**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC NHA TRANG**

**KHOA CÔNG NGHỆ THÔNG TIN**

****

**BÀI TẬP NHÓM MÔN HỌC ĐỒ ÁN PHÁT TRIỂN ỨNG DỤNG WEB**

**Tên dự án: FIT.NTU.CMS**

**Giảng viên hướng dẫn: Mai Cường Thọ**

**Lớp: 64.CNTT-CLC**

**Sinh viên thực hiện: 64130053 - Nguyễn Đức Anh**

**64130107 - Đinh Nhật Bảo**

**64130851 - Ngô Quang Huy**

**Khánh Hòa, tháng 1 năm 2024**

**MỤC LỤC**

**DANH MỤC HÌNH ẢNH**

# I. XÁC ĐỊNH YÊU CẦU

Trong thời đại công nghệ, các khoa trong trường đại học ngày càng cần có một hệ thống quản trị nội dung riêng biệt để công bố thông tin học tập, sự kiện, thông báo cho sinh viên và giảng viên. Tuy nhiên, đa phần hiện nay các khoa vẫn phụ thuộc vào trang web chung của trường, gây khó khăn trong việc cập nhật nội dung kịp thời và linh hoạt. Việc xây dựng một hệ thống CMS riêng, có khả năng tùy chỉnh theo từng khoa là một yêu cầu thực tế và cần thiết nhằm tăng hiệu quả truyền thông nội bộ và phát triển hệ sinh thái công nghệ của nhà trường.

## A. Mục tiêu và phạm vi dự án

### 1. Mục tiêu

* Xây dựng hệ thống CMS đơn giản nhưng đầy đủ chức năng cần thiết cho một Khoa trong trường Đại học.
* Cho phép người quản trị dễ dàng tạo bài viết, thông báo, sự kiện và quản lý menu website.
* Cung cấp khả năng kiểm duyệt nội dung bài viết trước khi đăng công khai.

### 2. Phạm vi

Xây dựng website CMS cho Khoa CNTT – Trường ĐH Nha Trang.

## B. Các vai trò chính:

* + Quản trị viên: Toàn quyền quản lý nội dung.
  + Kiểm duyệt viên: Kiểm tra và duyệt nội dung hiển thị
  + Người viết bài: Có thể tạo bài viết nhưng cần duyệt mới được hiển thị.
  + Người dùng truy cập: Xem nội dung công khai.

## C. Hệ thống hỗ trợ:

* + Tạo trang tĩnh (Giới thiệu, Liên hệ,...)
  + Tạo bài viết theo chuyên mục
  + Tạo thông báo và sự kiện
  + Tạo menu điều hướng
  + Responsive UI
  + Cơ sở dữ liệu: MySQL

# II. PHÂN TÍCH - THIẾT KẾ HỆ THỐNG

## A. Phân tích hệ thống

### 1. Yêu cầu chức năng:

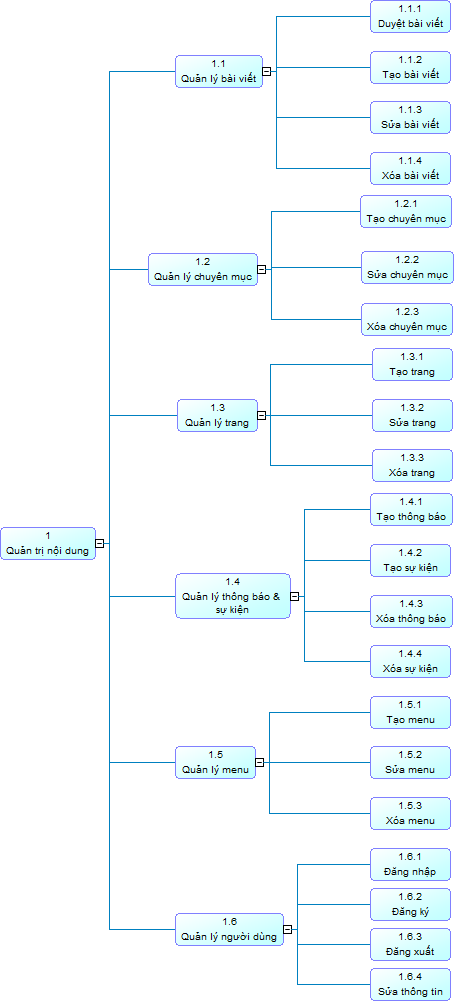
* Đăng nhập/đăng xuất quản trị viên
  + Cho phép quản trị viên sử dụng tài khoản để đăng nhập vào hệ thống
  + Đảm bảo bảo mật qua xác thực
  + Sau khi đăng nhập, có thể đăng xuất an toàn khỏi hệ thống
* Thêm/xóa/sửa bài viết theo chuyên mục
  + Cho phép tạo mới bài viết và gán vào chuyên mục phù hợp
  + Quản trị viên có thể sửa, xóa bài viết
  + Hệ thống lọc bài viết theo chuyên mục, ngày tạo hoặc trạng thái duyệt
* Kiểm duyệt bài viết
  + Mọi bài viết được tạo bởi người viết phải được quản trị viên duyệt trước khi hiển thị công khai
  + Chuyển trạng thái từ “Chờ duyệt” sang “Đã duyệt” hoặc “Từ chối”
* Tạo/trình bày trang tĩnh
  + Cho phép tạo trang như “Giới thiệu”, “Liên hệ”, “Hướng dẫn” với nội dung HTML tùy chỉnh
  + Các trang có thể được chỉnh sửa hoặc ẩn nếu cần
* Tạo thông báo và sự kiện
  + Hỗ trợ đăng tải thông báo ngắn hạn hoặc dài hạn, hiển thị theo ngày bắt đầu/kết thúc
  + Sự kiện có thể bao gồm thời gian, địa điểm, mô tả và hiển thị nổi bật
* Tạo và quản lý menu điều hướng
  + Quản trị viên có thể tạo, chỉnh sửa và sắp xếp các mục menu theo thứ tự hiển thị
  + Liên kết menu với trang tĩnh, bài viết, chuyên mục hoặc URL tùy chỉnh

### 2. Yêu cầu phi chức năng:

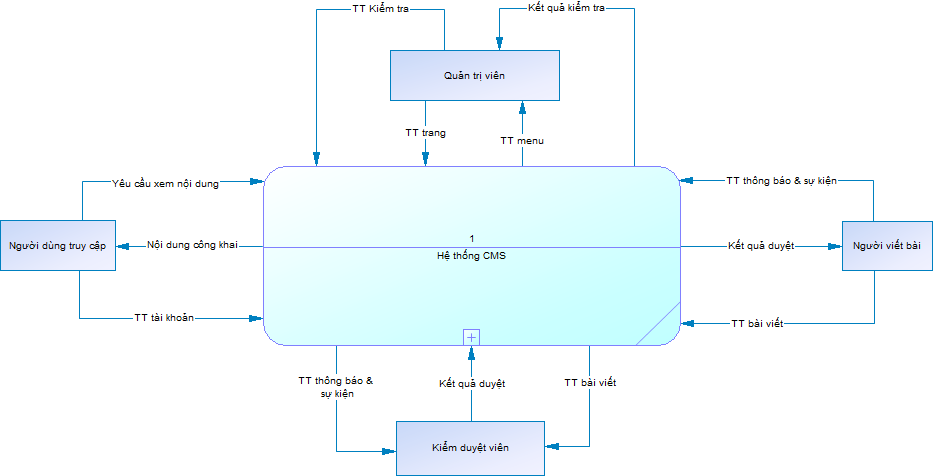
* Giao diện thân thiện, hiển thị tốt trên mọi thiết bị
  + Giao diện phù hợp với mọi loại thiết bị (máy tính, máy tính bảng, điện thoại)
  + Bố cục rõ ràng, dễ sử dụng, hỗ trợ cho người mới
* Hệ thống bảo mật
  + Sử dụng mã hóa mật khẩu cho tài khoản
  + Phân quyền người dùng rõ ràng
  + Ngăn chặn các lỗ hổng như SQL Injection, XSS
* Hiệu suất truy cập cao
  + Hệ thống xử lý tốt lượng truy cập lớn với thời gian phản hồi ngắn
  + Cấu trúc phần mềm hỗ trợ mở rộng tính năng trong tương lai
* Dễ bảo trì, mở rộng
  + Mã nguồn rõ ràng, có chú thích và tài liệu hướng dẫn sử dụng
  + Tách biệt các lớp chức năng: controller, service, repository (theo mô hình MVC)
* Dữ liệu được lưu trữ và truy vấn hiệu quả
  + Sử dụng cơ sở dữ liệu MySQL được tối ưu với chỉ mục
  + Hỗ trợ phân trang khi hiển thị dữ liệu lớn

## B. Thiết kế hệ thống

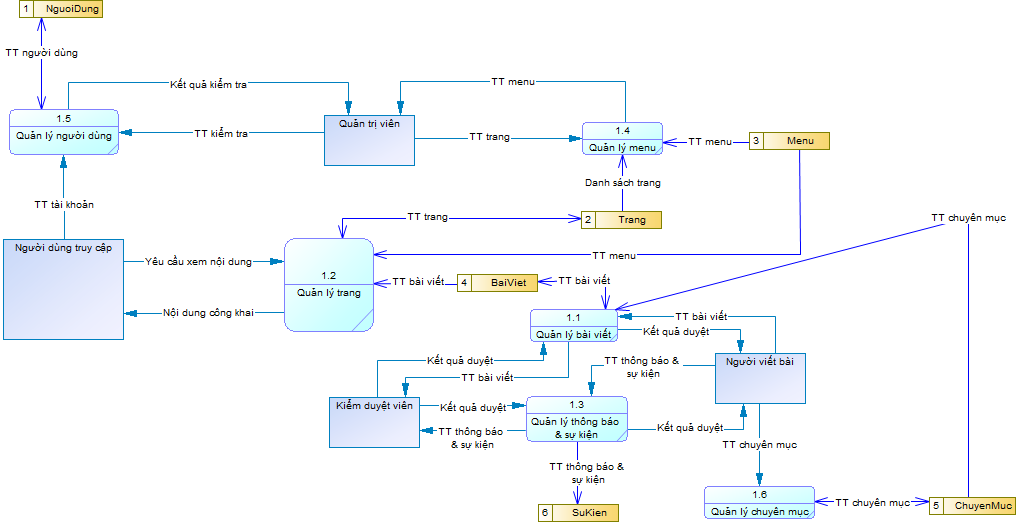
* + Sơ đồ phân rã chức năng (BFD): Hiển thị các chức năng chính và chức năng phụ trong hệ thống

****

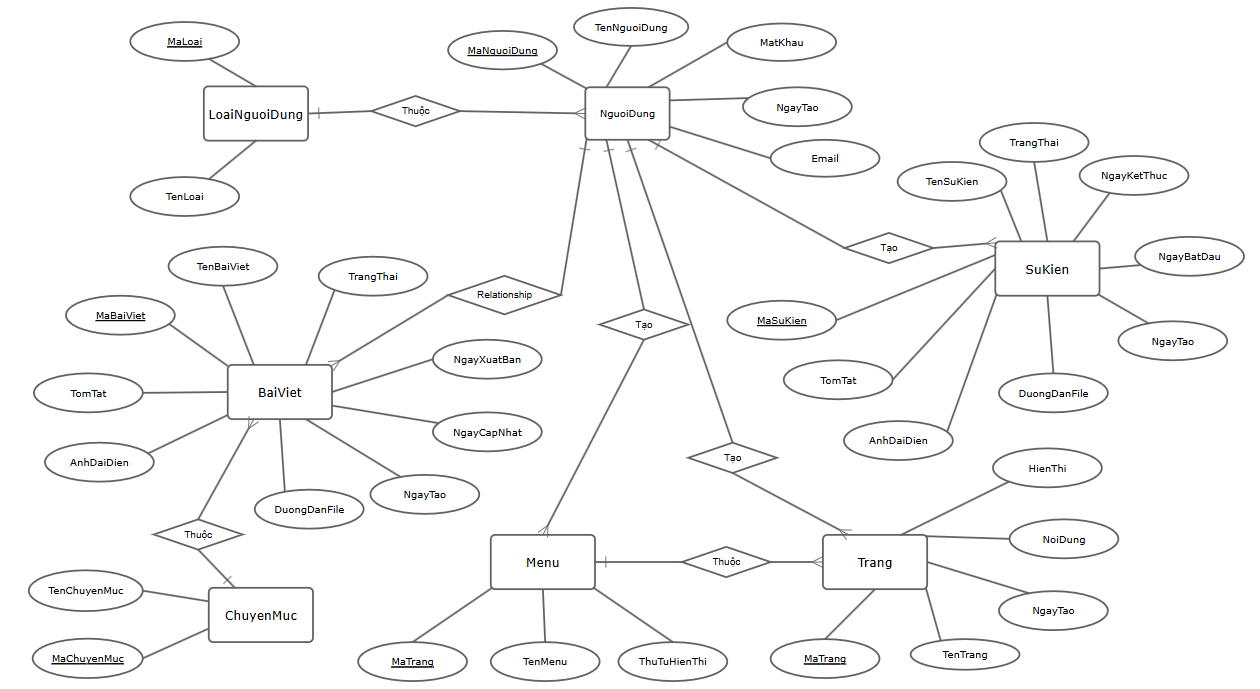
* + Sơ đồ luồng dữ liệu (DFD): Mô tả cách dữ liệu luân chuyển trong hệ thống
    - Sơ đồ luồng dữ liệu (DFD) mức 0:

****

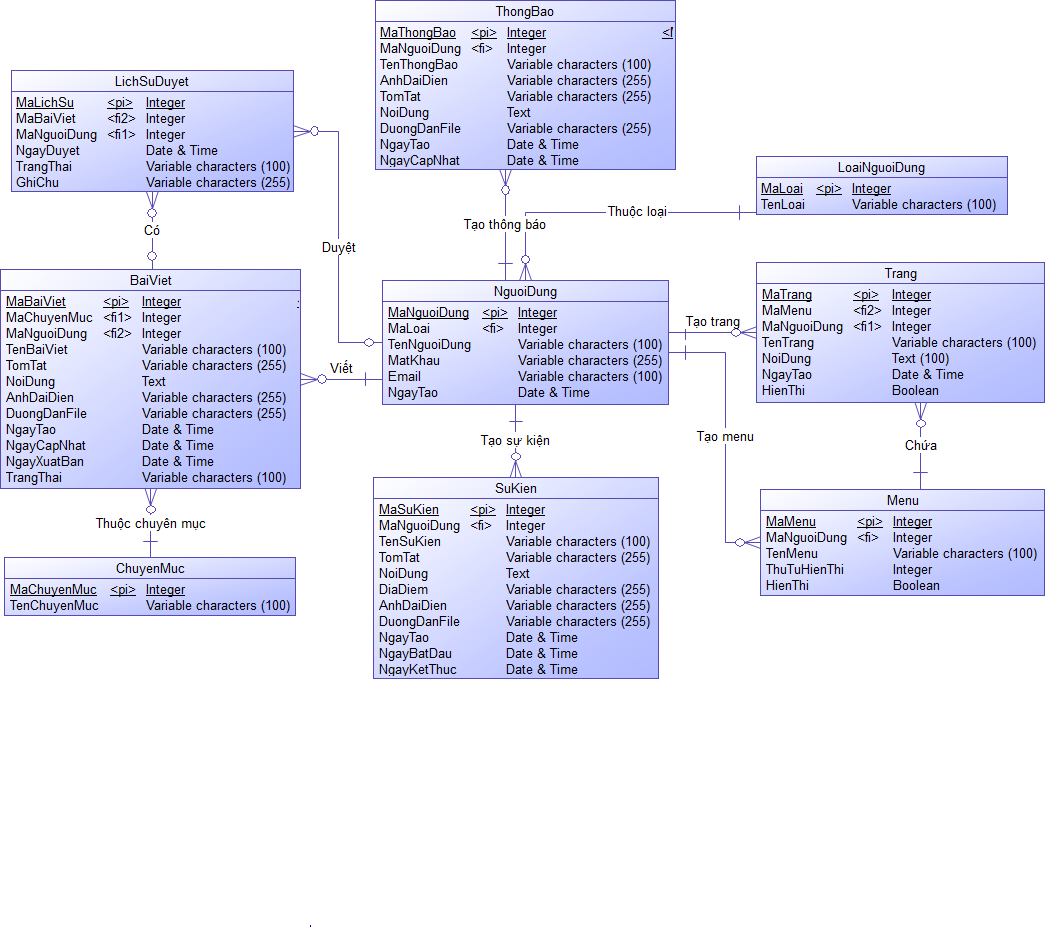
* + - Sơ đồ luồng dữ liệu (DFD) mức 1:

****

* + Sơ đồ ERD mức logic: Mô hình hóa quan hệ giữa các thực thể

****

* + S​ơ đồ ERD mức vật lý: Thể hiện chi tiết bảng, cột và mối quan hệ trong cơ sở dữ liệu

****

* Từ điển dữ liệu:

|  | Tên cột | Kiểu dữ liệu | Độ rộng | Mô tả |
| --- | --- | --- | --- | --- |
| NguoiDung  (Người dùng) | MaNguoiDung | int |  | Mã người dùng |
| MaLoai | int |  | Mã loại |
| TenDangNhap | char | 100 | Tên đăng nhập |
| MatKhau | char | 255 | Mật khẩu |
| Email | char | 100 | Đỉa chỉ email |
| NgayTao | datetime |  | Ngày tạo |
| SuKien  (Sự kiện) | MaSuKien | int |  | Mã sự kiện |
| TieuDe | char | 100 | Tiêu đề |
| TomTat | text |  | Tóm tắt |
| NoiDung | text |  | Nội dung |
| DiaDiem | char | 255 | Địa điểm |
| AnhDaiDien | char | 255 | Ảnh đại diện |
| DuongDanFile | char | 255 | Đường dẫn File |
| NgayTao | datetime |  | Ngày tạo sự kiện |
| NgayBatDau | datetime |  | Ngày bắt đầu sự kiện |
| NgayKetThuc | datetime |  | Ngày kết thúc |
| MaNguoiDung | int |  | Mã người dùng |
| HienThi | boolean |  | Hiển thị sự kiện |
| LichSuDuyet  (Lịch sử duyệt) | MaLichSu | int |  | Mã lịch sử |
| MaBaiViet | int |  | Mã bài viết |
| MaNguoiDung | int |  | Mã người dùng |
| NgayDuyet | datetime |  | Ngày duyệt |
| TrangThai | char | 100 | Trạng thái |
| GhiChu | char | 255 | Ghi chú |
| BaiViet  (Bài viết | MaBaiViet | int |  | Mã bài viết |
| MaNguoiDung | int |  | Mã người dùng |
| MaChuyenMuc | int |  | Mã chuyên mục |
| TomTat | text |  | Tóm tắt |
| TieuDe | char |  | Tiêu đề |
| NoiDung | text |  | Nội dung |
| AnhDaiDien | char | 255 | Ảnh đại diện |
| DuongDanFile | char | 255 | Đường dẫn File |
| NgayTao | datetime |  | Ngày tạo |
| NgayCapNhat | datetime |  | Ngày cập nhật |
| NgayXuatBan | datetime |  | Ngày xuất bản |
| TrangThai | char | 100 | Trạng thái |
| HienThi | boolean |  | Hiển thị bài viết |
| TuKhoaTimKiem | char | 255 | Từ khóa thuộc metadata |
| MoTa | char | 255 | Mô tả thuộc metadata |
| ThongBao  (Thông báo) | MaThongBao | int |  | Mã thông báo |
| MaNguoiDung | int |  | Mã người dùng |
| TieuDe | char | 100 | Tiêu đề |
| TomTat | text |  | Tóm tắt |
| NoiDung | text |  | Nội dung |
| AnhDaiDien | char | 255 | Ảnh đại diện |
| DuongDanFile | char | 255 | Đường dẫn File |
| NgayTao | datetime |  | Ngày tạo |
| NgayCapNhat | datetime |  | Ngày cập nhật |
| TrangThai | char | 100 | Trạng thái |
| HienThi | boolean |  | Hiển thị bài viết |
| ChuyenMuc  (Chuyên mục) | MaChuyenMuc | int |  | Mã chuyên mục |
| TenChuyenMuc | char | 100 | Tên chuyên mục |
| LoaiNguoiDung  (Loại người dùng) | MaLoai | int |  | Mã loại |
| TenLoai | char | 100 | Tên loại |
| Trang | MaTrang | int |  | Mã trang |
| MaMenu | int |  | Mã menu |
| MaNguoiDung | int |  | Mã người dùng |
| TieuDe | char | 100 | Tiêu đề |
| NgayTao | datetime |  | Ngày tạo |
| HienThi | boolean |  | Hiển thị |
| TuKhoaTimKiem | char | 255 | Từ khóa thuộc metadata |
| MoTa | char | 255 | Mô tả thuộc metadata |
| Menu | MaMenu | int |  | Mã menu |
| MaNguoiDung | int |  | Mã người dùng |
| TieuDe | char | 100 | Tiêu đề |
| ThuTuHienThi | int |  | Thứ tự hiển thị |

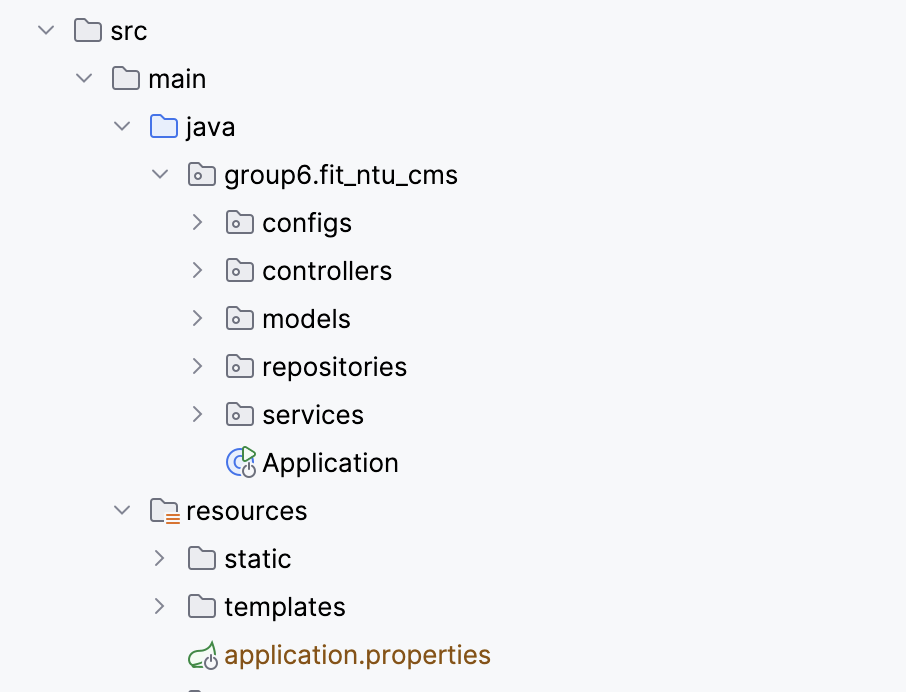
# III. TỔ CHỨC VÀ CÀI ĐẶT DỰ ÁN

## A. Tổ chức dự án

Dự án Faculty.NTU.CMS được lập trình sử dụng Spring Boot (Java) theo mô hình MVC (Model – View – Controller). Dữ liệu được tổ chức và lưu trữ trong hệ quản trị cơ sở dữ liệu MySQL.

Việc sử dụng Spring Boot giúp đơn giản hóa việc cấu hình ứng dụng và hỗ trợ tích hợp các thư viện cần thiết để phát triển một hệ thống web hiện đại. MVC là mô hình phổ biến giúp tách biệt các thành phần xử lý nghiệp vụ, giao diện người dùng và truy xuất dữ liệu.

1. Cấu trúc thư mục chính của dự án:



1. Lý do chọn Spring Boot và MySQL:

* Spring Boot hỗ trợ nhanh chóng xây dựng ứng dụng web hiện đại, có hệ sinh thái phong phú.
* MySQL là hệ quản trị cơ sở dữ liệu phổ biến, dễ sử dụng và tương thích tốt với các ứng dụng Spring Boot.
* Hệ thống dễ bảo trì, mở rộng và triển khai thực tế.
* Triển khai và hosting:
* Dự án dự kiến sẽ được triển khai và host online trên AWS EC2 để đảm bảo độ tin cậy, khả năng mở rộng và truy cập toàn cầu.

## B. Một số xử lý cơ bản

1. **Xử lý đăng ký tài khoản**

Trong chức năng này, FIT.CMS cho phép đăng ký qua các kênh: CMS, Facebook, Google.

Để làm được chức năng đăng ký bằng Facebook

Để làm được chức năng đăng ký bằng Google

| @PostMapping({"/register", "signup"})  public String register(@RequestParam String username,  @RequestParam String email,  @RequestParam String password,  @RequestParam String repeatPassword,  ModelMap model) {  if(!repeatPassword.equals(password)) {  model.addAttribute("error", "Password do not match.");  return "register";  }  UsersModel user = new UsersModel();  user.setUsername(username);  user.setEmail(email);  user.setPassword(password);  user.setCreatedDate(LocalDateTime.*now*());  user.setRole(Role.*USER*);  boolean result = usersService.register(user);  if(result) {  model.addAttribute("success", "Registration successfully!");  return "login";  } else {  return "register";  }  } |
| --- |

1. **Xử lý đăng nhập tài khoản**

Để quản lý phiên làm việc (phiên đăng nhập), ta dùng lớp HttpSession

| @PostMapping({"/login", "signin"})  public String login(@RequestParam String email,  @RequestParam String password,  ModelMap model,  HttpSession session) {  Optional<UsersModel> user = usersService.login(email, password);  if(user.isPresent()) {  session.setAttribute("username", user.get().getUsername());  session.setAttribute("role", user.get().getRole().name());  model.addAttribute("success","Login successfully!");  return "index";  } else {  model.addAttribute("error", "Incorrect email ỏ password!");  return "login";  }  } |
| --- |

1. **Xử lý quên mật khẩu:**

| @PostMapping("/forgot-password")  public String processForgotPassword(@RequestParam("email") String email, ModelMap model) {  Optional<UsersModel> userOpt = usersService.findByEmail(email);  if (userOpt.isEmpty()) {  model.addAttribute("error", "Email not found");  return "forgot-password";  }  UsersModel user = userOpt.get();  String token = UUID.*randomUUID*().toString();  user.setResetToken(token);  usersService.save(user);  String resetLink = "http://localhost:8080/reset-password?token=" + token;  sendResetEmail(email, resetLink);  model.addAttribute("success", "Successfully! Please check your email to reset your password.");  return "forgot-password";  }  private void sendResetEmail(String toEmail, String resetLink) {  SimpleMailMessage message = new SimpleMailMessage();  message.setTo(toEmail);  message.setSubject("Password reset request");  message.setText("Click the link to reset your password: " + resetLink);  mailSender.send(message);  } |
| --- |

# IV. KẾT QUẢ

* Một số màn hình chạy chương trình:

|  |
| --- |
| Giao diện đăng nhập. Người dùng có thể đăng nhập bằng tài khoản hệ thống hoặc xác thực qua Google hoặc Facebook |

|  |
| --- |
| Giao diện đăng ký tài khoản. Người dùng có thể đăng ký tài khoản |

|  |
| --- |
| Giao diện đăng nhập với vai trò ADMIN và giao diện đăng nhập với vai trò WRITER |

# V. TÀI LIỆU THAM KHẢO

# 

HỆ QUẢN TRỊ NỘI DUNG

**Mục tiêu:** Xây dựng một **Hệ quản trị nội dung cơ bản** cho một Khoa trong Trường Đại học (ví dụ Trường Đại học Nha Trang, Khoa Công nghệ thông tin)

***Tên dự án: Faculty.NTU.CMS***

# Yêu cầu

Phân tích, thiết kế và xây dựng một hệ quản trị nội dung, cho phép một đơn vị trong 1 trường đại học, có thể tùy biến và sử dụng để tạo ra trang web thông tin cho đơn vị mình

* Các yêu cầu tối thiểu:
  + Tạo trang
  + Tạo bài viết ( bài viết được phân theo chuyên mục, được kiểm duyệt nội dung, file đính kèm
  + Tạo các thông báo
  + Tạo các sự kiện (file đính kèm)
  + Tạo các menu
  + chuyên mục
  + upload file trong khi đăng bài để đính kèm file
  + …các chức năng khác, SV khảo sát và trình bày tại mục 2

# PHÂN TÍCH - THIẾT KẾ HỆ THỐNG

# TỔ CHỨC DỰ ÁN VÀ CÀI ĐẶT