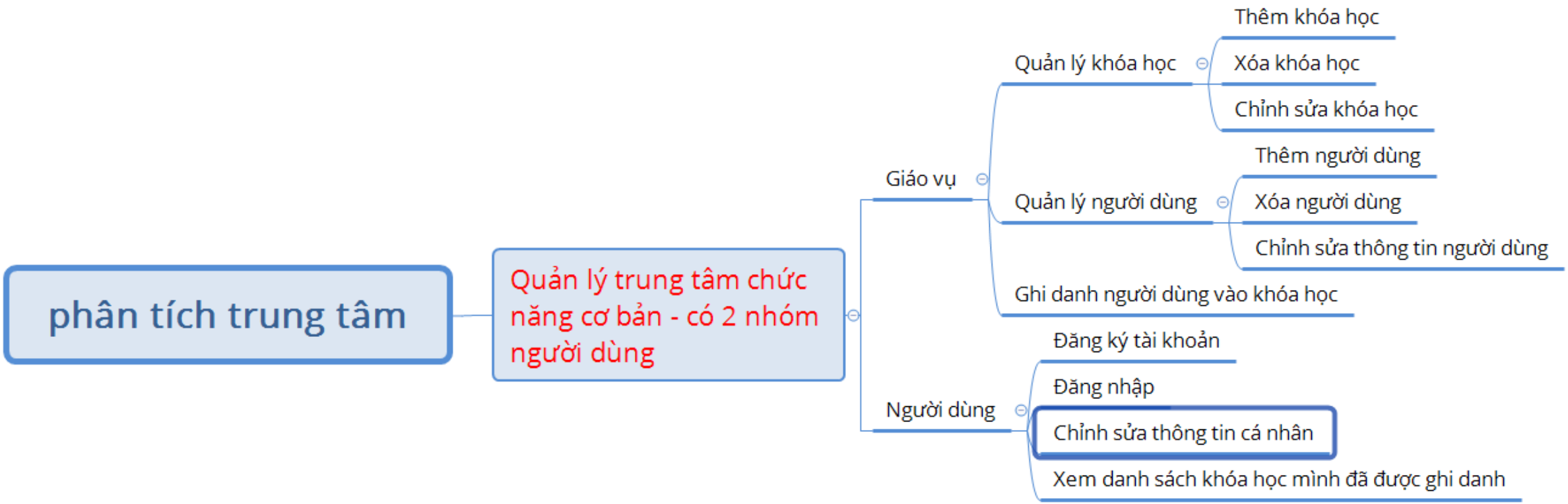
Đặc tả dự án Quản lý Trung Tâm

Hệ thống cho phép người dùng đăng kí thông tin và cho phép đăng nhập hệ thống.
Hệ thống chia làm 2 loại người dùng, Học viên và Giáo vụ.
Học viên sau khi đăng kí sẽ được lưu vào dữ liệu và có thông tin đăng nhập. Học viên đăng nhập sẽ có màn hình Dashboard gồm các chức năng:

+ Chỉnh sửa thông tin cá nhân,
+ Xem danh sách khóa học (hiển thị dạng thumnail có hình ảnh và mọt số thông tin cần thiết như tên, ngày tháng ghi danh học, số lượt học viên, lượt xem).
+ Học viên sẽ có nút Ghi danh, click vào và sẽ ghi danh vào khóa học.
+ Khóa học sẽ có mô tả vắn tắt và một số bài có link video youtube, click khóa học sẽ được xem chi tiết khóa học.

Khi giáo vụ đăng nhập sẽ có:

+ Dashboard highchart hiển thị tình hình ghi danh các khóa học.
+ Xem, thêm, xóa, sửa Học viên.
+ Thêm xóa sửa Danh sách khóa học.
+ Tạo một khóa học mới


phân tích trung tâm
Quản lý trung tâm chức năng cơ bản - có 2 nhóm người dùng
Giáo vụ
Quản lý khóa học
Thêm khóa học
Xóa khóa học
Chỉnh sửa khóa học
Quản lý người dùng
Thêm người dùng
Xóa người dùng
Chỉnh sửa thông tin người dùng
Ghi danh người dùng vào khóa học
Người dùng
Đăng ký tài khoản
Đăng nhập
Chỉnh sửa thông tin cá nhân
Xem danh sách khóa học mình đã được ghi danh

Sơ đồ lớp đối tượng

❖ Đặt tả lớp đối tượng

✓ Lớp đối tượng NguoiDung bao gồm các thuộc tính:

+ TaiKhoan: Mỗi người dùng cần xác định bởi 1 tài khoản (cần kiểm tra validation)
+ MatKhau: Mật khẩu người dung
+ HoTen: Họ tên người dung
+ Email: Email người dung
+ MaLoaiNguoiDung: Chỉ nhận 2 giá trị là
-HV: Học viên
-GV: Giáo vụ
+TenLoaiNguoiDung: Thuộc tính này dùng để lấy dữ liệu từ backend về sẽ chứa thông tin này.

✓ Từ prototype này ta có thể thiết kết prototype **DanhSachNguoiDung** chứa mảng NguoiDung để thao tác một số phương thức như sắp xếp theo họ tên, email, sdt v...v... Các phương thức tìm kiếm sinh viên theo tên hay tài khoản

NguoiDung
▪ TaiKhoan
▪ MatKhau
▪ HoTen
▪ Email
▪ SoDT
▪ MaLoaiNguoiDung
▪ TenLoaiNguoiDung

Sơ đồ lớp đối tượng

❖ Đặt tả lớp đối tượng Khóa học

✓ Lớp đối tượng KhoaHoc bao gồm các thuộc tính:

+MaKhoaHoc: Mỗi khóa học được xác định bởi MaKhoaHoc (Không trùng nhau)

+TenKhoaHoc: Tên của khóa học

+MoTa: Mô tả khóa học

+HinhAnh: Hình ảnh khóa học (Có thể copy link từ google khi thao tác thêm sửa).

+LuotXem: Lượt xem của khóa học đó

+NguoiTa: Thuộc tính người tạo này khi thực hiện tác vụ thêm sửa phải được lấy từ danh sách NguoiDung có thuộc tính MaLoaiNguoiDung = "GV".

KhoaHoc
▪ MaKhoaHoc
▪ TenKhoaHoc
▪ MoTa
▪ HinhAnh
▪ LuotXem
▪ NguoiTao

✓ Tương tự người dùng ta cũng cần có prototype **DanhSachKhoaHoc** để quản lý các đối tượng khóa học với một số chức năng như tìm kiếm theo TenKhoaHoc, MoTa ...

Api – Đặt tả chức năng

GET api/QuanLyTrungTam/DanhSachKhoaHoc

Lưu ý: Phần GET, POST, PUT, DELETE là những giao thức

Đường dẫn đầy đủ: http://sv.myclass.vn/api/QuanLyTrungTam/DanhSachKhoaHoc

Chức năng: Lấy về 1 mảng các đối tượng khóa học. Các bạn có thể console.log để xem kết quả trả về lấy dữ liệu gán đúng lớp đối tượng

KhoaHocService.js

```
function ServiceKhoaHoc()
{
    this.LayDanhSachKhoaHoc = function()
    {
        //Khai báo api
        urlAPI = `http://sv.myclass.vn/api/QuanLyTrungTam/DanhSachKhoaHoc`;
        //Yêu cầu api đến backend thông qua giao thức GET để lấy dữ liệu khóa học về
        return $.ajax({
            type:"GET",
            url: urlAPI,
        });
    }
}
```

Khi gọi service

```
//Khi sử dụng khởi tạo service khóa học
var serviceKH = new ServiceKhoaHoc();
//Từ service gọi phương thức lấy danh sách khóa học
serviceKH.LayDanhSachKhoaHoc().done(function(ketqua){
    //Nén kết quả lại lưu vào localstorage
    var jsonKhoaHoc = JSON.stringify(ketqua);
    //Lưu vào localstorage
    localStorage.setItem('DanhSachKhoaHoc',jsonKhoaHoc);
}).fail(function(error){
    console.log(error);
})
```

Api – Đặt tả chức năng

GET api/QuanLyTrungTam/ChiTietKhoaHoc/{id}

Lưu ý: Phần GET, POST, PUT, DELETE là những giao thức

Đường dẫn đầy đủ: http://sv.myclass.vn/api/QuanLyTrungTam/ChiTietKhoaHoc/{id}

Chức năng: Lấy về một đối tượng khóa học. Để gọi và sử dụng tương tự như service ở trên. Tượng tự các bạn có thể console.log() để kiểm tra kết quả.

KhoaHocService.js

```
this.ChiTietKhoaHoc = function(id)
{
    //Lưu ý đường dẫn có chứa tham số là id của khóa học
    urlAPI = `http://sv.myclass.vn/api/QuanLyTrungTam/ChiTietKhoaHoc/${id}`;
    return $.ajax({
        type:"GET",
        url: urlAPI
    });
}
```

Api – Đặt tả chức năng

POST api/QuanLyTrungTam/ThemKhoaHoc

Lưu ý: Phần GET, POST, PUT, DELETE là những giao thức

Đường dẫn đầy đủ: http://sv.myclass.vn/api/QuanLyTrungTam/ThemKhoaHoc

Chức năng: Gửi thông tin là đối tượng khóa học thông qua service tiến hành gửi thông tin khóa học lên server để thêm vào database. Tương tự có thể console.log(Kiểm tra kết quả)

KhoaHocService.js

```
this.ThemKhoaHoc = function(khoahoc) //Nhận vào đối tượng KhoaHoc
{
    //api thêm khóa học
    urlAPI = `http://sv.myclass.vn/api/QuanLyTrungTam/ThemKhoaHoc`;
    return $.ajax({
        type: "POST",
        url: urlAPI,
        dataType: "json",
        data: khoahoc,

    });
}
```

Api – Đặt tả chức năng

PUT api/QuanLyTrungTam/CapNhatKhoaHoc

Lưu ý: Phần GET, POST, PUT, DELETE là những giao thức

Đường dẫn đầy đủ: http://sv.myclass.vn/api/QuanLyTrungTam/CapNhatKhoaHoc

Chức năng: Gửi thông tin qua service tiến hành update dữ liệu của khóa học tại server(Chỉnh sửa thông tin khóa học). Cũng có thể console.log để kiểm tra kết quả.

KhoaHocService.js

```
this.CapNhatThongTinKhoaHoc = function(id,name,des,luotxem,creater)
{
    //Ổ đây các bạn có thể tận dụng lớp đối tượng khóa học truyền vào 1 đối tượng khóa học cũng được
    //Hoặc các bạn truyền từng tham số rồi sử dụng tạo object json theo kiểu này.
    //Đối với giao thức PUT cần nén dữ liệu objec json lại thành chuỗi
    var ngd = JSON.stringify({MaKhoaHoc:id,TenKhoaHoc:name,MoTa:des,LuotXem:luotxem,NguoiTao:creater});
    urlAPI = `http://sv.myclass.vn/api/QuanLyTrungTam/capnhatkhoahoc`;
    return $.ajax({
            type: "PUT",
            url: urlAPI,
            contentType: "application/json",
            dataType: "json",
            data: ngd,

}
```

Api – Đặt tả chức năng

DELETE api/QuanLyTrungTam/XoaKhoaHoc/{id}

Lưu ý: Phần GET, POST, PUT, DELETE là những giao thức

Đường dẫn đầy đủ: http://sv.myclass.vn/api/QuanLyTrungTam/XoaKhoaHoc/{id}

Chức năng: Dùng giao thức Delete kèm theo mã khóa học gửi về server để tiến hành xóa khóa học(Lưu ý: Nếu khóa học đó đã được ghi danh cho người dùng rồi thì sẽ không xóa được). Tương tự có thể kiểm tra console.log() kết quả

KhoaHocService.js

```
this.XoaKhoaHoc = function(id)
{
    var mangKQXoa = [];
    urlAPI = `http://sv.myclass.vn/api/QuanLyTrungTam/XoaKhoaHoc/${id}`;
    return $.ajax({
        type: "DELETE",
        url: urlAPI,
    });
}
```

Api – Đặt tả chức năng

POST api/QuanLyTrungTam/GhiDanhKhoaHoc

Lưu ý: Phần GET, POST, PUT, DELETE là những giao thức

Đường dẫn đầy đủ: http://sv.myclass.vn/api/QuanLyTrungTam/GhiDanhKhoaHoc

Chức năng: Dùng giao thức POST kèm theo mã khóa học và taikhoan để gửi về server thực hiện nghiệp vụ ghi danh học viên đó vào khóa học.

KhoaHocService.js

```
this.GhiDanhKhoaHoc = function(maKhoaHoc,taiKhoan)
{

    var model = JSON.stringify({MaKhoaHoc:maKhoaHoc, TaiKhoan:taiKhoan});
    urlAPI = `http://sv.myclass.vn/api/QuanLyTrungTam/GhiDanhKhoaHoc`;
    return $.ajax({
        type: "POST",
        url: urlAPI,
        contentType: "application/json",
        dataType: "json",
        data: model
    });
}
```

Một số api khác tương tự

GET api/QuanLyTrungTam/DanhSachHocvien

Lấy danh sách người dùng có MaLoaiNguoiDung là HV từ api.

GET api/QuanLyTrungTam/DanhSachNguoiDung

Lấy danh sách người dung bao gồm cả học viên và giáo vụ (Có MaLoaiNguoiDung = HV và GV).

GET api/QuanLyTrungTam/ThongTinNguoiDung?taikhoan={taikhoan}

Api nhận vào tham số qua link(url) là 1 tài khoản của người dùng. Trả về thông tin chi tiết của người dùng đó.

POST api/QuanLyTrungTam/ThemNguoiDung

Api nhận vào tham số là 1 tham số là đối tượng người dùng
Có định dạng json: {TaiKhoan:value, MatKhau:Value,HoTen: Value, Email:Value, SoDT:Value, MaLoaiNguoiDung:Value }
có thể sử dụng đối tượng NguoiDung nhưng không cần gán giá trị cho TenLoaiNguoiDung.

PUT api/QuanLyTrungTam/CapNhatThongTinNguoiDung

Api nhận vào tham số là 1 tham số là đối tượng người dùng
Có định dạng json: {TaiKhoan:value, MatKhau:Value, Email:Value, SoDT:Value, MaLoaiNguoiDung:Value } =>
Tuyên nhiên cần nén đối tượng này lại bằng phương thức JSON.stringify(nguoiDung). Vì giao thức là PUT

DELETE api/QuanLyTrungTam/XoaNguoiDung/{id}

Api nhận tham số qua link(url) là taikhoan của người dùng để tiến hành xóa người dùng.

Một số api khác tương tự

POST api/QuanLyTrungTam/DangKy

Api nhận vào tham số là 1 tham số là đối tượng người dùng
Có định dạng json: {TaiKhoan:value, MatKhau:Value, Email:Value, SoDT:Value, MaLoaiNguoiDung:Value } có thể sử dụng đối tượng NguoiDung nhưng không cần gán giá trị cho TenLoaiNguoiDung.

GET api/QuanLyTrungTam/DangNhap?taikhoan={taikhoan}&matkhau={matkhau}

Api nhận vào tham số là 1 tham số là đối tượng người dùng
Có định dạng json: {TaiKhoan:value, MatKhau:Value, Email:Value, SoDT:Value, MaLoaiNguoiDung:Value } có thể sử dụng đối tượng NguoiDung nhưng không cần gán giá trị cho TenLoaiNguoiDung.

GET api/QuanLyTrungTam/LayThongtinKhoaHoc?taikhoan={taikhoan}

Api nhận vào tham số từ link là taikhoan. Sẽ trả về danh sách các khóa học mà người dung này đã ghi danh

Lưu ý: Để kiểm tra kết quả