|  |  |
| --- | --- |
| NGUYỄN THỊ CHANG |  |
| **BỘ CÔNG THƯƠNG**  **TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**  **---------------------------------** |
|  |
|  |
| ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC NGÀNH HỆ THỐNG THÔNG TIN |
| **XÂY DỰNG WEBSITE BÁO ĐIỆN TỬ TỈNH HẢI DƯƠNG SỬ DỤNG NGÔN NGỮ PYTHON** |
|  |
|  |
| **CBHD: ThS. NGUYỄN XUÂN HOÀNG** |
| HỆ THỐNG THÔNG TIN | **Sinh viên: Nguyễn Thị Chang** |
| **Mã sinh viên: 2018603223** |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
| *Hà Nội – Năm 2022* |

**Lời mở đầu**

Ngày nay thế giới hoàn toàn dựa vào các phương tiện điện tử trong cuộc phiêu lưu hàng ngày của nó. Mọi người không có thời gian để cập nhật thông tin qua báo chí hoặc xem hoặc nghe tin tức trên truyền hình hoặc đài. Mọi người ngày nay cần được cập nhật thông tin hàng ngày trong thế giới cạnh tranh này. Hầu hết mọi người nhận được thông tin về thế giới xung quanh thông qua Internet, một cách nhanh chóng, dễ tiếp cận và đáng tin cậy. WWW (World Wide Web) là trung tâm dịch vụ thông tin toàn cầu khổng lồ, được phân phối rộng rãi, dành cho các dịch vụ Thông tin: tin tức, quảng cáo, thông tin người tiêu dùng, quản lý tài chính, giáo dục, chính phủ, thương mại điện tử, v.v., thông tin siêu liên kết, thông tin truy cập và sử dụng. “Cổng thông tin Lathel” là một dịch vụ được giới thiệu để đáp ứng yêu cầu trên và giúp mọi người cập nhật tin tức, quan điểm, đánh giá, tin nóng và các tiêu đề mới nhất trong các lĩnh vực khác nhau cũng như các phát minh mới trên thế giới.

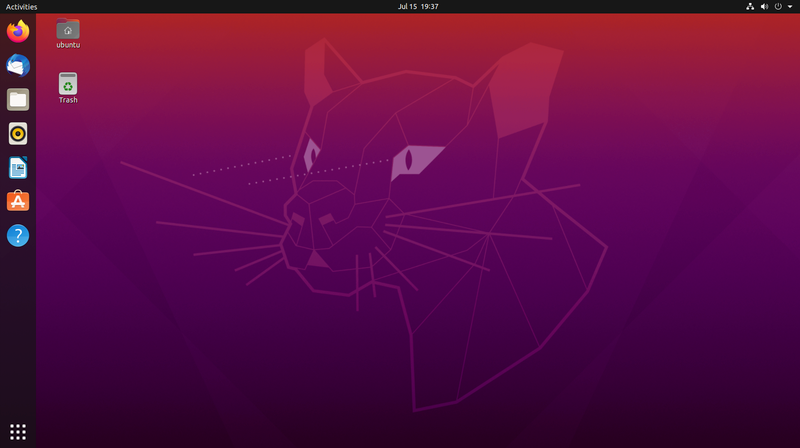
# **CHƯƠNG 1. TỔNG QUAN**

## **1.1. Đặt vấn đề**

## **1.2. Giới thiệu về môi trường Ubuntu**

### 1.2.1. Khái niệm và lịch sử cơ bản

a. Khái niệm



Hình 2. 1 Giao diện mặc định của ubuntu

[Ubuntu](https://bizflycloud.vn/tin-tuc/ubuntu-la-gi-co-nen-su-dung-ubuntu-khong-20180908094633985.htm) là [hệ điều hành máy tính](https://bizflycloud.vn/tin-tuc/he-dieu-hanh-la-gi-muc-tieu-va-chuc-nang-cua-he-dieu-hanh-20181018151637002.htm) dựa trên Debian GNU/Linux, một bản phân phối Linux thông dụng.

Ubuntu là phần mềm mã nguồn mở tự do, người dùng được tự do chạy, sao chép, phân phối, nghiên cứu, thay đổi và cải tiến phần mềm theo điều khoản của giấy phép GNU GPL. Ubuntu được tài trợ bởi Canonical Ltd (chủ sở hữu là một người Nam Phi Mark Shuttleworth).

Mục đích của Ubuntu là cung cấp một hệ điều hành ổn định, cập nhật cho người dùng thường, tập trung vào sự tiện dụng và dễ dàng cài đặt. Ubuntu được đánh xếp hạng là bản phân phối Linux thông dụng nhất cho máy tính để bàn, chiếm khoảng 30% số bản Linux được cài đặt trên máy tính để bàn năm 2007.

b. Lịch sử phát triển của Ubuntu

Bản phát hành đầu tiên của Ubuntu là vào 20 tháng 10 năm 2004, bắt đầu bằng việc tạo ra một nhánh tạm thời của dự án Debian Linux. Việc này đã được thực hiện để một phiên bản mới của Ubuntu có thể được phát hành mỗi 6 tháng, tạo ra một hệ điều hành được cập nhật thường xuyên hơn. Bản phát hành Ubuntu luôn gồm bản GNOME mới nhất, và được lên lịch phát hành khoảng 1 tháng sau GNOME. Khác với các nhánh có mục đích chung trước của Debian - như MEPIS, Xandros, Linspire, Progeny và Libranet, phần nhiều trong số chúng dựa vào các phần mềm bổ sung có mã đóng mô hình của một doanh nghiệp. Ubuntu lại giống với triết lý của Debian hơn và dùng các phần mềm miễn phí (libre) vào mọi thời điểm. Các gói của Ubuntu nói chung dựa trên các gói từ nhánh không ổn định của Debian: cả 2 bản phân phối đều dùng gói có định dạng deb của Debian và APT/Synaptic để quản lý các gói đã cài. Ubuntu đã đóng góp trực tiếp và lập tức tất cả thay đổi đến Debian, chứ không chỉ tuyên bố chúng lúc phát hành, mặc dù các gói của Debian và Ubuntu không cần thiết "tương thích nhị phân" với nhau. Nhiều nhà phát triển Ubuntu cũng là người duy trì các gói khoá (gói chủ chốt) của chính Debian. Dù sao, Ian Murdock, nhà sáng lập của Debian, đã chỉ trích Ubuntu vì sự không tương thích giữa các gói của Ubuntu và Debian, ông nói rằng Ubuntu đã làm sai lệch quá xa so với Debian Sarge, do đó không còn giữ được sự tương thích.

### 1.2.2. Tính năng của Ubuntu

a. Đặc điểm

Ubuntu kết hợp những đặc điểm nổi bật chung của hệ điều hành nhân Linux, như tính bảo mật trước mọi virus và malware, khả năng tùy biến cao, tốc độ, hiệu suất làm việc, và những đặc điểm riêng tiêu biểu của Ubuntu như giao diện bắt mắt, bóng bẩy, cài đặt ứng dụng đơn giản, sự dễ dàng trong việc sao lưu dữ liệu và sự hỗ trợ của một cộng đồng người dùng khổng lồ.

b. Cài đặt

Mỗi phiên bản phát hành có một đĩa chạy trực tiếp, cho phép người dùng xem xét phần cứng của họ có tương thích với hệ điều hành hay không trước khi cài đặt lên đĩa cứng bằng phần mềm Ubiquity. Tập tin ảnh đĩa có thể được tải về từ trang chủ Ubuntu, và các đĩa cài đặt có thể được cung cấp bởi bên thứ ba.

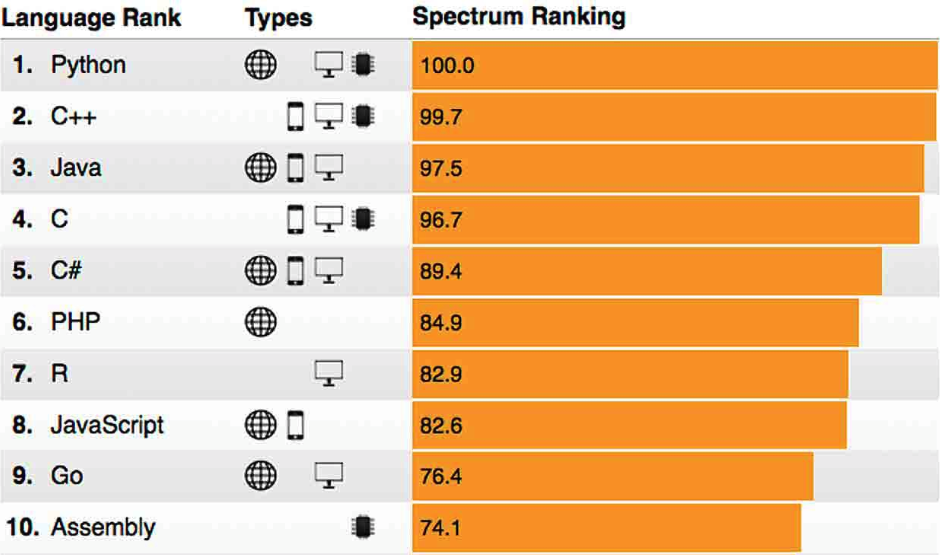
Đĩa cài đặt Ubuntu yêu cầu máy tính có từ 256 MB RAM trở lên. Quá trình cài đặt Ubuntu sử dụng giao diện đồ họa, tốc độ cài đặt phụ thuộc vào cấu hình máy tính, trung bình là từ 20 – 30 phút.

Ngoài ra, người dùng có thể cài đặt Ubuntu lên một đĩa ảo bên trong Windows bằng bộ cài đặt Wubi. Mặc dù tốc độ xử lý có thể bị giảm sút so với cài đặt đầy đủ, nhưng bù lại, cài đặt bên trong Windows không yêu cầu người dùng phải phân vùng lại đĩa cứng và có thể được gỡ bỏ dễ dàng từ ngay trong Windows.

**1.3. Tổng quan về ngôn ngữ Python**

### 1.3.1. Giới thiệu

Python là một ngôn ngữ lập trình thông dịch, hướng đối tượng, ngôn ngữ lập trình cấp cao được giải thích với ngữ nghĩa động. Python với triết lý thiết kế của nó rất thuận tiện cho việc đọc hiểu code, đơn giản và rõ ràng được thiết kế bởi **Guido van Rossum**. Thiết kế bắt đầu vào cuối những năm 1980 và được phát hành lần đầu tiên vào tháng**2 năm 1991**. Đến nay thì cộng đồng người sử dụng ngôn ngữ này rất đông, nếu so sánh từ bảng xếp hạng các ngôn ngữ năm 2018 thì Python đã leo lên vị trí số 1 trên bảng xếp hạng những ngôn ngữ lập trình phổ biến.



Hình 2. 2 IEEE Spectrum The Top Programming languages 2018

Python hoàn toàn tạo kiểu động và sử dụng cơ chế cấp phát bộ nhớ tự động. Ngôn ngữ này có cấu trúc dữ liệu cấp cao mạnh mẽ và cách tiếp cận đơn giản nhưng hiệu quả đối với lập trình hướng đối tượng. Do đó nó tương tự như Perl, Ruby, Scheme, Smalltalk, và Tcl. Ngôn ngữ này được phát triển trong một dự án mã mở, do tổ chức phi lợi nhuận Python Software Foundation quản lý. Theo đánh giá của Eric S. Raymond, Python là ngôn ngữ có hình thức rất sáng sủa, cấu trúc rõ ràng, thuận tiện cho người mới học lập trình. Cấu trúc của nó còn cho phép người sử dụng viết mã lệnh với số lần gõ phím tối thiểu, như nhận định của chính Guido van Rossum trong một bài phỏng vấn ông.

1.3.2. Tính năng

Ngôn ngữ lập trình đơn giản, dễ học: So với các ngôn ngữ lập trình khác như C++ hay Java, Python có cú pháp rất đơn giản, rõ ràng, dễ đọc và dễ viết hơn rất nhiều. Việc học lập trình sẽ trở nên thú vụ hơn vì Pythong cho phép bạn tập trung vào những giải pháp, chứ không phải cú pháp.

Khả năng di chuyển tốt: Bạn có thể di chuyển các chương trình của Python từ nền tảng này sang nền tảng khác và chạy nó mà không có bất kì sự thay đổi nào. Nó chạy liền mạch trên hầu hết tất cả nền tảng: Windows, macOS, Linux.

Miễn phí và mã nguồn mở: Python một cộng đồng lớn, không ngừng cải thiện và cập nhật cho phiên bản tốt hơn. Thế nên, bạn có thể sử dụng và phân phối Python, hoặc có thể dùng cho cả mục đích thương mại. Hơn thế nữa, bạn còn có thể thay đổi mã nguồn của nó.

Khả năng mở ộng và có thể nhúng: Bạn có thể dễ dàng kết hợp các phần code bằng C/C++ và ngôn ngữ khác vào code Python đối với các ứng dụng đòi hỏi sự phức tạp lớn. Điều này sẽ giúp cho ứng dụng của bạn có các tính năng tốt hơn và có khả năng cripting – một điều mà các ngôn ngữ khác khó làm được.

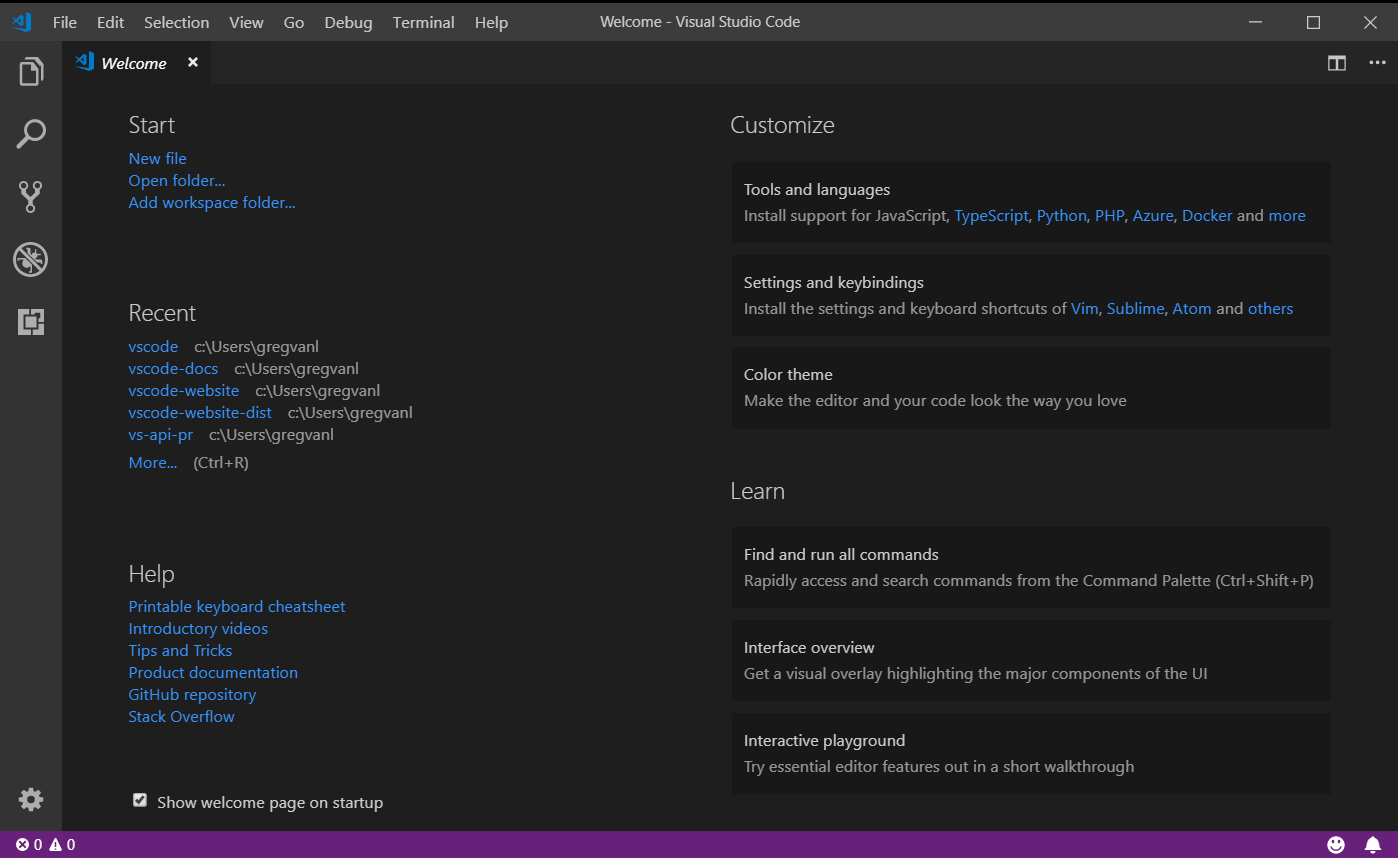
Ngôn ngữ thông dịch cấp cao: Khi sử dụng Python, chúng ta không phải lo lắng những nhiệm vụ như: quản lý bộ nhớ, dọn dẹp những dữ liệu vô nghĩa,… Khi chạy code Python, ngôn ngữ này sẽ sẽ tự chuyển động đổi code sang ngôn ngữ máy tính có thể hiểu. Bạn không cần lo lắng về bất kì hoạt động ở cấp thấp nào.

Thư viện tiêu chuẩn lớn: Thư viện tiêu chuẩn với số lượng vô cùng lớn của Python sẽ giúp cho công việc lập trình của bạn trở nên đơn giản hơn rất nhiều và chúng ta cũng không cần phải tự viết tất cả code. Những thư viện này được sử dụng bởi một cộng đồng lớn và luôn được kiểm tra kỹ lưỡng.

Hướng đối tượng: Mọi thứ trong Python đều là hướng đối tượng, lập trình hướng đối tượng (OPP) sẽ giúp giải quyết những vấn đề phức tạp một cách trực quan. Bạn có thể phân chia những vấn đề phức tạp thành những tập nhỏ hơn bằng cách tạo ra đối tượng với OPP.

## **1.4. Giới thiệu về Visual Studio Code**

Visual Studio Code là trình biên tập code hoàn toàn miễn phí, dành cho các hệ điều hành macOS, Windows và Linux được phát triển bởi Microsoft. Visual Code Studio hỗ trợ các chức năng như debug, tự hoàn thành mã thông minh, cải tiến mã nguồn, cho phép người dùng thay đổi theme, phím tắt…



Hình 2. 3 Giao diện welcome Visual Studio Code

**Một số tính năng nổi bật của Visual Studio Code**

* Hỗ trợ nhiều ngôn ngữ lập trình như C/C++, Visual Basic, Java, Python…
* Hoạt động tốt trên cả ba nền tảng macOS, Windows và Linux.
* Kho tiện ích mở rộng phong phú.
* Kho lưu trữ an toàn, tiện lợi, dễ dàng kết nối với Git. Lưu trữ dữ liệu dạng phân cấp.
* Hỗ trợ nhiều ứng dụng web, có sẵn trình soạn thảo và thiết kế web.
* Hỗ trợ viết code, hỗ trợ thiết bị đầu cuối.
* Màn hình đa nhiệm, dễ thao tác chuyển đổi.

**Một số tính năng nổi bật của Visual Studio Code**

Extension được Microsoft phát triển, có hỗ trợ gợi nhớ code (suggestion), tự động hoàn thành code, chạy ứng dụng Python và debug.

*Lệnh cài đặt nhanh*: ext install ms-python.python

[**HTML CSS Support**](https://marketplace.visualstudio.com/items?itemName=ecmel.vscode-html-css)

Extension giúp viết mã HTML/CSS dành cho các lập trình viên sử dụng ngôn ngữ Python để phát triển website

*Lệnh cài đặt nhanh*: ext install ecmel.vscode-html-css

[**Remote Development**](https://vinasupport.com/huong-dan-code-remote-su-dung-visual-studio-code/)

Extension hỗ trợ code, debug, chạy chương trình Python từ xa

*Lệnh cài đặt nhanh*: ext install ms-vscode-remote.vscode-remote-extensionpack

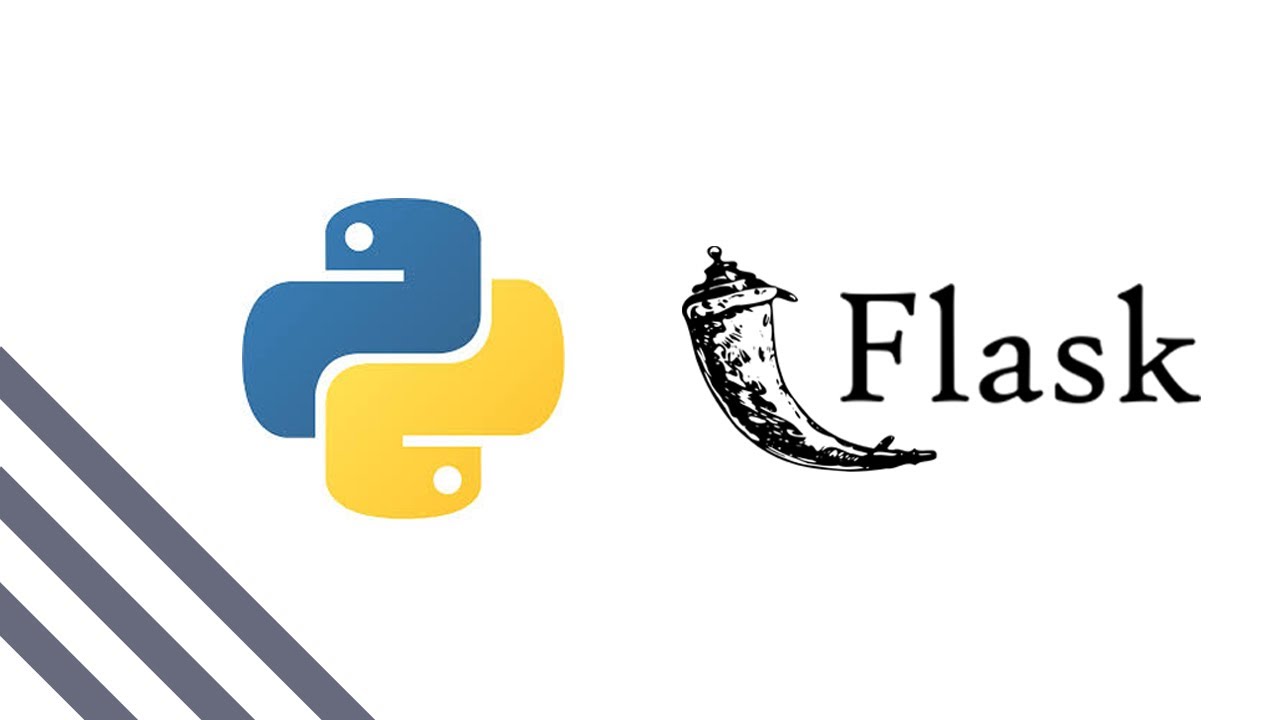


Hình 2. 4 Minh họa tính năng của Visual Studio Code

**2.5. Framework Flask trong Python**

1.5.1. Framework Flask

Flask là một web frameworks, nó thuộc loại micro-framework được xây dựng bằng ngôn ngữ lập trình Python. Flask cho phép bạn xây dựng các ứng dụng web từ đơn giản tới phức tạp. Nó có thể xây dựng các api nhỏ, ứng dụng web chẳng hạn như các trang web, blog, trang wiki hoặc một website dựa theo thời gian hay thậm chí là một trang web thương mại. Flask cung cấp cho bạn công cụ, các thư viện và các công nghệ hỗ trợ bạn làm những công việc trên.



Hình 2. 5 Flask trong Python

Flask là một micro-framework. Điều này có nghĩa Flask là một môi trường độc lập, ít sử dụng các thư viện khác bên ngoài. Do vậy, Flask có ưu điểm là nhẹ, có rất ít lỗi do ít bị phụ thuộc cũng như dễ dàng phát hiện và xử lý các lỗi bảo mật.

1.5.2. Flask trong Python

Lập trình viên rất dễ dàng để bắt đầu với Flask, vì framework này không có quá nhiều khối lượng kiến thức để học. Để tạo một ứng dụng “Hello World” với Flask, bạn chỉ cần một vài dòng code.

*Ví dụ như sau:*

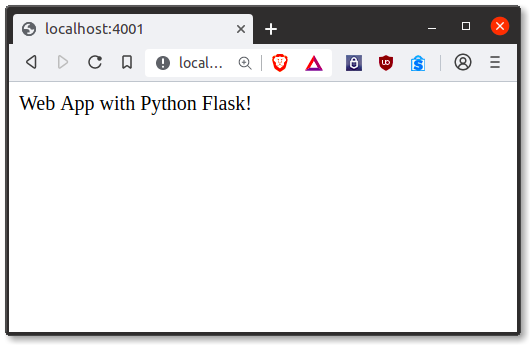
from flask import Flask  
app = Flask(\_\_name\_\_)  
  
@app.route('/')  
def hello\_world():  
 return 'Hello World!'  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 app.run()

Nếu bạn muốn phát triển ứng dụng từ local computer, bạn cũng có thể thực hiện một cách dễ dàng. Lưu chương trình này thành server.py và chạy nó cùng với python server.py

|  |
| --- |
| $ python server.py  \* Serving Flask app "hello"  \* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit) |

Sau đó nó sẽ khởi động một máy chủ web chỉ khả dụng trên máy tính của bạn. Trong trình duyệt web, mở localhost trên cổng 5000(urk) và bạn sẽ thấy “Hello World” được hiển thị.

Để lưu và tiếp tục phát triển chương trình hay ứng dụng của mình một cách trực tuyến, bạn có thể sử dụng PythonAnyWhere



Hình 2. 6 Ví dụ về Output của Flask

Mặc dù Flask là Microframework Python nhưng điều này không có nghĩa là toàn bộ ứng dụng của bạn phải nằm trong một tệp Python duy nhất. Bạn có thể và nên sử dụng nhiều tệp cho các chương trình lớn hơn để dễ dàng xử lý độ phức tạp. Thêm vào đó, lập trình viên cũng có thể mở rộng Framework Flask, lựa chọn việc sử dụng cơ sở dữ liệu, sử dụng ORM và nhiều lựa chọn khác.

## **1.6. Giới thiệu về Rational Rose**

### 1.6.1. Giới thiệu

Rational Rose là phần mềm công cụ mạnh hỗ trợ phân tích hệ thống phần mềm theo đối tượng. Nó giúp ta mô hình hóa hệ thống trước khi viết mã trình.

Rational Rose hỗ trợ cho việc làm mô hình doanh nghiệp, giúp người dùng hiểu được hệ thống của mô hình doanh nghiệp, giúp chúng t phân tích hệ thống và làm cho chúng ta có thể thiết kế được mô hình.

Mô hình Rose là bức tranh của một hệ thống từ những phối cảnh khác nhau nó bao gồm tất cả các mô hình UML, actors, use cases, objects, component và deployment nodes trong hệ thống. Nó mô tả chi tiết mà hệ thống bao gồm và nó sẽ làm việc như thế nào vì thế người lập trình có thể dùng mô hình như một bản thiết kế cho việc xây dựng hệ thống.

### 1.6.2. Tính năng

Rational Rose cung cấp những tính năng sau đây để tạo điều kiện thuận lợi cho việc phân tích thiết kế và xây dựng ứng dụng:

* Mô hình hướng đối tượng.
* Mô hình cung cấp cho UML, COM, OMT và Booch ’93.
* Kiểm tra ngữ nghĩa
* Hỗ trợ phát sinh mã cho một số ngôn ngữ.
* Hỗ trợ việc phát triển cho việc kiểm soát lặp đi lặp lại.
* Phát triển cho nhiều người dùng và cung cấp cho cá nhân.
* Hợp nhất những công cụ làm mô hình dữ liệu.
* Phát sinh tài liệu.
* Nhiều nền tảng có sẵn.

# **CHƯƠNG 2. PHÂN TÍCH THIẾT KẾ HỆ THỐNG**

## **2.1. Hoạt động của hệ thống**

* + 1. Miêu tả - Description

Một trang báo tiện tử cơ bản mà mọi người có thể truy cập hằng ngày để đọc những tin tức mới. Ở đây, các bài báo được bố trí một cách rõ ràng với Ảnh, tiêu đề, chuyên mục và thời gian đăng bài. Bạn đọc có thể xem các bài báo ưa thích theo chuyên mục hoặc tìm kiếm đầu báo cụ thể trên thanh tìm kiếm. Hệ thống cho phép nhiều biên tập và biên soạn có thể làm việc và tạo ra nhiều bài báo mới tức thì trong đó biên soan viên sẽ là người tự tạo ra một bài báo hoặc lấy từ nguồn có sẵn và lưu thành bản thảo và sau đó biên tập viên sẽ kiểm duyệt bài báo đó và xuất bản nó đến với bạn đọc đảm bảo tính chuyên nghiệp trong việc đăng bài báo. Quản trị viên sẽ là người nắm quyền cáo nhất với khả năng quản lý tài khoản và các chuyên mục.

* + 1. Mục tiêu - Objectives

Bạn đọc có thể đọc nhiều bài báo mới hằng ngày

Các trang báo được hiển thị một cách rõ ràng và dễ sử dụng

Dễ dàng thao tác với giao diện

Cho phép nhiều biên soạn, biên tập có thể hoạt động để tạo các bài báo

Các bài báo được kiểm duyệt và có người chịu trách nghiệm trước khi công bố đến bạn đọc.

Dễ dàng quản lý, bảo trì và nâng cấp

* + 1. Tính năng - Features

Bạn đọc:

* Xem tin tức theo danh mục
* Tìm kiếm tin tức
* Xem nội dung tin tức

Biên soạn viên:

* Tạo bài báo bản thảo

Biên tập viên:

* Tạo bài báo bản thảo
* Chỉnh sửa bài báo
* Xóa bài báo
* Xuất bản bài báo

Quản trị viên:

* Quản lý người dùng
* Quản lý chuyên mục
  + 1. Cơ sở dữ liệu - Database

Users(id\_user, username, password, name, postion)

Sections(id\_section, name)

Articles(id\_article, headline, byline, uploaded\_by, section, body, last\_edited)

## 2.2. **Các yêu cầu phi chức năng**

Bảng 2.1: Các yêu cầu phi chức năng

|  |  |  |
| --- | --- | --- |
| Mục | Tên yêu cầu | Mô tả yêu cầu |
| 1 | Giao diện | Giao diện hệ thống phải dễ sử dụng, trực quan, thân thiện với mọi người dùng. |
| 2 | Tốc độ xử lý | Hệ thống phải xử lý nhanh chóng và chính xác |
| 3 | Tương thích | Tương thích với đa phần các trình duyệt web hiện tại |

## **Mô hình hóa Usecase**

* + 1. Xác định các Usecase

|  |  |  |
| --- | --- | --- |
| **Actor** | **Use case** | **Mô tả** |
| Quản trị viên | Đăng nhập | Cho phép admin đăng nhập vào hệ thống để thực hiện chức năng của admin. |
| Bảo trì người dùng | Cho phép admin thêm, sửa, xóa, disable Biên soạn viên và Biên tập viên |
| Bảo trì chuyên mục | Cho phép admin thêm, sửa, xóa các chuyên mục |
| Đăng xuất | Cho phép admin đăng xuất ra khỏi hệ thống của admin. |
| Biên tập viên | Đăng nhập | Cho phép biên tập viên đăng nhập vào hệ thống để thực hiện chức năng của Biên tập viên |
| Đăng xuất | Cho phép biên tập viên đăng xuất ra khỏi hệ thống của Biên tập viên |
| Xem báo theo chuyên mục | Cho phép biên tập viên xem báo theo các chuyên mục |
| Xem báo theo trạng thái | Cho phép biên tập viên xem báo theo trạng thái (xuất bản/ bản thảo) |
| Xem chi tiết thông tin báo | Cho phép biên tập viên xem chi tiết thông tin bài báo (Tiêu đề, tác giả, ảnh bìa,…) |
| Tìm kiếm báo | Cho phép biên tập viên tìm kiếm báo |
| Tạo bài báo bản thảo | Cho phép biên tập viên tạo mới bản thảo |
| Xem trang cá nhân | Cho phép biên tập viên xem trang cá nhân của họ |
| Đổi mật khẩu | Cho phép biên tập viên đổi mật khẩu tài khoản của họ |
| Xóa bài báo | Cho phép biên tập viên xóa bài báo (ở cả trạng thái Xuất bản hoặc Bản thảo) |
| Xuất bản bài báo | Cho phép biên tập viên Xuất bản bài báo khi bài báo đang ở trạng thái Bản thảo |
| Chỉnh sửa bài báo | Cho phép biên tập viên chỉnh sửa bài báo (ở cả trạng thái Xuất bản hoặc Bản thảo) |
| Biên soạn viên | Đăng nhập | Cho phép biên soạn viên đăng nhập vào hệ thống để thực hiện chức năng của Biên soạn viên |
| Đăng xuất | Cho phép biên soạn viên đăng xuất ra khỏi hệ thống của Biên soạn viên |
| Xóa bài báo bản thảo của họ | Cho phép biên soạn viên xóa bài báo của chính họ tạo |
| Chỉnh sửa bài bảo bản thảo của họ | Cho phép biên soạn viên chỉnh sửa bài báo của chính họ tạo |
| Xem báo theo chuyên mục | Cho phép biên soạn viên xem báo theo các chuyên mục |
| Xem báo theo trạng thái | Cho phép biên soạn viên xem báo theo trạng thái (xuất bản/ bản thảo) |
| Xem chi tiết thông tin báo | Cho phép biên soạn viên xem chi tiết thông tin bài báo (Tiêu đề, tác giả, ảnh bìa,…) |
| Tìm kiếm báo | Cho phép biên soạn viên tìm kiếm báo |
| Tạo bài báo bản thảo | Cho phép biên soạn viên tạo mới bản thảo |
| Xem trang cá nhân | Cho phép biên soạn viên xem trang cá nhân của họ |
| Đổi mật khẩu | Cho phép biên soạn viên đổi mật khẩu tài khoản của họ |
| Độc giả | Xem các bài báo theo chuyên mục | Cho phép độc giả xem các bài báo theo chuyên mục |
| Tìm kiếm bài báo | Cho phép độc giả tìm kiếm bài báo |
| Xem chi tiết bài bao | Cho phép độc giả xem chi tiết bài báo |

### Biểu đồ Usecase

Biểu đồ Usecase tổng quát



Biểu đồ Usecase của độc giả (người xem)



Biểu đồ Usecase của Biên soạn viên



Biểu đồ Usecase của Biên tập viên

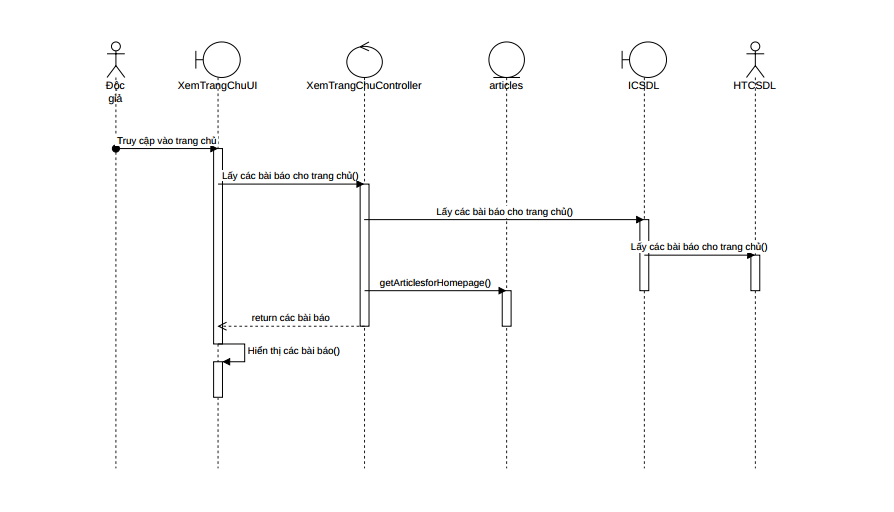


Biểu đồ Usecase của Quản trị viên

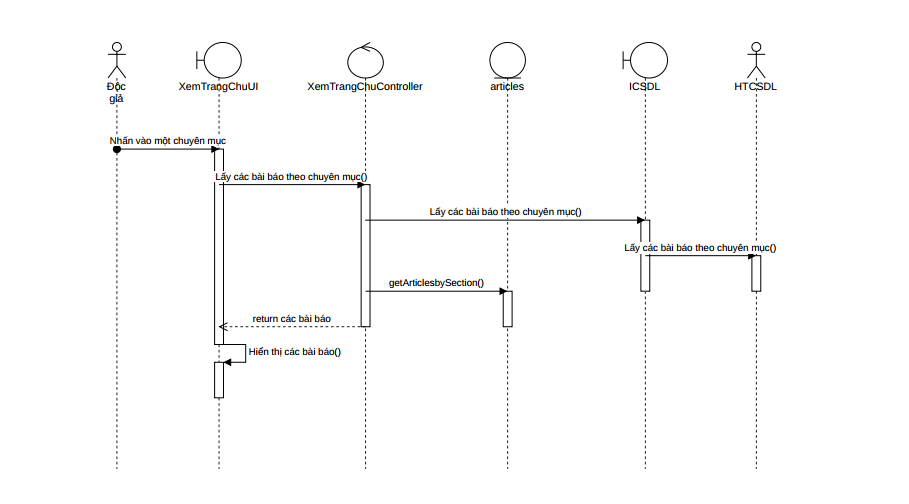


### 2.2.3. Biểu đồ trình tự

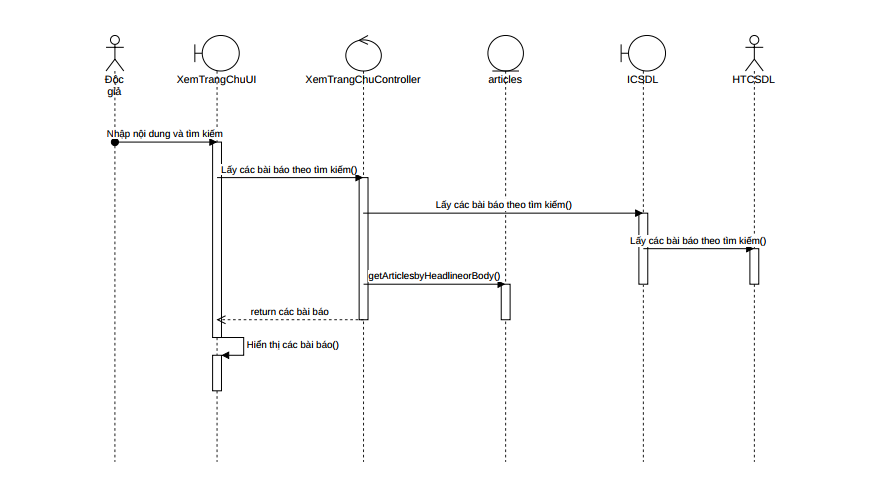
### **Biểu đồ trình tự của “Độc giả”**



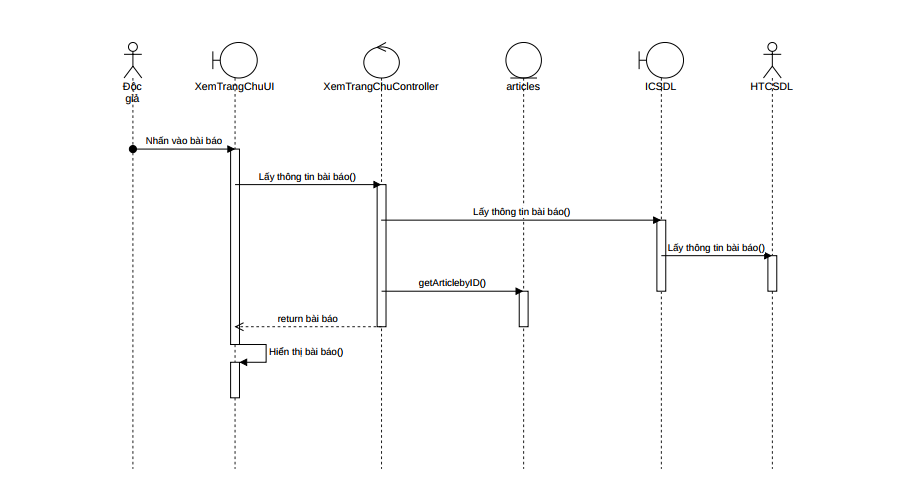
Biểu đồ trình tự Độc gỉa truy cập trang chủ



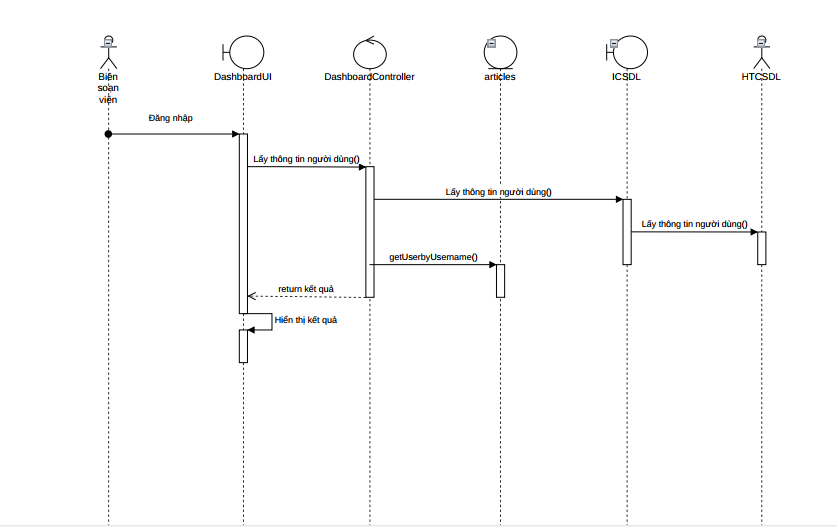
Biểu đồ trình tự Độc giả truy cập chuyên mục báo



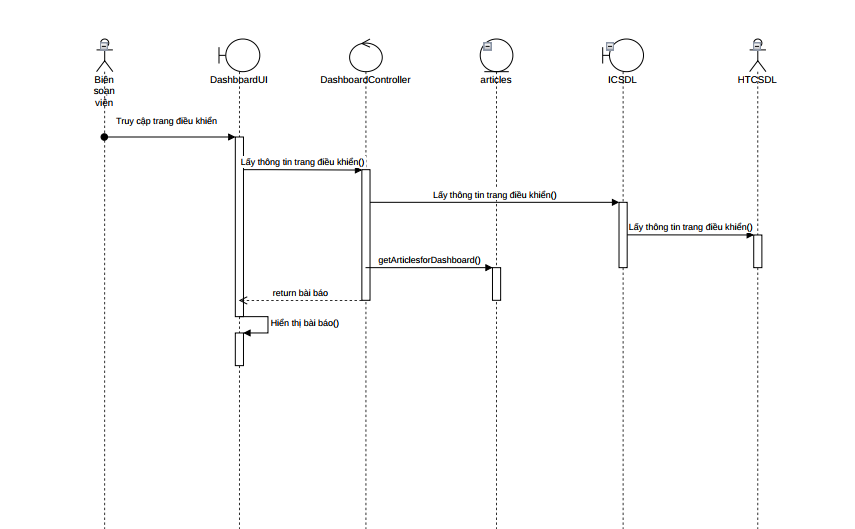
Biểu đồ trình tự Độc giả tìm kiếm

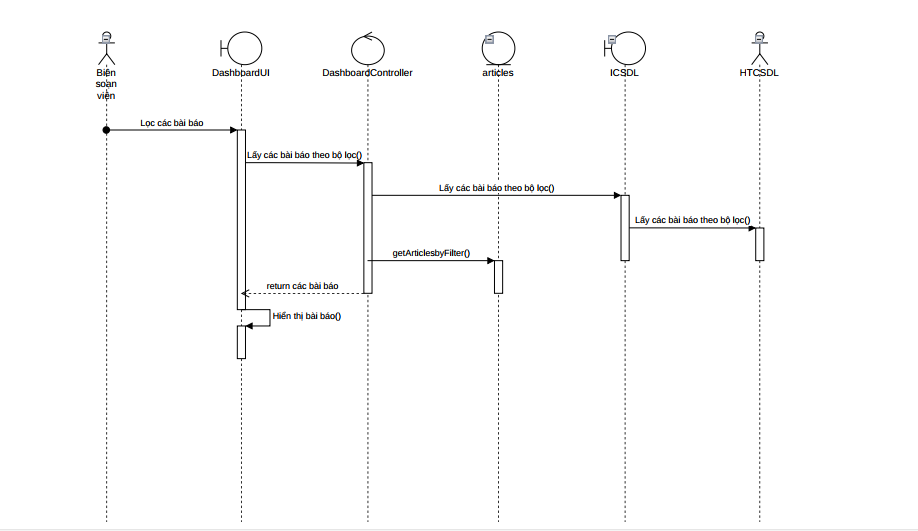
Biểu đồ trình tự Độc giả xem bài báo

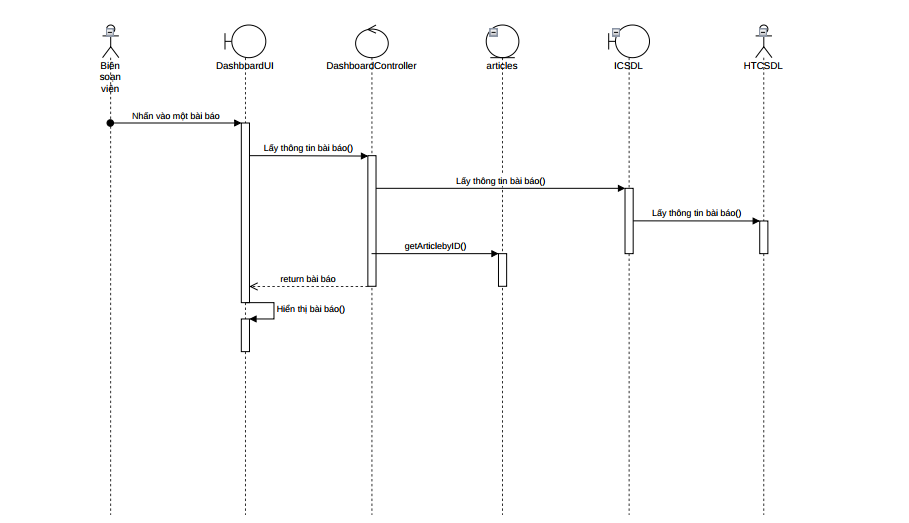
### **Biểu đồ trình tự của “Biên soạn viên”**



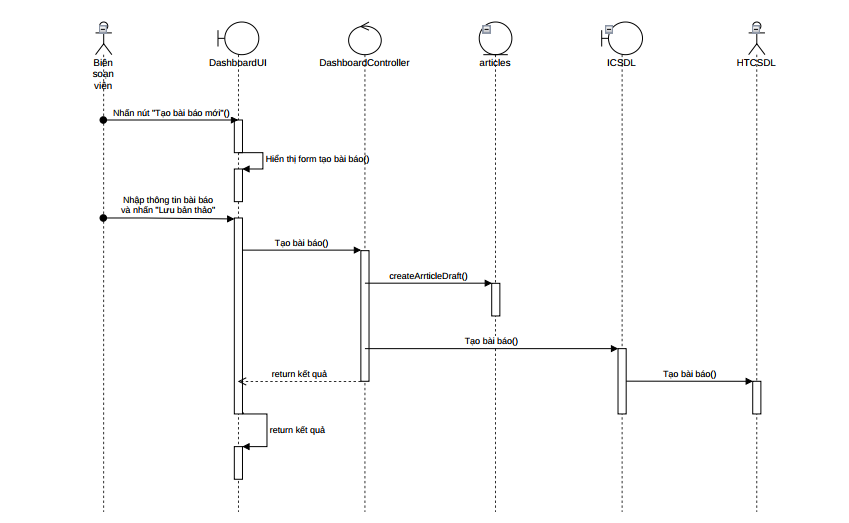
Biểu đồ trình tự Biên soạn viên đăng nhập vào hệ thống

Biểu đồ trình tự Biên soạn viên truy cập trang điều khiển

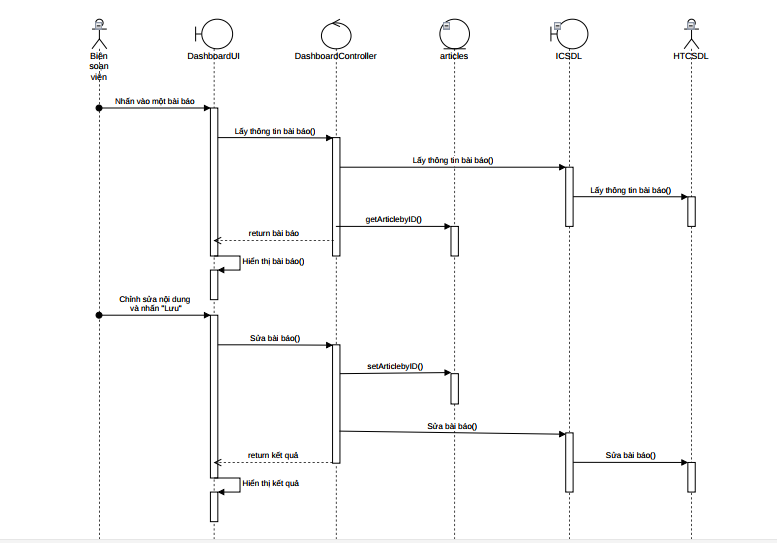
Biểu đồ trình tự Biên soạn viên lọc các bài báo



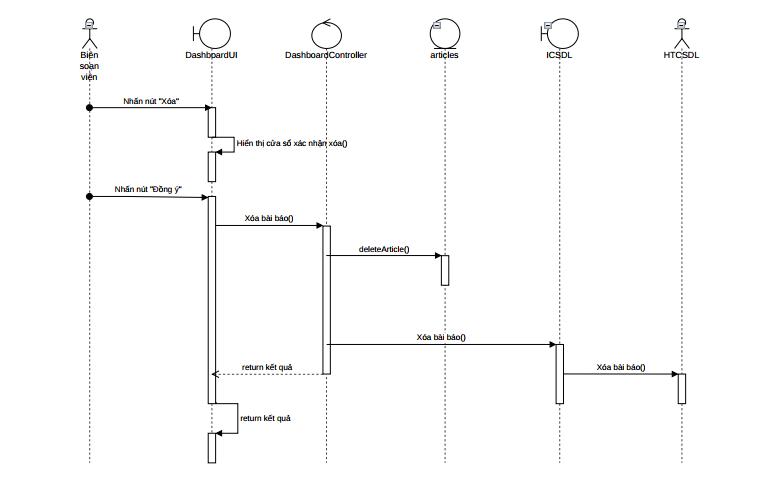
Biểu đồ trình tự Biên soạn viên xem một bài báo



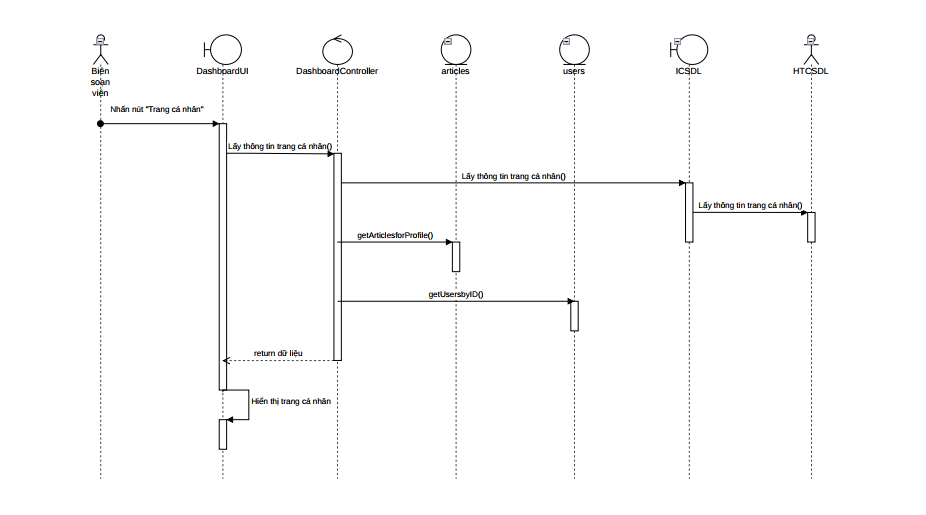
Biểu đồ trình tự Biên soạn viên tạo mới bài báo



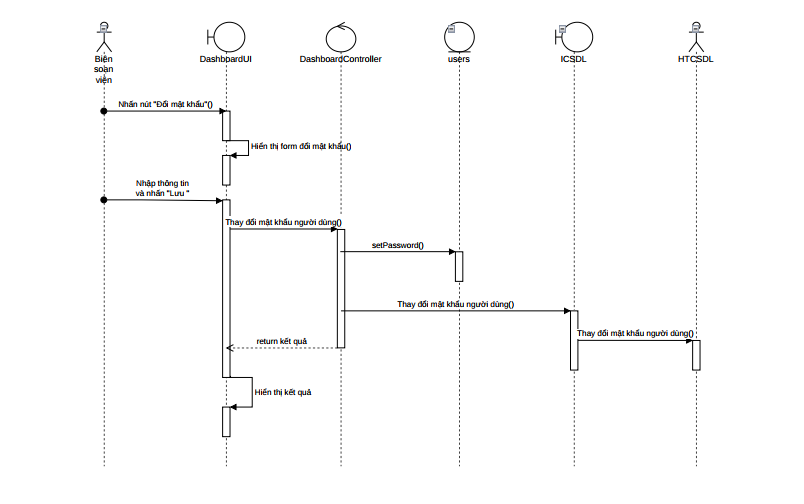
Biểu đồ trình tự Biên soạn viên chỉnh sửa một bài báo



Biểu đồ trình tự Biên soạn viên xóa một bài báo

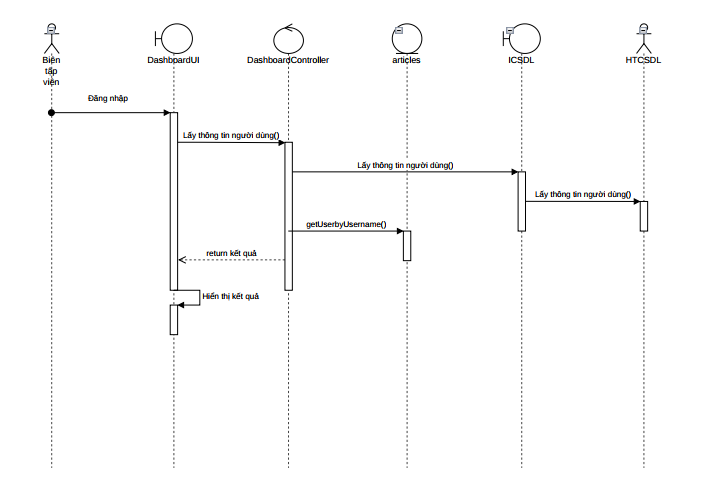


Biểu đồ trình tự Biên soạn viên xem thông tin cá nhân

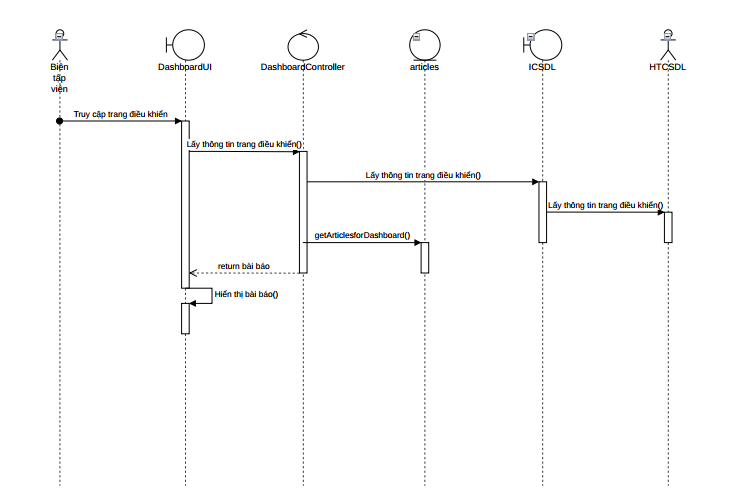


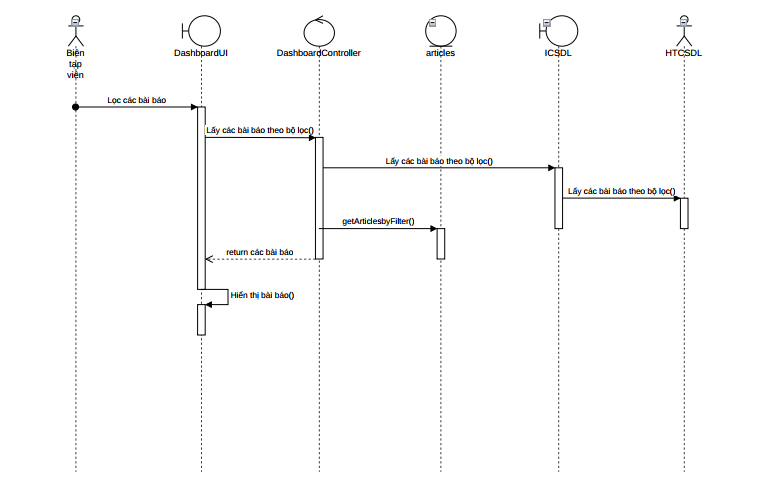
Biểu đồ trình tự Biên soạn viên đổi mật khẩu

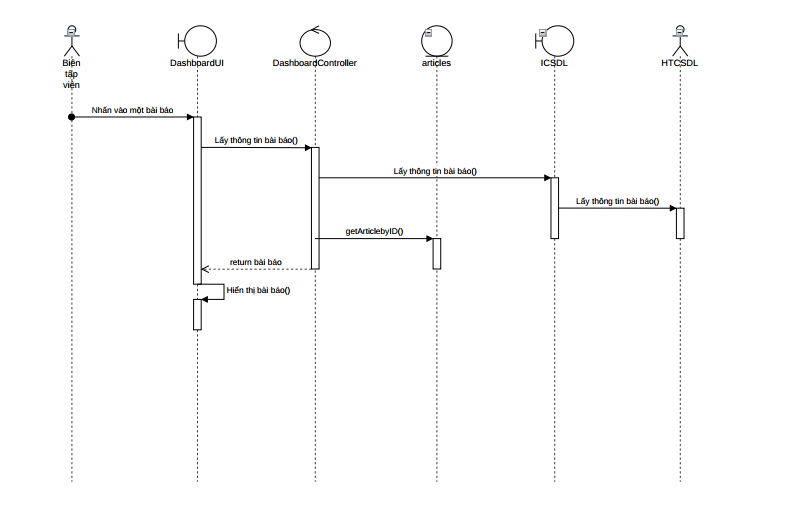
### **Biểu đồ trình tự của “Biên tập viên”**

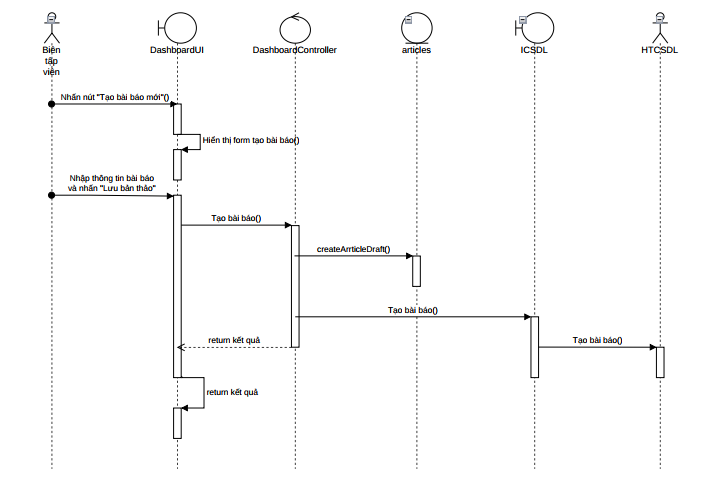


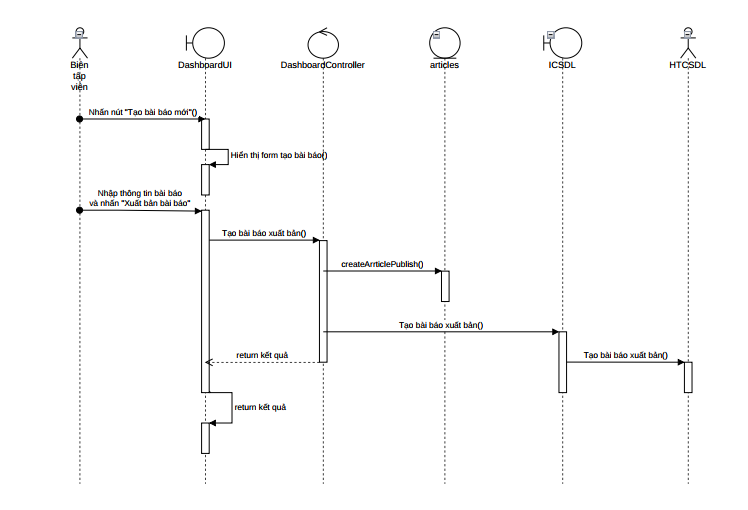
Biểu đồ trình tự Biên tập viên đăng nhập vào hệ thống

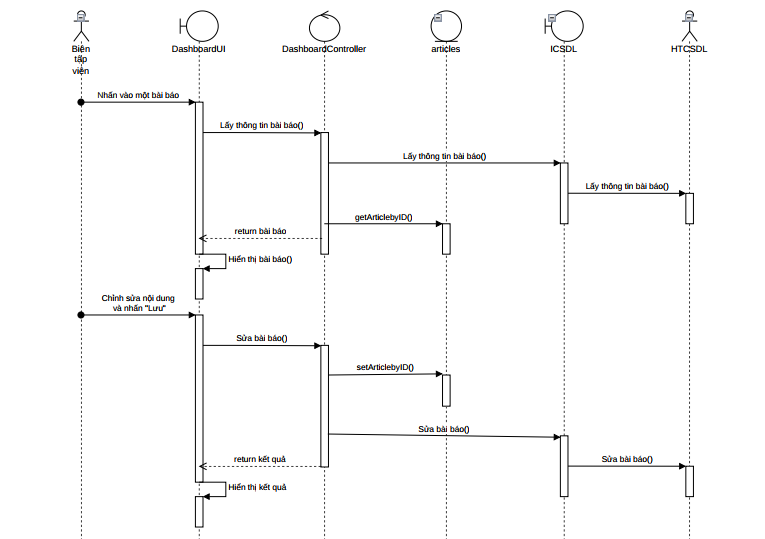
Biểu đồ trình tự Biên tập viên truy cập trang điều khiển

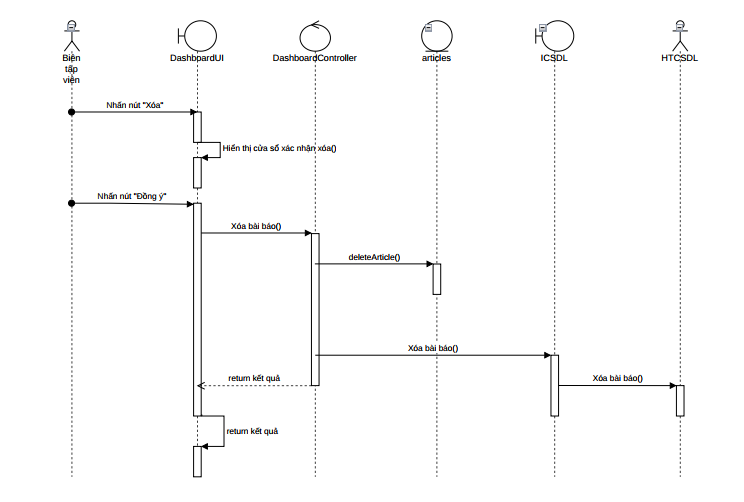
Biểu đồ trình tự Biên tập viên lọc các bài báo

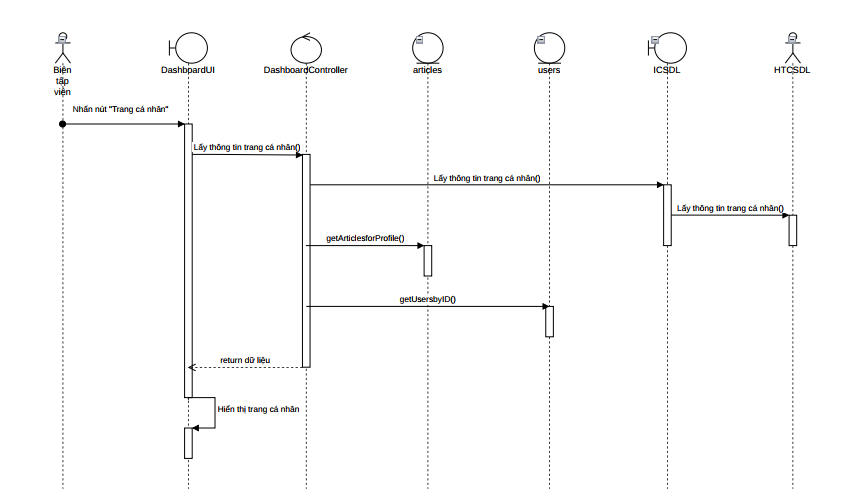
Biểu đồ trình tự Biên tập viên xem chi tiết bài báo

Biểu đồ trình tự Biên tập viên tạo mới một bài báo – bản thảo

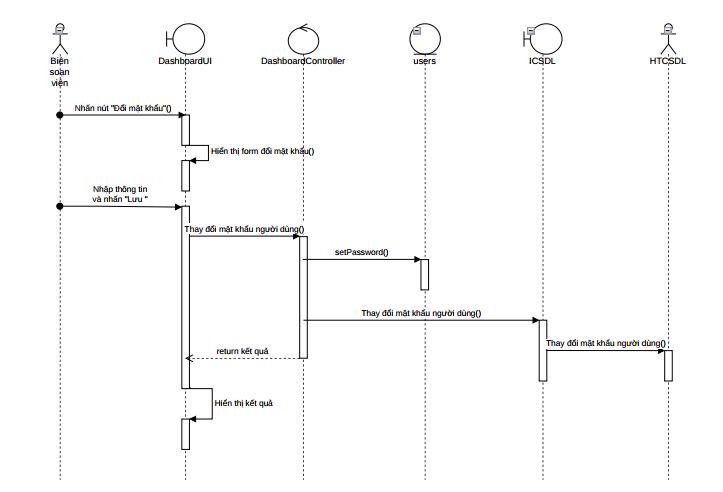
Biểu đồ trình tự Biên tập viên xuất bản một bài báo

Biểu đồ trình tự Biên tập viên chỉnh sửa một bài báo

Biểu đồ trình tự Biên tập viên xóa một bài báo

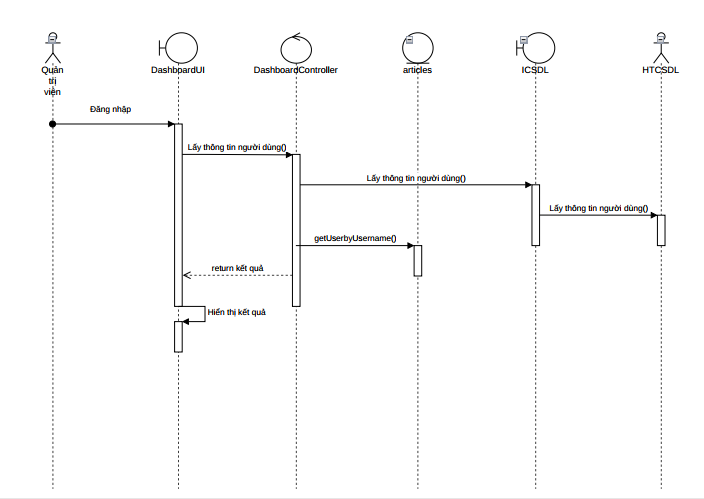


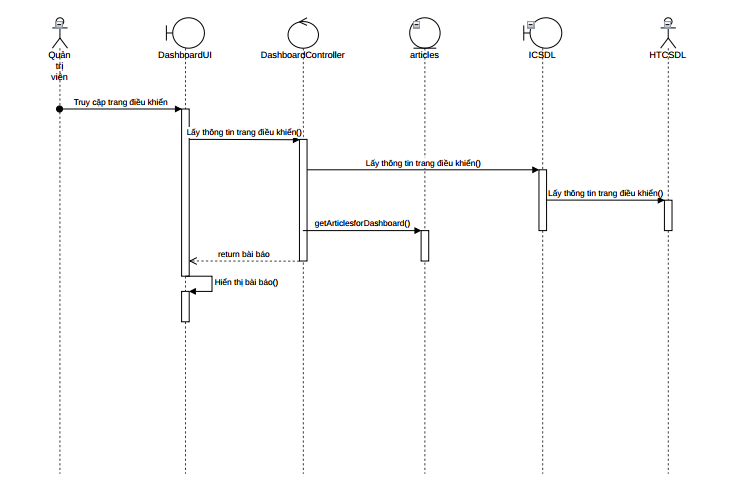
Biểu đồ trình tự Biên tập viên xem trang cá nhân



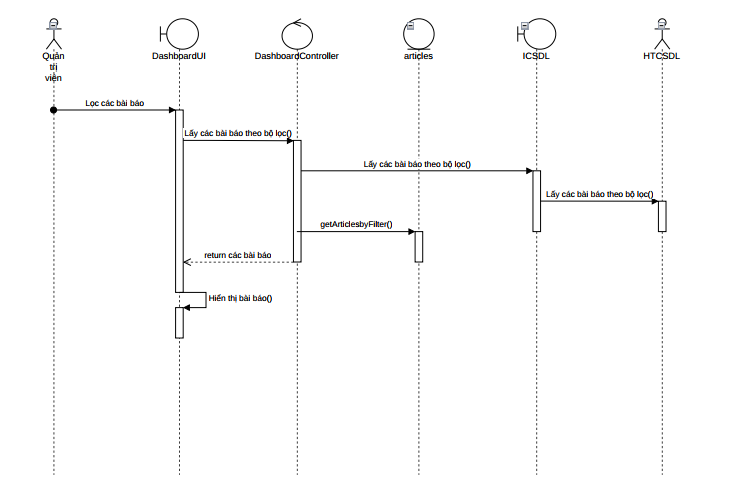
Biểu đồ trình tự Biên tập viên đổi mật khẩu

### **Biểu đồ trình tự của “Quản trị viên”**

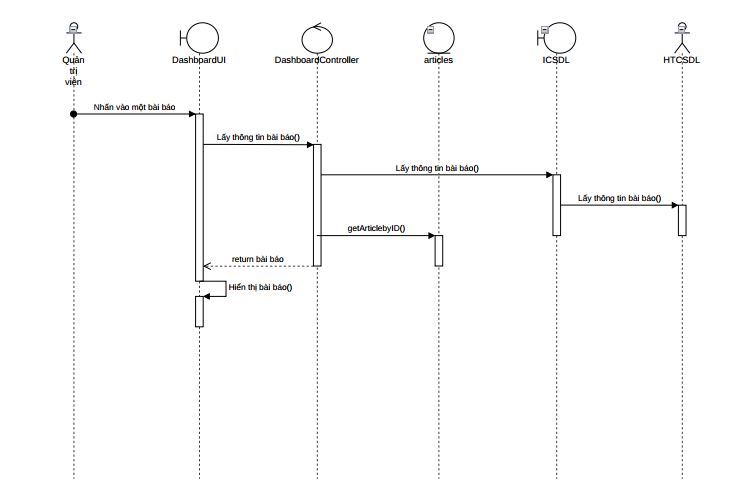
Biểu đồ trình tự Quản trị viên đăng nhập



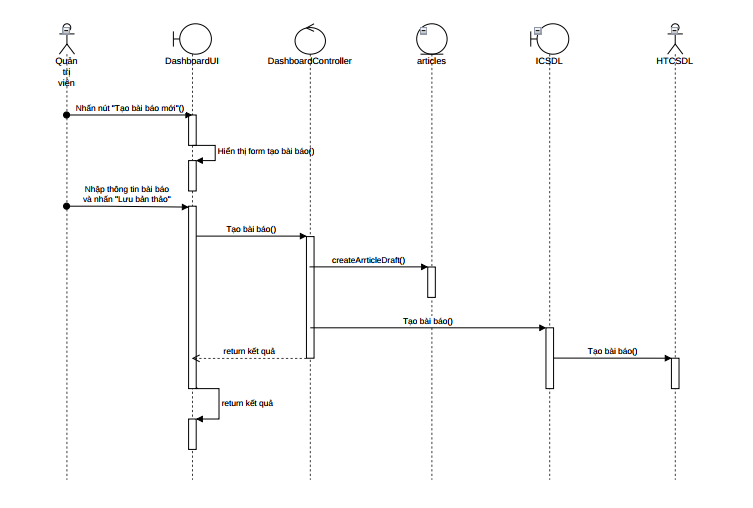
Biểu đồ trình tự Quản trị viên truy cập trang điều khiển



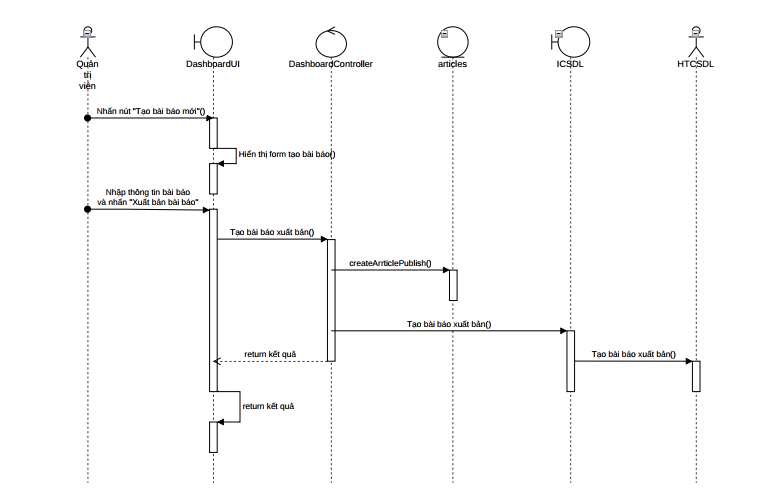
Biểu đồ trình tự Quản trị viên lọc các bài báo



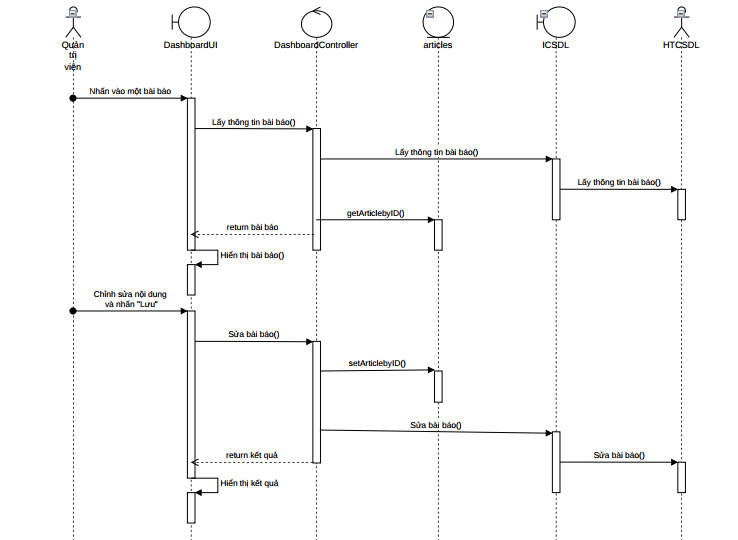
Biểu đồ trình tự Quản trị viên xem chi tiết bài báo



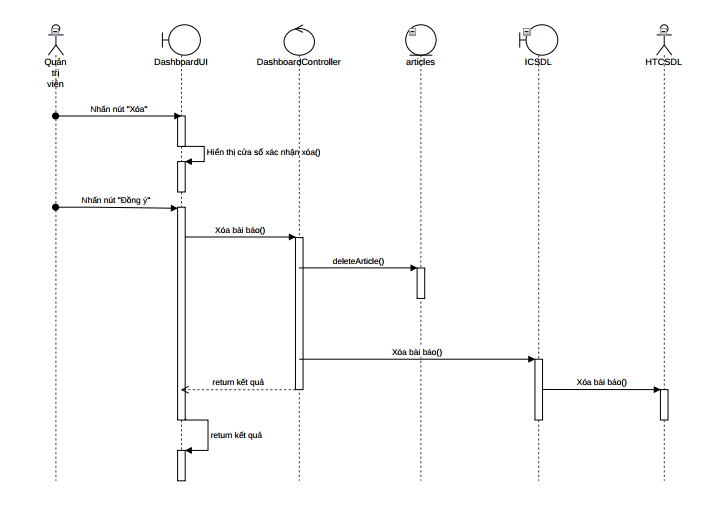
Biểu đồ trình tự Quản trị viên tạo mới một bài báo – bản thảo



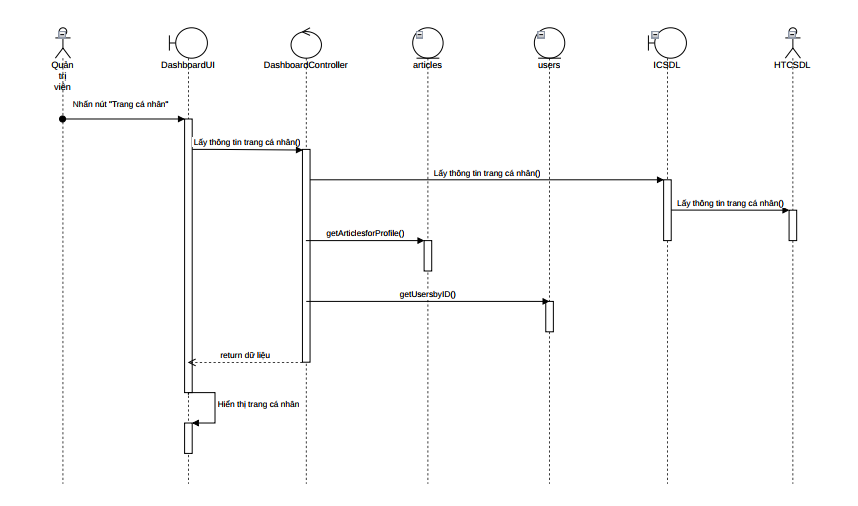
Biểu đồ trình tự Quản trị viên tạo mới một bài báo – xuất bản



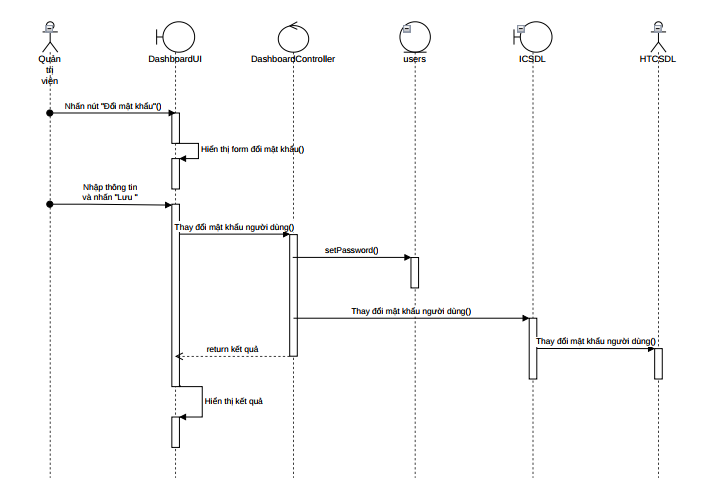
Biểu đồ trình tự Quản trị viên chỉnh sửa bài báo



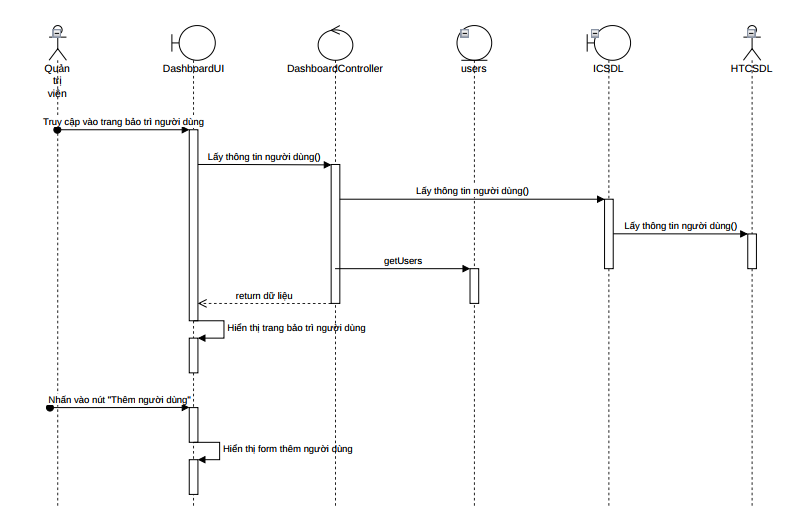
Biểu đồ trình tự quản trị viên xóa bài báo

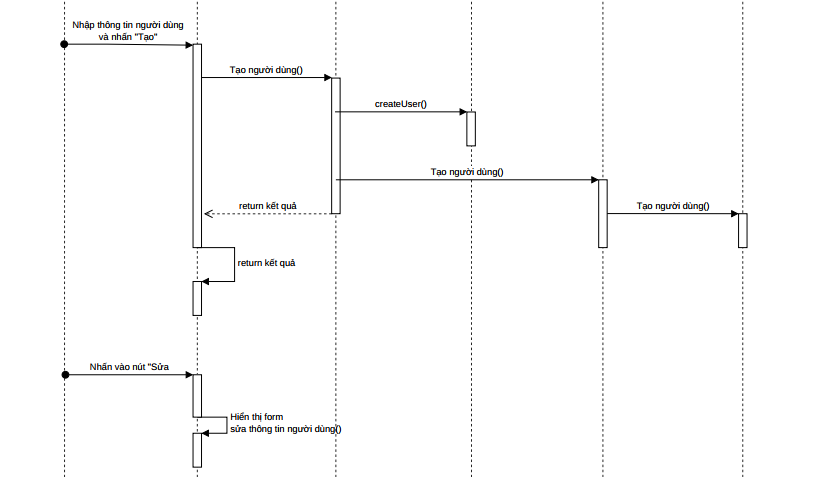


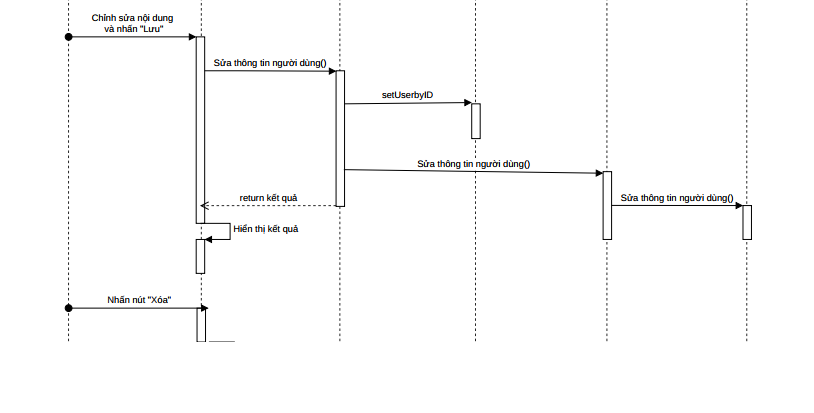
Biểu đồ trình tự Quản trị viên xem trang cá nhân



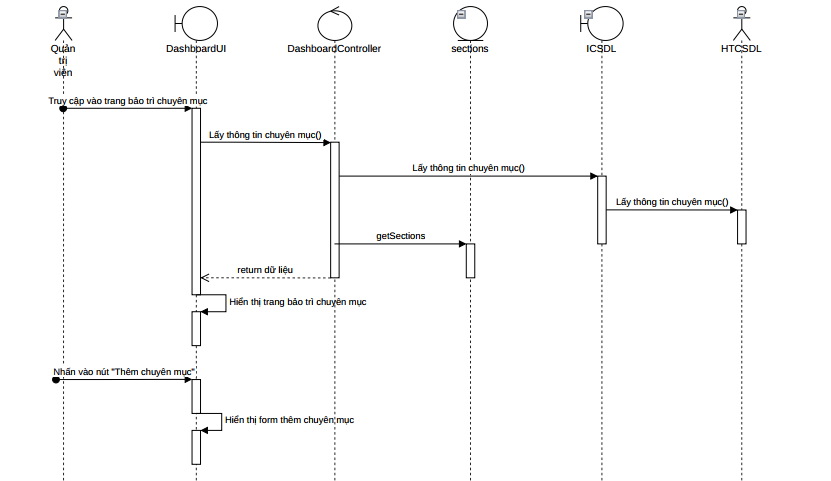
Biểu đồ trình tự Quản trị viên đổi mật khẩu

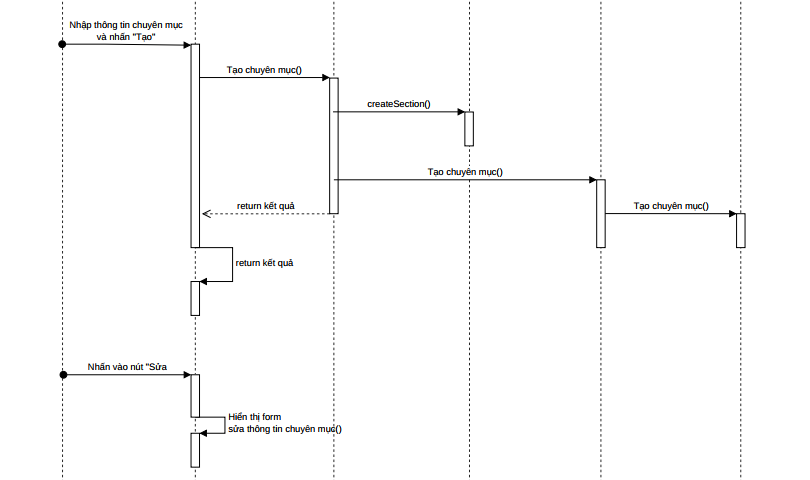


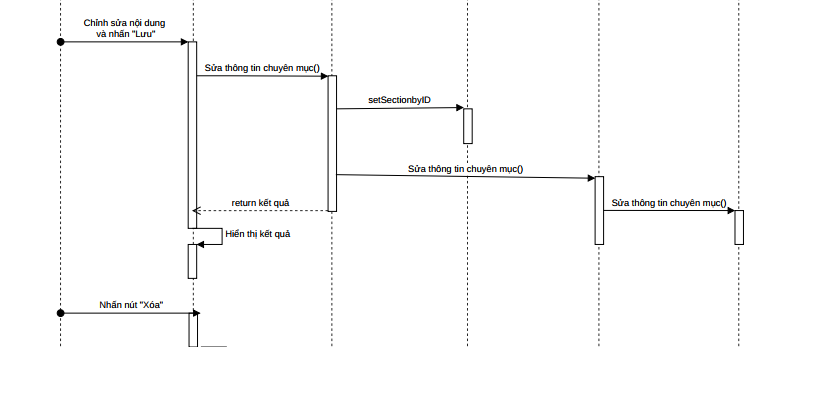




Biểu đồ trình tự Quản trị viên truy cập trang bảo trì người dùng



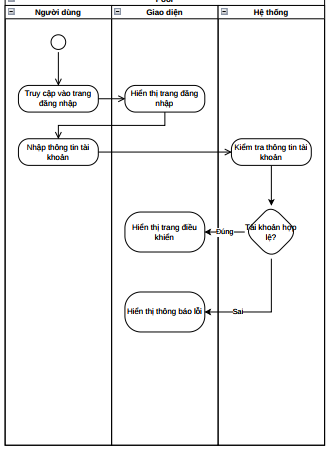




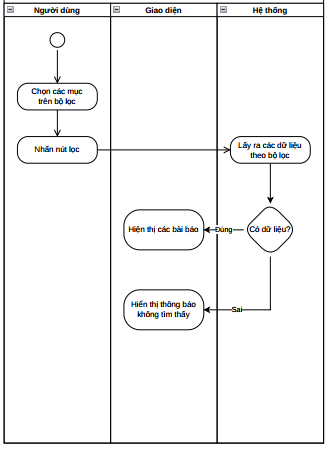
Biểu đồ trình tự Quản trị viên truy cập vào trang bảo trì chuyên mục

### 2.2.4. Biểu đồ hoạt động

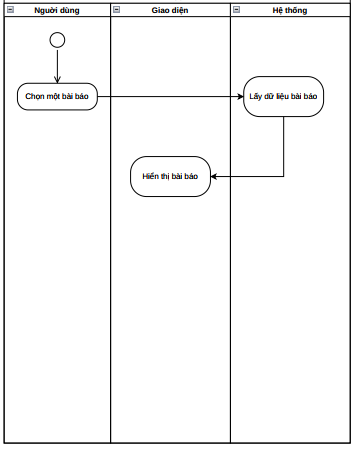
### Biểu đồ hoạt động “Đăng nhập”



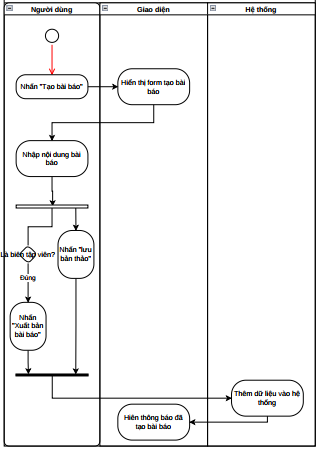
### Biểu đồ hoạt động Lọc bài báo



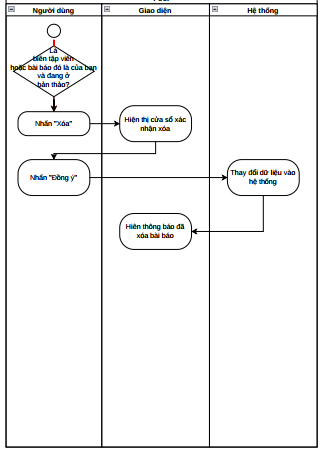
### Biểu đồ hoạt động Xem bài báo



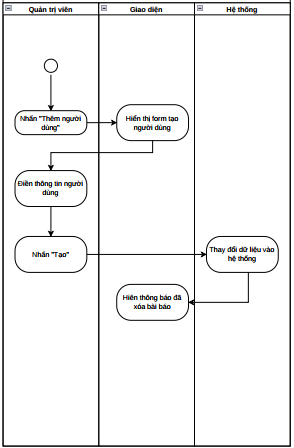
### Biểu đồ hoạt động Thêm bài báo



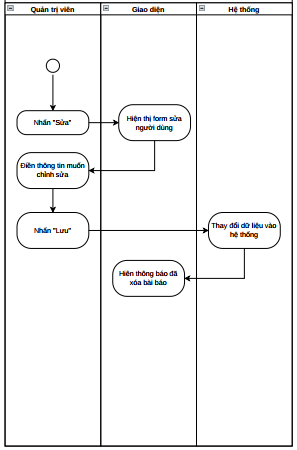
### Biểu đồ hoạt động Xóa bài báo



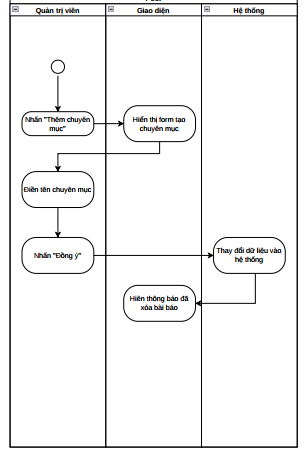
### Biểu đồ hoạt động Thêm người dung



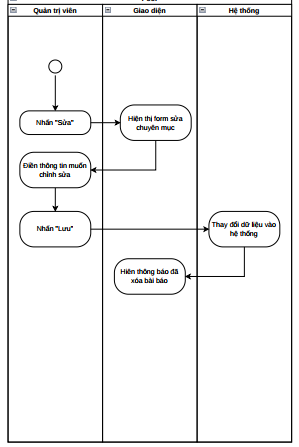
### Biểu đồ hoạt động Sửa người dùng



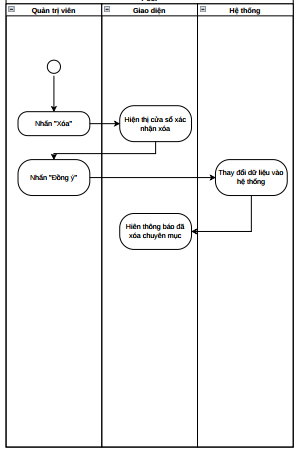
### Biểu đồ hoạt động Thêm chuyên mục



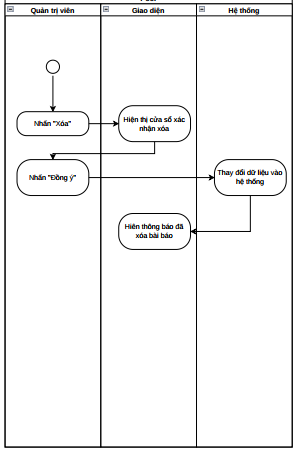
### Biểu đồ hoạt động Sửa chuyên mục



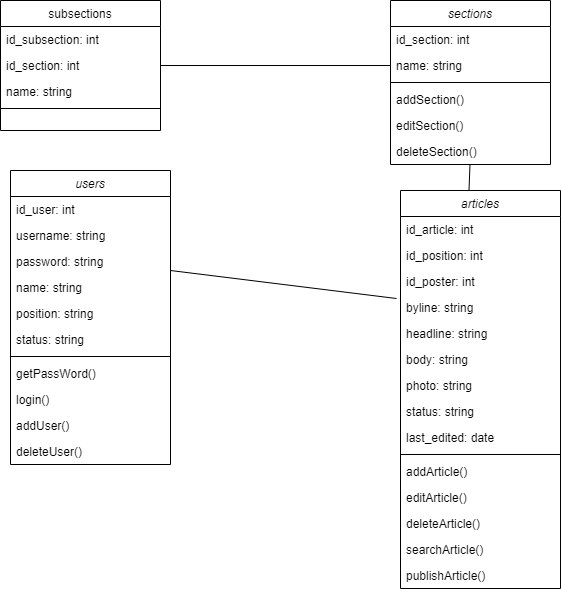
### Biểu đồ hoạt động Xóa chuyên mục



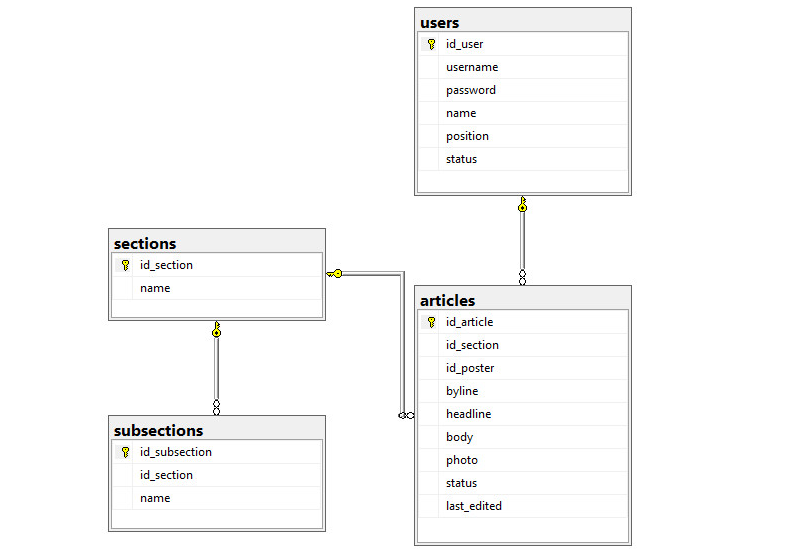
### Biểu đồ hoạt động Xóa người dùng



### 3.2.5. Biểu đồ lớp



Biểu đồ lớp hệ thống



Diagram

## **2.3. Thiết kế cơ sở dữ liệu**

### Bảng Sections

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên trường | Kiểu dữ liệu | Kích thước | Ràng buộc | Mô tả |
| id\_section | int |  | PRIMARY KEY  AUTO\_INCREMENT | Khóa chính của bảng |
| name | varchar | 255 | NOT NULL | Tên chuyên mục |

### Bảng Subsections

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên trường | Kiểu dữ liệu | Kích thước | Ràng buộc | Ghi chú |
| id\_subsection | int |  | PRIMARY KEY  AUTO\_INCREMENT | Khóa chính của bảng |
| id\_section | int |  | FOREIGN KEY | Khóa ngoài của bảng liên kết với bảng Sections |
| name | varchar | 255 | NOT NULL | Tên tiểu mục |

### Bảng Users

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên trường | Kiểu dữ liệu | Kích thước | Ràng buộc | Ghi chú |
| id\_user | int |  | PRIMARY KEY  AUTO\_INCREMENT | Khóa chính của bảng |
| username | varchar | 255 | NOT NULL | Tên tài khoản người dùng |
| password | varchar | 255 | NOT NULL | Mật khẩu của tài khoản |
| name | varchar | 255 | NOT NULL | Tên người dùng |
| position | varchar | 255 | NOT NULL | Vị trí người dùng |
| status | varchar | 255 | NOT NULL | Trạng thái người dùng |

### Bảng Articles

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên trường | Kiểu dữ liệu | Kích thước | Ràng buộc | Ghi chú |
| id\_article | int |  | PRIMARY KEY  AUTO\_INCREMENT | Khóa chính của bảng |
| id\_section | int |  | FOREIGN KEY | Khóa ngoài thứ 1 của bảng liên kết với bảng Sections |
| id\_poster | int |  | FOREIGN KEY | Khóa ngoài thứ 2 của bảng liên kết với bảng Users |
| byline | varchar | 255 | NOT NULL | Người đăng bài báo |
| headline | longtext |  |  | Tiêu đề bài báo |
| body | longtext |  | NOT NULL | Nội dung bài báo |
| photo | varchar | 255 |  | Hình ảnh trang bìa bài báo |
| status | varchar | 255 | NOT NULL | Trạng thái của bài báo |
| last\_edited | timestamp |  | DEFAULT  CURRENT\_TIMESTAMP | Thời gian chỉnh sửa cuối cùng của bài báo |

# CHƯƠNG 3. KIỂM THỬ VÀ THỰC HIỆN, ĐÁNH GIÁ CHƯƠNG TRÌNH

## 3.1. Kiểm thử phần mềm

### 3.1.1 Kế hoạch kiểm thử

Kiểm thử phần mềm sẽ giúp đảm bảo hoàn thiện chức năng của hệ. Kiểm tra phần mềm cũng để chắc chắn rằng hệ thống đã sẵn sàng cho sử dụng. Ngoài ra, kiểm thử phần mềm xác nhận rằng hệ thống đáp ứng các yêu cầu khác nhau: bao gồm hiệu suất, độ tin cậy, an toàn, khả năng ứng dụng. Việc xác nhận này được thực hiện để đảm bảo rằng hệ thống đang được xây dựng đúng. Tiến hành kiểm tra các chức năng của hệ thống, đảm bảo hệ thống hoạt động đúng như các yêu cầu đặt ra trong mô tả.

Cần kiểm tra các hoạt động chức năng:

Bạn đọc:

* Xem tin tức theo danh mục
* Tìm kiếm tin tức
* Xem nội dung tin tức

Biên soạn viên:

* Tạo bài báo bản thảo
* Chỉnh sửa bản thảo của chính họ
* Xóa bản thảo của chính họ

Biên tập viên:

* Tạo bài báo bản thảo
* Chỉnh sửa bài báo
* Xóa bài báo
* Xuất bản bài báo

Quản trị viên:

* Quản lý người dùng
* Quản lý chuyên mục

Kiểm tra lần lượt qua từng trang của hệ thống

* Test trang chủ
* Test trang chi tiết
* Test trang tìm kiếm

Kiểm tra chức năng của người quản trị

* Biên soạn viên
* Biên tập viên
* Quản trị viên

### 3.1.2. Chiến lược kiểm thử

* Kiểm thử mức hệ thống và kiểm thử chấp nhận
* Kiểm thử giao diện và từng chức năng
* Việc kiểm thử bắt đầu khi đã hoàn thiện bộ test case để kiểm thử giao diện và hệ thống.

### 3.1.3. Thiết kế testcase

### 3.1.4. Tiến hành kiểm thử

* Thực hiện kiểm thử dựa trên các test case đã viết và ghi nhận kết quả.
* Chuẩn bị test data
* Thiết kế và phân loại các trường hợp kiểm thử dựa theo độ ưu tiên của từng trường hợp kiểm thử
* Tự động hoá cho các trường hợp kiểm thử cần thiết
* Chạy các test case theo các bước đã định ra trước đó
* Chạy lại các case failed trước đó để xác nhận là case đó đã được sửa.
* So sánh kết quả thực tế với kết quả mong muốn.
* Đánh giá kết quả kiểm thử cho các trường hợp kiểm thử.

### 3.1.5. Kết quả đạt được

a. Test report

b. Nhận xét đánh giá và đề xuất

* Chương trình được xây dựng vẫn chưa đạt được mức hoàn thiện các chức năng
* Test Case vẫn còn chưa bao phủ hết hệ thống chương trình

## 3.2. Hiện thực chương trình

### 3.2.1. Giao diện dành cho độc giả

### 3.2.2. Giao diện dành cho biên soạn viên, biên tập viên

### 3.2.3. Giao diện dành cho quản trị viên

# **KẾT LUẬN**

# **TÀI LIỆU THAM KHẢO**

### **PHỤ LỤC**