để tiện cho việc chỉnh sửa mà không ảnh hưởng đến các module khác. Cấu trúc bao gồm 4 phần:
    - action: thực hiện các lời gọi đến các hành động mà người dùng yêu cầu
    - constants: định nghĩa tên cho các hành động mà người dùng sẽ gọi đến.
    - reducers: xử lý dữ liệu nhận được và lưu lại trong store.
    - services: các file service thực hiện các nhiệm vụ gửi request đến server và nhận response để lấy dữ liệu cho người dùng.

1. **Cách xây dựng 1 module trong project**

* Server

Để tạo 1 module mới cho bên server -> vào thư mục modules và thực hiện lần lượt theo các bước sau:

* Tạo một module mới bằng cách copy từ folder \_samle-module. Đặt tên cho thư mục bằng chữ thường không dấu, nếu tên của module bao gồm nhiều từ thì cách giữa các từ ta sử dụng dấu gạch dưới:

VD: ***management\_tasktemplate***

* Đặt tên cho route, controller và service theo quy tắc :

**tên module . route/controller/service . js**

VD: user.route.js , user.controller.js , user.service.js

* Định nghĩa các route cho module
* Định nghĩa các controller tương ứng với các route
* Viết các service trong file service của module.

* Client

Cấu trúc của một module bên client bao gồm 4 phần như mô tả ở mục số 2 - Cấu trúc thư mục cho phần client.

Các bước để xây dựng 1 module mới :

* Viết các service cần thiết của module vào file service.js trong thư mục redux của module
* Định nghĩa tên cho các hành động mà người dùng sẽ gọi đến trong file constants.js
* Viết các hàm tương ứng với các lời gọi mà người dùng yêu cầu.
* Định nghĩa cho cấu trúc state cho module sẽ được lưu trong store như thế nào. Sử dung cấu trúc switch-case để tách bạch rõ ràng với các action mà người dùng sẽ gọi để xử lý dữ liệu nhận được.

Về phần giao diện cho module thì sẽ viết và lưu trong folder components của module tương ứng.