TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

─────── \* ───────

ĐỒ ÁN

**TỐT NGHIỆP ĐẠI HỌC**

NGÀNH CÔNG NGHỆ THÔNG TIN

**WEBSITE HỎI ĐÁP VỀ CÁC VẤN ĐỀ CÔNG NGHỆ THÔNG TIN**

Sinh viên thực hiện: **Vũ Hữu Lượng**

Lớp CN-CNTT02 – K58

Giáo viên hướng dẫn: ThS. **Nguyễn Tiến Thành**

HÀ NỘI 5-2017

# PHIẾU GIAO NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP

1. Thông tin về sinh viên

Họ và tên sinh viên: Vũ Hữu Lượng

Điện thoại liên lạc: 01694029022

Email: luongvu15595@gmail.com

Lớp: CN-CNTT02-k58

Đồ án tốt nghiệp được thực hiện tại: Viện Công nghệ thông tin và Truyền thông – Đại học Bách Khoa Hà Nội.

Thời gian làm ĐATN: Từ ngày 14/02/2017 đến 20/5/2017

1. Mục đích nội dung của ĐATN

Xây dựng một website hỏi và giải đáp các thắc mắc về những khó khăn trong quá trình làm việc, học tập trong ngành CNTT.

1. Các nhiệm vụ cụ thể của ĐATN

Đồ án tốt nghiệp được thực hiện nhằm đạt được những kết quả sau:

Đồ án tốt nghiệp phải phát biểu được bài toán đặt ra và thể hiện được mục đích cụ thể của đồ án là “Xây dựng website hỏi đáp Askme” với đối tượng là những người có đam mê, mong muốn tìm hiểu, học hỏi, mở rộng thêm kiến thức về công nghệ thông tin.

1. Lời cam đoan của sinh viên

Tôi - Vũ Hữu Lượng - cam kết ĐATN là công trình nghiên cứu của bản thân tôi dưới sự hướng dẫn của **ThS. Nguyễn Tiến Thành**. Các kết quả nêu trong ĐATN là trung thực, không phải là sao chép toàn văn của bất kỳ công trình nào khác.

|  |  |
| --- | --- |
|  | Hà Nội, ngày tháng 5 năm 2017  Tác giả ĐATN  Vũ Hữu Lượng |

1. Xác nhận của giáo viên hướng dẫn về mức độ hoàn thành của ĐATN và cho phép bảo vệ

|  |  |
| --- | --- |
|  | Hà Nội, ngày tháng 5 năm 2017  Giáo viên hướng dẫn |

ThS. Nguyễn Tiến Thành

# TÓM TẮT NỘI DUNG ĐỒ ÁN

**Chương 1: Đặt vấn đề và định hướng giải pháp**

Tìm hiểu qua về mục đích của đồ án, giới thiệu tổng quan về đề tài và dự kiến kết quả thu được. Những khái niệm lý thuyết quan trọng tạo điều kiện cho việc phân tích thiết kế hệ thống.

**Chương 2: Phân tích thiết kế hệ thống**

Phần này bao gồm:

* Phân tích ứng dụng, mô tả chức năng của trang web “Askme”.
* Phân tích và thiết kế hệ thống:

+ Xác định các tác nhân của hệ thống.

+ Thiết kế và mô tả Use Case.

+ Thiết kế biểu đồ hoạt động cho từng Use Case.

+ Biểu đồ lớp.

**Chương 3: Kiểm thử hệ thống bằng selenium**

Chúng ta sẽ kiểm thử một số chức năng trong hệ thống.

**Chương 4: Kết quả thực nghiệm**

Các hình giao diện của chương trình.

**Chương 5: Kết luận**

* Đánh giá chung kết quả đạt được.
* Đánh giá ưu điểm, khuyết điểm của hệ thống.
* Định hướng phát triển trong tương lai.

# LỜI CẢM ƠN

Lời đầu tiên, em xin gửi lời cảm ơn sâu sắc đến **ThS. Nguyễn Tiến Thành**, người đã trực tiếp hướng dẫn và tạo mọi điều kiện thuận lợi cho em trong quá trình thực hiện đồ án tốt nghiệp.

Em cũng xin gửi lời cảm ơn tới các thầy cô trong trường đại học Bách Khoa Hà Nội, đặc biệt là các thầy cô thuộc Viện công nghệ thông tin và truyền thông đã truyền đạt cho em những kiến thức và kinh nghiệm làm nền tảng cho việc thực hiện đồ án tốt nghiệp cũng như trong quá trình công tác sau này.

Cuối cùng, em xin cảm ơn đến gia đình, bạn bè đã giúp đỡ, động viên và đóng góp ý kiến để em có thể hoàn thành đồ án tốt nghiệp này.

Em xin chân thành cảm ơn!

# MỤC LỤC

Table of Contents

[PHIẾU GIAO NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP 2](#_Toc483454995)

[TÓM TẮT NỘI DUNG ĐỒ ÁN 3](#_Toc483454996)

[LỜI CẢM ƠN 4](#_Toc483454997)

[MỤC LỤC 5](#_Toc483454998)

[DANH MỤC HÌNH ẢNH 6](#_Toc483454999)

[DANH MỤC BẢNG BIỂU 8](#_Toc483455000)

[DANH MUC TỪ VIẾT TẮT 9](#_Toc483455001)

[CHƯƠNG 1. ĐẶT VẤN ĐỀ VÀ ĐỊNH HƯỚNG GIẢI PHÁP 10](#_Toc483455002)

[1. ĐẶT VẤN ĐỀ: 10](#_Toc483455003)

[2. MỤC TIÊU VÀ HƯỚNG GIẢI 10](#_Toc483455004)

[2.1. Mục tiêu của đề tài 10](#_Toc483455005)

[2.2. Hướng giải quyết 10](#_Toc483455006)

[3. CƠ SỞ LÝ THUYẾT 10](#_Toc483455007)

[3.1. Giới thiệu Java 10](#_Toc483455008)

[3.2. Giới thiệu về AngularJS 12](#_Toc483455009)

[3.3. Giới thiệu về Spring MVC 13](#_Toc483455010)

[3.4. Giới thiệu về Hibernate 16](#_Toc483455011)

[CHƯƠNG 2: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG 17](#_Toc483455012)

[1. MÔ TẢ CHỨC NĂNG 17](#_Toc483455013)

[1.1. Sơ đồ phân rã chức năng 17](#_Toc483455014)

[1.2. Tương tác người dùng 18](#_Toc483455015)

[1.3. Tương tác quản trị 18](#_Toc483455016)

[2. PHÂN TÍCH THIẾT KẾ 19](#_Toc483455017)

[2.1. Các tác nhân 19](#_Toc483455018)

[2.2. Biểu đồ Use Case 19](#_Toc483455019)

[2.3. Biểu đồ hoạt động 28](#_Toc483455020)

[2.4. Biểu đồ lớp 45](#_Toc483455021)

[2.5. Thiết kế cơ sở dữ liệu 45](#_Toc483455022)

[Chương 3: KIỂM THỬ TỰ ĐỘNG 50](#_Toc483455023)

[1. KHÁI NIỆM VỀ SELENIUM 50](#_Toc483455024)

[2. TÌM HIỂU VỀ SELENIUM 50](#_Toc483455025)

[3. KIỂM THỬ CHỨC NĂNG BẰNG SELENIUM 51](#_Toc483455026)

[3.1. Chức năng đăng kí 51](#_Toc483455027)

[3.2. Chức năng đăng nhập 52](#_Toc483455028)

[3.3. Chức năng đặt câu hỏi 52](#_Toc483455029)

[3.4. Chức năng trả lời câu hỏi 54](#_Toc483455030)

[3.5. Chức năng bình chọn câu hỏi 55](#_Toc483455031)

[3.6. Chức năng bình chọn câu trả lời 56](#_Toc483455032)

[3.7. Sửa câu trả lời 57](#_Toc483455033)

[3.8. Xóa câu trả lời 58](#_Toc483455034)

[3.9. Sửa câu hỏi 59](#_Toc483455035)

[3.10. Xóa câu hỏi 60](#_Toc483455036)

[3.11. Báo cáo câu hỏi 61](#_Toc483455037)

[CHƯƠNG 4: KẾT QUẢ THỰC NGHIỆM 63](#_Toc483455038)

[1. GIAO DIỆN CHỨC NĂNG 63](#_Toc483455039)

[1.1. giao diện trang đăng kí: 63](#_Toc483455040)

[1.2. giao diện trang đăng nhập 63](#_Toc483455041)

[1.3. Giao diện trang đặt câu hỏi 64](#_Toc483455042)

[1.4. Giao diện trang chủ: 64](#_Toc483455043)

[1.5. Giao diện trang chi tiết câu hỏi: 65](#_Toc483455044)

[1.6. Giao diện trang danh sách người dùng: 65](#_Toc483455045)

[1.7. Giao diện trang danh sách tags: 65](#_Toc483455046)

[1.8. Giao diện trang quản lý người dùng: 66](#_Toc483455047)

[1.9. Giao diện trang quản lý tags: 66](#_Toc483455048)

[1.10. Giao diện trang quản lý báo cáo: 67](#_Toc483455049)

[CHƯƠNG 5: KẾT LUẬN 68](#_Toc483455050)

[1. Đánh giá kết quả đạt được 68](#_Toc483455051)

[2. Kết luận rút ra: 68](#_Toc483455052)

[3. Định hướng phát triển 68](#_Toc483455053)

# DANH MỤC HÌNH ẢNH

[Hình 1: Các tính năng cốt lõi của AngularJS 14](#_Toc483493650)

[Hình 2: Mô hình MVC 15](#_Toc483493651)

[Hình 3: Mô hình hoạt động của DispatcherServlet 16](#_Toc483493652)

[Hình 4: File web.xml đơn giản 17](#_Toc483493653)

[Hình 5: Sơ đồ phân rã chức năng 19](#_Toc483493654)

[Hình 6: Sơ đồ Use-Case tổng quát 21](#_Toc483493655)

[Hình 7: Use-Case quản lý câu hỏi cá nhân 24](#_Toc483493656)

[Hình 8: Use-Case quản lý câu trả lời cá nhân 25](#_Toc483493657)

[Hình 9: Use-Case bình chọn 25](#_Toc483493658)

[Hình 10: Sơ đồ hoạt động đăng kí 30](#_Toc483493659)

[Hình 11: Sơ đồ hoạt động đăng nhập 31](#_Toc483493660)

[Hình 12: Sơ đồ hoạt động đặt câu hỏi 32](#_Toc483493661)

[Hình 13: Sơ đồ hoạt động xóa câu hỏi cá nhân 33](#_Toc483493662)

[Hình 14: Sơ đồ hoạt động sửa câu hỏi cá nhân 34](#_Toc483493663)

[Hình 15: Sơ đồ hoạt động thêm mới câu trả lời 35](#_Toc483493664)

[Hình 16: Sơ đồ hoạt động xóa câu trả lời cá nhân 36](#_Toc483493665)

[Hình 17: Sơ đồ hoạt động sửa câu trả lời cá nhân 37](#_Toc483493666)

[Hình 18: Sơ đồ hoạt động báo cáo câu hỏi 38](#_Toc483493667)

[Hình 19: Sơ đồ hoạt động báo cáo câu hỏi 39](#_Toc483493668)

[Hình 20: Sơ đồ hoạt động tìm kiếm 40](#_Toc483493669)

[Hình 21: Sơ đồ hoạt động tìm kiếm 41](#_Toc483493670)

[Hình 22: Sơ đồ hoạt động đổi tên người dùng 42](#_Toc483493671)

[Hình 23: Sơ đồ hoạt động xóa người dùng 43](#_Toc483493672)

[Hình 24: Sơ đồ hoạt động xóa tags 44](#_Toc483493673)

[Hình 25: Sơ đồ hoạt động quản lý báo cáo 45](#_Toc483493674)

[Hình 26: Sơ đồ lớp 46](#_Toc483493675)

[Hình 27: Kiểm thử chức năng đăng kí 52](#_Toc483493676)

[Hình 28: Kiểm thử chức năng đăng nhập 53](#_Toc483493677)

[Hình 29: Kiểm thử chức năng đặt câu hỏi 54](#_Toc483493678)

[Hình 30: Kiểm thử chức năng trả lời câu hỏi 56](#_Toc483493679)

[Hình 31: Kiểm thử chức năng bình chọn câu hỏi 56](#_Toc483493680)

[Hình 32: Kiểm thử chức năng bình chọn câu trả lời 57](#_Toc483493681)

[Hình 33: Kiểm thử chức năng sửa câu trả lời 59](#_Toc483493682)

[Hình 34: Kiểm thử chức năng xóa câu trả lởi 60](#_Toc483493683)

[Hình 35: Kiểm thử chức năng sửa câu hỏi 61](#_Toc483493684)

[Hình 36: Kiểm thử chức năng xóa câu hỏi 62](#_Toc483493685)

[Hình 37: Kiểm thử chức năng báo cáo câu hỏi 63](#_Toc483493686)

[Hình 38: Giao diện trang đăng kí 64](#_Toc483493687)

[Hình 39: Giao diện đăng nhập 64](#_Toc483493688)

[Hình 40: Giao diện trang đặt câu hỏi 65](#_Toc483493689)

[Hình 41: Giao diện trang chủ 65](#_Toc483493690)

[Hình 42: Giao diện trang chi tiết câu hỏi 66](#_Toc483493691)

[Hình 43: Giao diện trang danh sách người dùng 66](#_Toc483493692)

[Hình 44: Giao diện trang danh sách tags 67](#_Toc483493693)

[Hình 45: Giao diện trang quản lý người dùng 67](#_Toc483493694)

[Hình 46: Giao diện trang quản lý tags 68](#_Toc483493695)

[Hình 47: Giao diện trang quản lý báo cáo 68](#_Toc483493696)

# DANH MỤC BẢNG BIỂU

[Bảng 1. Bảng mô tả cơ sở dữ liệu cho user 47](#_Toc483493697)

[Bảng 2.Bảng mô tả cơ sở dữ liệu choquestions 47](#_Toc483493698)

[Bảng 3. Bảng mô tả cơ sở dữ liệu cho answers 47](#_Toc483493699)

[Bảng 4. Bảng mô tả cơ sở dữ liệu cho followed 48](#_Toc483493700)

[Bảng 5.Bảng mô tả cơ sở dữ liệu cho following 48](#_Toc483493701)

[Bảng 6.Bảng mô tả cơ sở dữ liệu choreport 48](#_Toc483493702)

[Bảng 7.Bảng mô tả cơ sở dữ liệu chorole 49](#_Toc483493703)

[Bảng 8. Bảng mô tả cơ sở dữ liệu cho tags 49](#_Toc483493704)

[Bảng 9. Bảng mô tả cơ sở dữ liệu cho tag\_question 49](#_Toc483493705)

[Bảng 10.Bảng mô tả cơ sở dữ liệu cho user\_role 50](#_Toc483493706)

[Bảng 11.Bảng mô tả cơ sở dữ liệu cho vote\_answers 50](#_Toc483493707)

[Bảng 12**.** Bảng mô tả cơ sở dữ liệu chovote\_questions 50](#_Toc483493708)

# DANH MUC TỪ VIẾT TẮT

# CHƯƠNG 1. ĐẶT VẤN ĐỀ VÀ ĐỊNH HƯỚNG GIẢI PHÁP

## 1. ĐẶT VẤN ĐỀ:

Ngày nay, công nghệ thông tin đã có những bước phát triển vượt bậc. Máy tính điện tử giờ đây không còn là một phương tiện quý hiếm mà ngày càng trở nên phổ biến, trở thành công cụ làm việc và giải trí thông dụng của con người, không chỉ ở nơi làm việc mà ngay cả ở trong gia đình. Không những vậy công nghệ thông tin còn được áp dụng trên nhiều lĩnh vực như kinh tế, chính trị, xã hội… Ứng dụng của công nghệ thông tin và tin học hóa được xem là một trong những yếu tố mang tính quyết định trong hoạt động của quốc gia, tổ chức.

Chính vì sự phát triển mạnh mẽ đó của công nghệ thông tin mà nhu cầu của con người về tìm hiểu công nghệ thông tin ngày càng lớn. Với các công cụ tìm kiếm hiện nay thì việc tìm kiếm những thông tin về công nghệ thông tin là không quá khó khăn. Nhưng có vấn đề được đặt ra là lượng thông tin tìm thấy quá lớn và đôi khi không đúng với điều mình mong muốn khiến chúng ta gặp khó khăn trong việc giải quyết các vấn đề mà mình thắc mắc. Đặc biệt, có những vấn đề phải là những người đã có kinh nghiệm mới giải quyết được. Vì vậy, vấn đề được đặt ra là cần có một nơi để mọi người có thể tìm kiếm thông tin dễ dàng, cùng nhau trao đổi để tìm ra giải pháp giải quyết vấn đề một cách nhanh chóng và hiệu quả nhất.

Từ thực tế nêu trên nên Website “Askme” đã được ra đời. Askme là nơi mọi người có thể hỏi, giải đáp, chia sẻ kinh nghiệm của mình cho mọi người để giải quyết vấn những vấn để mà mọi người gặp phải trong quá trình học tập, làm việc nghiên cứu lĩnh vực công nghệ thông tin.

## 2. MỤC TIÊU VÀ HƯỚNG GIẢI

### 2.1. Mục tiêu của đề tài

Xây dựng được một trang web đáp ứng các yêu cầu

* Tạo được sự thuận tiện dễ dàng cho mọi người khi xem, đánh giá, tham gia đặt câu hỏi cũng như trả lời.
* Xây dựng được trang quản trị quản lý hiệu quả website, dễ sử dụng, đáp ứng được nhu cầu quản trị website.
* Đáp ứng được các yêu cầu về kĩ thuật thông tin chính xác được chia sẻ giữa các thành viên.

### 2.2. Hướng giải quyết

Xây dựng lên một hệ thống website “Askme” đáp ứng nhu cầu sử dụng của người dùng.

## 3. CƠ SỞ LÝ THUYẾT

### 3.1. Giới thiệu Java

#### 3.1.1. Định nghĩa

Ngôn ngữ lập trình hướng đối tượng, được thiết kế độc lập với hệ điều hành, cho phép người lập trình viết chương trình một lần và có thể sửa dụng tại bất kì đâu.

#### 3.1.2. Lịch sử ra đời

Java được khởi đầu bởi James Gosling và bạn đồng nghiệp ở Sun MicroSystem năm 1991. Ban đầu Java được tạo ra nhằm mục đích viết phần mềm cho các sản phẩm gia dụng và có tên là Oak. Java được phát hành năm 1994, đến năm 2010 được Oracle mua lại từ Sun MicroSystem.

#### 3.1.3. Đặc điểm của java

* **Độc lập nền (architecture neutral):** Một chương trình viết bằng ngôn ngữ Java có thể chạy trên nhiều máy tính có hệ điều hành khác nhau (Windows, Unix, Linux,…) với điều kiện ở đó có cài đặt máy ảo Java (Java Virual Machine).
* **Thông dịch (Interpreter):** Java là một ngôn ngữ lập trình vừa biên dịch vừa thông dịch. Chương trình nguồn viết bằng ngôn ngữ Java có đuôi \*.java đầu tiên được biên dịch thành tập tin có đuôi \*.class và sau đó sẽ được trình thông dịch thành mã máy.
* **Hướng đối tượng (OOP – object-oriented programming):** Hướng đối tượng trong Java tương tự như C++ nhưng Java là một ngôn ngữ lập trình hướng đối tượng hoàn toàn. Tất cả mọi thứ đề cập đến trong Java đều liên quan đến các đối tượng được định nghĩa trước, thậm chí hàm chính của một chương trình viết bằng Java (đó là hàm main) cũng phải đặt bên trong một lớp. Hướng đối tượng trong Java không có tính đa kế thừa (multi inheritance) như trong C++ mà thay vào đó Java đưa ra khái niệm interface để hỗ trợ tính đa kế thừa.
* **Đa nhiệm- đa luồng (MultiTasking- Multithreading):** java hỗ trợ lập trình đa nhiệm, đa luồng cho phép nhiều tiến trình, tiến trình có thể chạy song song cùng một thời điểm và tương tác với nhau.
* **Linh hoạt (portable):** Chương trình ứng dụng viết bằng ngôn ngữ Java chỉ cần chạy được trên máy ảo Java là có thể chạy được trên bất kỳ máy tính, hệ điều hành nào có máy ảo Java. “Viết một lần, chạy mọi nơi” (write once, run anywhere).

#### 3.1.4. Tác dụng của Java

* **Java được sử dụng để**

**Viết ứng dụng web (J2EE):** Java thường được sử dụng để xây dựng các hệ thống web lớn đòi hỏi độ bảo mật cao, số lượng người dùng lớn như ngân hàng, phần mềm quản lý bệnh viện, CRM, HRM, … Đối với các website nhỏ thông thường rất ít viết bằng Java.

**Viết ứng dụng mobile (J2ME):** trước đây nền tảng J2ME thường được sử dụng để viết game và app cho di động feature phone (file .jar) và giờ đây khi điện thoại thông minh sử dụng hệ điều hành Android ra đời Java lại tiếp tục được sử dụng để viết app và game cho nền tảng Android (file .apk).

**Viết ứng dụng desktop (J2SE):** Các ứng dụng desktop viết bằng java thật sự không nhiều có thể kể đến một số phần mềm như JMeter hoặc Designer Vista. Lợi thế lớn nhất của ứng dụng java là bạn chỉ viết một lần và sau đó có thể đem chương trình lên windows, linux hay mac để chạy mà không cần phải viết lại. Tuy nhiên do chạy trên JVM nên hiệu xuất của ứng dụng thấp hơn so với các ngôn nhữ như C/C++, C#.

### 3.2. Giới thiệu về AngularJS

#### 3.2.1. Định nghĩa

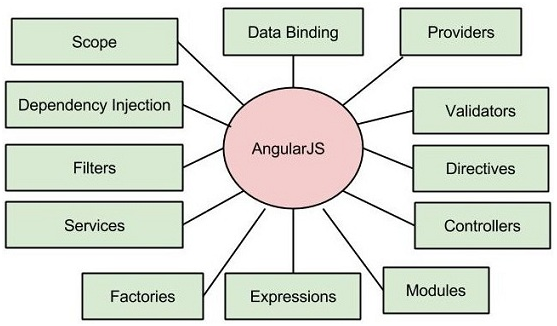
Là một Framework có cấu trúc cho các ứng dụng web động. Nó cho phép bạn sử dụng HTML như là ngôn ngữ mẫu và cho phép bạn mở rộng cú pháp của HTML để diễn đạt các thành phần ứng dụng của bạn một cách rõ ràng và súc tích. Hai tính năng cốt lõi: Data binding và Dependency injection của AngularJS loại bỏ phần lớn code mà bạn thường phải viết. Nó xảy ra trong tất cả các trình duyệt, làm cho nó trở thành đối tác lý tưởng của bất kỳ công nghệ server nào.

#### 3.2.2. Đặc trưng của AngularJS

* Phát triển dựa trên Javascript.
* Tạo các ứng dụng client-side theo mô hình MVC.
* Khả năng tương thích cao, tự động xử lý mã Javascript để phù hợp với mỗi trình duyệt.
* Mã nguồn mở, miễn phí hoàn toàn và được sử dụng rộng rãi.

#### 3.2.3. Các tính năng cốt lõi của AngularJS

* **Data-binding:** Nó tự động đồng bộ hóa dữ liệu giữa thành phần model và view.
* **Scope**: Là đối tượng có nhiệm vụ giao tiếp giữa controller và view của ứng dụng.
* **Controller:** xử lý dữ liệu cho đối tượng $scope, từ đây bên views sẽ sử dụng các dữ liệu trong scope để hiển thị ra tương ứng.
* **Service:** là singleton object được khởi tạo 1 lần duy nhất cho mỗi ứng dụng, cung cấp các phương thức lưu trữ có sẵn. Ví dụ $http để tạo ra XMLHttpRequests.
* **Filter:** Lọc các tập con từ tập item trong các mảng và trả về các mảng mới.
* **Directive:** dùng để tạo các thẻ HTML riêng phục vụ những mục đích riêng. AngularJS có những directive có sẵn như ng-bind, ng-model…
* **Template:** một thành phần của view, hiển thị thông tin từ controller.
* **Routing:** chuyển đổi giữa các action trong controller, qua lại giữa các view.
* **MVC:** mô hình thiết kế để phân chia ứng dụng thành nhiều phần khác nhau. Mỗi phần có một nhiệm vụ nhất định. AngularJS không triển khai MVC theo cách truyền thống, mà gắn liền hơn với Model-View-ViewModel.
* **Deep-link:** liên kết sâu, cho phép mã hóa trạng thái ứng dụng trong các URL để nó có thể bookmark với công cụ tìm kiếm. Các ứng dụng có thể được phục hồi lại từ các địa chỉ URL với cùng một trạng thái.
* **Dependency Injection:** AngularJS có sẵn một hệ thống con dependency Injection để giúp các lập trình viên tạo ra các ứng dụng dễ phát triển, dễ hiểu và dễ kiểm tra.



Hình 1: Các tính năng cốt lõi của AngularJS

#### 3.2.4. Các components chính

* **Ng-app**: định nghĩa và liên kết một ứng dụng AngularJS tới HTML.
* **Ng-model**: gắn kết giá trị của dữ liệu ứng dụng AngularJS đến các điều khiển đầu vào HTML.
* **Ng-bind**: gắn dữ liệu ứng dụng AngularJS đến các thẻ HTML.

#### 3.2.5. Ưu điểm của AngularJS

* Cung cấp khả năng tạo ra các Single Page Aplication dễ dàng.
* Cung cấp khả năng data binding tới HTML, khiến cho người dùng cảm giác linh hoạt thân thiện.
* Dễ dàng Unit test.
* Dễ tái sử dụng component.
* Giúp lập trình viên viết code ít hơn với nhiều chức năng hơn.
* Chạy được trên các loại trình duyệt, trên cả PC lẫn mobile.

#### 3.2.6. Nhược điểm của AngularJS

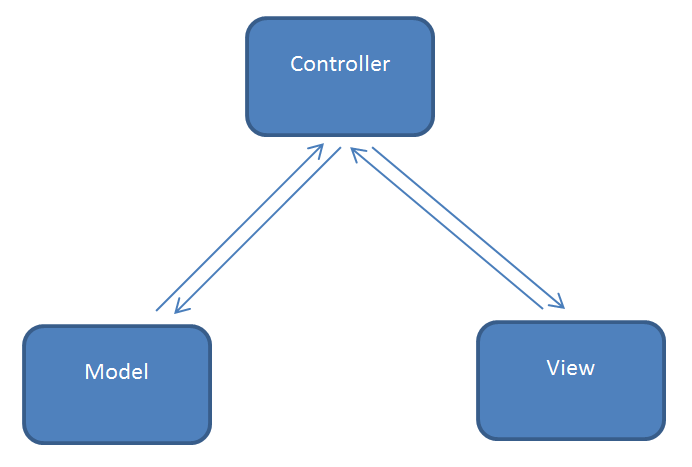
* Không an toàn: được phát triển từ Javascript cho nên ứng dụng được viết bởi AngularJS không an toàn. Nên có sự bảo mật và xác thực phía server sẽ giúp ứng dụng an toàn hơn.
* Nếu người sử dụng vô hiệu hóa JavaScript thì sẽ chỉ nhìn thấy trang cơ bản.

### 3.3. Giới thiệu về Spring MVC

#### 3.3.1. Giới thiệu về mô hình MVC

*3.3.1.1. Định nghĩa*

Viết tắt của Model – View – Controller. Là một kiến trúc phần mềm hay mô hình thiết kế được sử dụng trong công nghệ phần mềm. Mô hình MVC tách biệt phần xử lý dữ liệu ra khỏi phần giao diện, cho phép phát triển, kiểm tra và bảo trì các thành phần một cách độc lập.

****

Hình 2: Mô hình MVC

* **Model:** thể hiện các cấu trúc dữ liệu. Các lớp thuộc thành phần model thường thực hiện các tác vụ như truy vấn, thêm, xóa, cập nhật dữ liệu. Khi dữ liệu trong Model thay đổi, thành phần View sẽ được cập nhật lại.
* **View:** là thành phần thể hiện dữ liệu trong Model thành các giao diện tương tác với người sử dụng. Một mô hình có nhiều View phụ thuộc vào các mục đích khác nhau.
* **Controller** đóng vai trò trung gian giữa Model và View. Thông tin người dùng từ View được gửi cho Controller xử lý, sau đó Controller tương tác với Model để lấy dữ liệu được yêu cầu, sau cùng Controller trả dữ liệu này về cho View.

*3.3.1.2. Ưu nhược điểm của MVC*

* **Ưu điểm**

Thể hiện tính chuyên nghiệp trong lập trình, phân tích thiết kế. Do được chia thành các thành phần độc lập nên giúp phát triển ứng dụng nhanh, đơn giản, dễ nâng cấp, bảo trì.

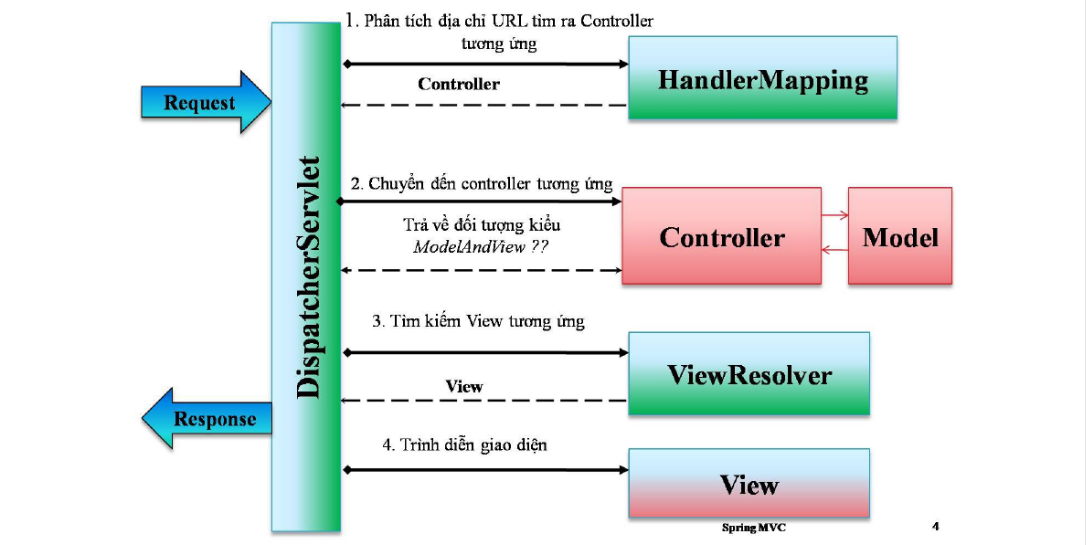
* **Nhược điểm**

Đối với dự án nhỏ việc áp dụng mô hình MVC gây cồng kềnh, tốn thời gian trong quá trình phát triển. Tốn thời gian trung chuyển dữ liệu của các thành phần.

#### 3.3.2. Giới thiệu SpringMVC

*3.3.2.1. Định nghĩa*

SpringMVC Framework cài đặt đầy đủ đặc tính của MVC pattern để xây dựng các ứng dụng Web và được thiết kế xung quanh một DispatcherServlet xử lý hết tất cả các HTTP yêu cầu và HTTP phản hồi lại. Các luồng xử lý của Spring MVC được mô tả trong sơ đồ sau:



Hình 3: Mô hình hoạt động của DispatcherServlet

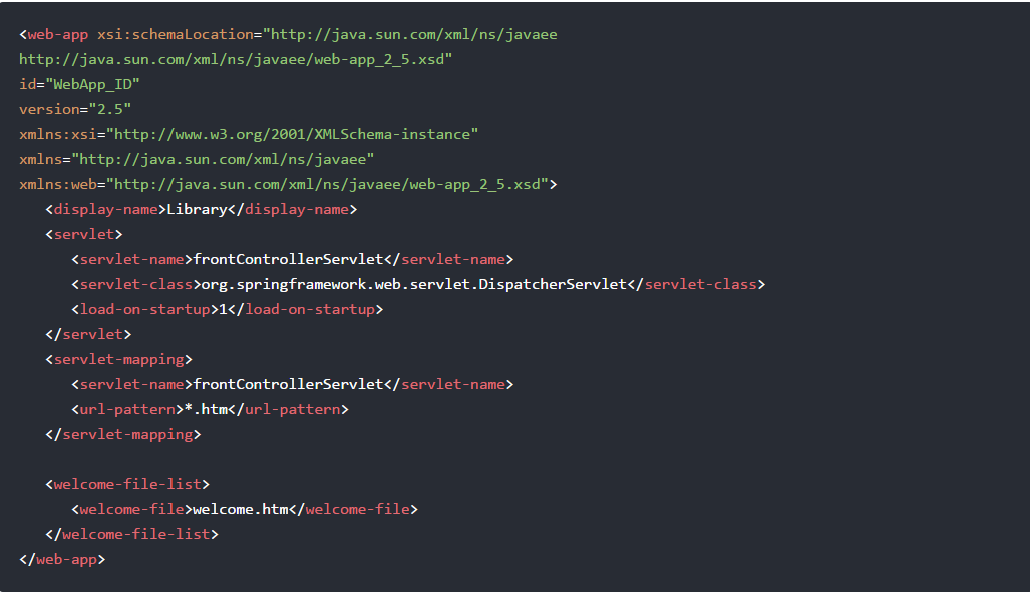
* Sau khi nhận được 1 yêu cầu HTTP, DispatcherServlet sẽ chỉ định cho HandlerMapping gọi Controller thích hợp.
* Controller sẽ nhận yêu cầu và gọi các Service tương ứng thích hợp dựa trên phương thức GET được sử dụng hoặc phương thức POST. Các phương thích service này sẽ thiết lập một nhóm các dữ liệu Model được định nghĩa theo logic business và trả về tên View cho DispatcherServlet.
* Các DispatcherServlet sẽ được các ViewResolver hỗ trợ để chọn được View đã định nghĩa tương ứng với Request.
* Khi View được hoàn thiện, các DispatcherServlet sẽ chuyển dữ liệu Model tới View và render trên trình duyệt.

Tất cả các thành phần nêu trên, ví dụ như HandlerMapping, Controller và ViewResolver là bộ phận của WebApplicationContext một mở rộng của ApplicationContext với một số tính năng bổ sung cần thiết cho các ứng dụng web.

*3.3.2.2. DispatcherServlet:*

* DispatcherServlet đóng vai trò như quả tim của Spring MVC, hoạt động giống như Front Controller, tất cả các request của người dùng được giữ lại ở servlet này. DispatcherServlet cũng như những servlet khác, nó được cấu hình trong file triển khai của ứng dụng được mô tả trong file web.xml.

Dưới đây là một file web.xml đơn giản:

\

Hình 4: File web.xml đơn giản

* Trong file chúng ta cấu hình tên của servlet là frontControllerServlet, tức là chúng ta đang cấu hình load nội dung của file frontControllerServlet-servlet.xml. Và mẫu URL trong phần servlet-mapping phải gồm đuôi “.htm”.

*3.3.2.3. Spring Application Context*

* Default Application context file: Mặc định DispatcherServlet sẽ load nội dung của các file xml có tên dạng [tên servlet]-servlet.xml. Như vậy có nghĩa là khi servlet của chúng ta frontControllerServlet được load nó sẽ load nội dung từ file xml: /WEB-INF/frontControllerServlet-servlet.xml.
* User definded application context file: chúng ta có thể ghi đè tên và vị trí của file xml mặc định bằng cách cung cấp khởi tạo tham số cho DispatcherServlet. Tên của tham số khởi tạo (<param-name>) là contextConfigLocation, giá trị tham số khởi tạo (<param-value>) được quy định cụ thể bởi tên và vị trí của file xml mà chúng ta cần được load bởi container.

### 3.4. Giới thiệu về Hibernate

Là một Framework cho persistence layer, là một dịch vụ lưu trữ và truy vấn dữ liệu quan hệ mạnh mẽ và nhanh chóng. Hibernate giúp người phát triển các class dùng để lưu trữ dữ liệu theo cách thức hướng đối tượng. Hibernate cho phép thực hiện các câu truy vấn dữ liệu theo cách thức hướng đối tượng. Hibernate cho phép thực hiện các câu truy vấn dữ liệu bằng cách sử dụng ngôn ngữ SQL mở rộng của Hibernate (HQL) hoặc là ngôn ngữ SQL nguyên thủy cũng như là sử dụng các API.

Kiến trúc của Hibernate có 3 phần chính:

* **Quản lý kết nối:** Hibernate cung cấp dịch vụ kết nối cơ sở dữ liệu hiệu quả. Kết nối cơ sở dữ liệu là phần tốn kém tài nguyên và hiệu năng vì nó đòi hỏi nhiều tài nguyên phải mở và đóng kết nối cơ sở dữ liệu.
* **Quản lý giao dịch:** Quản lý dịch vụ giao dịch cung cấp khả năng người sử dụng để thực hiện nhiều hơn một câu lệnh tại một thời điểm.
* **Đối tượng bản đồ quan hệ:** Đối tượng bản đồ quan hệ là kỹ thuật ánh xạ biểu diễn dữ liệu từ một mô hình đối tượng cho đến một mô hình dữ liệu quan hệ. Đối tượng này được sử dụng để thực hiện các thao tác như truy vấn, thêm mới, cập nhật và xóa các bản ghi dạng bảng.

#### 3.4.1. Ưu điểm của Hibernate

* **Mã nguồn mở và nhẹ:** Hibernate Framework là mã nguồn mở theo LGPL licence và dung lượng nhỏ.
* **Hiệu xuất nhanh:** Hiệu xuất của Hibernate Framwork là nhanh bởi bộ nhớ cache được sử dụng trong nội bộ Hibernate Framwork. Có 2 loại bộ nhớ cache trong Hibernate Framework, gồm bộ nhớ cache cấp một và bộ nhớ cache cấp hai. Bộ nhớ cache cấp một được bật bằng lệnh mặc định.
* **Truy vấn cơ sở dữ liệu độc lập:** HQL (Hibernate Query Language) là phiên bản hướng đối tượng của SQL. Nó tạo ra các truy vấn cơ sở dữ liệu độc lập. Vì vậy, bạn không cần phải viết các truy vấn cơ sở dữ liệu cụ thể.
* **Tạo bảng tự động:** Hibernate Framework cung cấp phương tiện để tạo ra các bảng cơ sở dữ liệu tự động. Vì vậy, không cần phải tạo ra các bảng trong cơ sở dữ liệu bằng tay.
* **Đơn giản lệnh join phức tạp:** Có thể lấy dữ liệu từ nhiều bảng một cách dễ dàng với Hibernate Framework.
* **Cung cấp thống kê truy vấn và trạng thái cơ sở dữ liệu:** Hibernate hỗ trợ bộ nhớ cache truy vấn và cung cấp số liệu thống kê về truy vấn và trạng thái cơ sở dữ liệu.

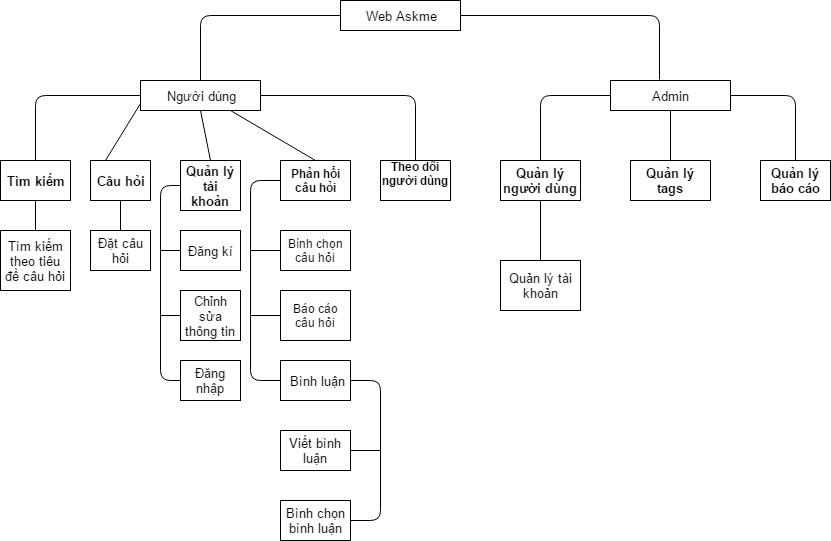
#### 3.4.2. Nhược điểm

* Chậm hơn JDBC do phải tạo mã SQL.

# CHƯƠNG 2: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

## 1. MÔ TẢ CHỨC NĂNG

### 1.1. Sơ đồ phân rã chức năng



Hình 5: Sơ đồ phân rã chức năng

### 1.2. Tương tác người dùng

#### 1.2.1. Tìm kiếm câu hỏi

* Tìm kiếm theo tiêu đề câu hỏi.

#### 1.2.2. Câu hỏi

* Người dùng có thể đặt câu hỏi mới, chỉnh sửa, xóa câu hỏi mà mình đã đặt.

#### 1.2.3. Quản lý tài khoản

* Đăng ký tài khoản.
* Chỉnh sửa thông tin cá nhân.
* Đăng nhập vào website.

#### 1.2.4. Phản hồi câu hỏi

* Bình chọn câu hỏi .
* Báo cáo câu hỏi
* Bình luận câu hỏi

#### 1.2.5. Theo dõi người dùng

* Người dùng có thể theo dõi người dùng khác trong hệ thống.

### 1.3. Tương tác quản trị

#### 1.3.1. Quản lý người dùng

* Quản lý tài khoản: quản trị có thể xóa tài khoản

#### 1.3.2. Quản lý tags

* Người quản trị có thể xóa tags

#### 1.3.3. Quản lý báo cáo

* Người quản trị có thể thay đổi trạng thái báo cáo hoặc xóa câu hỏi bị báo cáo.

## 2. PHÂN TÍCH THIẾT KẾ

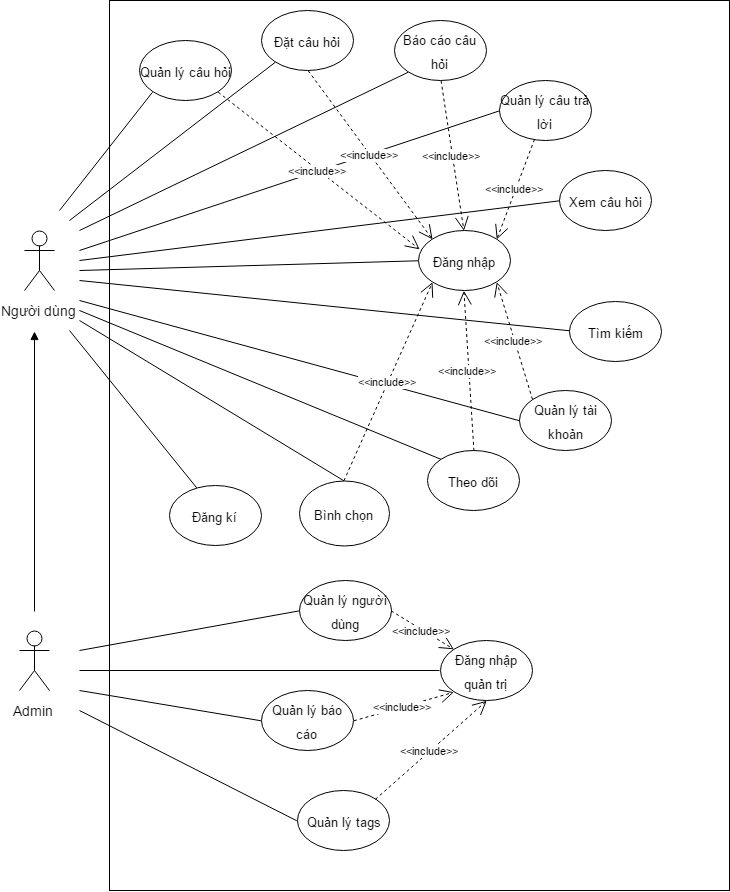
### 2.1. Các tác nhân

Hệ thống website Askme bao gồm các tác nhân:

* Người dùng: Là đối tượng truy cập vào website, tìm kiếm, xem các câu hỏi có trong hệ thống, đặt câu hỏi mới, bình luận, bình chọn cho câu hỏi, bình luận có trong hệ thống,
* Người quản trị: điều hành, quản lý và theo dõi mọi hoạt động của hệ thống.

### 2.2. Biểu đồ Use Case

#### 2.2.1. Biểu đồ Use Case tổng quan



Hình 6: Sơ đồ Use-Case tổng quát

Hệ thống được chia làm 15 chức năng chính, chi tiết các use case như sau:

* **Đăng kí:** người sử dụng phải đăng kí tài khoản nếu như muốn sử dụng một số chức năng trong hệ thống như: đặt câu hỏi, bình luận, bình chọn…
* **Đăng nhập:** người dùng đăng nhập vào hệ thống.
* **Tìm kiếm:** khả năng tìm kiếm áp dụng cho tất cả đối tượng sử dụng, từ khóa tìm kiếm xoay quanh tiêu đề các câu hỏi có trong cơ sở dữ liệu.
* **Xem câu hỏi:** người dùng có thể truy cập vào hệ thống để xem thông tin câu hỏi câu trả lời trong mỗi câu hỏi có trong trang web.
* **Theo dõi:** người dùng có thể theo dõi người dùng khác trong hệ thống
* **Báo cáo câu hỏi:** khi thấy câu hỏi trong hệ thống vi phạm về nội dung mà trang web hướng tới người dùng có thể báo cáo câu hỏi cho người quản trị của hệ thống.
* **Bình chọn:** người dùng có thể bình chọn cho câu hỏi, câu trả lời có trong hệ thống
* **Quản lý câu hỏi cá nhân:** người dùng có thể sửa, xóa câu hỏi của mình đã đăng trong hệ thống.
* **Quản lý câu trả lời:** người dùng có thể thêm, sửa, xóa câu trở lời cho các câu hỏi có trong hệ thống.
* **Đặt câu hỏi:** Người dùng thêm mới câu hỏi vào trong hệ thống.
* **Đăng nhập quản trị:** Người dùng đăng nhập vào hệ thống với quyền quản trị
* **Quản lý tài khoản**: người dùng có thể sửa đổi thông tin cá nhân.
* **Quản lý người dùng:** Chức năng của người quản trị. Người quản trị hệ thống có thể thực hiện thao tác xóa người dùng ra khỏi hệ thống.
* **Quản lý báo cáo:** Người quản trị hệ thống có thể xem nội dung report để đưa ra cách thức xử lý đối với câu hỏi và phần report cho phù hợp.
* **Quản lý tags:** Người quản trị có thể xóa các tags có trong hệ thống.

#### 2.2.2. Use case phía người dùng

##### 2.2.2.1. Chức năng đăng kí

|  |  |
| --- | --- |
| Tên Use Case | Đăng kí thành viên |
| Tác nhân chính | Người dùng |
| Mô tả tóm tắt | Người dùng ấn vào đăng kí trên thanh công cụ và cung cấp các thông tin cá nhân cho hệ thống để tạo tài khoản đăng nhập |
| Đầu vào | Người dùng (người chưa có tài khoản) |
| Kịch bản chính | * Trên giao diện chính khi truy cập vào trang web, người dùng ấn vào nút “đăng kí”, hệ thống sẽ chuyển đến trang đăng kí tài khoản * Trên trang đăng kí tài khoản được bố trí các dòng nhập. Người dùng nhập đầy đủ và chính xác vào các thông tin: email, họ và tên, mật khẩu, xác thực mật khẩu. * Người dùng bấm vào nút đăng kí để hoàn tất đăng kí. |
| Đầu ra | Tài khoản của người dùng được khởi tạo. |
| Ngoại lệ | Người dùng cung cấp thông tin không hợp lệ, hệ thống đưa ra thông báo lỗi và yêu cầu người dùng nhập lại.  Người dùng hủy bỏ đăng kí bằng cách chuyển trang khác. |

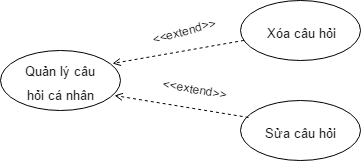
##### 2.2.2.2. Chức năng đăng nhâp

|  |  |
| --- | --- |
| Tên Use-case | Đăng nhập |
| Tác nhân chính | Người dùng, người quản trị |
| Mô tả tóm tắt | Người dùng nhập thông tin tài khoản và đăng nhập |
| Đầu vào | Người dùng đã có tài khoản trong hệ thống |
| Kịch bản chính | * Trên giao diện chính, người dùng chọn đăng nhập trên thanh công cụ, hệ thống sẽ chuyển sang trang đăng nhập. * Người dùng nhập email và mật khẩu sau đó chọn “Đăng nhập”. |
| Đầu ra | Hệ thống sẽ chuyển về trang chính với trạng thái đã đăng nhập, tên người dùng sẽ được hiển thị tại góc trên bên phải của website. |
| Ngoại lệ | Sai mật khẩu hoặc email không đúng. Hệ thống sẽ báo lỗi và yêu cầu nhập lại. Người dùng có thể thoát ca sử dụng bằng cách chọn di chuyển đến trang khác trên thanh công cụ của hệ thống. |

##### 2.2.2.3. Đặt câu hỏi

|  |  |
| --- | --- |
| Tên Use Case | Đặt câu hỏi |
| Tác nhân chính | Người dùng, người quản trị |
| Mô tả tóm tăt | Người dùng đăng nhập sau đó đặt câu hỏi |
| Đầu vào | Người dùng đã đăng nhập |
| Kịch bản chính | * Sau khi đăng nhập vào hệ thống, người dùng ấn vào “Hỏi” trên thanh công cụ hoặc ấn vào nút đặt câu hỏi ở trên cùng bên trái của trang chính. * Hệ thống sẽ chuyển đến trang đặt câu hỏi, ở đây người dùng sẽ điền đầy đủ thông tin câu hỏi * Ấn nút “submit” |
| Đầu ra | Câu hỏi được thêm mới vào CSDL |
| Ngoại lệ | Nhập sai thông tin hệ thống sẽ thông báo lỗi. |

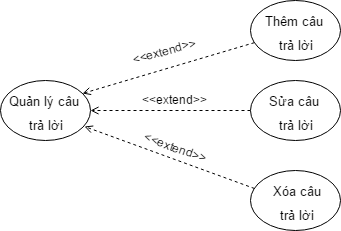
##### 2.2.2.4. Quản lý câu hỏi cá nhân



Hình 7: Use-Case quản lý câu hỏi cá nhân

|  |  |
| --- | --- |
| Tên Use Case | Quản lý câu hỏi cá nhân |
| Tác nhân chính | Người dùng, người quản trị |
| Mô tả tóm tăt | Người dùng đăng nhập sau đó có thể xóa hoặc sửa câu hỏi của mình |
| Đầu vào | Người dùng đã đăng nhập |
| Kịch bản chính | * Sau khi đăng nhập vào hệ thống, người dùng chọn câu hỏi của mình muốn xóa. * Trang chi tiết câu hỏi sẽ được hiển thị. Người dùng chọn xóa ở phía dưới cùng câu hỏi. * Một thông báo sẽ hiện lên ”Bạn có muốn xóa câu hỏi…”. * Nhấn nút “xóa”. |
| Đầu ra | Câu hỏi bị xóa khỏi hệ thống |

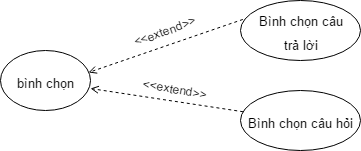
##### 2.2.2.5. Quản lý câu trả lời



Hình 8: Use-Case quản lý câu trả lời cá nhân

|  |  |
| --- | --- |
| Tên Use Case | Quản lý câu trả lời |
| Tác nhân chính | Người dùng, người quản trị |
| Mô tả tóm tăt | Người dùng đăng nhập sau đó có thể thêm, sửa, xóa câu trả lời của mình. |
| Đầu vào | Người dùng đã đăng nhập |
| Kịch bản chính | * Sau khi đăng nhập vào hệ thống, người dùng chọn câu hỏi. * Chọn trả lời của mình muốn xóa. Rồi nhấn “xóa” * Một thông báo sẽ hiện lên ”Bạn có muốn xóa câu trả lại…”. * Nhấn nút “xóa”. |
| Đầu ra | Câu trả lời bị xóa khỏi hệ thống |

##### 2.2.2.6. Bình chọn



Hình 9: Use-Case bình chọn

|  |  |
| --- | --- |
| Tên Use Case | Bình chọn |
| Tác nhân chính | Người dùng, người quản trị |
| Mô tả tóm tăt | Người dùng đăng nhập rồi chọn câu hỏi sau đó có thể bình chọn cho câu hỏi hoặc câu trả lời |
| Đầu vào | Người dùng đã đăng nhập |
| Kịch bản chính | * Sau khi đăng nhập vào hệ thống, người dùng chọn câu hỏi. * Ấn nút “Upvote” hoặc “Downvote” phía bên trái cạnh nội dung câu hỏi |
| Đầu ra | Trạng thái của nút “Upvote” hoặc “Downvote” sẽ bị thay đổi |
| Ngoại lệ | Nếu đã “Upvote” (“Downvote”) thì click tiếp “Upvote” (“Downvote”) trạng thái nút sẽ không thay đổi. |

##### 2.2.2.7. Báo cáo vi phạm

|  |  |
| --- | --- |
| Tên Use Case | Báo cáo vi phạm |
| Tác nhân chính | Người dùng |
| Mô tả tóm tăt | Người dùng đăng nhập rồi chọn câu hỏi không phải của mình rồi chọn báo cáo câu hỏi |
| Đầu vào | Người dùng đã đăng nhập |
| Kịch bản chính | * Sau khi đăng nhập vào hệ thống, người dùng chọn câu hỏi. * Trang hiển thị chi tiết câu hỏi sẽ hiện ra. Người dùng click vào nút “báo cáo” * Trang báo cáo sẽ hiện ra. Người dùng sẽ nhập nội dung báo cáo vào. * Rồi ấn nút ”submit” |
| Đầu ra | Nội dung báo cáo được gửi lên cho người quản trị xử lý |

##### 2.2.2.8. Tìm kiếm

|  |  |
| --- | --- |
| Tên Use Case | Tìm kiếm |
| Tác nhân chính | Người dùng, người quản trị |
| Mô tả tóm tăt | Người dùng tìm kiếm câu hỏi theo yêu cầu |
| Đầu vào | Từ khóa tìm kiếm của người dùng |
| Kịch bản chính | * Sau khi truy cập vào hệ thống người dùng chọn ô tìm kiếm trên thanh công cụ của trang web * Người dùng nhập từ khóa tìm kiếm vào ấn nút “Submit” |
| Đầu ra | Trả về trang tìm kiếm với danh sách các câu hỏi có chứa từ khóa tìm kiếm |

##### 2.2.2.9. Theo dõi người dùng

|  |  |
| --- | --- |
| Tên Use Case | Theo dõi người dùng |
| Tác nhân chính | Người dùng, người quản trị |
| Mô tả tóm tăt | Người dùng đăng nhập vào hệ thống sau đó truy cập vào trang danh sách người dùng và có thể theo dõi người dùng |
| Đầu vào | Đã đăng nhập vào hệ thống |
| Kịch bản chính | * Sau khi đăng nhập thành công vào hệ thống người dùng chọn “Users” ở trên thanh công cụ. * Danh sách các người dùng sẽ hiện ra người dùng sẽ chọn người muốn theo dõi vào ấn nút “follow” để theo dõi |
| Đầu ra | Theo dõi thành công nút “follow” sẽ được đổi màu và trở thành nút “following” |

##### 2.2.2.10 Đổi mật khẩu

|  |  |
| --- | --- |
| Tên Use Case | Đổi mật khẩu |
| Tác nhân chính | Người dùng, người quản trị |
| Mô tả tóm tăt | Người dùng đăng nhập vào hệ thống sau đó truy cập vào trang cá nhân chọn đổi mật khẩu và có thể đổi mật khẩu của mình |
| Đầu vào | Đã đăng nhập vào hệ thống |
| Kịch bản chính | * Sau khi đăng nhập thành công vào hệ thống người dùng chọn profile * Trang cá nhân sẽ hiển thị ra người dùng sẽ chọn đổi mật khẩu ở menu chức năng bên trái màn hình. * Trang đổi mật khẩu sẽ hiện ra người dùng nhập đầy đủ thông tin và ấn nút “submit” |
| Đầu ra | Hiển thị thông báo đổi mật khẩu thành công |

#### 2.2.3. Use Case phía người quản trị

##### 2.2.3.1. Quản lý người dùng

|  |  |
| --- | --- |
| Tên Use Case | Quản lý người dùng |
| Tác nhân chính | Người quản trị |
| Mô tả tóm tăt | Người quản trị truy cập vào hệ thống sau đó chọn quản lý người dùng và có thể xóa người dùng trong hệ thống |
| Đầu vào | Đã đăng nhập vào hệ thống |
| Kịch bản chính | * Sau khi đăng nhập vào hệ thống người quản trị chọn “quanlyuser” trên thanh công cụ của trang web * Một danh sách người dùng sẽ được hiển thị * Người quản trị chọn người dùng muốn xóa và ấn nút “xóa” |
| Đầu ra | Xóa thành công người dùng khỏi hệ thống |

##### 2.2.3.2 Quản lý tags

|  |  |
| --- | --- |
| Tên Use Case | Quản lý tag |
| Tác nhân chính | Người quản trị |
| Mô tả tóm tăt | Người quản trị truy cập vào hệ thống sau đó chọn quản lý tag và có thể xóa tag có trong hệ thống |
| Đầu vào | Đã đăng nhập vào hệ thống |
| Kịch bản chính | * Sau khi đăng nhập vào hệ thống người quản trị chọn “quanlytags” trên thanh công cụ của trang web * Một danh sách tag sẽ được hiển thị * Người quản trị chọn tag muốn xóa và ấn nút “xóa” |
| Đầu ra | Xóa thành công tag khỏi hệ thống |

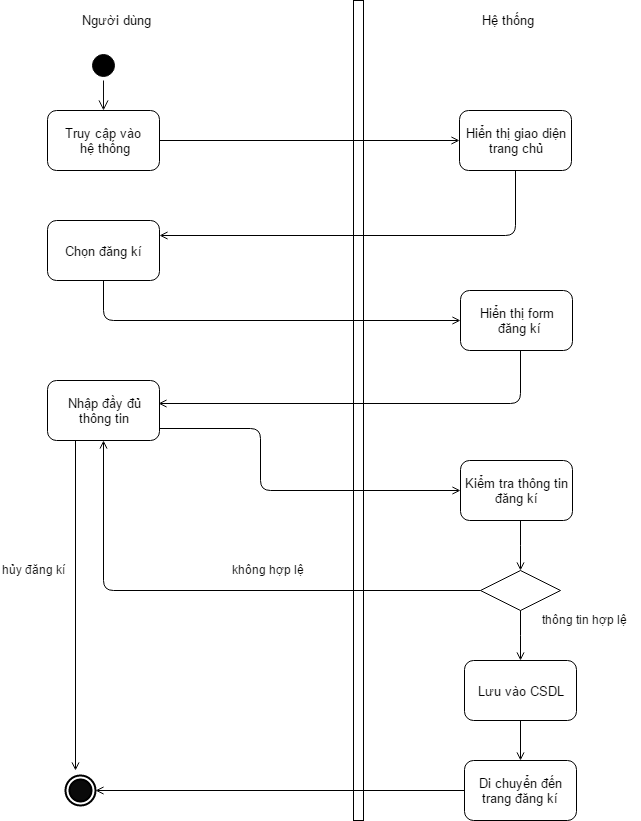
##### 2.2.3.3. Quản lý báo cáo

|  |  |
| --- | --- |
| Tên Use Case | Quản lý báo cáo |
| Tác nhân chính | Người quản trị |
| Mô tả tóm tăt | Người quản trị truy cập vào hệ thống sau đó chọn quản lý tag và có thể thay đổi trạng thái của báo cáo |
| Đầu vào | Đã đăng nhập vào hệ thống |
| Kịch bản chính | * Sau khi đăng nhập vào hệ thống người quản trị chọn “baocao” trên thanh công cụ của trang web * Một danh sách các báo cáo sẽ được hiện lên. * Người quản trị sẽ xem chi tiết câu hỏi bị báo cáo * Bài viết vi phạm thì người quản trị sẽ chọn nút “xóa” để tiến hành xóa câu hỏi. Nếu không thì người quản trị quay trở lại trang “báo cáo” và tiến hành ấn nút “cancel” |
| Đầu ra | Bài đăng vi phạm sẽ bị xóa khỏi hệ thống hoặc bài báo cáo sẽ được điều chỉnh trạng thái thành “đã xử lý”. |

### 2.3. Biểu đồ hoạt động

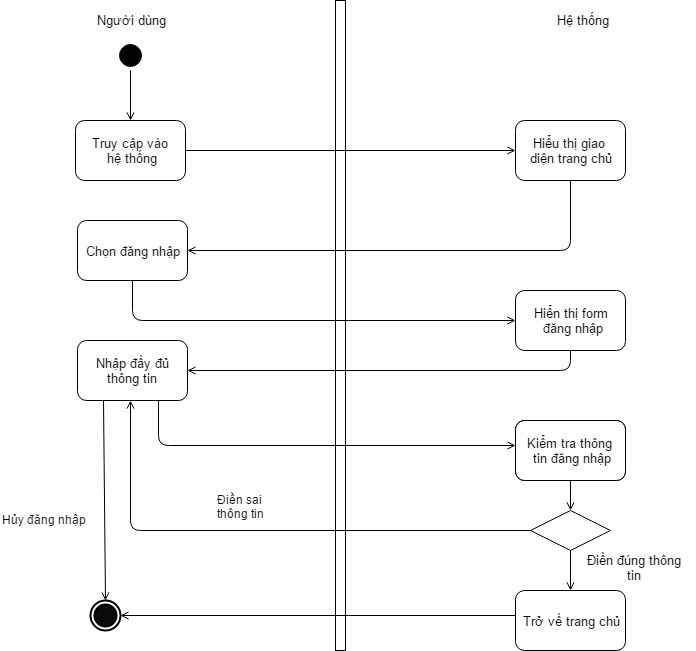
#### 2.3.1. Biểu đồ hoạt động người dùng

##### 2.3.1.1 Đăng ký

**

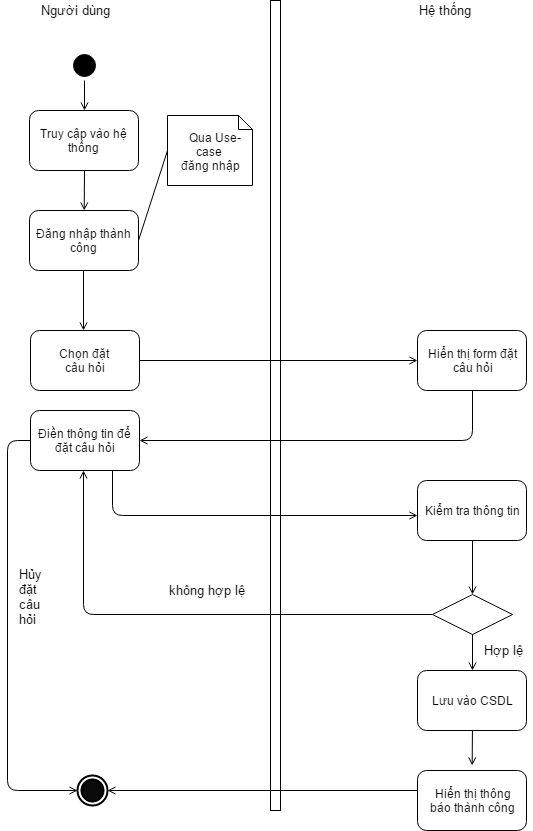
Hình 10: Sơ đồ hoạt động đăng kí

##### 2.3.1.2 Đăng nhập

**

Hình 11: Sơ đồ hoạt động đăng nhập

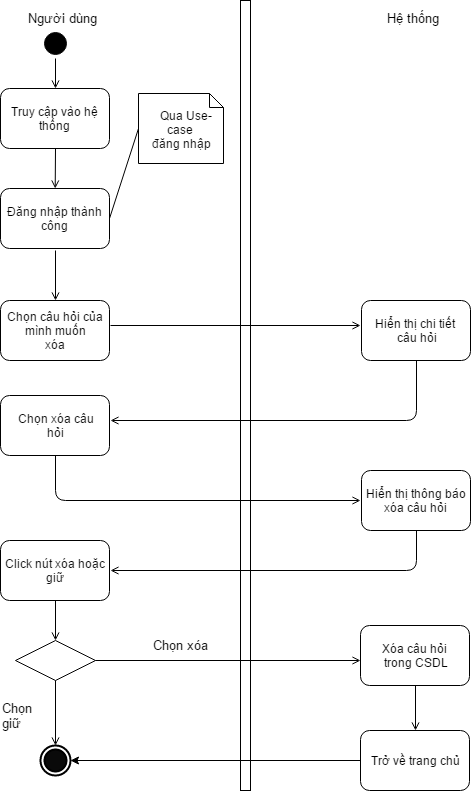
##### 2.3.1.3 Đặt câu hỏi

**

Hình 12: Sơ đồ hoạt động đặt câu hỏi

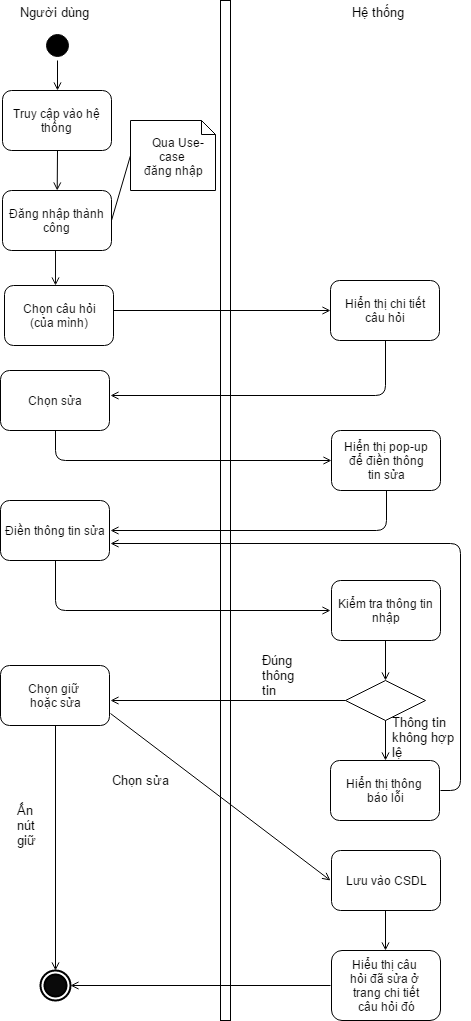
##### 2.3.1.4 Quản lý câu hỏi cá nhân

* Xóa câu hỏi cá nhân

**

Hình 13: Sơ đồ hoạt động xóa câu hỏi cá nhân

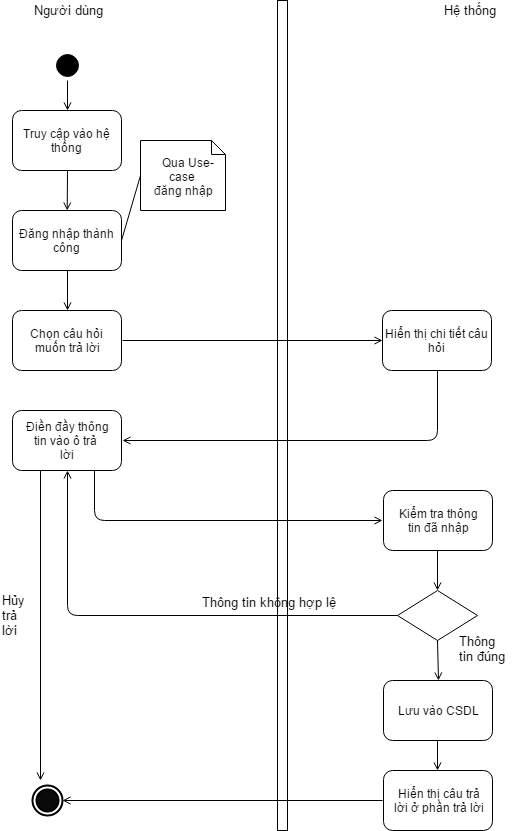
* Sửa câu hỏi cá nhân

**

Hình 14: Sơ đồ hoạt động sửa câu hỏi cá nhân

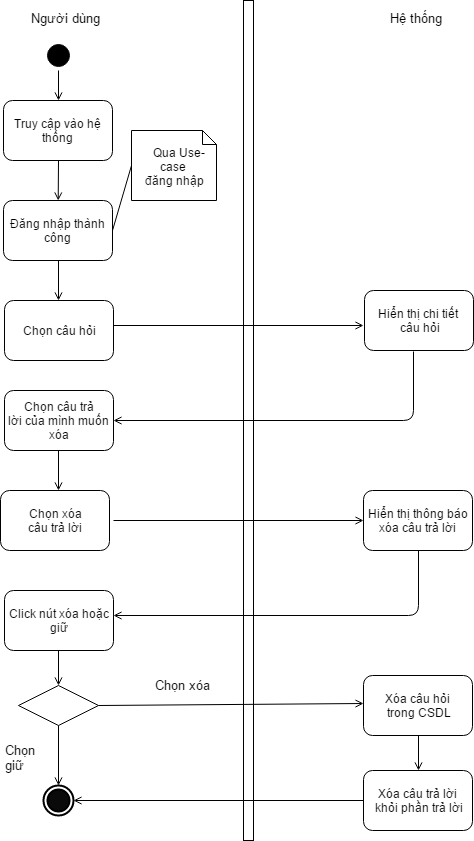
##### 2.3.1.5 Quản lý câu trả lời cá nhân

* Thêm mới câu trả lời

**

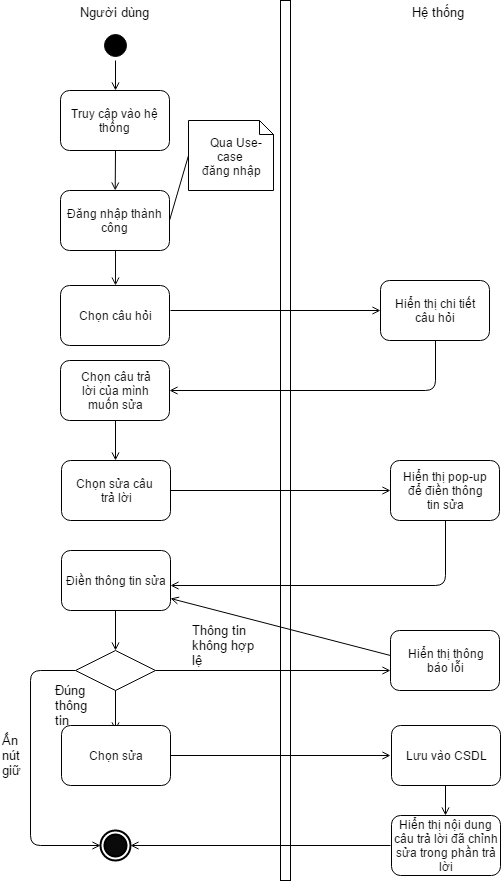
Hình 15: Sơ đồ hoạt động thêm mới câu trả lời

* Xóa câu trả lời cá nhân

**

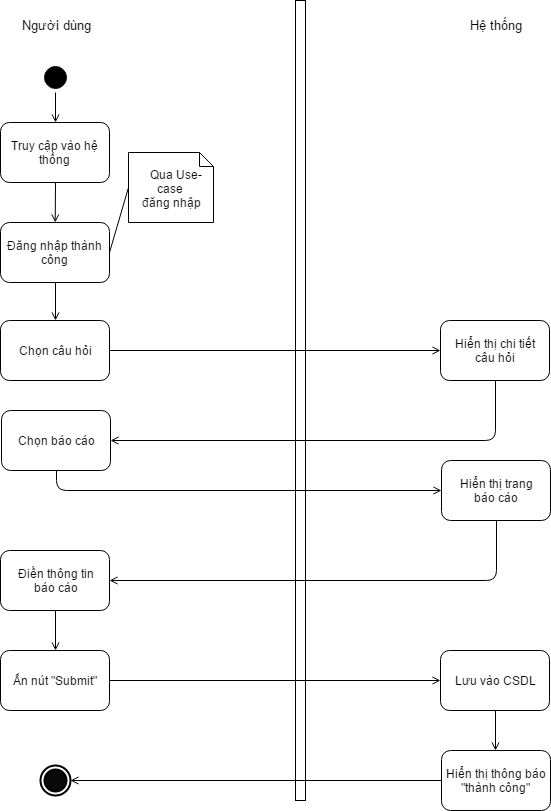
Hình 16: Sơ đồ hoạt động xóa câu trả lời cá nhân

* Sửa câu trả lời cá nhân

**

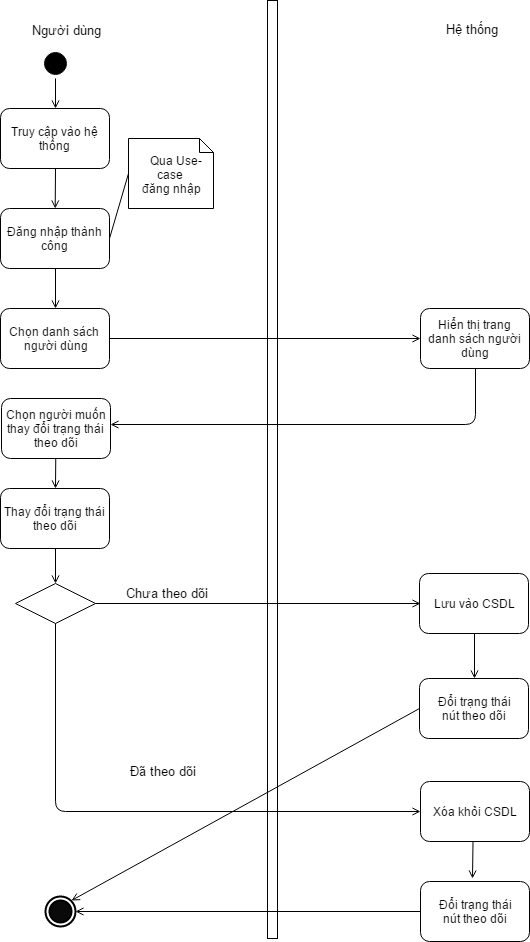
Hình 17: Sơ đồ hoạt động sửa câu trả lời cá nhân

##### 2.3.1.6 Báo cáo

**

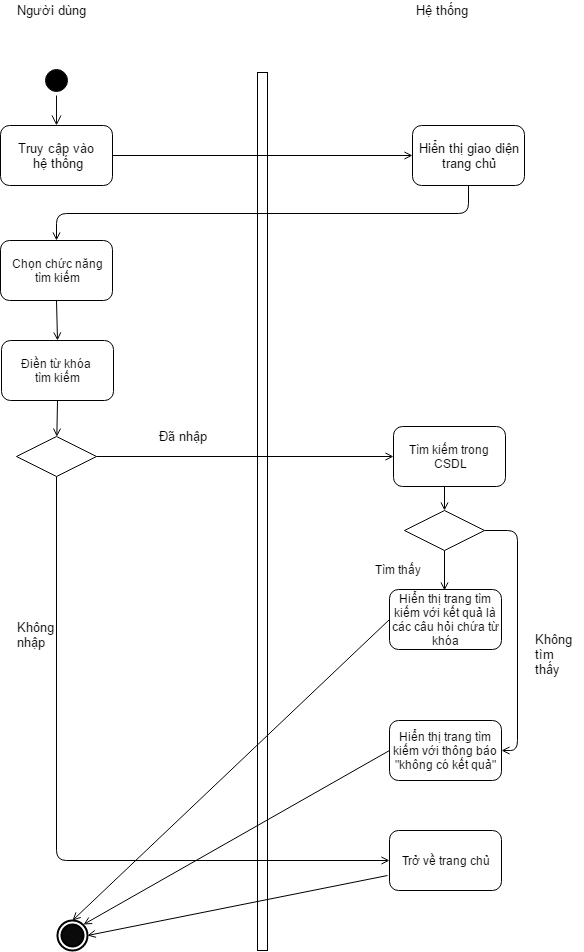
Hình 18: Sơ đồ hoạt động báo cáo câu hỏi

##### 2.3.1.7 Theo dõi người dùng

**

Hình 19: Sơ đồ hoạt động báo cáo câu hỏi

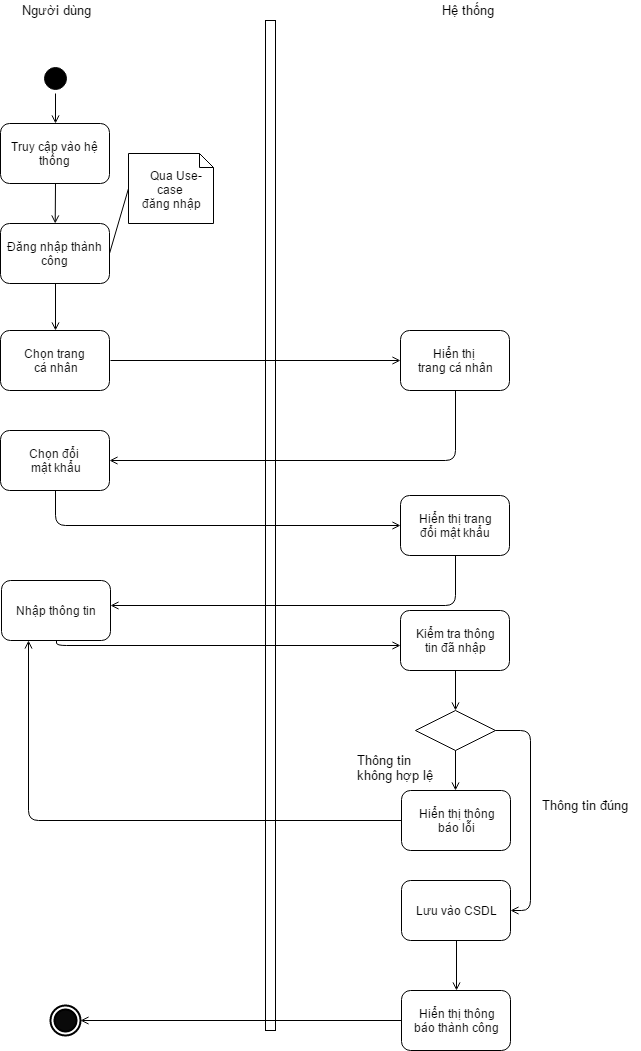
##### 2.3.1.8. Tìm kiếm

**

Hình 20: Sơ đồ hoạt động tìm kiếm

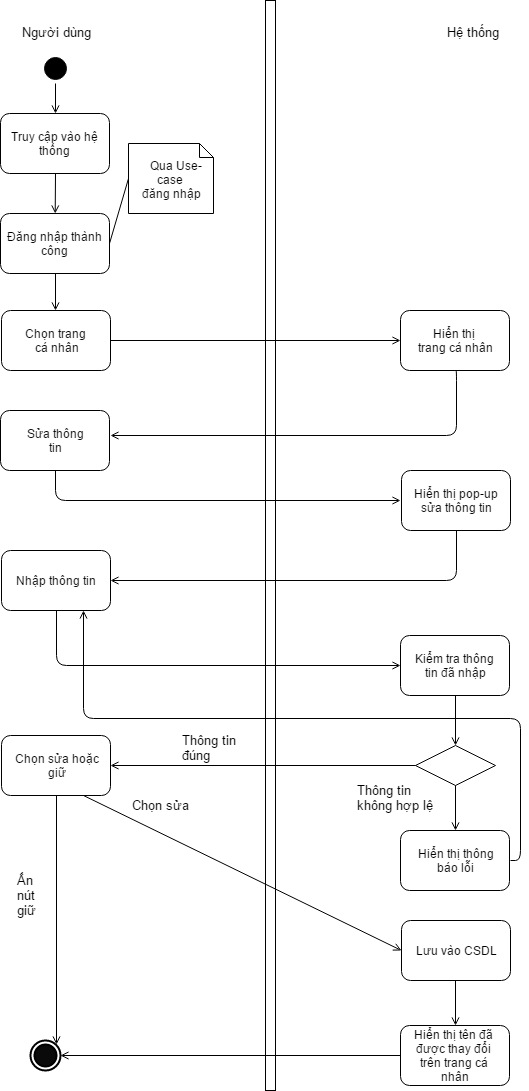
##### 2.3.1.9 Quản lý tài khoản

* Đổi mật khẩu

**

Hình 21: Sơ đồ hoạt động tìm kiếm

* Đổi tên người dùng

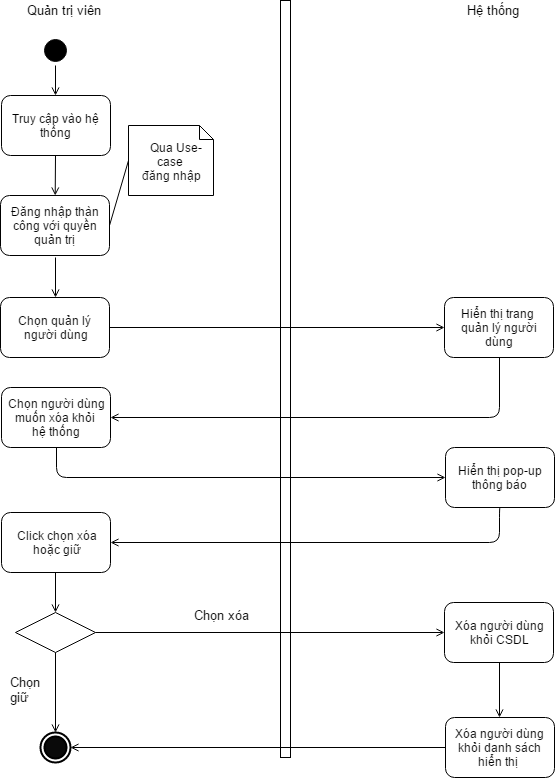
**

Hình 22: Sơ đồ hoạt động đổi tên người dùng

#### 2.3.2. Biểu đồ hoạt động quản trị

##### 2.3.2.1 Quản lý người dùng

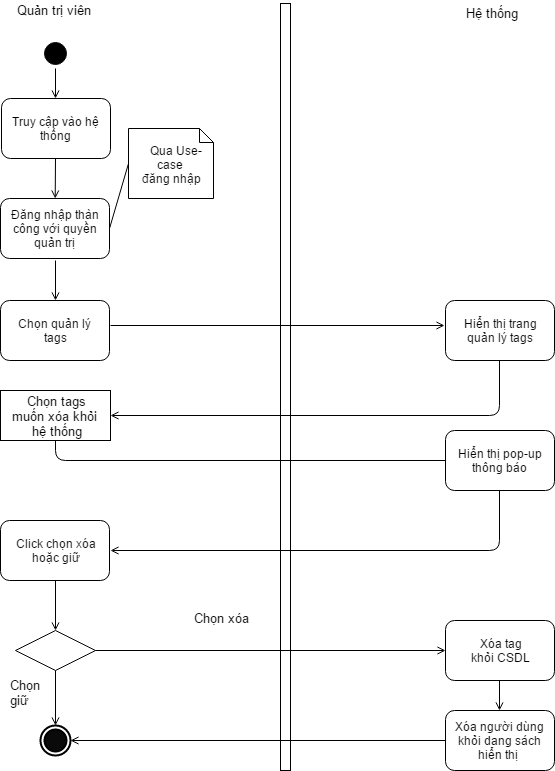
* Xóa người dùng



Hình 23: Sơ đồ hoạt động xóa người dùng

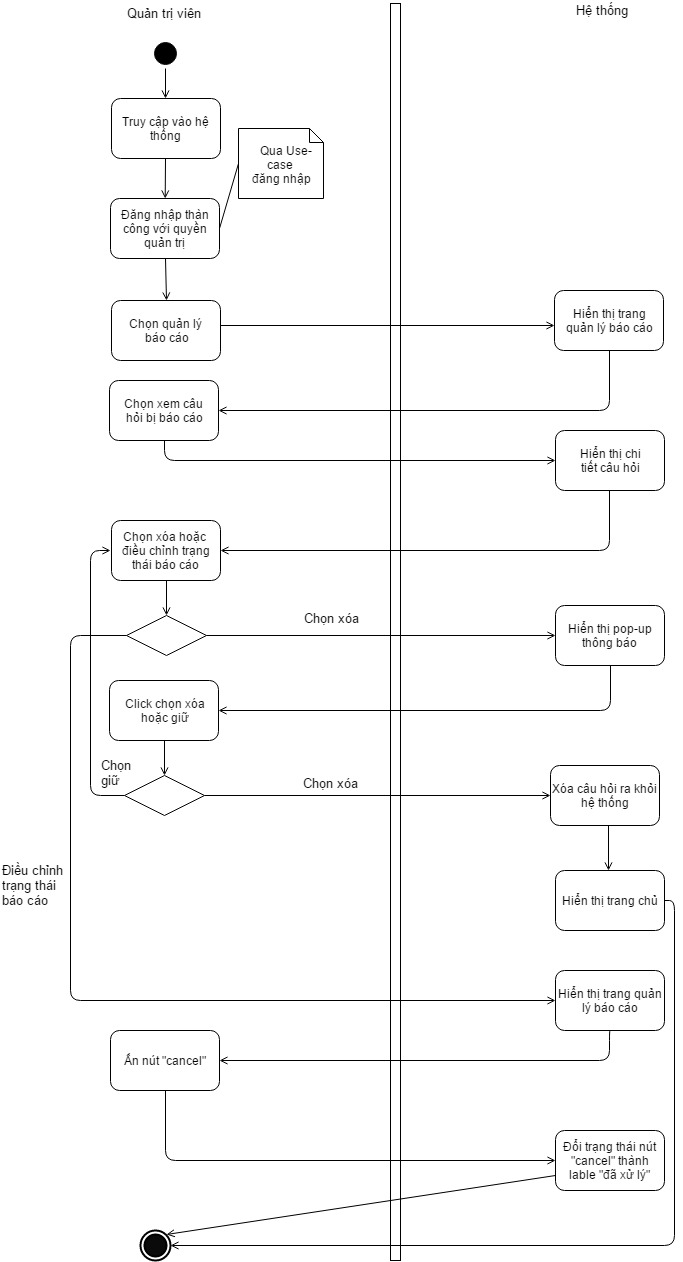
##### 2.3.2.2 Quản lý tags

* Xóa tags

**

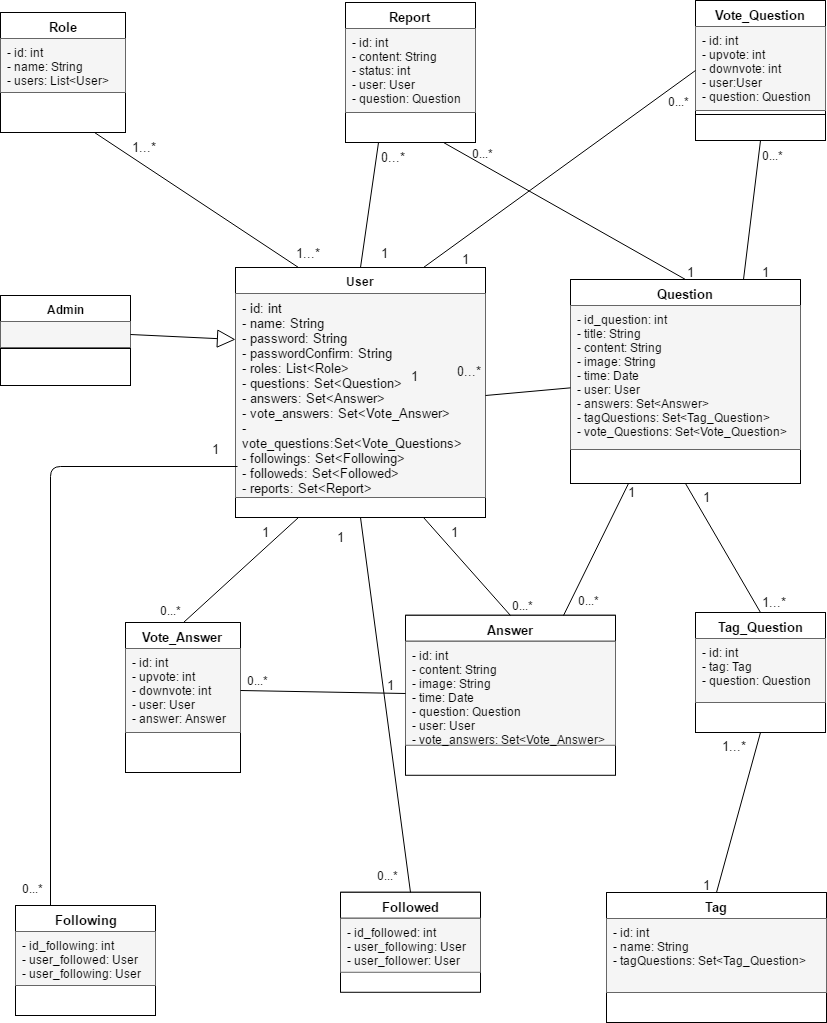
Hình 24: Sơ đồ hoạt động xóa tags

##### 2.3.2.3 Quản lý báo cáo

**

Hình 25: Sơ đồ hoạt động quản lý báo cáo

### 2.4. Biểu đồ lớp

****

Hình 26: Sơ đồ lớp

### 2.5. Thiết kế cơ sở dữ liệu

Cơ sở dữ liệu được thiết kế có 12 bảng:

Bảng 1. Bảng mô tả cơ sở dữ liệu cho user

|  |  |  |  |
| --- | --- | --- | --- |
| **Trường** | **Kiểu dữ liệu** | **Ghi chú** | **Mô tả** |
| Id\_user | Int(11) | Primary key, tăng tự động | Mã người dùng |
| Name | Varchar(50) |  | Tên người dùng |
| Password | Varchar(100) |  | Mật khẩu người dùng |
| Email | Varchar(50) |  | Địa chỉ email của người dùng |
| image | Varchar(50) |  | ảnh đại diện của người dùng |

Bảng 2.Bảng mô tả cơ sở dữ liệu choquestions

|  |  |  |  |
| --- | --- | --- | --- |
| **Trường** | **Kiểu dữ liệu** | **Ghi chú** | **Mô tả** |
| Id\_question | Int(11) | Primary key, tăng tự động | Mã câu hỏi |
| title | Varchar(100) |  | Tiêu đề câu hỏi |
| content | Varchar(500) |  | Nội dung câu hỏi |
| image | Varchar(50) | Có thể null | ảnh của câu hỏi |
| id\_user | Int(11) | Foreign Key | Người đặt câu hỏi |
| time | datetime |  | Thời gian đặt câu hỏi |

Bảng 3. Bảng mô tả cơ sở dữ liệu cho answers

|  |  |  |  |
| --- | --- | --- | --- |
| **Trường** | **Kiểu dữ liệu** | **Ghi chú** | **Mô tả** |
| Id\_answer | Int(11) | Primary key, tăng tự động | Mã câu trả lời |
| content | Varchar(500) |  | Nội dung câu trả lời |
| image | Varchar(50) | Có thể null | Ảnh câu trả lời |
| Id\_question | Int(11) | Foreign Key | Mã câu hỏi trả lời |
| id\_user | Int(11) | Foreign Key | Mã người trả lời |
| time | datetime |  | Thời gian trả lời |

Bảng 4. Bảng mô tả cơ sở dữ liệu cho followed

|  |  |  |  |
| --- | --- | --- | --- |
| **Trường** | **Kiểu dữ liệu** | **Ghi chú** | **Mô tả** |
| Id\_followed | Int(11) | Primary key, tăng tự động | Mã followed |
| User\_id | Int(11) | Foreign Key | Người theo dõi |
| Followed\_id | Int(11) | Foreign Key | Người được theo dõi |

Bảng 5.Bảng mô tả cơ sở dữ liệu cho following

|  |  |  |  |
| --- | --- | --- | --- |
| **Trường** | **Kiểu dữ liệu** | **Ghi chú** | **Mô tả** |
| Id\_following | Int(11) | Primary key, tăng tự động | Mã following |
| User\_id | Int(11) | Foreign Key | Người theo dõi |
| Following\_id | Int(11) | Foreign Key | Người được theo dõi |

Bảng 6.Bảng mô tả cơ sở dữ liệu choreport

|  |  |  |  |
| --- | --- | --- | --- |
| **Trường** | **Kiểu dữ liệu** | **Ghi chú** | **Mô tả** |
| Id\_report | Int(11) | Primary key, tăng tự động | Mã báo cáo |
| User\_id | Int(11) | Foreign Key | Mã người dùng báo cáo |
| Question\_id | Int(11) | Foreign Key | Mã câu hỏi bị báo cáo |
| content | Varchar(100) |  | Nội dung báo cáo |
| status | Int(11) |  | Trạng thái báo cáo |

Bảng 7.Bảng mô tả cơ sở dữ liệu chorole

|  |  |  |  |
| --- | --- | --- | --- |
| **Trường** | **Kiểu dữ liệu** | **Ghi chú** | **Mô tả** |
| Id | Int(11) | Primary key, tăng tự động | Mã vai trò |
| name | Varchar(45) |  | Tên vai trò |

Bảng 8. Bảng mô tả cơ sở dữ liệu cho tags

|  |  |  |  |
| --- | --- | --- | --- |
| **Trường** | **Kiểu dữ liệu** | **Ghi chú** | **Mô tả** |
| Id\_tag | Int(11) | Primary key, tăng tự động | Mã tag |
| name | Varchar(50) |  | Tên tag |

Bảng 9. Bảng mô tả cơ sở dữ liệu cho tag\_question

|  |  |  |  |
| --- | --- | --- | --- |
| **Trường** | **Kiểu dữ liệu** | **Ghi chú** | **Mô tả** |
| Id\_tag\_question | Int(11) | Primary key, tăng tự động | Mã tag\_question |
| Id\_question | Int(11) | Foreign Key | Mã câu hỏi |
| Id\_tag | Int(11) | Foreign Key | Mã tag |

Bảng 10.Bảng mô tả cơ sở dữ liệu cho user\_role

|  |  |  |  |
| --- | --- | --- | --- |
| **Trường** | **Kiểu dữ liệu** | **Ghi chú** | **Mô tả** |
| User\_id | Int(11) | primary key | Mã người dùng |
| Role\_id | Int(11) | primary key | Mã vai trò |

Bảng 11.Bảng mô tả cơ sở dữ liệu cho vote\_answers

|  |  |  |  |
| --- | --- | --- | --- |
| **Trường** | **Kiểu dữ liệu** | **Ghi chú** | **Mô tả** |
| Id\_vote\_answer | Int(11) | Primary key, tăng tự động | Mã bình chọn câu trả lời |
| Id\_user | Int(11) | Foreign Key | Mã người bình chọn |
| Id\_answer | Int(11) | Foreign Key | Mã câu trả lời bình chọn |
| upvote | Tinyint(1) | Mặc định là 0 | Trạng thái upvote |
| downvote | Tinyint(1) | Mặc định là 0 | Trạng thái downvote |

Bảng 12**.** Bảng mô tả cơ sở dữ liệu chovote\_questions

|  |  |  |  |
| --- | --- | --- | --- |
| **Trường** | **Kiểu dữ liệu** | **Ghi chú** | **Mô tả** |
| Id\_vote\_question | Int(11) |  | Mã bình chọn câu hỏi |
| Id\_user | Int(11) | Foreign Key | Mã người bình chọn |
| Id\_question | Int(11) | Foreign Key | Mã câu hỏi được bình chọn |
| upvote | Tinyint(1) | Mặc định là 0 | Trạng thái upvote |
| downvote | Tinyint(1) | Mặc định là 0 | Trạng thái downvote |

# Chương 3: KIỂM THỬ TỰ ĐỘNG

## 1. KHÁI NIỆM VỀ SELENIUM

Selenium là một tập hợp mạnh mẽ của các công cụ hỗ trợ phát triển nhanh chóng của các thử nghiệm tự động hóa cho các ứng dụng dựa trên web. Selenium cung cấp một tập hợp các thử nghiệm chức năng đặc biệt hướng đến các nhu cầu của các thử nghiệm của một ứng dụng web. Các hoạt động này là rất linh hoạt, cho phép nhiều tùy chọn vị trí các thành phần UI và so sánh kết quả thử nghiệm dự kiến.

* Phân loại Selenium:
  + Selenium IDE
  + Selenium Webdriver (Selenium 2)
  + Selenium Gird
  + Selenium RC
* Những đặc điểm nổi bật của Selenium:
* Selenium là một công cụ mã nguồn mở
* Chạy trên các trình duyệt web hiện đại như Firefox, IE, Chrome,..
* Kiểm thử có thể xuất ra hầu hết các ngôn ngữ như HTML, Java, Dot Net, … để tạo kịch bản kiêm thử (testcase) kết hợp với việc sử dụng vòng lặp, các điều kiện… khiến cho script trở nên chính xác hơn.
* Selenium Webdriver được phát triển tốt hơn để hỗ trợ cho các trang web động
* Mục đích của Webdriver là hỗ trợ cho các vấn đề kiểm thử web-app hiện nay.
* Có thể thực hiện các kịch bản trên các hệ điều hành khác nhau
* Selenium có sự hỗ trợ của một số các nhà cung cấp trình duyệt lớn.

Trong đồ án này chọn 2 loại selenium được sử dụng nhiều nhất cho kiểm thử website là: Selenium IDE và Selenium Webdriver (Selenium 2) để giới thiệu một cách chi tiết

## 2. TÌM HIỂU VỀ SELENIUM

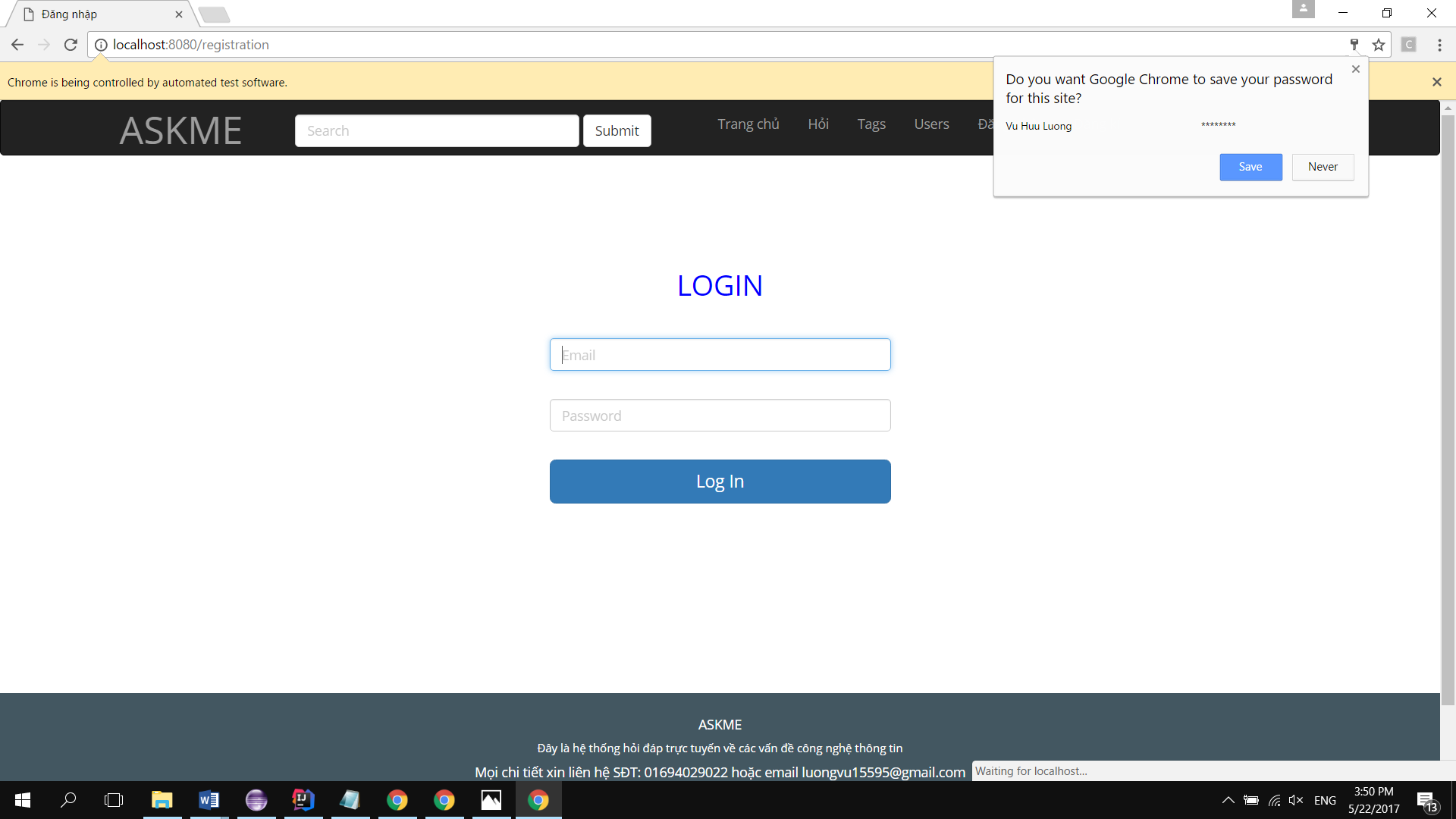
* Selenium Webdriver kế thừa và phát triển từ Selenium IDE, Selenium Remote Control (RC), Seleniu Gird.
* Selenium Webdriver làm việc trực tiếp với trình duyệt ở mức độ điều hành
* Với selenium webdriver chúng ta không thể tương tác với các đối tượng UI ẩn / vô hiệu hóa trong mã nguồn web.
* Thiết kế dễ hiểu, chương trình test dễ thực hiện
* Selenium được phát triển tốt hơn để hỗ trợ các trang web động
* Selenium Webdrier có tính năng tương tác gần với thực tế hơn vì nó thực hiện các hành động trực tiếp với trình duyệt giống như các end- user trong thực tế.
* Tốc độ thực thi và nhận phản hồi từ trình duyệt khi sử dụng selenium webdriver sẽ nhanh hơn vì nó thực hiện mọi hành động trực tiếp với trình duyệt.

## 3. KIỂM THỬ CHỨC NĂNG BẰNG SELENIUM

### 3.1. Chức năng đăng kí

|  |  |  |  |
| --- | --- | --- | --- |
| **Mã test case** | **Tên chức năng** | **Các bước thực hiện** | **Kết quả thực hiện** |
| TC1 | Đăng kí tài khoản | 1. Vào link: <http://localhost:8080/> 2. Click vào link đăng kí trên thanh công cụ 3. Input email nhập vào [123@gmail.com](mailto:123@gmail.com) 4. Input họ và tên nhập vào “Vu Huu Luong” 5. Input password nhập vào “12345678” 6. Input nhập lại password nhập vào “12345678” 7. Ấn nút đăng kí | Đăng kí thành công tài khoản và chuyển về trang đăng nhập |

* Kết quả sau khi thực hiện:

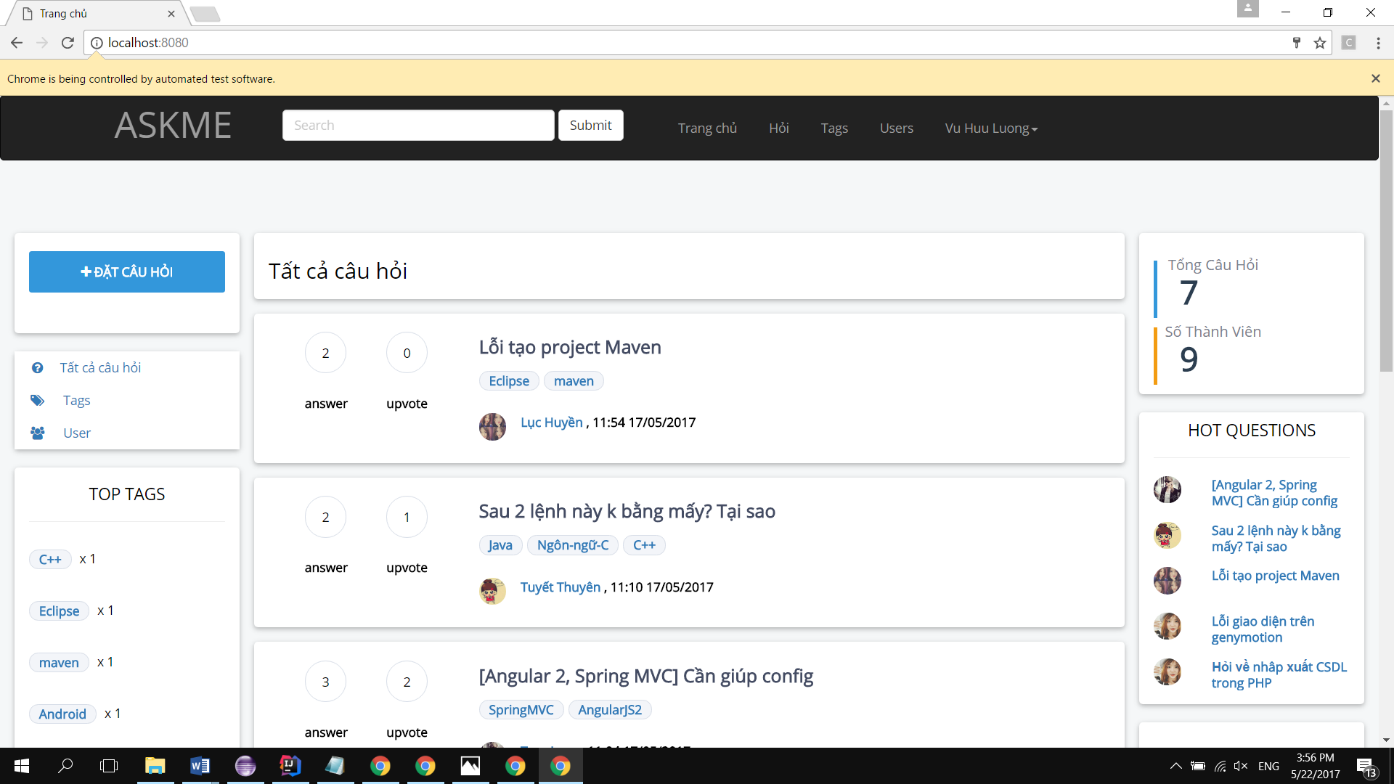


Hình 27: Kiểm thử chức năng đăng kí

### 3.2. Chức năng đăng nhập

|  |  |  |  |
| --- | --- | --- | --- |
| **Mã test case** | **Tên chức năng** | **Các bước thực hiện** | **Kết quả thực hiện** |
| TC2 | Đăng nhập vào hệ thống | 1. Vào link: <http://localhost:8080/>. 2. Click vào link đăng nhập trên thanh công cụ 3. Input email nhập vào [123@gmail.com](mailto:123@gmail.com). 4. Input password nhập vào “12345678”. 5. Rồi ấn login. | Đăng kí nhập thành công và chuyển về trang chủ với trạng thái đã đăng nhập |

* Kết quả sau khi thực hiện:

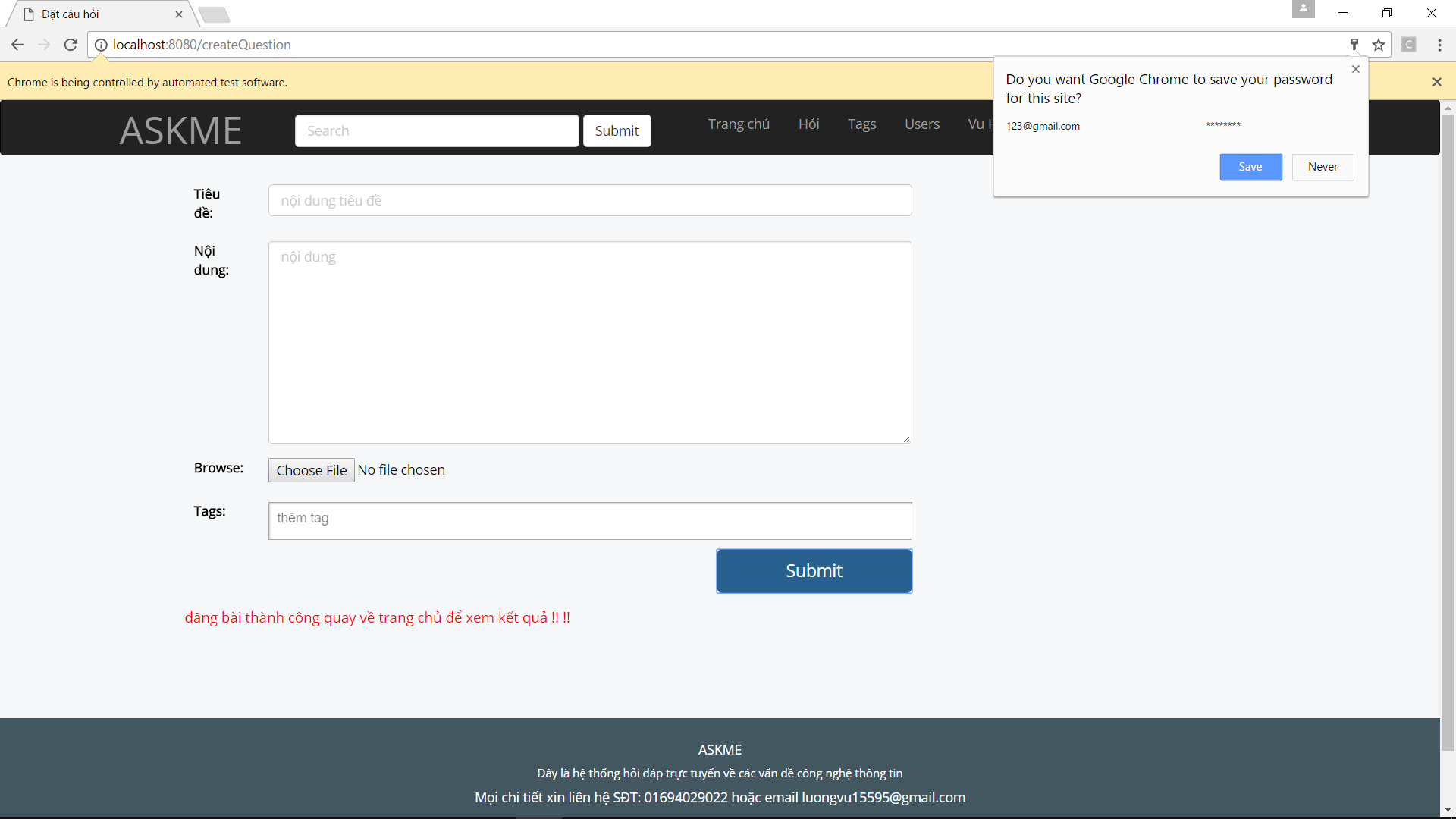


Hình 28: Kiểm thử chức năng đăng nhập

### 3.3. Chức năng đặt câu hỏi

|  |  |  |  |
| --- | --- | --- | --- |
| **Mã test case** | **Tên chức năng** | **Các bước thực hiện** | **Kết quả thực hiện** |
| TC3 | Đặt câu hỏi | 1. Vào link: <http://localhost:8080/> 2. Click vào link hỏi trên thanh công cụ 3. Input email nhập vào [123@gmail.com](mailto:123@gmail.com) 4. Input password nhập vào “12345678” 5. Rồi ấn login 6. Input tiêu đề nhập vào “Đây là tiêu đề câu hỏi” 7. Input nội dung nhập vào “Đây là nội dung câu hỏi” 8. Input tag nhập vào “Tags” 9. Rồi ấn submit | Hiển thị thông báo đăng bài thành công. |

* Kết quả sau khi thực hiện:

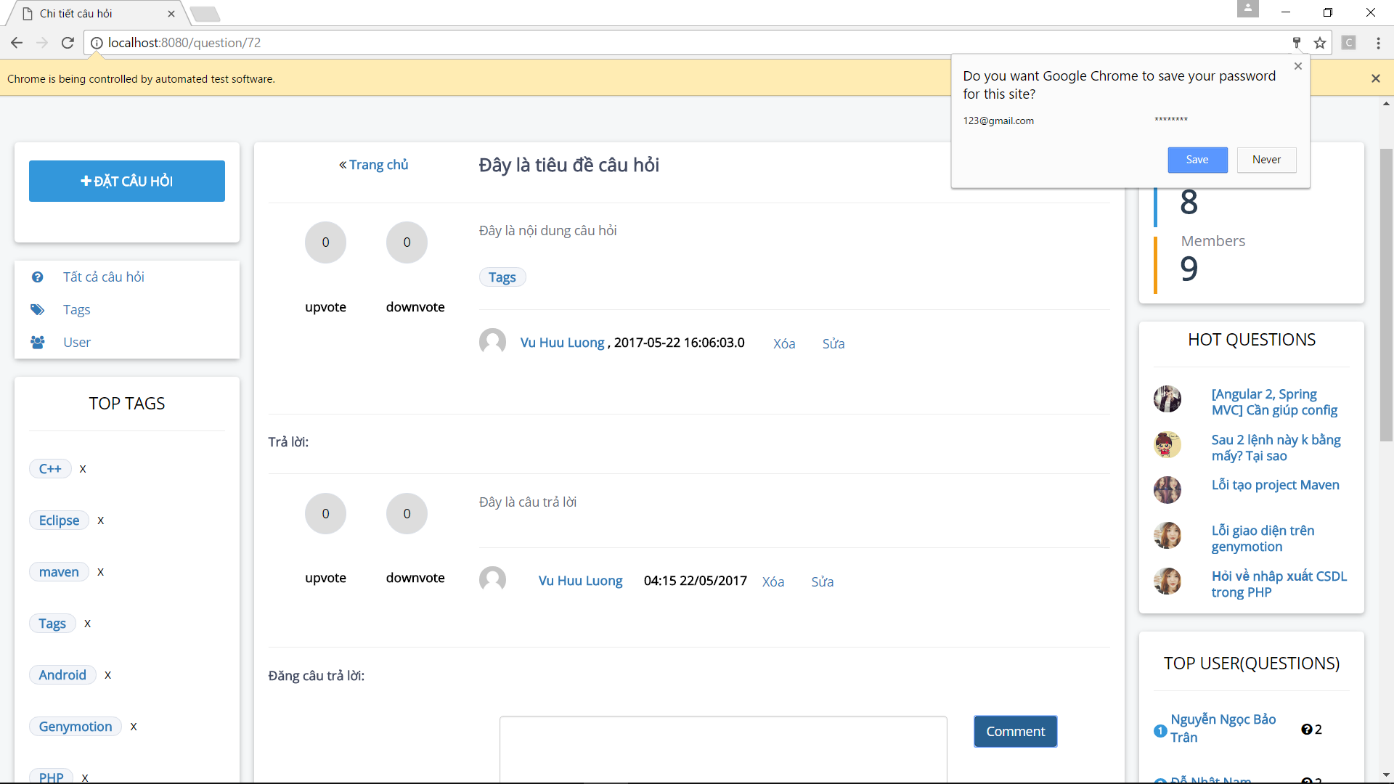


Hình 29: Kiểm thử chức năng đặt câu hỏi

### 3.4. Chức năng trả lời câu hỏi

|  |  |  |  |
| --- | --- | --- | --- |
| **Mã test case** | **Tên chức năng** | **Các bước thực hiện** | **Kết quả thực hiện** |
| TC4 | Trả lời câu hỏi | 1. Vào link: <http://localhost:8080/> 2. Click vào link đăng nhập trên thanh công cụ 3. Input email nhập vào [123@gmail.com](mailto:123@gmail.com) 4. Input password nhập vào “12345678” 5. Rồi ấn login 6. Click vào câu hỏi có tiêu đề “Đây là tiêu đề câu hỏi” 7. Input trả lời nhập vào “Đây là câu trả lời” 8. Rồi ấn submit | Hiển thị câu trả lời ngay phía trên. |

* Kết quả sau khi thực hiện:

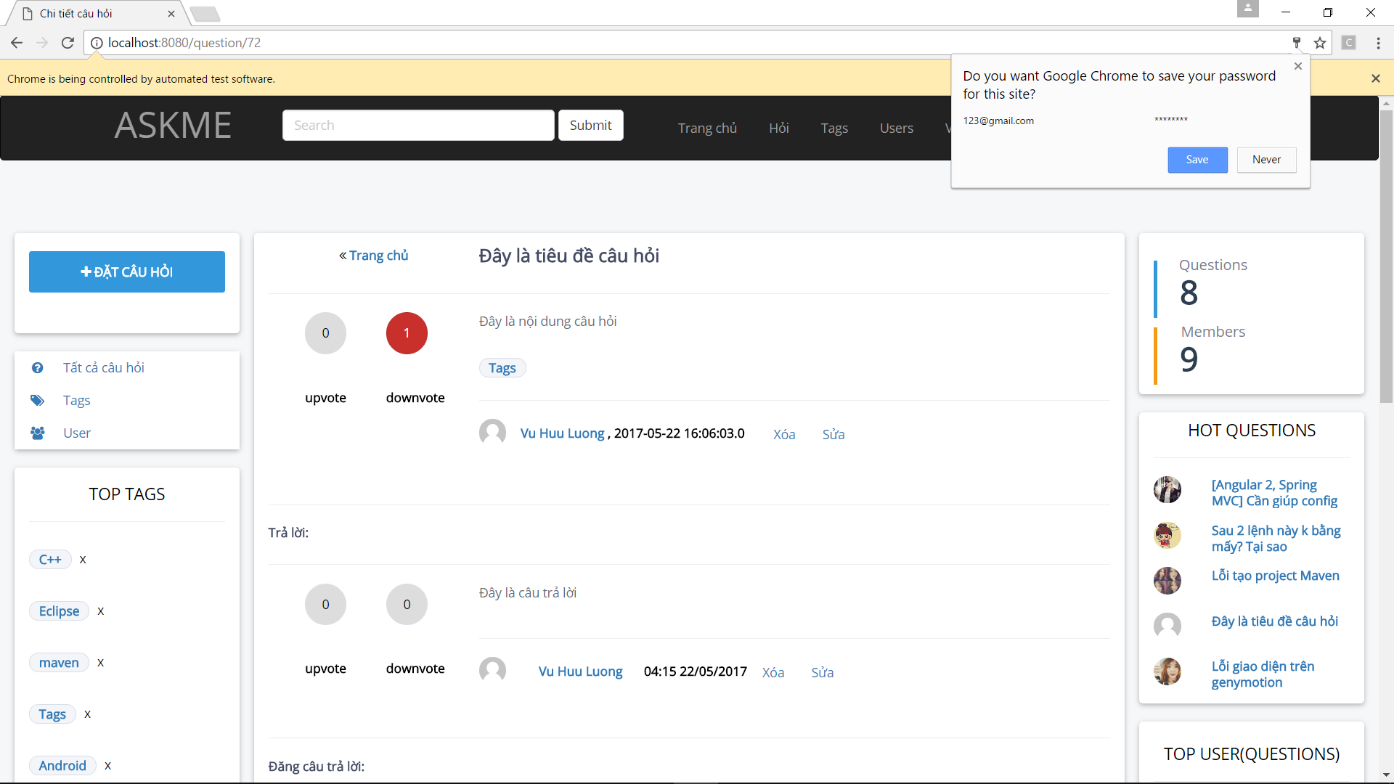


Hình 30: Kiểm thử chức năng trả lời câu hỏi

### 3.5. Chức năng bình chọn câu hỏi

|  |  |  |  |
| --- | --- | --- | --- |
| **Mã test case** | **Tên chức năng** | **Các bước thực hiện** | **Kết quả thực hiện** |
| TC5 | Bình chọn câu hỏi | 1. Vào link: <http://localhost:8080/> 2. Click vào link đăng nhập trên thanh công cụ 3. Input email nhập vào [123@gmail.com](mailto:123@gmail.com) 4. Input password nhập vào “12345678” 5. Rồi ấn login 6. Click vào câu hỏi có tiêu đề “Đây là tiêu đề câu hỏi” 7. ấn nút downvote | Trạng thái nút downvote sẽ chuyển sang màu đỏ |

* Kết quả sau khi thực hiện:

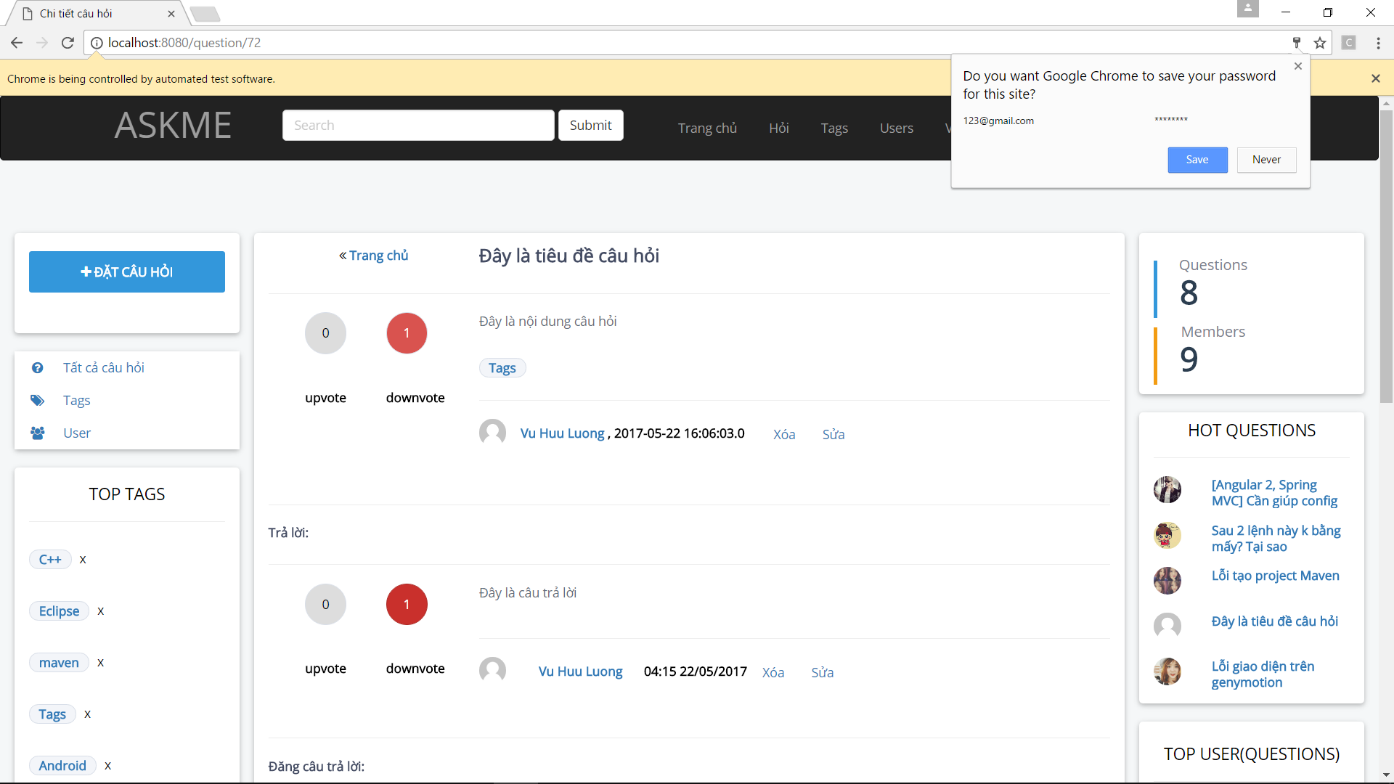
****

Hình 31: Kiểm thử chức năng bình chọn câu hỏi

### 3.6. Chức năng bình chọn câu trả lời

|  |  |  |  |
| --- | --- | --- | --- |
| **Mã test case** | **Tên chức năng** | **Các bước thực hiện** | **Kết quả thực hiện** |
| TC6 | Bình chọn câu trả lời | 1. Vào link: <http://localhost:8080/> 2. Click vào link đăng nhập trên thanh công cụ 3. Input email nhập vào [123@gmail.com](mailto:123@gmail.com) 4. Input password nhập vào “12345678” 5. Rồi ấn login 6. Click vào câu hỏi có tiêu đề “Đây là tiêu đề câu hỏi” 7. Chọn câu trả lời “Đây là câu trả lời” 8. Ấn nút downvote | Trạng thái nút downvote sẽ chuyển sang màu đỏ |

* Kết quả sau khi thực hiện:

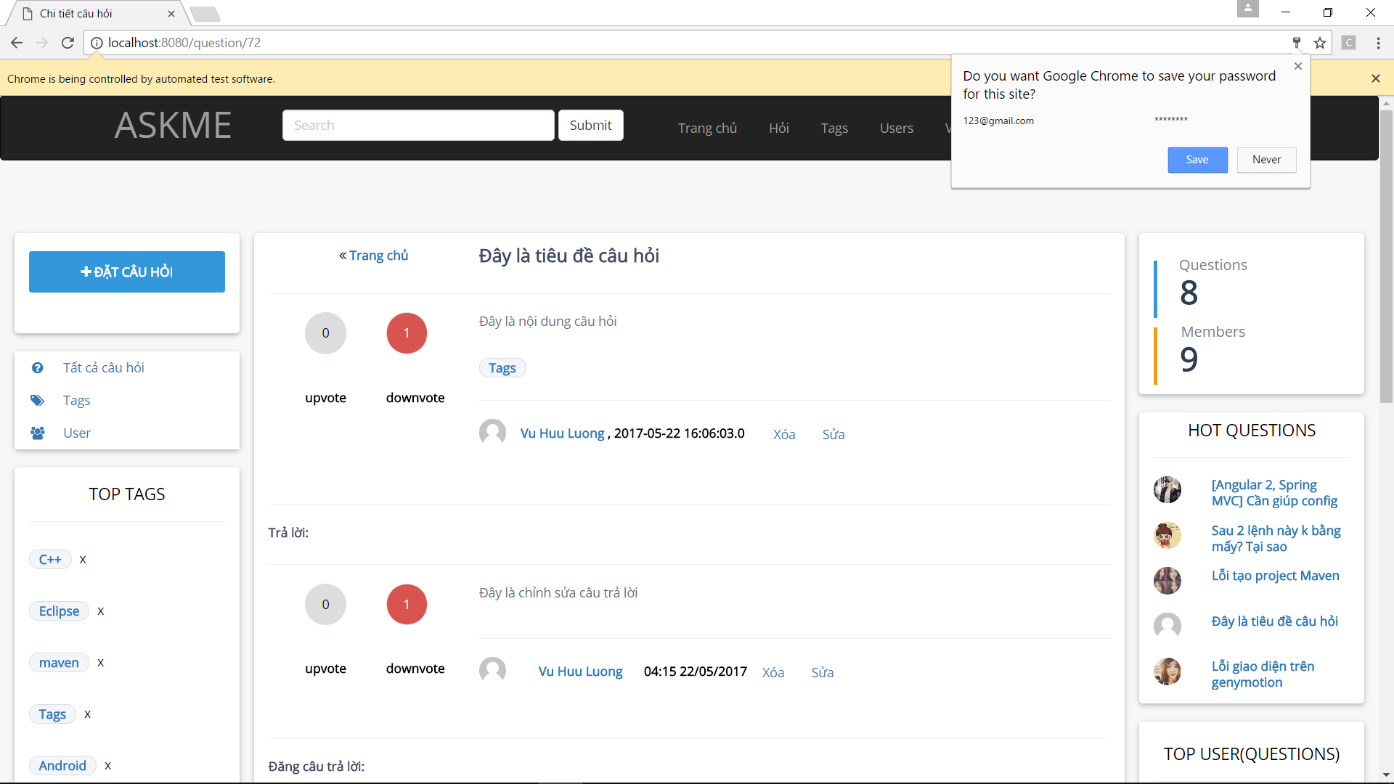
****

Hình 32: Kiểm thử chức năng bình chọn câu trả lời

### 3.7. Sửa câu trả lời

|  |  |  |  |
| --- | --- | --- | --- |
| **Mã test case** | **Tên chức năng** | **Các bước thực hiện** | **Kết quả thực hiện** |
| TC7 | Sửa câu trả lời | 1. Vào link: <http://localhost:8080/> 2. Click vào link đăng nhập trên thanh công cụ 3. Input email nhập vào [123@gmail.com](mailto:123@gmail.com) 4. Input password nhập vào “12345678” 5. Rồi ấn login 6. Click vào câu hỏi có tiêu đề “Đây là tiêu đề câu hỏi” 7. Chọn câu trả lời “Đây là câu trả lời” 8. Ấn nút sửa 9. Xóa nội dung câu trả lời 10. Nhập vào nội dung “Đây là chỉnh sửa câu trả lời” 11. Ấn nút sửa | Câu trả lời được sửa nội dung thành“Đây là chỉnh sửa câu trả lời” |

* Kết quả sau khi thực hiện:

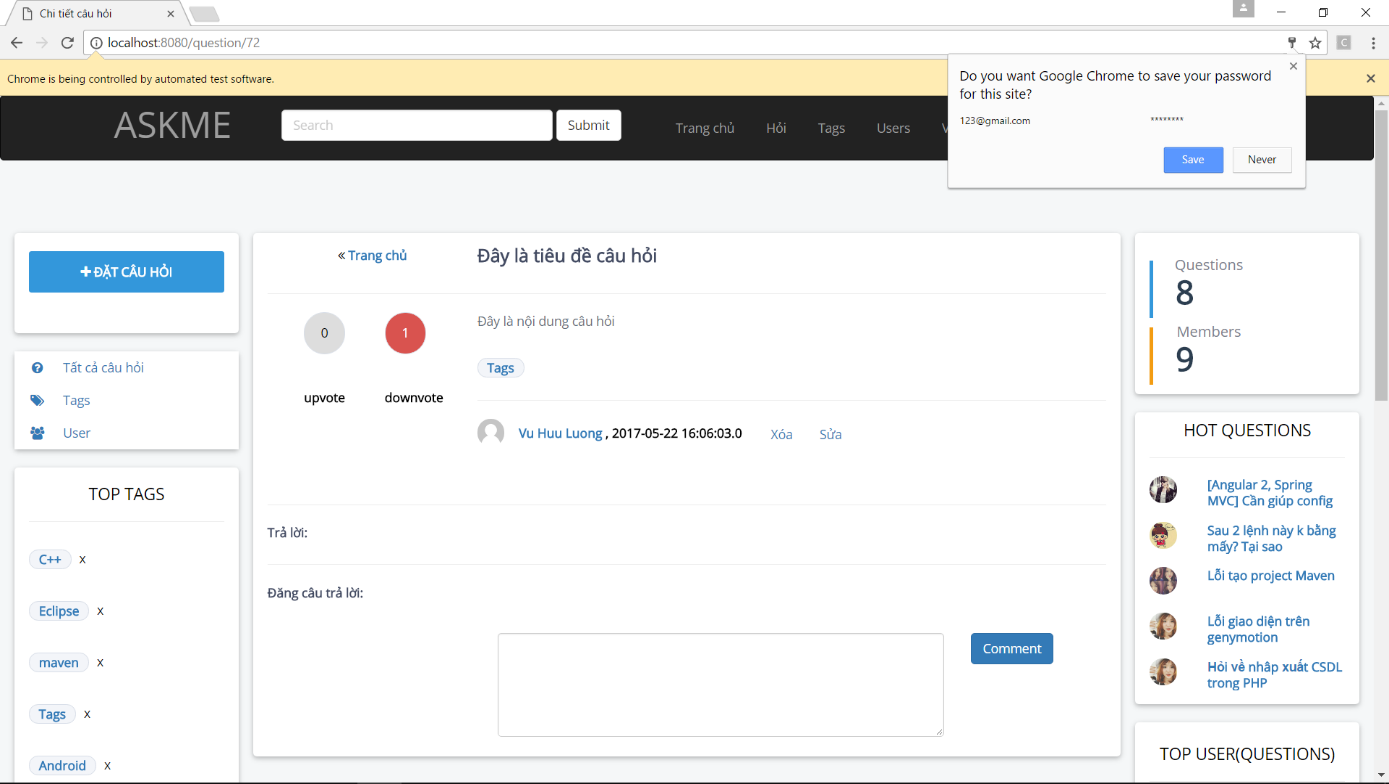


Hình 33: Kiểm thử chức năng sửa câu trả lời

### 3.8. Xóa câu trả lời

|  |  |  |  |
| --- | --- | --- | --- |
| **Mã test case** | **Tên chức năng** | **Các bước thực hiện** | **Kết quả thực hiện** |
| TC8 | Xóa câu trả lời | 1. Vào link: <http://localhost:8080/> 2. Click vào link đăng nhập trên thanh công cụ 3. Input email nhập vào [123@gmail.com](mailto:123@gmail.com) 4. Input password nhập vào “12345678” 5. Rồi ấn login 6. Click vào câu hỏi có tiêu đề “Đây là tiêu đề câu hỏi” 7. Chọn câu trả lời “Đây là chỉnh sửa câu trả lời” 8. Ấn nút xóa 9. Ấn tiếp nút xóa | Câu trả lời bị xóa khỏi hệ thống |

* Kết quả sau khi thực hiện:

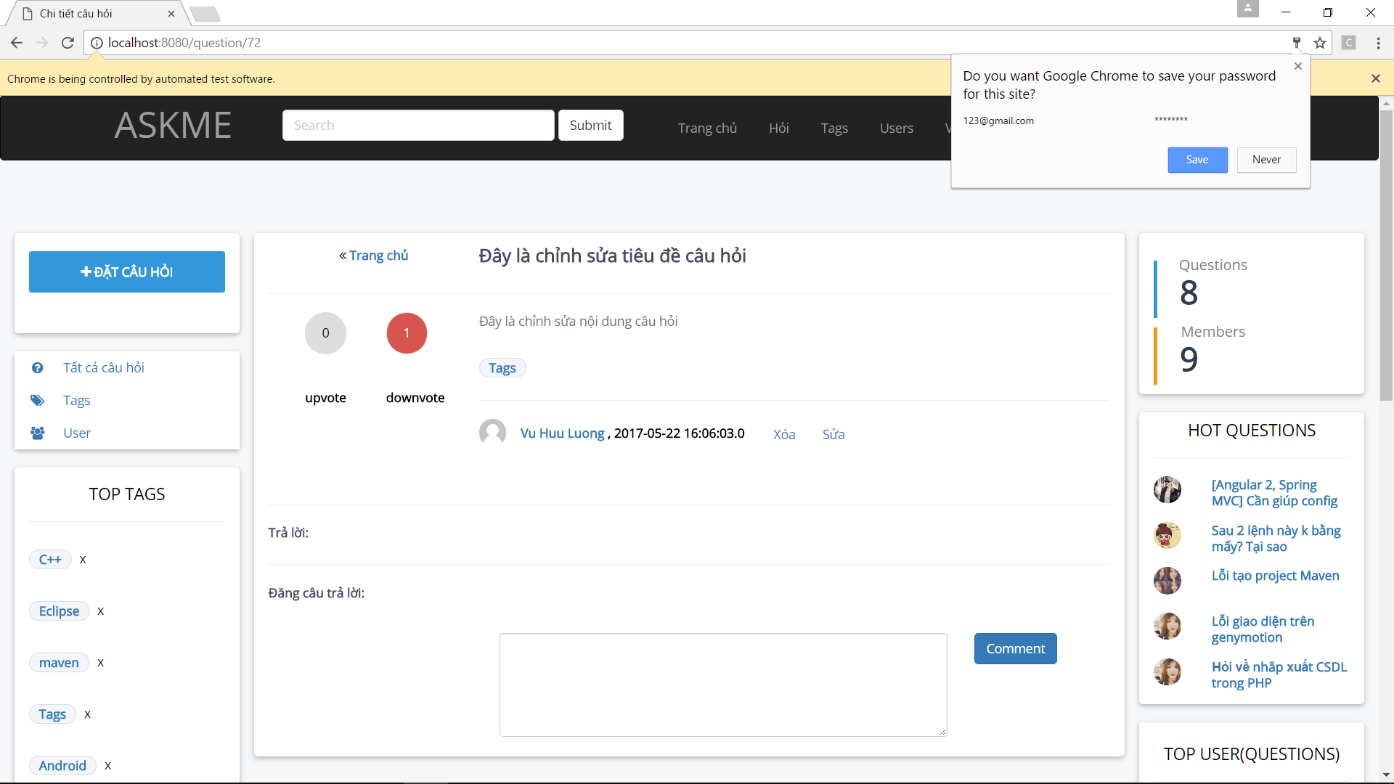


Hình 34: Kiểm thử chức năng xóa câu trả lởi

### 3.9. Sửa câu hỏi

|  |  |  |  |
| --- | --- | --- | --- |
| **Mã test case** | **Tên chức năng** | **Các bước thực hiện** | **Kết quả thực hiện** |
| TC9 | Sửa câu hỏi | 1. Vào link: <http://localhost:8080/> 2. Click vào link đăng nhập trên thanh công cụ 3. Input email nhập vào [123@gmail.com](mailto:123@gmail.com) 4. Input password nhập vào “12345678” 5. Rồi ấn login 6. Click vào câu hỏi có tiêu đề “Đây là tiêu đề câu hỏi” 7. Ấn nút sửa 8. Xóa nội tiêu đề 9. Nhập vào nội dung “Đây là chỉnh sửa tiêu đề câu hỏi” 10. Xóa ô nhập nội dung 11. Nhập vào Input nội dung “Đây là chỉnh sửa nội dung câu hỏi” 12. Ấn sửa | Tiêu đề và nội dung của câu hỏi đã bị thay đổi |

* Kết quả sau khi thực hiện:

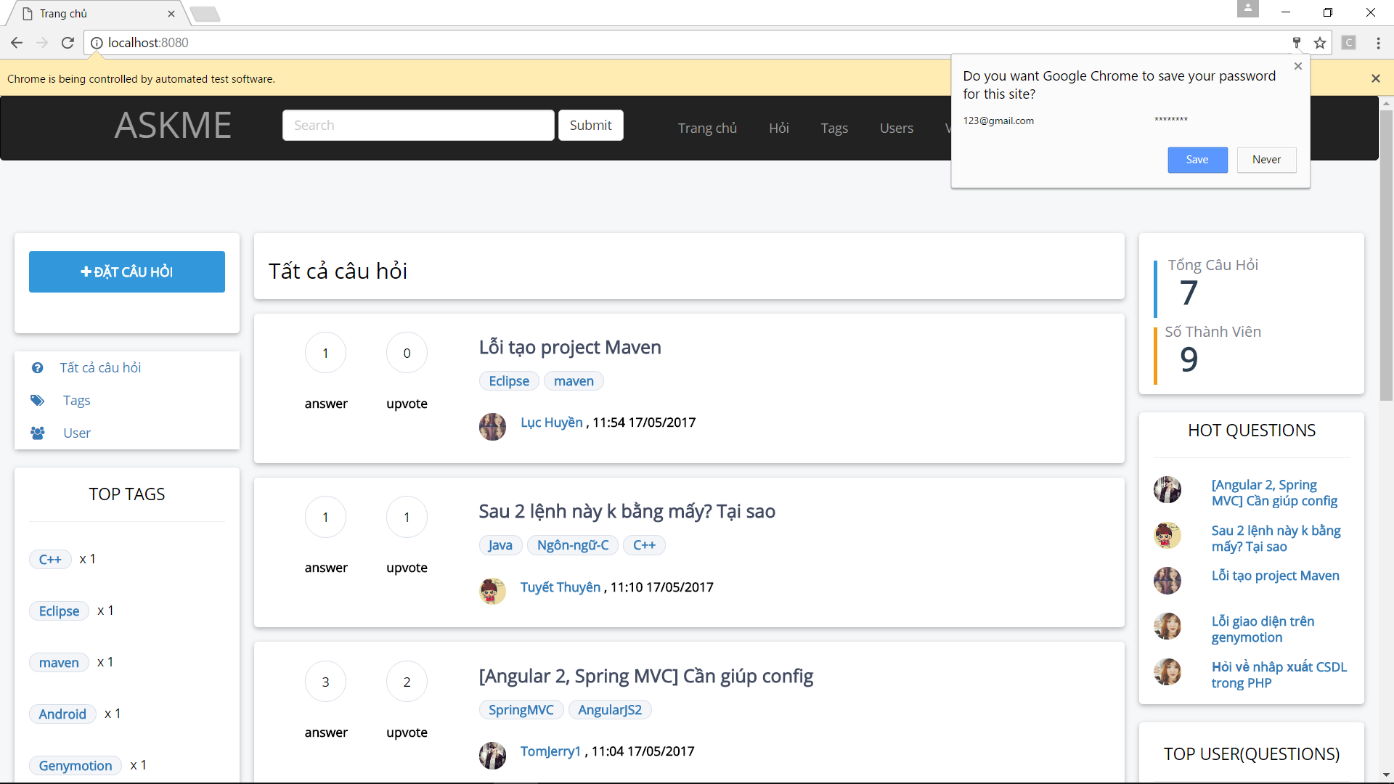
****

Hình 35: Kiểm thử chức năng sửa câu hỏi

### 3.10. Xóa câu hỏi

|  |  |  |  |
| --- | --- | --- | --- |
| **Mã test case** | **Tên chức năng** | **Các bước thực hiện** | **Kết quả thực hiện** |
| TC10 | Xóa câu hỏi | 1. Vào link: <http://localhost:8080/> 2. Click vào link đăng nhập trên thanh công cụ 3. Input email nhập vào [123@gmail.com](mailto:123@gmail.com) 4. Input password nhập vào “12345678” 5. Rồi ấn login 6. Click vào câu hỏi có tiêu đề “Đây là chỉnh sửa tiêu đề câu hỏi” 7. Ấn nút xóa 8. Ấn tiếp nút xóa | Trở về trang chủ và câu hỏi đã bị xóa khỏi hệ thống |

* Kết quả sau khi thực hiện:

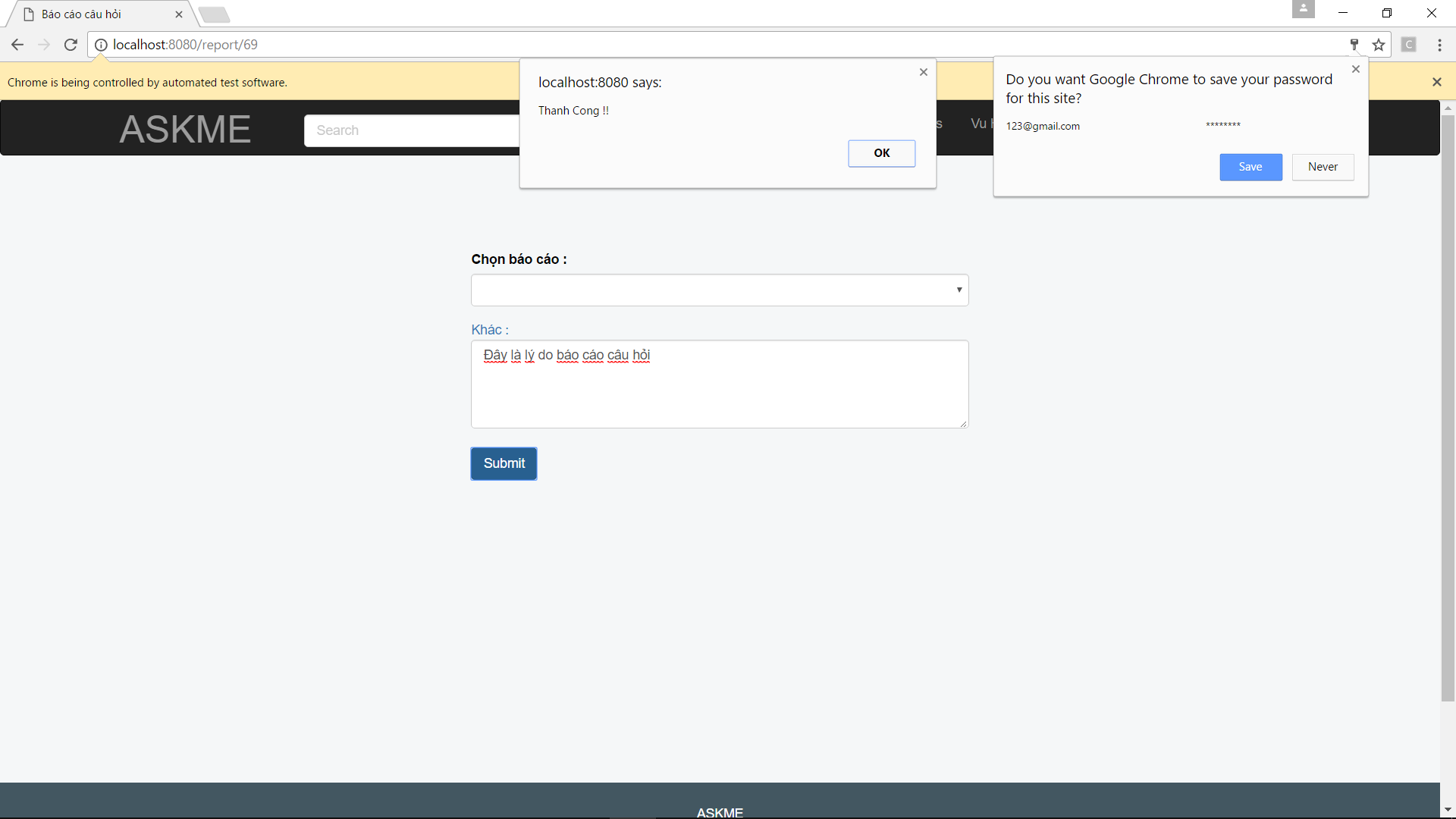


Hình 36: Kiểm thử chức năng xóa câu hỏi

### 3.11. Báo cáo câu hỏi

|  |  |  |  |
| --- | --- | --- | --- |
| **Mã test case** | **Tên chức năng** | **Các bước thực hiện** | **Kết quả thực hiện** |
| TC11 | Báo cáo câu hỏi | 1. Vào link: <http://localhost:8080/> 2. Click vào link đăng nhập trên thanh công cụ 3. Input email nhập vào [123@gmail.com](mailto:123@gmail.com) 4. Input password nhập vào “12345678” 5. Rồi ấn login 6. Click vào câu hỏi có tiêu đề “[Angular 2, Spring MVC] Cần giúp config” 7. Ấn nút báo cáo 8. Chọn khác 9. Nhập vào input báo cáo nội dung “Đây là lý do báo cáo câu hỏi” 10. Ấn nút Submit | Hiển thị thông báo thành công |

* Kết quả sau khi thực hiện:

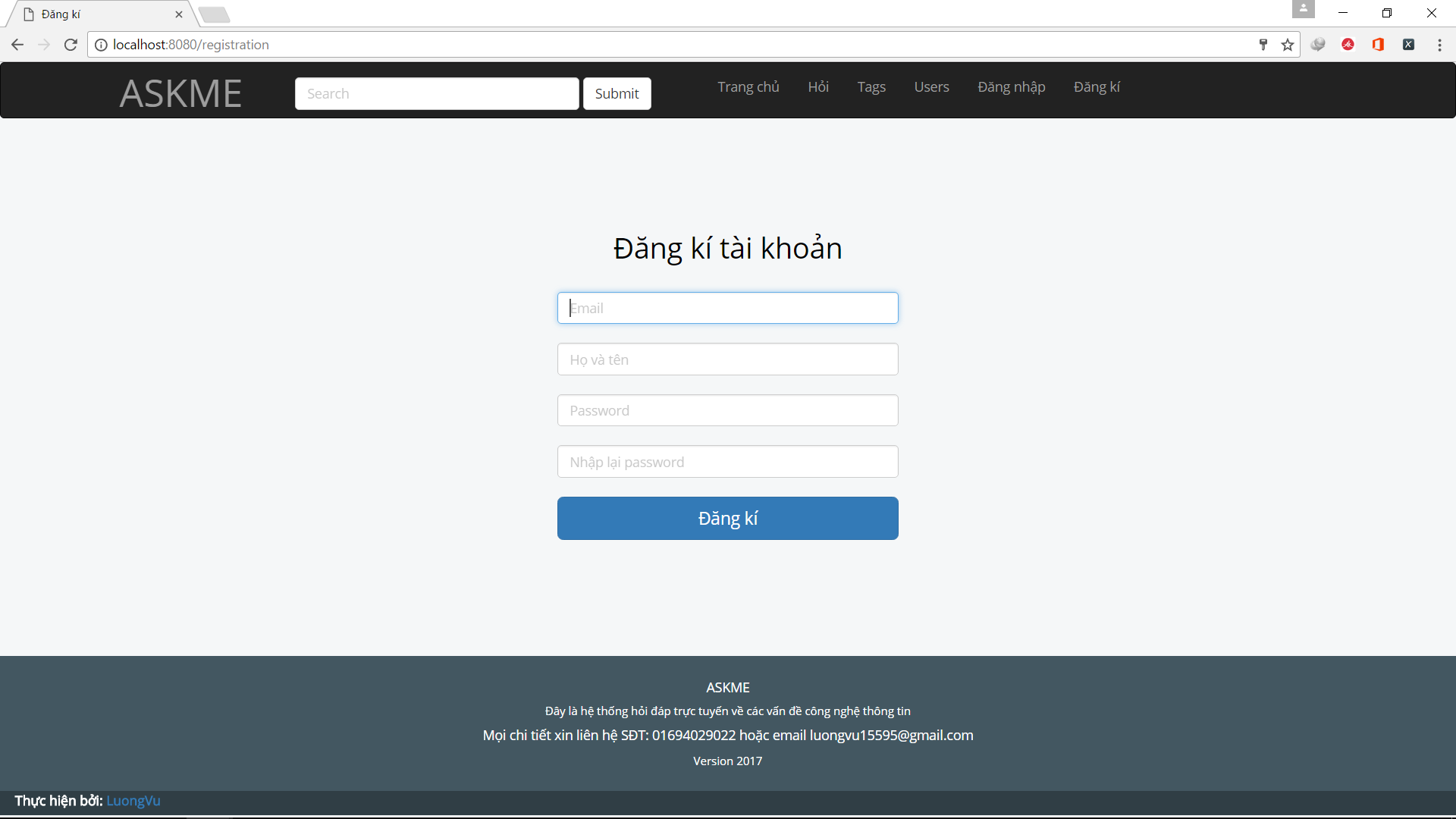


Hình 37: Kiểm thử chức năng báo cáo câu hỏi

# CHƯƠNG 4: KẾT QUẢ THỰC NGHIỆM

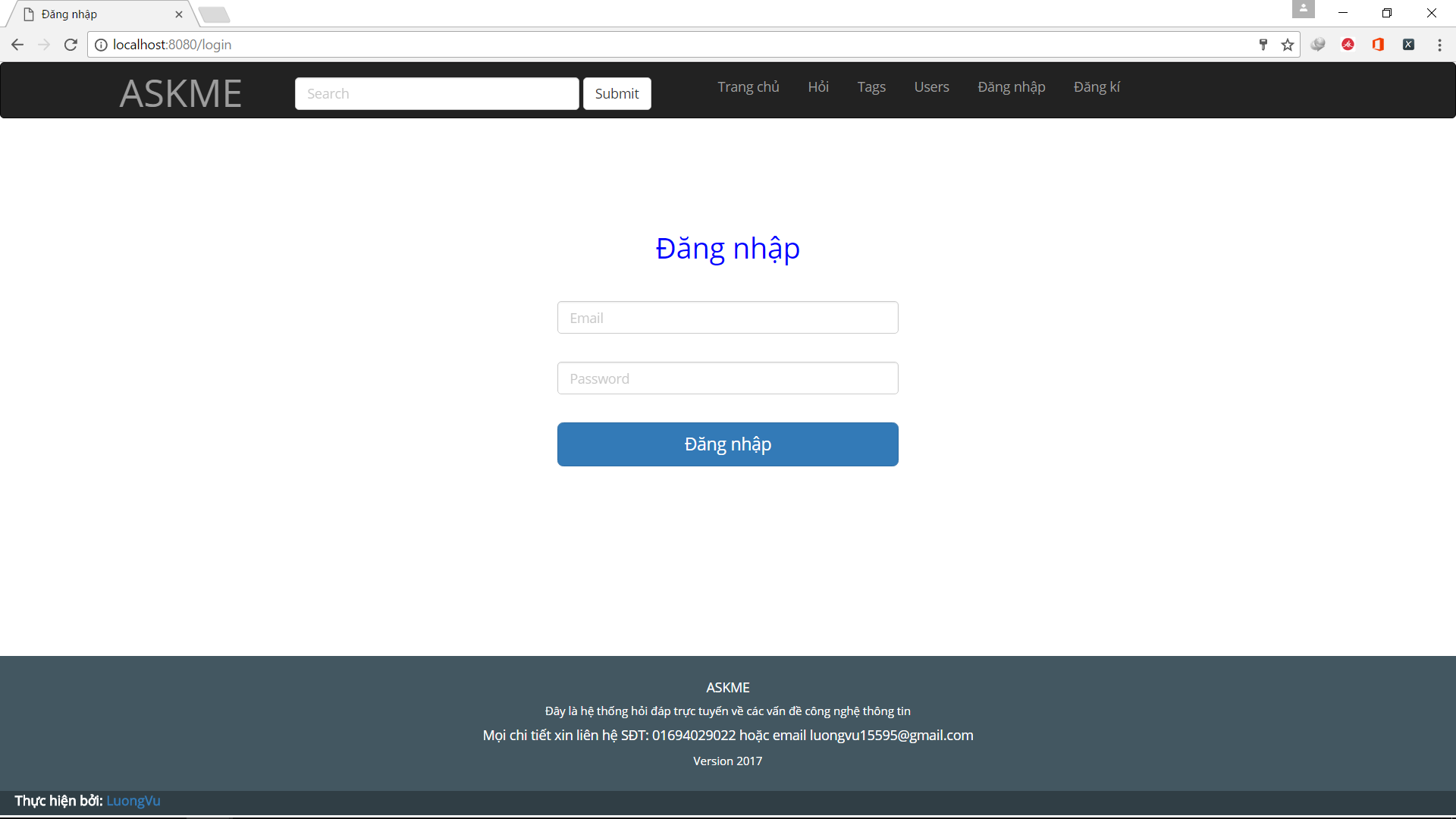
## 1. GIAO DIỆN CHỨC NĂNG

### 1.1. giao diện trang đăng kí:



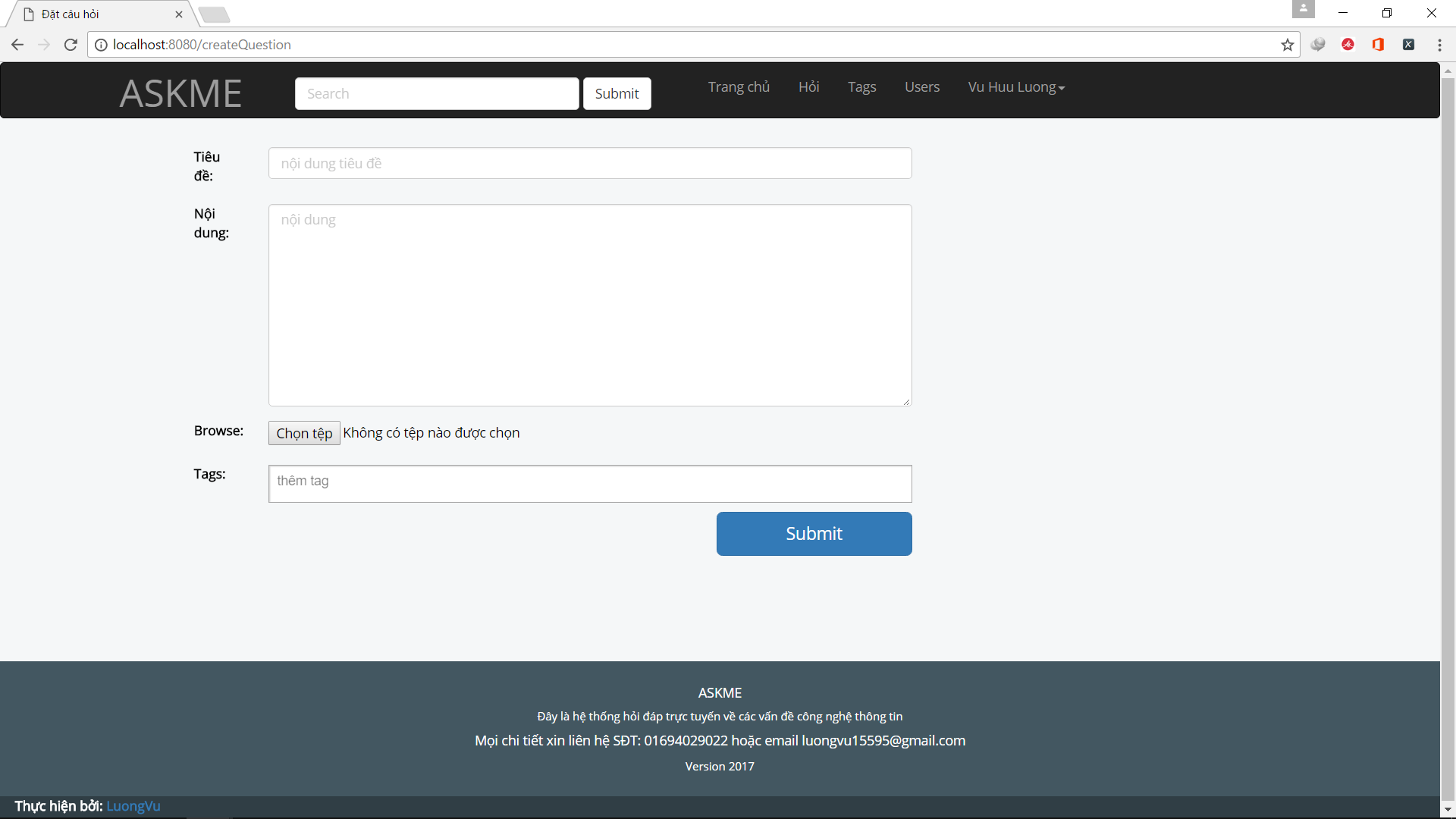
Hình 38: Giao diện trang đăng kí

### 1.2. giao diện trang đăng nhập

****

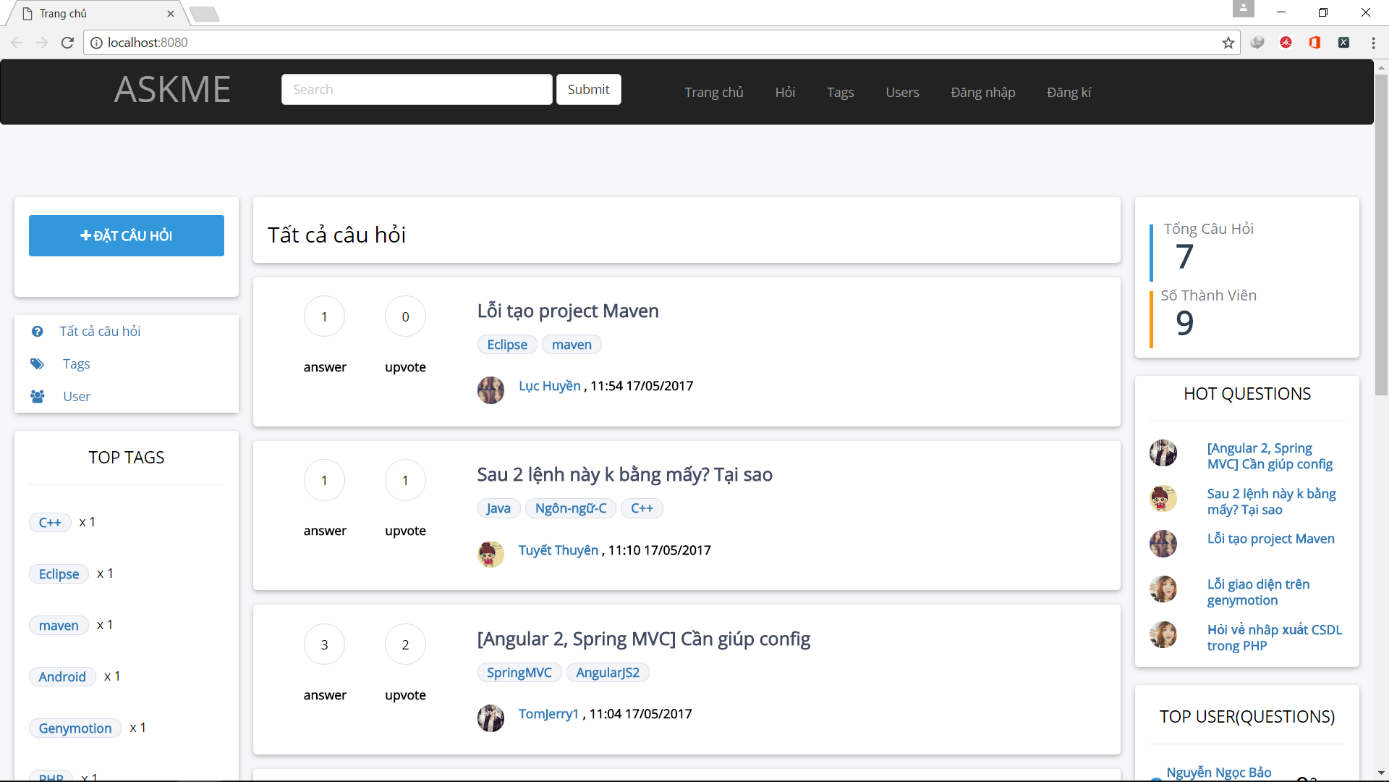
Hình 39: Giao diện đăng nhập

### 1.3. Giao diện trang đặt câu hỏi

****

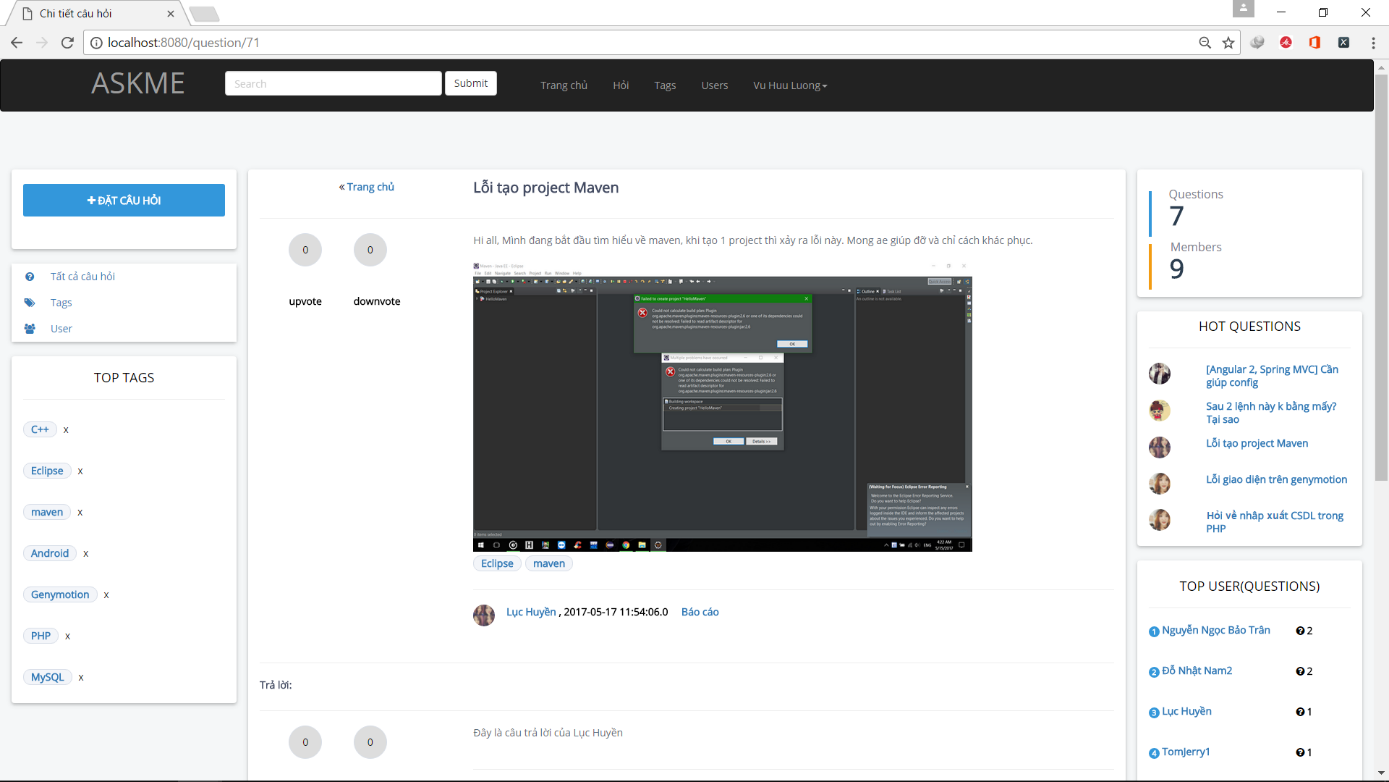
Hình 40: Giao diện trang đặt câu hỏi

### 1.4. Giao diện trang chủ:



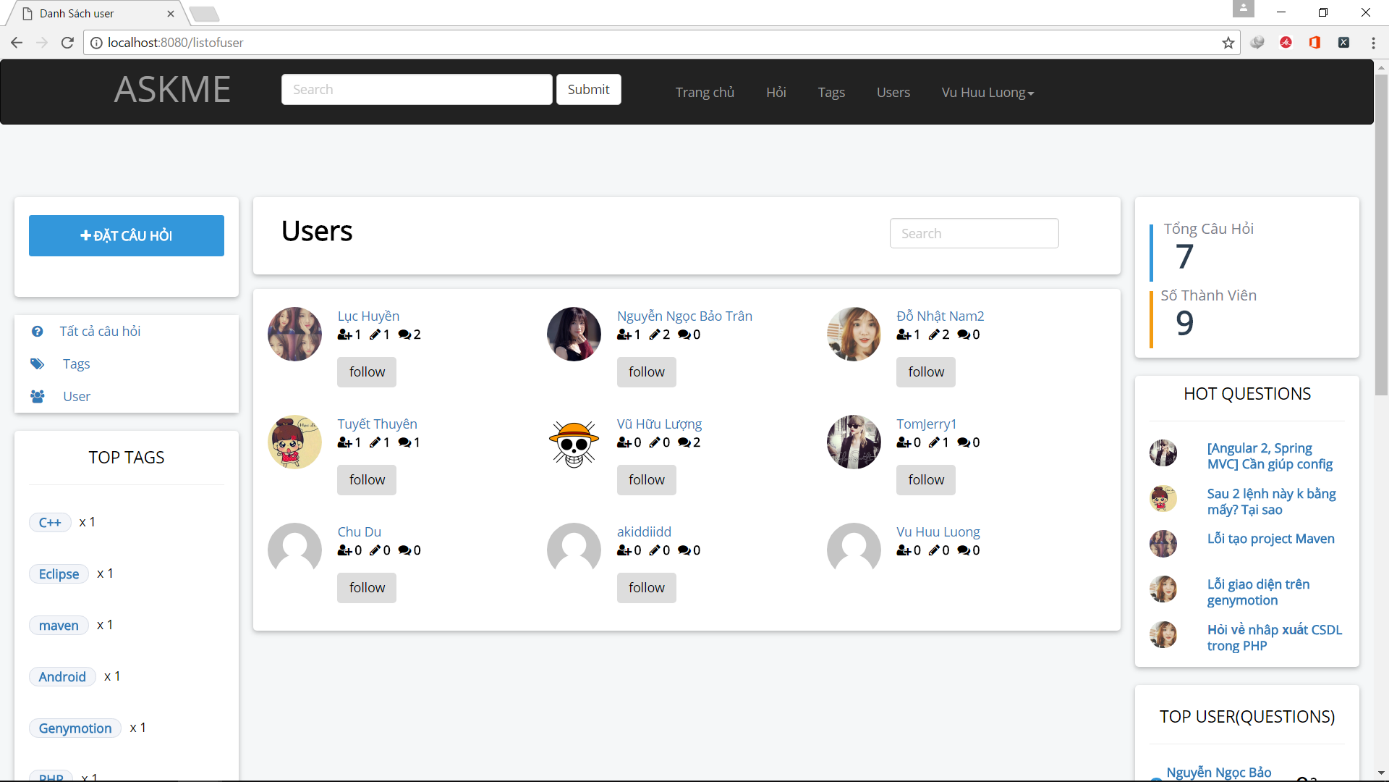
Hình 41: Giao diện trang chủ

### 1.5. Giao diện trang chi tiết câu hỏi:



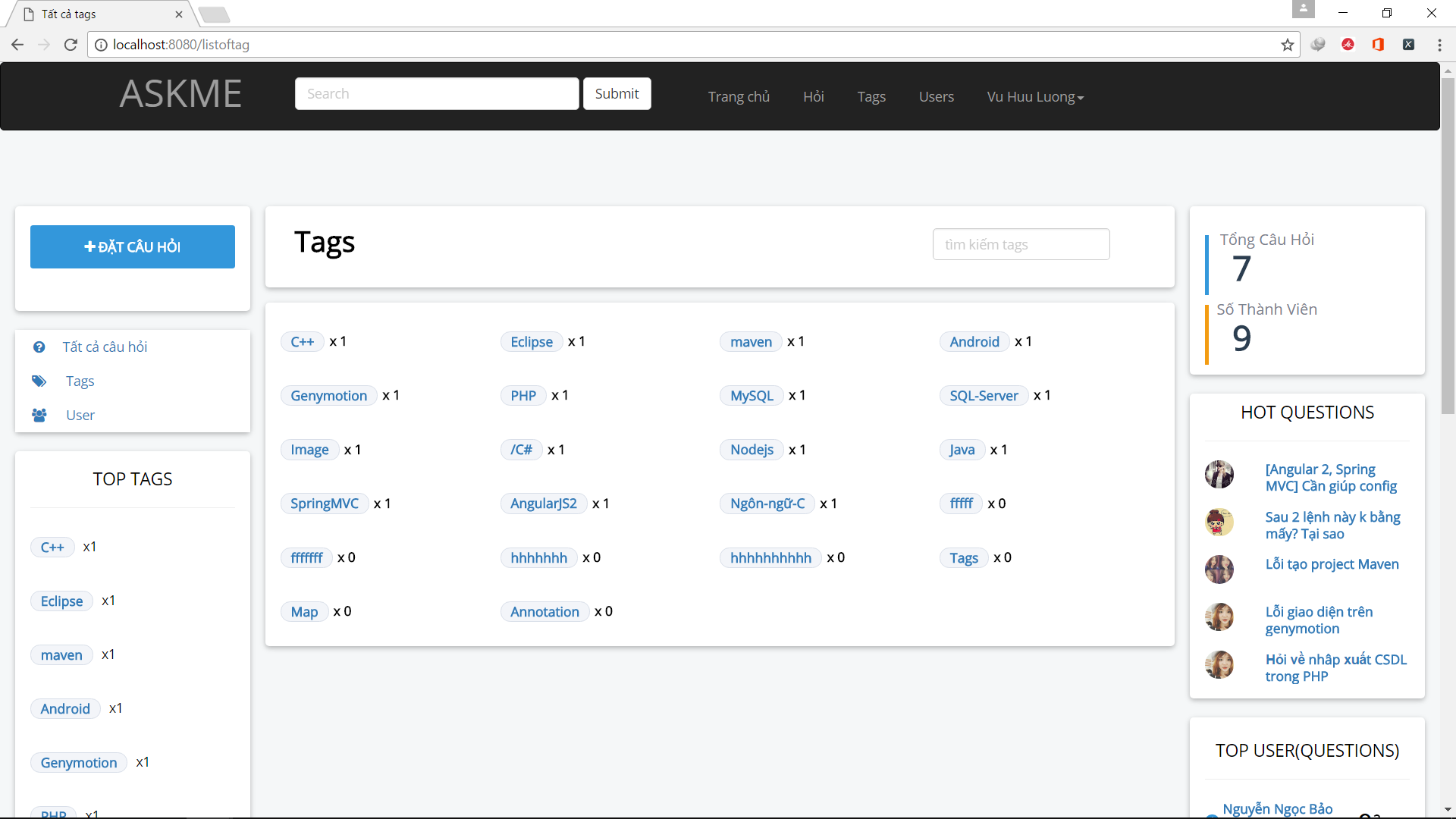
Hình 42: Giao diện trang chi tiết câu hỏi

### 1.6. Giao diện trang danh sách người dùng:

**

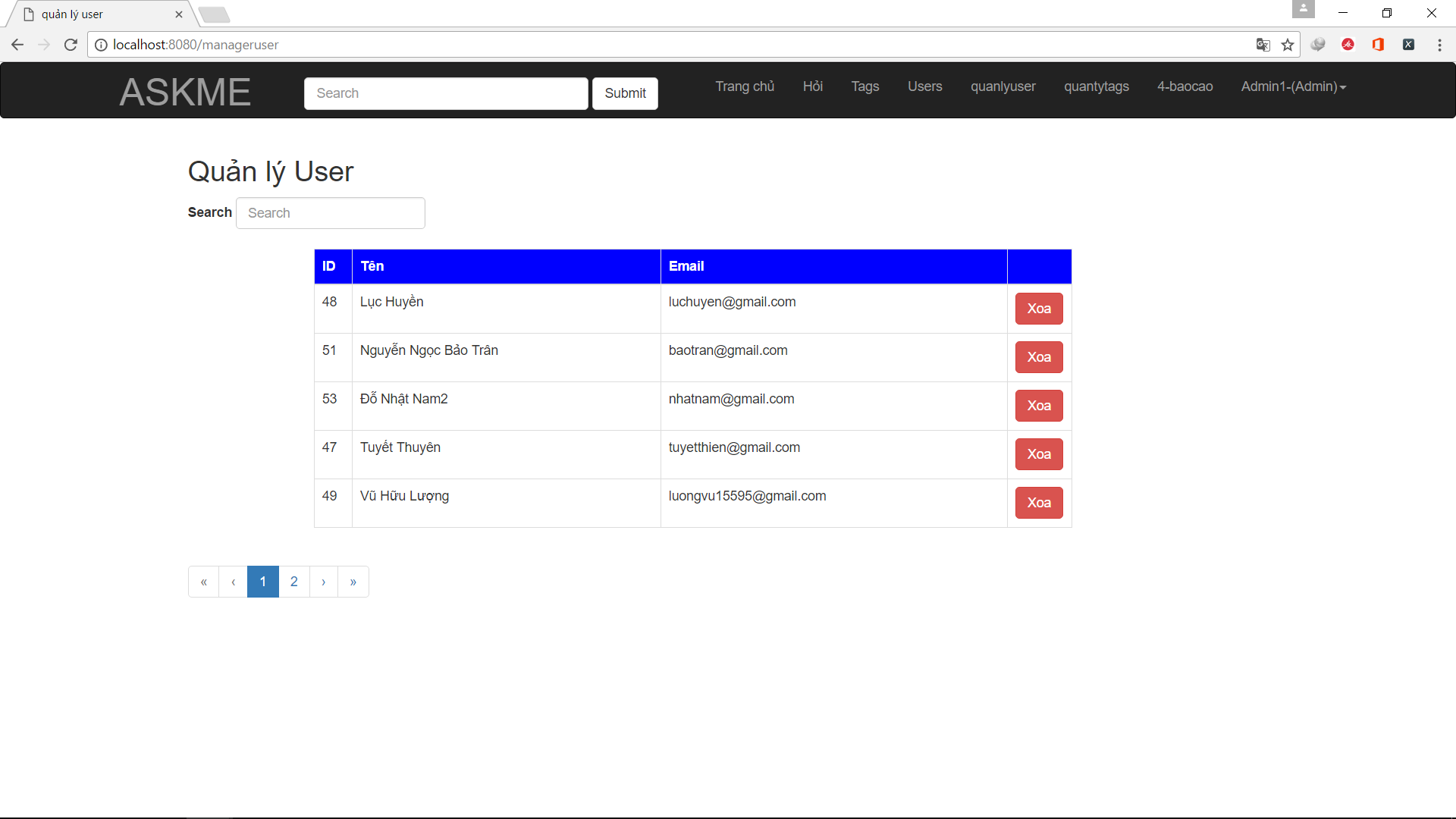
Hình 43: Giao diện trang danh sách người dùng

### 1.7. Giao diện trang danh sách tags:



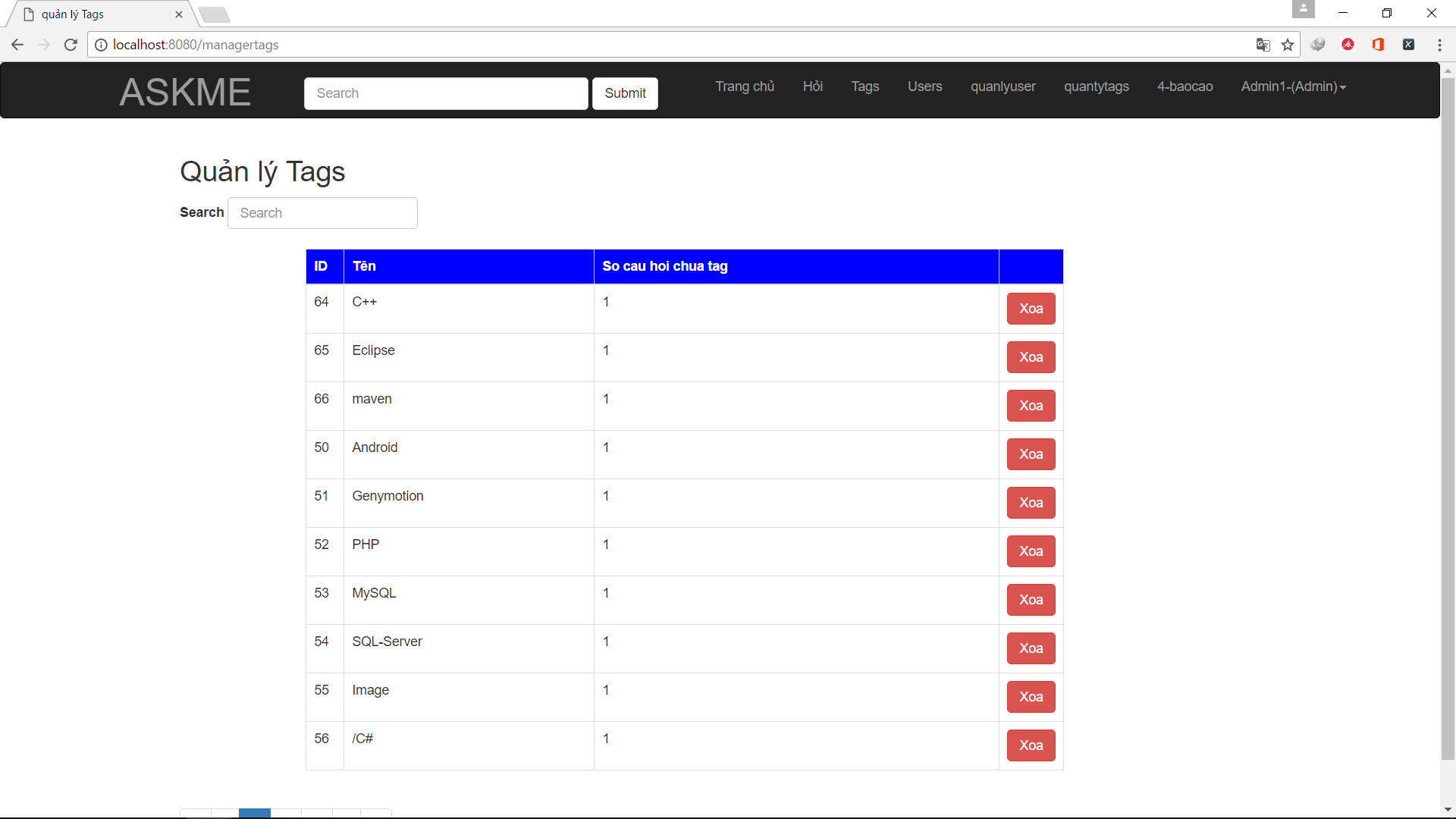
Hình 44: Giao diện trang danh sách tags

### 1.8. Giao diện trang quản lý người dùng:



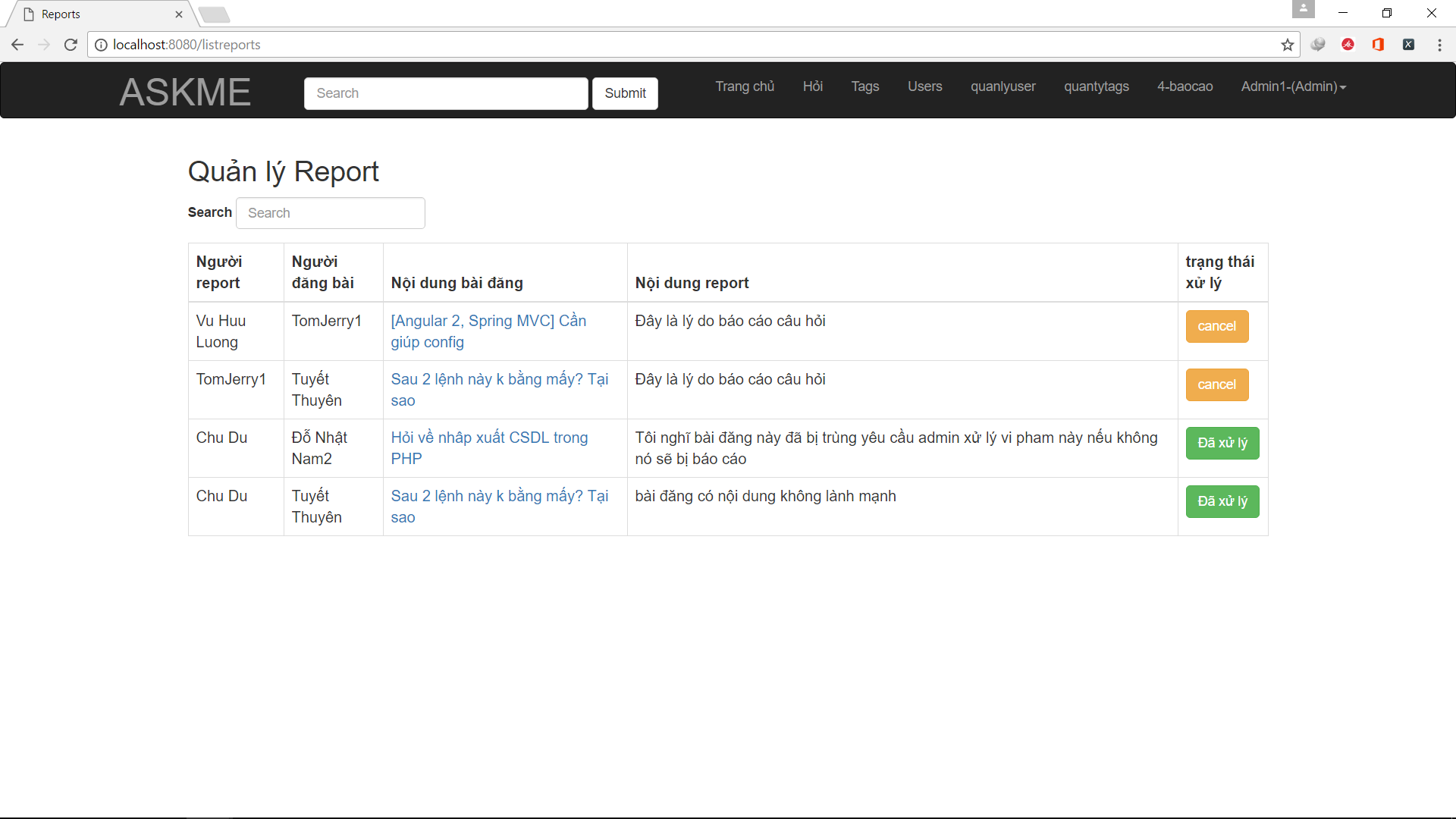
Hình 45: Giao diện trang quản lý người dùng

### 1.9. Giao diện trang quản lý tags:



Hình 46: Giao diện trang quản lý tags

### 1.10. Giao diện trang quản lý báo cáo:



Hình 47: Giao diện trang quản lý báo cáo

# CHƯƠNG 5: KẾT LUẬN

## 1. Đánh giá kết quả đạt được

Qua quá trình tìm hiểu và phân tích thiết kế hệ thống, em đã hiểu được quá trình phân tích, thiết kế và xây dựng từng bước của một hệ thống quản trị nội dung. Ngoài ra còn nắm được cách xây dựng một hệ thống quản trị nội dung sử dụng các cơ sở dữ liệu tiên tiến cụ thể là back-end sử dụng công nghệ SpringMVC, Hibernate và front-end sử dụng công nghệ AngularJS.

*1.1. Ưu điểm*

* Website được xây dựng với đầy đủ các chức năng theo đặt ra, đáp ứng được nhu cầu xem, người đặt câu hỏi, người trả lời và người quản trị website
* Giao diện thân thiện, dễ sử dụng, dễ thao tác trên website
* Chương trình về cơ bản là ổn định, hệ thống hoạt động tốt không xảy ra những lỗi lớn. Do sử dạng các công nghệ tiên tiến.

*1.2. Nhược điểm*

* Giao diện vẫn còn đơn giản, hệ thống chưa có độ phức tạp cao.
* Chưa thử nghiệm với một lượng dữ liệu lớn. Hệ thống mới chỉ đáp ứng được một số mong muốn của người dùng.
* Một số tính năng chưa thật hoàn thiện

## **2. Kết luận rút ra**:

Đồ án đã đạt được các mục tiêu cơ bản đặt ra khi thực hiện:

* Tìm hiểu và triển khai website bằng ngôn ngữ Spring MVC ở Backend và AngularJS ở Front-end.
* Tìm hiểu đầy đủ và hiểu rõ quy trình thiết kế và xây dựng một trang web có tính thực thế cao, ứng dụng cho việc phát triển những dự án mới trong tương lai.

## 3. Định hướng phát triển

Website tuy đã có thể đáp ứng được các yêu cầu của đề tài, tuy nhiên để có thể triển khai ra thực tế thì còn cần phải phát triển thêm nhiều tính năng mới, nghiên cứu thực tế kĩ lưỡng hơn, chương trình có độ phức tạp cao hơn.

Các mục tiêu cần phải đạt được khi phát triển hệ thống trong tương lai:

* Triển khai hệ thống ra thực tế, tìm hiểu và rút ra những vấn đề mới gặp phải khi triển khai ra thực tế.
* Tiếp tục thiết kế và nâng cấp giao diện để làm vừa lòng người sử dụng.
* Nghiên cứu và tìm hiểu phản hồi của người dùng, từ đó đánh giá chính xác và khách quan hơn về hệ thống và có những điều chỉnh thích hợp hơn.
* Chia sẻ các câu hỏi lên mạng xã hội.
* Đăng nhập bằng google hoặc facbook.

## 