{Trang trắng này dùng để dán bản Nhận xét của người hướng dẫn, hoặc thay trang này bằng Nhận xét của người hướng dẫn}

{Trang trắng này dùng để dán bản Nhận xét của người phản biện, hoặc thay trang này bằng Nhận xét của người phản biện}

# TÓM TẮT

Tên đề tài: Xây dựng website thi và chấm trắc nghiệm Tin học đại cương tại trường Đại học Bách khoa

Sinh viên thực hiện: Đỗ Đình Vĩnh

Số thẻ sinh viên: 102152101167 Lớp: 10T2

Kết quả của đề tài này là xây dựng một website dùng để quản lý việc tổ chức thi và chấm điểm theo hình thức trắc nghiệm cho môn học **Tin học đại cương** tại trường Đại học Bách khoa. Hệ thống website này được quản lý bởi Admin – người có quyền hạn cao nhất. Admin là đối tượng tạo ra tài khoản cho giảng viên và sau đó giảng viên sẽ là đối tượng thực hiện việc tạo ra các nhóm thi, đề thi cũng như cho phép sinh viên thực hiện bài thi. Sinh viên là đối tượng sẽ thực hiện bài thi trắc nghiệm môn học Tin học đại cương trên hệ thống sau khi được giảng viên cho phép.

|  |  |
| --- | --- |
| ĐẠI HỌC ĐÀ NẴNG  **TRƯỜNG ĐẠI HỌC BÁCH KHOA**  KHOA CÔNG NGHỆ THÔNG TIN | **CỘNG HÒA XÃ HÔI CHỦ NGHĨA VIỆT NAM**  Độc lập - Tự do - Hạnh phúc |

# NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP

Họ tên sinh viên: Đỗ Đình Vĩnh Số thẻ sinh viên: 102152101167

Lớp: 10T2 Khoa: Công nghệ thông tin Ngành: Công nghệ phần mềm

1. *Tên đề tài đồ án:*

Xây dựng website thi và chấm trắc nghiệm Tin học đại cương tại trường Đại học Bách khoa

1. *Đề tài thuộc diện:*  *Có ký kết thỏa thuận sở hữu trí tuệ đối với kết quả thực hiện*
2. *Các số liệu và dữ liệu ban đầu:*

* File excel chứa thông tin về các nhóm thi
* File word chứa thông tin về đề thi trắc nghiệm

1. *Nội dung các phần thuyết minh và tính toán:*

…...………………………………………………………………………………………

…...………………………………………………………………………………………

…...………………………………………………………………………………………

…...………………………………………………………………………………………

…...………………………………………………………………………………………

1. *Các bản vẽ, đồ thị ( ghi rõ các loại và kích thước bản vẽ ):*

…...………………………………………………………………………………………

…...………………………………………………………………………………………

…...………………………………………………………………………………………

…...………………………………………………………………………………………

1. *Họ tên người hướng dẫn:* ThS. Nguyễn Văn Nguyên
2. *Ngày giao nhiệm vụ đồ án:*  *……../……./2017*
3. *Ngày hoàn thành đồ án: ……../……./2017*

|  |  |
| --- | --- |
|  | *Đà Nẵng, ngày tháng năm 2017* |
| **Trưởng Bộ môn** …………………….. | **Người hướng dẫn** |

# LỜI NÓI ĐẦU

Để hoàn thành đề tài tốt nghiệp này em đã nhận được sự hướng dẫn, giúp đỡ và động viên tận tình từ nhiều phía. Có những lúc gặp khó khăn tưởng chừng như bỏ cuộc nhưng nhờ sự động viên của giảng viên hướng dẫn, gia đình và bàn bè đã giúp em hoàn thành đồ án này. Tất cả những điều đó đã trở thành một động lực rất lớn để em tiến tới một tương lai tươi sáng phía trước. Với tất cả sự cảm kích và trân trọng, em xin được gửi lời cảm ơn đến tất mọi người.

Ngoài sự nỗ lực của bản thân, em xin chân thành cảm ơn ThS.Nguyễn Văn Nguyên đã tận tình giúp đỡ em trong suốt thời gian thực hiện đề tài.

Em xin chân thành cảm ơn các thầy cô giáo của trường Đại học Bách Khoa Đà Nẵng, xin cảm ơn quý thầy cô đã tận tình dạy bảo, giúp đỡ em trong suốt thời gian em học tại trường.

Xin cảm ơn tất cả các sinh viên, bạn bè và gia đình những người đã luôn sát cánh cùng mình, chia sẻ, ủng hộ và giúp đỡ mình trong thời gian thực hiện đề tài.

Xin trân trọng cảm ơn!

Sinh viên

Đỗ Đình Vĩnh

# CAM ĐOAN

Tôi xin cam đoan :

* 1. Những nội dung trong luận văn này là do tôi thực hiện dưới sự hướng dẫn trực tiếp của ThS.Nguyễn Văn Nguyên.
  2. Mọi tham khảo dùng trong luận văn đều được trích dẫn rõ ràng tên tác giả, tên công trình, thời gian, địa điểm công bố.
  3. Mọi sao chép không hợp lệ, vi phạm quy chế đào tạo, hay gian trá, tôi xin chịu hoàn toàn trách nhiệm.

Sinh viên thực hiện

Đỗ Đình Vĩnh

# MỤC LỤC

[TÓM TẮT](#_Toc499804804)

[NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP](#_Toc499804805)

[LỜI NÓI ĐẦU i](#_Toc499804806)

[CAM ĐOAN ii](#_Toc499804807)

[MỤC LỤC iii](#_Toc499804808)

[DANH SÁCH CÁC BẢNG, HÌNH VẼ v](#_Toc499804809)

[DANH SÁCH CÁC KÝ HIỆU, CHỮ VIẾT TẮT viii](#_Toc499804810)

[Chương 1: TỔNG QUAN VỀ ĐỀ TÀI 3](#_Toc499804811)

[**1.** **Tổng quan** 3](#_Toc499804812)

[***1.1*** ***Giới thiệu về đề tài*** 3](#_Toc499804813)

[***1.2*** ***Mô hình ứng dụng*** 3](#_Toc499804814)

[**2.** **Yêu cầu của đề tài** 3](#_Toc499804815)

[Chương 2: CƠ SỞ LÝ THUYẾT 5](#_Toc499804816)

[**1.** **Tổng quan về giao thức TCP/IP** 5](#_Toc499804817)

[***1.1*** ***Tổng quan*** 5](#_Toc499804818)

[***1.2*** ***Phương thức hoạt động*** 6](#_Toc499804819)

[**2.** **Tổng quan về giao thức HTTP** 8](#_Toc499804820)

[***2.1*** ***Tổng quan*** 8](#_Toc499804821)

[***2.2*** ***Kết nối HTTP*** 9](#_Toc499804822)

[***2.3*** ***Thông điệp HTTP*** 10](#_Toc499804823)

[***2.4*** ***Gói tin HTTP*** 14](#_Toc499804824)

[Chương 3: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG 16](#_Toc499804825)

[**1.** **Tổng quan về hệ thống** 16](#_Toc499804826)

[***1.1*** ***Giới thiệu về hệ thống*** 16](#_Toc499804827)

[***1.2*** ***Cấu trúc hệ thống*** 16](#_Toc499804828)

[**2.** **Thiết kế hệ thống** 16](#_Toc499804829)

[***2.1*** ***Biểu đồ use case*** 17](#_Toc499804830)

[***2.2*** ***Biểu đồ hoạt động*** 19](#_Toc499804831)

[***2.3*** ***Biểu đồ tuần tự*** 24](#_Toc499804832)

[***2.4*** ***Biểu đồ lớp*** 31](#_Toc499804833)

[**3.** **Thiết kế cơ sở dữ liệu** 31](#_Toc499804834)

[Chương 4: TRIỂN KHAI VÀ ĐÁNH GIÁ KẾT QUẢ 37](#_Toc499804835)

[**1.** **Các thuật toán triển khai trong ứng dụng** 37](#_Toc499804836)

[***1.1*** ***Thuật toán xáo trộn đề thi*** 37](#_Toc499804837)

[***1.2*** ***Thuật toán tính điểm thi sinh viên*** 37](#_Toc499804838)

[**2.** **Yêu cầu hệ thống** 38](#_Toc499804839)

[***2.1*** ***Các công cụ sử dụng*** 38](#_Toc499804840)

[***2.2*** ***Yêu cầu cấu hình máy tính*** 38](#_Toc499804841)

[**3.** **Minh họa quy trình build và run ứng dụng** 38](#_Toc499804842)

[**4.** **Đánh giá kết quả** 39](#_Toc499804843)

[KẾT LUẬN 46](#_Toc499804844)

[TÀI LIỆU THAM KHẢO 47](#_Toc499804845)

# DANH SÁCH CÁC BẢNG, HÌNH VẼ

[BẢNG 3.1 Bảng users 32](#_Toc499804449)

[BẢNG 3.2 Bảng user\_role 32](#_Toc499804450)

[BẢNG 3.3 Bảng roles 33](#_Toc499804451)

[BẢNG 3.4 Bảng questions 33](#_Toc499804452)

[BẢNG 3.5 Bảng answers 33](#_Toc499804453)

[BẢNG 3.6 Bảng group\_question 34](#_Toc499804454)

[BẢNG 3.7 Bảng groups 34](#_Toc499804455)

[BẢNG 3.8 Bảng students 35](#_Toc499804456)

[BẢNG 3.9 Bảng teachers 36](#_Toc499804457)

[HÌNH 2.1 Các tầng trong bộ giao thức TCP/IP 5](#_Toc499804458)

[HÌNH 2.2 Quá trình đóng mở gói dữ liệu trong TCP/IP 7](#_Toc499804459)

[HÌNH 2.3 Cấu trúc gói dữ liệu trong TCP/IP 7](#_Toc499804460)

[HÌNH 2.4 Mô hình client-server 8](#_Toc499804461)

[HÌNH 2.5 Quy trình hoạt động kết nối HTTP không bền vững 9](#_Toc499804462)

[HÌNH 2.6 Cấu trúc chung của thông điệp HTTP yêu cầu 11](#_Toc499804463)

[HÌNH 2.7 Ví dụ về thông điệp HTTP yêu cầu 12](#_Toc499804464)

[HÌNH 2.8 Cấu trúc chung của một thông điệpHTTP trả lời 13](#_Toc499804465)

[HÌNH 2.9 Ví dụ thông điệp HTTP trả lời 13](#_Toc499804466)

[HÌNH 2.10 Gói tin HTTP 14](#_Toc499804467)

[HÌNH 3.1 Mô hình MVC 16](#_Toc499804468)

[HÌNH 3.2 Biểu đồ use case người dùng 17](#_Toc499804469)

[HÌNH 3.3 Biểu đồ use case quản lý 17](#_Toc499804470)

[HÌNH 3.4 Biểu đồ use case giảng viên 18](#_Toc499804471)

[HÌNH 3.5 Biểu đồ use case sinh viên 19](#_Toc499804472)

[HÌNH 3.6 Biểu đồ hoạt động cho use case đăng nhập 19](#_Toc499804473)

[HÌNH 3.7 Biểu đồ hoạt động cho use case đăng xuất 20](#_Toc499804474)

[HÌNH 3.8 Biểu đồ hoạt động cho use case quản lý admin 20](#_Toc499804475)

[HÌNH 3.9 Biểu đồ hoạt động cho use case admin – quản lý teacher 21](#_Toc499804476)

[HÌNH 3.10 Biểu đồ hoạt động cho use case teacher – quản lý teacher 21](#_Toc499804477)

[HÌNH 3.11 Biểu đồ hoạt động cho use case quản lý nhóm sinh viên 22](#_Toc499804478)

[HÌNH 3.12 Biểu đồ hoạt động cho use case quản lý đề thi 22](#_Toc499804479)

[HÌNH 3.13 Biểu đồ hoạt động cho use case xuất file 23](#_Toc499804480)

[HÌNH 3.14 Biểu đồ hoạt động cho use case làm bài thi 23](#_Toc499804481)

[HÌNH 3.15 Biểu đồ tuần tự cho use case login 24](#_Toc499804482)

[HÌNH 3.16 Biểu đồ tuần tự cho use case logout 24](#_Toc499804483)

[HÌNH 3.17 Biểu đồ tuần tự cho use case quản lý admin 25](#_Toc499804484)

[HÌNH 3.18 Biểu đồ tuần tự cho use case admin – quản lý teacher 26](#_Toc499804485)

[HÌNH 3.19 Biểu đồ tuần tự cho use case teacher – quản lý teacher 27](#_Toc499804486)

[HÌNH 3.20 Biểu đồ tuần tự cho use case quản lý nhóm sinh viên 28](#_Toc499804487)

[HÌNH 3.21 Biểu đồ tuần tự cho use case quản lý đề thi 29](#_Toc499804488)

[HÌNH 3.22 Biểu đồ tuần tự cho use case xuất file 29](#_Toc499804489)

[HÌNH 3.23 Biểu đồ tuần tự cho use case làm bài thi 30](#_Toc499804490)

[HÌNH 3.24 Biểu đồ lớp thiết kế 31](#_Toc499804491)

[HÌNH 3.25 Mối liên kết các bảng trong CSDL 32](#_Toc499804492)

[HÌNH 4.1 Hướng dẫn build source code 39](#_Toc499804493)

[HÌNH 4.2 Kết quả sau khi build source code 39](#_Toc499804494)

[HÌNH 4.3 Hướng dẫn run ứng dụng 39](#_Toc499804495)

[HÌNH 4 4 Một phần của file excel chứa thông tin nhóm sinh viên 40](#_Toc499804496)

[HÌNH 4.5 Một phần của file word chứa đề thi 40](#_Toc499804497)

[HÌNH 4.6 Màn hình trang chủ của ứng dụng 41](#_Toc499804498)

[HÌNH 4.7 Màn hình đăng nhập của ứng dụng 41](#_Toc499804499)

[HÌNH 4.8 Giao diện chính của người dùng admin 42](#_Toc499804500)

[HÌNH 4.9 Giao diện chính của người dùng teacher 42](#_Toc499804501)

[HÌNH 4.10 Giao diện chính của người dùng student 43](#_Toc499804502)

[HÌNH 4.11 Màn hình sinh viên làm bài thi 43](#_Toc499804503)

[HÌNH 4.12 Màn hình sau khi sinh viên đệ trình bài thi 44](#_Toc499804504)

[HÌNH 4.13 Màn hình dùng để xuất kết quả ra file excel 44](#_Toc499804505)

[HÌNH 4.14 Kết quả xuất ra file excel theo định dạng file đầu vào 45](#_Toc499804506)

[HÌNH 4.15 Kết quả xuất ra file excel theo định dạng bảng điểm 45](#_Toc499804507)

# DANH SÁCH CÁC KÝ HIỆU, CHỮ VIẾT TẮT

|  |  |
| --- | --- |
| **Từ viết tắt** | **Diễn giải** |
| SQL | Structured Query Language |
| CSDL | Cơ sở dữ liệu |
| TCP | Transmission Control Protocol (Giao thức điều khiển truyền tin) |
| HTTP | User Datagram Protocol |
| IP | Internet Protocol (Giao thức liên mạng) |
| CRUD | Create – Read – Update - Delete |
| DB | Database |

**MỞ ĐẦU**

1. **Bối cảnh ra đời của đề tài**

Cùng với sự phát triển mạnh mẽ của các ngành trong thời kỳ công nghiệp hóa, hiện đại hóa đất nước, công nghệ thông tin luôn là một trong những ngành tiên phong cho việc góp phần thúc đẩy sự phát triển ấy. Chính vì vậy việc ứng dụng công nghệ thông tin vào trong mọi mặt đời sống nói chung và cho công tác quản lý nói riêng đã trở nên quen thuộc với tất cả các cơ quan, đoàn thể cũng như các doanh nghiệp hay bất kỳ cá nhân nào.

Trường Đại học Bách khoa Đà Nẵng – một trong những trường đại học danh tiếng bậc nhất trong việc đạo tạo kỹ sư tại Việt Nam cũng đã ứng dụng rất nhiều dịch các vụ tin học. Một trong nhưng dịch vụ ấy là việc ứng dụng quản lý thi và chấm trắc nghiệm cho bộ môn Tin học đại cương của trường. Hiện tại, hệ thống quản lý ấy đang được phát triển dựa trên hệ thống Moodle – một hệ thống quản lý học tập mã nguồn mở khá phổ biến. Mặc dù Moodle khá tốt và tiện dụng, song những chức năng của nó vẫn chưa thể đáp ứng hoàn toàn những yêu cầu cụ thể mà hệ thống cần phải có. Chính bởi vậy, cùng sự góp ý của ThS. Nguyễn Văn Nguyên, em đã quyết định thực hiện đề tài “Xây dựng website quản lý việc thi và chấm trắc nghiệm Tin học đại cương tại trường Đại học bách khoa Đà Nẵng” cho đồ án tốt nghiệp của mình.

1. **Mục đích và ý nghĩa**
2. **Mục đích**

Xây dựng được một website mà có thể ứng dụng nó vào việc quản lý thi và chấm bài thi trắc nghiệm cho môn học Tin học đại cương của trường Đại học Bách khoa Đà Nẵng.

1. **Ý nghĩa**

Giảm thiểu tối đa các thao tác thủ công trong việc tổ chức thi và chấm trắc nghiệm môn học Tin học đại cương cho giảng viên phụ trách bộ môn này tại trường Đại học Bách khoa Đà Nẵng. Điều đó góp phần tiết kiệm thời gian và công sức trong việc quản lý.

1. **Phạm vi và đối tượng nghiên cứu**

Đề tài được nghiên cứu trong phạm vi nhà trường (trường Đại học Bách khoa Đà Nẵng) với đối tượng chính là về môn học Tin học đại cương.

1. **Phương pháp nghiên cứu**

Để hoàn thành đồ án này, em đã sử dụng các phương pháp nghiên cứu sau:

* Phương pháp lấy yêu cầu.
* Phương pháp phân tích và thiết kế.
* Phương pháp nghiên cứu tài liệu.

1. **Cấu trúc của đồ án tốt nghiệp**

**Chương 1: Tổng quan về đề tài**

Tổng quan.

Yêu cầu đề tài.

**Chương 2: Cơ sở lý thuyết.**

Tổng quan về giao thức TCP/IP

Tổng quan về giao thức HTTP

**Chương 3: Phân tích và thiết kế hệ thống.**

Tổng quan về hệ thống

Thiết kế hệ thống.

Thiết kế CSDL.

**Chương 4: Triển khai và đánh giá kết quả.**

Các thuật toán triển khai trong ứng dụng

Yêu cầu hệ thống.

Minh họa quy trình build và run ứng dụng

Đánh giá kết quả

**Kết luận**

**Tài liệu tham khảo**

# Chương 1: TỔNG QUAN VỀ ĐỀ TÀI

1. **Tổng quan**
2. ***Giới thiệu về đề tài***

Đề tài xây dựng website để quản lý việc thi và chấm bài thi trắc nghiệm môn học Tin học đại cương tại trường Đại học Bách khoa.

Đề tài được hiểu như sau:

Giảng viên sẽ là người trực tiếp quản lý việc thi và chấm bài thi trắc nghiệm và việc quản lý này sẽ thực hiện theo các nhóm sinh viên.

Đầu tiên giảng viên sẽ import các nhóm thi – được cho bởi các sheet trong một file excel theo định dạng cho trước. Việc này cũng sẽ tạo ra tài khoản cho sinh viên.

Sau đó, ứng với mỗi nhóm thi, giảng viên sẽ import các đề thi – được cho bởi file word theo định dạng cho trước.

Các sinh viên tương ứng theo nhóm sẽ vào phòng thi để truy cập vào website và thực hiện bài thi dưới sự hướng dẫn của giảng viên coi thi.

Sinh viên kết thúc bài thi bằng cách nhấn vào button submit trên màn hình trong quá trình thi.

Việc chấm điểm sẽ được hệ thống tự động chấm và sau đó giảng viên có thể xuất bảng điểm theo từng nhóm thi bởi việc sử dụng tính năng tương ứng có trong hệ thống.

1. ***Mô hình ứng dụng***

Ứng dụng được xây dựng dựa trên mô hình clien – server theo giao thức HTTP và TCP/IP:

* Client thực hiện gửi request đến server
* Server sẽ nhận request từ client, sau đó thực hiện xử lý và cuối cùng gửi trả kết quả về lại client.
* Kết thúc quá trình xử lý.

1. **Yêu cầu của đề tài**

Đề tài cần phải:

* Thực hiện chính xác 100% việc import các file excel (chứa các nhóm sinh viên) và file word (chứa đề thi) theo định dạng cho trước.
* Thực hiện chính xác tối thiểu 99.9% việc chấm điểm bài thi
* Thực hiện chính xác 100% việc export các file excel theo định dạng cho trước.
* Có đầy đủ các chức năng dùng để quản lý: thống kê, thêm, xem, xóa, sửa …

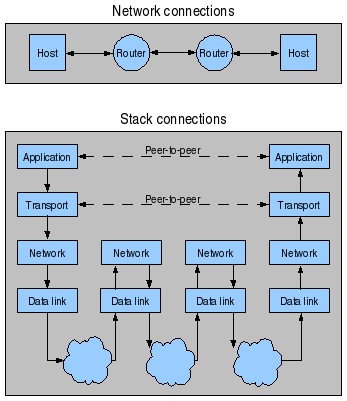
# Chương 2: CƠ SỞ LÝ THUYẾT

1. **Tổng quan về giao thức TCP/IP**
2. ***Tổng quan***

TCP/IP là bộ giao thức cho phép kết nối các hệ thống mạng không đồng nhất với nhau. Ngày nay TCP/IP được sử dụng rộng rãi trong mạng cục bộ cũng như mạng toàn cầu.

TCP/IP được xem như giản lược của mô hình tham chiếu OSI với 4 tầng như sau:

* Tầng Liên Kết (Datalink Layer)
* Tầng Mạng (Internet Layer)
* Tầng Giao Vận (Transport Layer)
* Tầng Ứng Dụng (Application Layer)



HÌNH 2.1 Các tầng trong bộ giao thức TCP/IP

* Tầng liên kết:

Tầng liên kết (còn được gọi là tầng liên kết dữ liệu hay tầng giao tiếp mạng) là tầng thấp nhất trong mô hình TCP/IP, bao gồm các thiết bị giao tiếp mạng và các chương trình cung cấp các thông tin cần thiết để có thể hoạt động, truy nhập đường truyền vật lý qua các thiết bị giao tiếp mạng đó.

* Tầng internet:

Tầng Internet ( hay còn gọi là tầng Mạng) xử lý quá trình truyền gói tin trên mạng, các giao thức của tầng này bao gồm : IP ( Internet Protocol) , ICMP ( Internet Control Message Protocol) , IGMP ( Internet Group Message Protocol )

* Tầng giao vận:

Tầng giao vận phụ trách luồng dữ liệu giữa 2 trạm thực hiện các ứng dụng của tầng trên, tầng này có 2 giao thức chính là TCP ( Transmisson Control Protocol) và UDP ( User Datagram Protocol )

TCP cung cấp luồng dữ liệu tin cậy giữa 2 trạm, nó sử dụng các cơ chế như chia nhỏ các gói tin ở tầng trên thành các gói tin có kích thước thích hợp cho tầng mạng bên dưới, báo nhận gói tin, đặt hạn chế thời gian timeout để đảm bảo bên nhân biết được các gói tin đã gửi đi. Do tầng này đảm bảo tính tin cậy nên tầng trên sẽ không cần quan tâm đến nữa

UDP cung cấp một dịch vụ rất đơn giản hơn cho tầng ứng dụng . Nó chỉ gửi dữ liệu từ trạm này tới trạm kia mà không đảm bảo các gói tin đến được tới đích. Các cơ chế đảm bảo độ tin cậy được thực hiện bởi tầng trên

* Tầng ứng dụng:

Là tầng trên của mô hình TCP/IP bao gồm các tiến trình và các ứng dụng cung cấp cho người sử dụng để truy cập mạng. Có rất nhiều ứng dụng được cung cấp trong tầng này , mà phổ biến là Telnet: sử dụng trong việc truy cập mạng từ xa, FTP ( File Transport Protocol ) dịch vụ truyền tệp tin., EMAIL : dịch vụ truyền thư tín điện tử. WWW ( Word Wide Web )

1. ***Phương thức hoạt động***

****

HÌNH 2.2 Quá trình đóng mở gói dữ liệu trong TCP/IP

Cũng tương tự như trong mô hình OSI, khi truyền dữ liệu , quá trình tiến hành từ tầng trên xuống tầng dưới, qua mỗi tầng dữ liệu được thêm vào thông tin điều khiển gọi là Header. Khi nhận dữ liệu thì quá trình xảy ra ngược lại. dữ liệu được truyền từ tấng dưới lên và qua mỗi tầng thì phần header tương ứng sẽ được lấy đi và khi đến tầng trên cùng thì dữ liệu không còn phần header nữa.

****

HÌNH 2.3 Cấu trúc gói dữ liệu trong TCP/IP

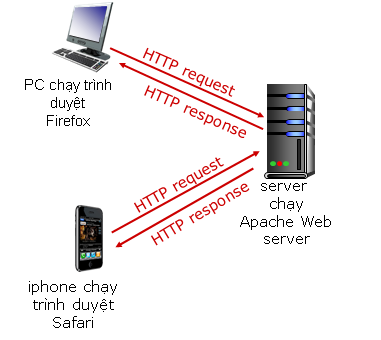
Hình trên cho ta thấy lược đồ dữ liệu qua các tầng. Trong hình ta thấy tại các tầng khác nhau dữ liệu được mang những thuật ngữ khác nhau:

* Trong tầng ứng dụng: dữ liệu là các luồng được gọi là stream.
* Trong tầng giao vận: đơn vị dữ liệu mà TCP gửi xuống gọi là TCP segment.
* Trong tầng mạng, dữ liệu mà IP gửi xuống tầng dưới gọi là IP Datagram
* Trong tầng liên kết, dữ liệu được truyền đi gọi là frame.

1. **Tổng quan về giao thức HTTP**
2. ***Tổng quan***

Hypertext Transfer Protocol (HTTP) là giao thức thuộc lớp ứng dụng web. Sử dụng kết nối TCP cổng 80. HTTP được định nghĩa trong RFC 1945 (HTTP 1.0) và RFC 2616 (HTTP 1.1).

HTTP hoạt động dựa trên mô hình client-server. Trình duyệt client thực hiện yêu cầu, nhận và hiển thị đối tượng web (gồm dữ liệu HTML, hình ảnh JPEG, Java applet, video, âm thanh, …). Trong khi, web server sẽ gửi trả lời khi nhận được yêu cầu từ client.



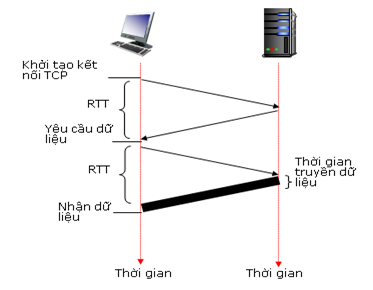
HÌNH 2.4 Mô hình client-server

1. ***Kết nối HTTP***

Có hai loại kết nối HTTP là kết nối không bền vững và kết nối bền vững.

* Kết nối không bền vững: sau khi, server gửi đi một đối tượng thì kết nối TCP sẽ được đóng. Như vậy, mỗi kết nối TCP chỉ truyền được duy nhất một yêu cầu từ client và nhận lại một thông điệp trả lời từ server.
* Kết nối bền vững: server sẽ duy trì kết nối TCP cho việc gửi nhiều đối tượng. Như vậy, sẽ có nhiều yêu cầu từ client được gửi đến server trên cùng một kết nối. Thông thường kết nối TCP này sẽ được đóng lại trong một khoảng thời gian định trước.

Quy trình hoạt động của kết nối HTTP không bền vững:



HÌNH 2.5 Quy trình hoạt động kết nối HTTP không bền vững

Hình 2.5 thể hiện chi tiết các bước hoạt động trong kết nối HTTP không bền vững giữa HTTP client (Firefox, Chrome,…) và HTTP server (ví dụ www.hutech.edu.vn):

* Bước 1: client khởi tạo kết nối TCP bằng việc gửi yêu cầu đến server. Server nhận được yêu cầu và chấp nhận kết nối bằng việc gửi trả lời về cho client. Nếu sau khoảng thời gian RTT mà không nhận được trả lời từ phía server thì client sẽ gửi lại yêu cầu.
* Bước 2: sau khi kết nối được thiết lập, client sẽ gửi thông điệp yêu cầu chứa tên đường dẫn của các đối tượng ( ví dụ: www.hutech.edu.vn/homepage/index.php) đến server. Server nhận được thông điệp yêu cầu và tiến hành lấy ra các đối tượng được yêu cầu. Sau đó, các đối tượng được đóng gói thành thông điệp trả lời và gửi đến client.
* Bước 3: server đóng kết nối TCP (Lưu ý: server chỉ đóng kết nối TCP khi chắc chắn rằng client nhận được thông điệp trả lời)
* Bước 4: client nhận thông điệp trả lời chứa tập tin HTML và hiển thị các đối tượng.
* Bước 5: lặp lại các bước từ 1-4 cho các đối tượng khác.
* Lưu ý: Trong kết nối HTTP không bền vững cần có một RTT (Round Trip Time) để khởi tạo kết nối TCP. Ngoài ra, cần có một RTT cho thông điệp HTTP yêu cầu và một vài byte đầu tiên của thông điệp HTTP trả lời được trả về.Tổng thời gian truyền tập tin = 2RTT+ thời gian truyền.

Thời gian đáp ứng RTT làthời gian gửi một gói tin cơ bản từ client đến server rồi quay ngược lại. RTTbao gồmđộ trễtruyền gói tin và hàng đợi, độ trễtrongcác bộ định tuyếntrung gian, chuyển mạch vàxử lýgói tin.

Qui trình hoạt động của kết nối HTTP bền vững:

Kết nối bền vững không có pipelining:

Client phát ra yêu cầu mới, chỉ khi đáp ứng trước đó đã nhận xong.

RTT cho mỗi đối tượng tham chiếu.

Kết nối bền vững có pipelining:

Mặc định có trong HTTP/1.1.

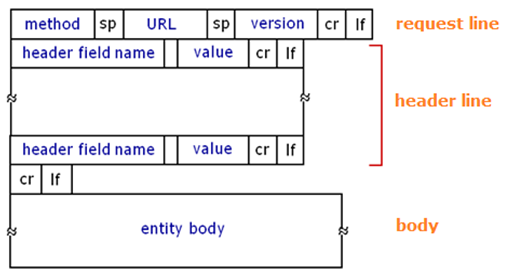
Client gửi yêu cầu ngay sau khi gặp một đối tượng tham chiếu.

Ít nhất 1 RTT cho tất cả đối tượng tham chiếu.

1. ***Thông điệp HTTP***

Thông điệp HTTP được viết bằng văn bản ASCII thông thường, do vậy người dùng có thể đọc được. Có 2 dạng thông điệp HTTP: thông điệp HTTP yêu cầu và thông điệp HTTP trả lời.

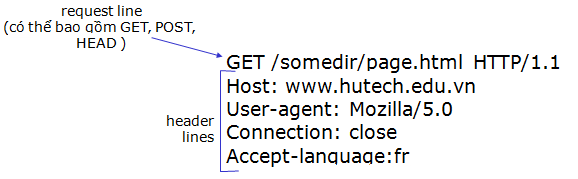
Thông điệp HTTP yêu cầu:



HÌNH 2.6 Cấu trúc chung của thông điệp HTTP yêu cầu

* Request line: dòng đầu tiên của thông điệp HTTP yêu cầu. Request line bao gồm có 3 trường như: cách thức (method), URL, phiên bản (version). Trường cách thức (method) có thể chứa các giá trị khác nhau, bao gồm GET, POST, HEAD. Phần lớn các thông điệp HTTP yêu cầu điều sử dụng phương thức GET. Các phương thức GET được sử dụng khi trình duyệt yêu cầu một đối tượng được xác định trong trường URL. Phương thức POST là những phương thức mà HTTP client sử dụng khi người dùng điền vào một biểu mẫu (form) nào đó, chẳng hạn như người dung muốn nhập một từ khóa nào đó vào google.com đề tìm kiếm thông tin. Nếu phương thức là POST thì Entity body của thông điệp yêu cầu sẽ chứa thông tin mà người dùng đã điền. Phương thức HEAD tương tự như GET, nhưng Khi server nhận được yêu cầu bằng phương thức HEAD, nó sẽ trả về thông điệp HTTP và không chứa đối tượng được yêu cầu.
* Header line: là các dòng tiếp theo.
* Sp: bao gồm các giá trị về khoảng trống.
* Blank line: bao gồm các giá trị điều khiển trở về đầu dòng, xuống hang (cr,lf).
* Entity Body (nếu có): là phần thân của thông điệp HTTP yêu cầu.

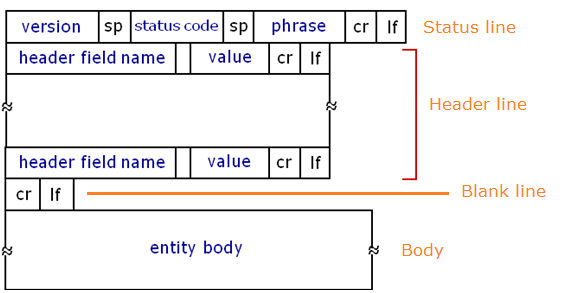
Sau đây là một ví dụ về thông điệp HTTP yêu cầu:



HÌNH 2.7 Ví dụ về thông điệp HTTP yêu cầu

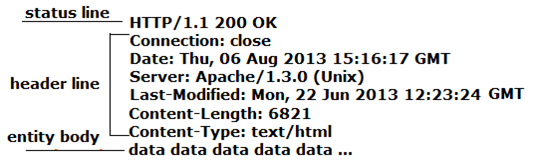
Hình 2.7 giải thích và làm rõ được các thông số sau:

* GET: Trong ví dụ này, trình duyệt đang yêu cầu các đối tượng /somedir/page.html.
* Host www.hutech.edu.vn: chỉ địnhcác servermà trên đó cácđối tượng được lưu trữ.
* Connection: close trình duyệt báo cho máy chủ là nó không muốn sử dụng kết nối liên tục và muốn máy chủ đóng kết nối sau khi gửi các đối tượng được yêu cầu.
* User-agent : chỉ thị loại trình duyệt gửi yêu cầu đến server, ở đây dùng trình duyệt Mozilla/5.0.
* Accept-language (fr): chỉ thị người dùng muốn nhận được phiên bản tiếng Pháp của đối tượng chứa trên server, nếu không server sẽ gửi phiên bản mặc định của đối tượng.
* Thông điệp HTTP trả lời
* Thông điệp HTTP trả lời có ba phần: dòng trạng thái (status line), dòng tiêu đề (header lines), thân thông điệp (entity body). Thân (Body) là thành phần chính của thông điệp. Status line có 3 trường: phiên bản của giao thức (version), mã trạng thái (status code), trạng thái tương ứng (phrase) và các giá trị khoảng trống (sp), điều khiển trở về đầu dòng, xuống hàng (cr,lf).



HÌNH 2.8 Cấu trúc chung của một thông điệpHTTP trả lời

Sau đây là ví dụ về thông điệp HTTP trả lời



HÌNH 2.9 Ví dụ thông điệp HTTP trả lời

Trong ví dụ hình 2.9, cho thấy:

* Status line thể hiện server đang sử dụng HTTP/1.1.
* Header line bao gồm:
* Connection: close báo cho client rằng server sẽ đóng kết kết TCP sau khi gửi thông điệp.
* Date: cho biết thời gian mà thông điệp HTTP trả lời được tạo và gửi bởi server. Đó là thời gian mà server lấy đối tượng từ hệ thống tập tin của nó, chèn vào thông điệp và gửi cho client.
* Server: cho biết đây là Apache Web server, tương tự như User-agent trong thông điệp yêu cầu.
* Last-Modified: cho biết thời gian đối tượng được tạo hay sửa đổi lần cuối.
* Content-Length: cho biết số bytes của đối tượng được gửi.
* Content-Type: cho biết đối tượng trong phần entity body là HTML.
* Entity body chứa đối tượng yêu trong trường hợp ví dụ trong hình 9.6 được đại diện bởi các data data data data…

Mã trạng thái trả lời:

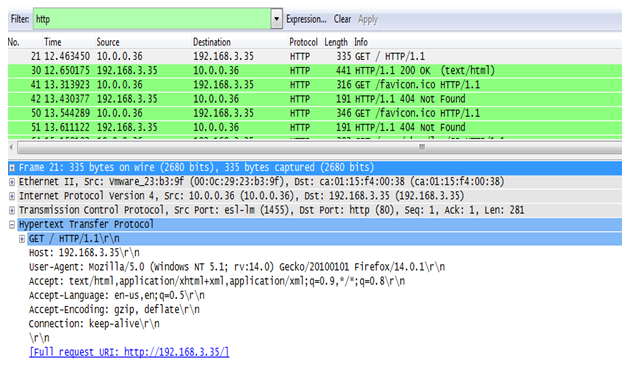
Các giá trị chữ số đầu tiên của mã trang thái (status code) có 5 giá trị:

* 1xx\_Thông tin: khôngđược sử dụng, chỉ dự phòngtrong tương lai.
* 2xx\_Thành công: hành độngđã nhận đượcthành công và đươc chấp nhận.
* 3xx\_Chuyển hướng: hành động tiếp theo phải được thực hiện để hoàn tất yêu cầu.
* 4xx\_Clien lỗi: chứa cú pháp sai có thể không thực hiện được.
* 5xx\_ Server lỗi: các máy server không thực hiện một yêu cầu rõ ràng đối với các yêu cầu hợp lệ.

Một vài mã trạng thái thông dụng thường gặp:

* 200 OK: Yêu cầu thành công
* 301 Moved Permanently: đối tượng yêu cầu đã được chuyển.
* 400 Bad Request: server không hiểu được thông điệp yêu cầu.
* 404 Not Found: đối tượng được yêu cầu không có trong server.
* 505 HTTP Version Not Supported: server không hỗ trợ phiên bản giao thức HTTP này.

1. ***Gói tin HTTP***

****

HÌNH 2.10 Gói tin HTTP

Hình wireshark trên cho thấyHTTP đang sử dụng phiên bản HTTP 1.1 dùng phương thức GET.

* Có các mã trạng thái hồi đáp như:
* 200 OK
* 404 Not Found
* Địa chỉ IP nguồn: 10.0.0.36
* Địa chỉ IP đích: 192.168.3.35
* Cổng HTTP: 80
* Sử dụng phương thức TCP
* Trang web đăng nhập là (host) http://192.168.3.35 không sử dụng DNS để phân giải địa chỉ IP 192.168.3.35.
* User-agent: loại trình duyệt gửi yêu cầu đến server ở đây dùng trình duyệt Mozilla/5.0.
* Accept-language: sử dụng ngôn ngữ tiếng anh (en-us).

# Chương 3: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

1. **Tổng quan về hệ thống**
2. ***Giới thiệu về hệ thống***

Hệ thống được thiết kế dựa trên mô hình client – server, bao gồm:

* Client gửi yêu cầu đến server
* Server tiếp nhận và xử lý yêu cầu, sau đó gửi trả kết quả về cho client

Toàn bộ người dùng đều tương tác trực tiếp với hệ thống

1. ***Cấu trúc hệ thống***

Hệ thống được xây dựng dựa trên kiến trúc mô hình MVC nổi tiếng.



HÌNH 3.1 Mô hình MVC

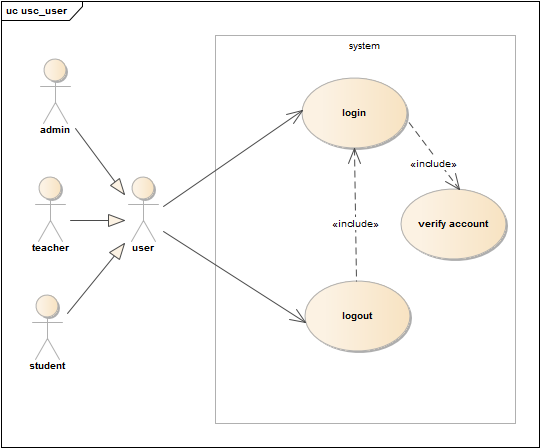
1. **Thiết kế hệ thống**

Để thiết kế được một hệ thống hoàn chỉnh yêu cầu người lập trình phải phân tích và thiết kế một cách chi tiết:

* Yêu cầu người dùng.
* Yêu cầu hệ thống.
* Sơ đồ hoạt động.
* Biểu đồ tuần tự.
* Biểu đồ lớp.
* Biểu đồ luồng dữ liệu.

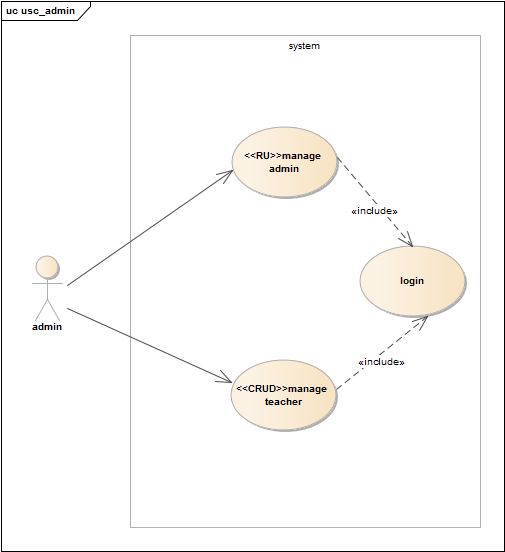
1. ***Biểu đồ use case***

* *Biểu đồ use case người dùng*

****

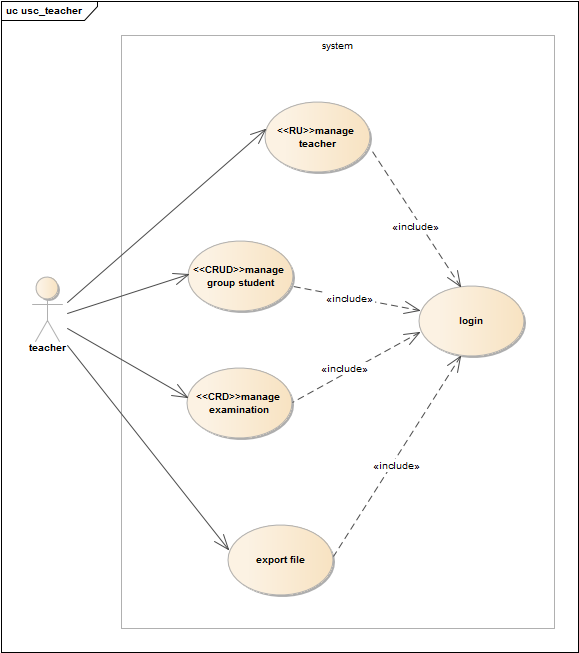
HÌNH 3.2 Biểu đồ use case người dùng

* *Biểu đồ use case quản lý*

****

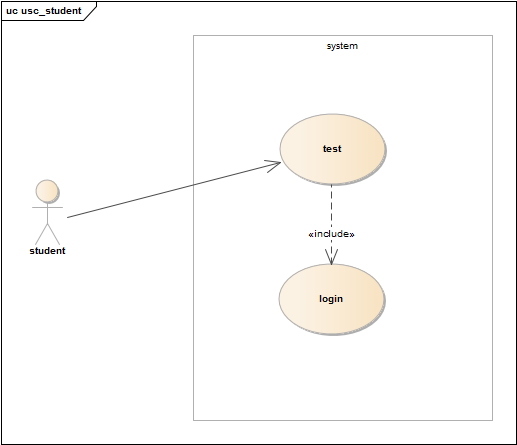
HÌNH 3.3 Biểu đồ use case quản lý

* *Biểu đồ use case giảng viên*

****

HÌNH 3.4 Biểu đồ use case giảng viên

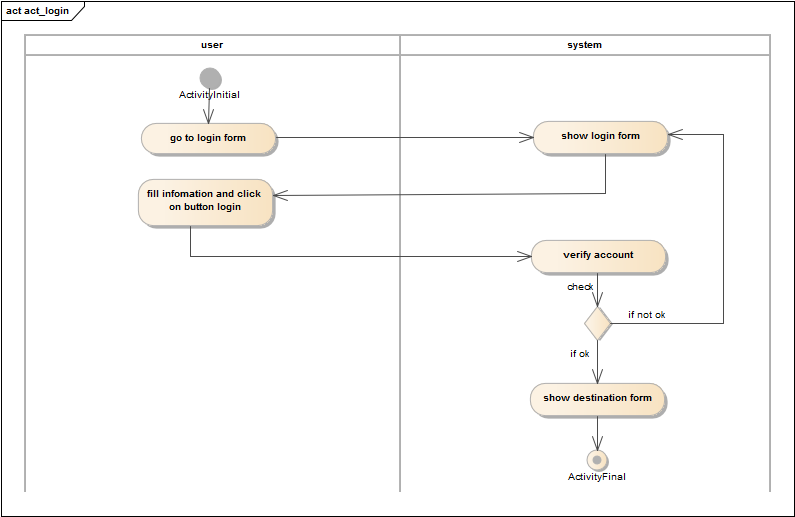
* *Biểu đồ use case sinh viên*

****

HÌNH 3.5 Biểu đồ use case sinh viên

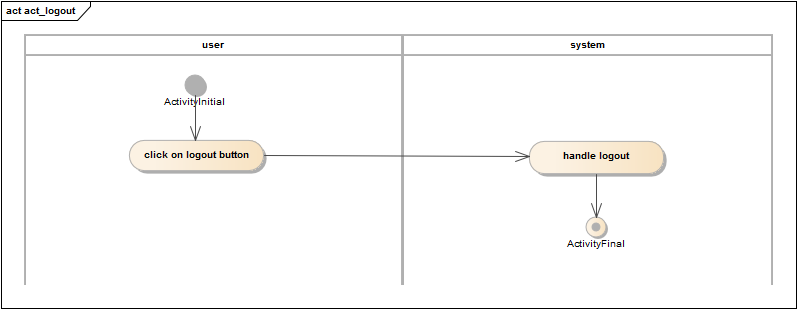
1. ***Biểu đồ hoạt động***

* *Biểu đồ hoạt động cho use case đăng nhập*

****

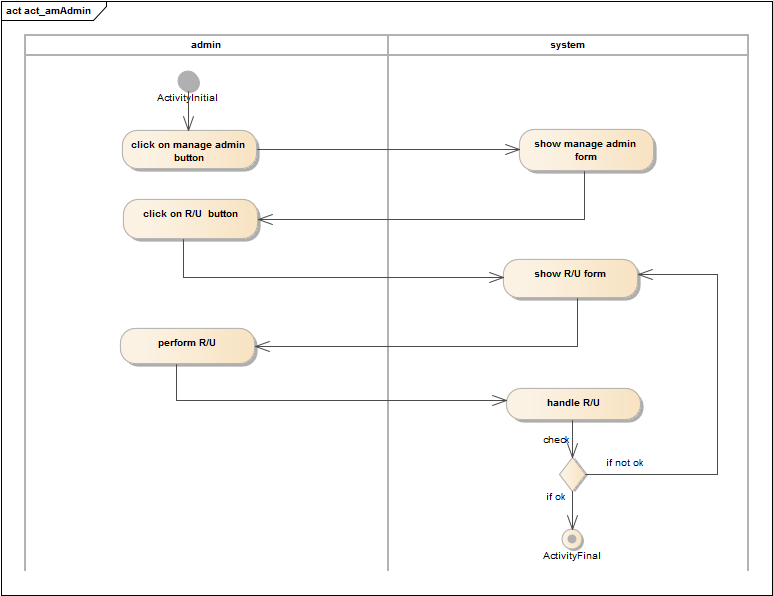
HÌNH 3.6 Biểu đồ hoạt động cho use case đăng nhập

* *Biểu đồ hoạt động cho use case đăng xuất*

****

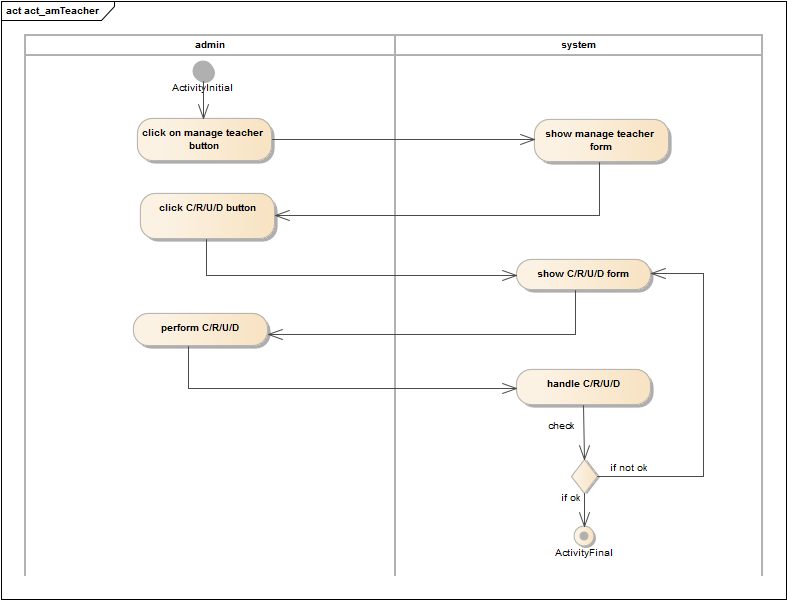
HÌNH 3.7 Biểu đồ hoạt động cho use case đăng xuất

* *Biểu đồ hoạt động cho use case quản lý admin*

****

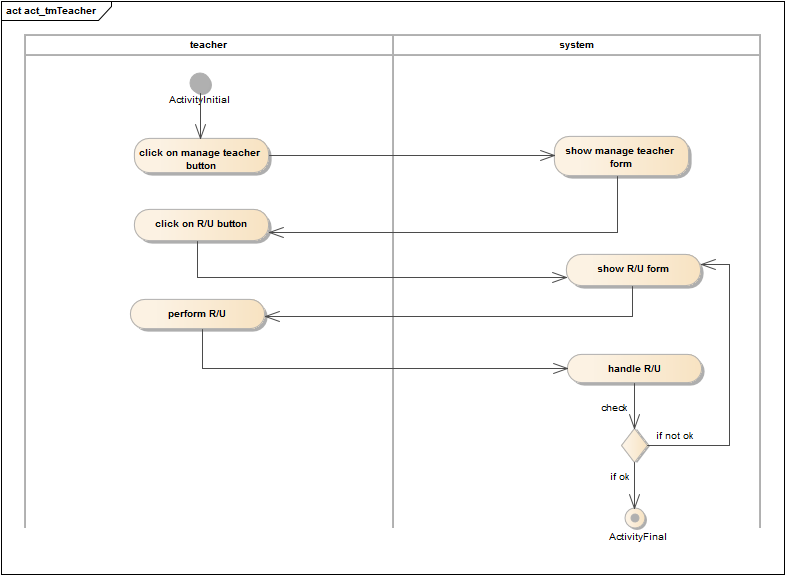
HÌNH 3.8 Biểu đồ hoạt động cho use case quản lý admin

* *Biểu đồ hoạt động cho use case admin – quản lý teacher*

****

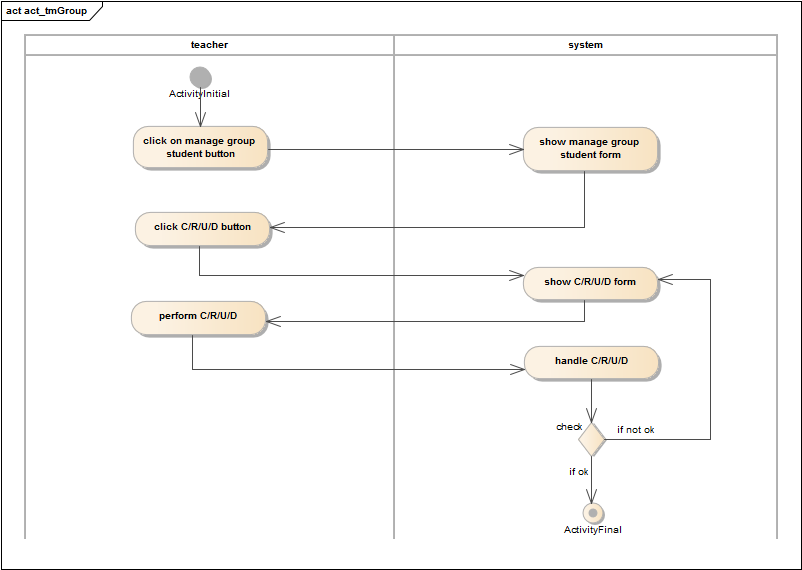
HÌNH 3. Biểu đồ hoạt động cho use case admin – quản lý teacher

* *Biểu đồ hoạt động cho use case teacher – quản lý teacher*

****

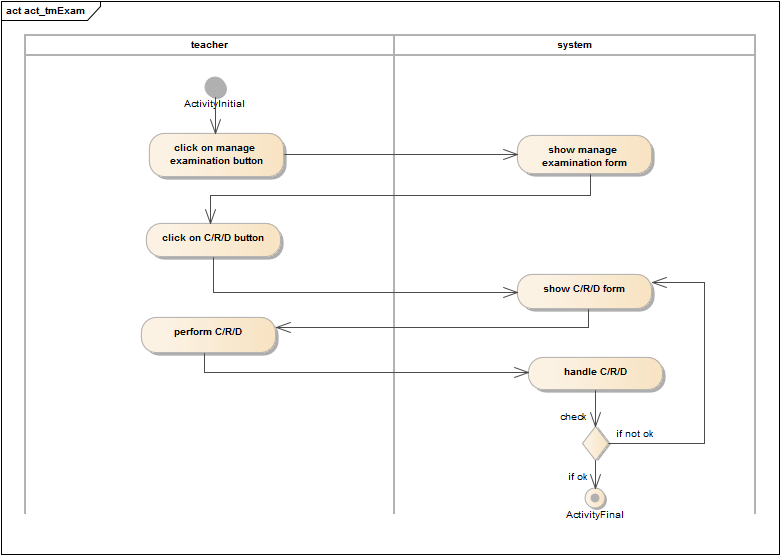
HÌNH 3. Biểu đồ hoạt động cho use case teacher – quản lý teacher

* *Biểu đồ hoạt động cho use case quản lý nhóm sinh viên*

****

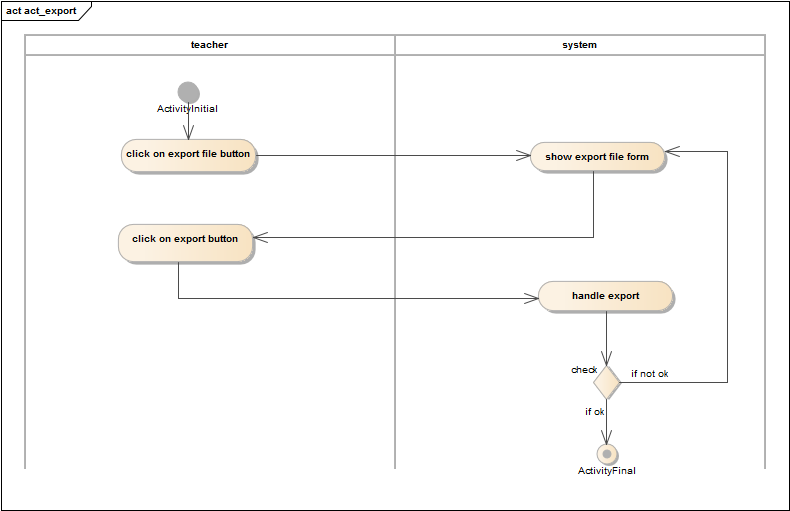
HÌNH 3. Biểu đồ hoạt động cho use case quản lý nhóm sinh viên

* *Biểu đồ hoạt động cho use case quản lý đề thi*

****

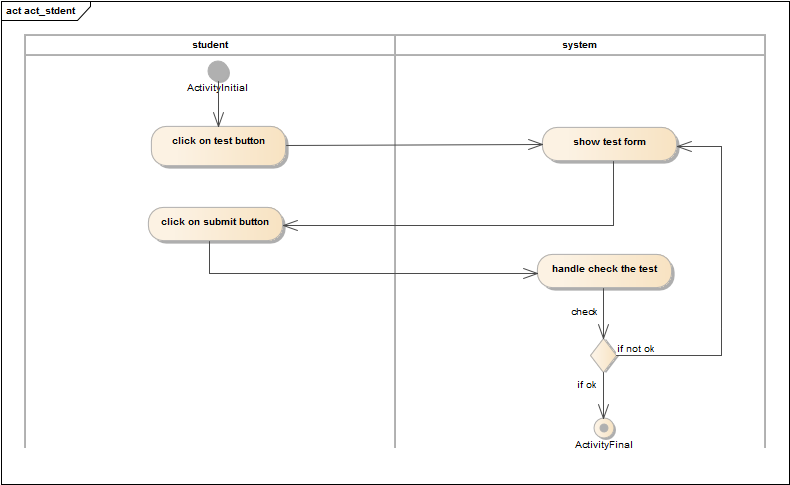
HÌNH 3. Biểu đồ hoạt động cho use case quản lý đề thi

* *Biểu đồ hoạt động cho use case xuất file*

****

HÌNH 3. Biểu đồ hoạt động cho use case xuất file

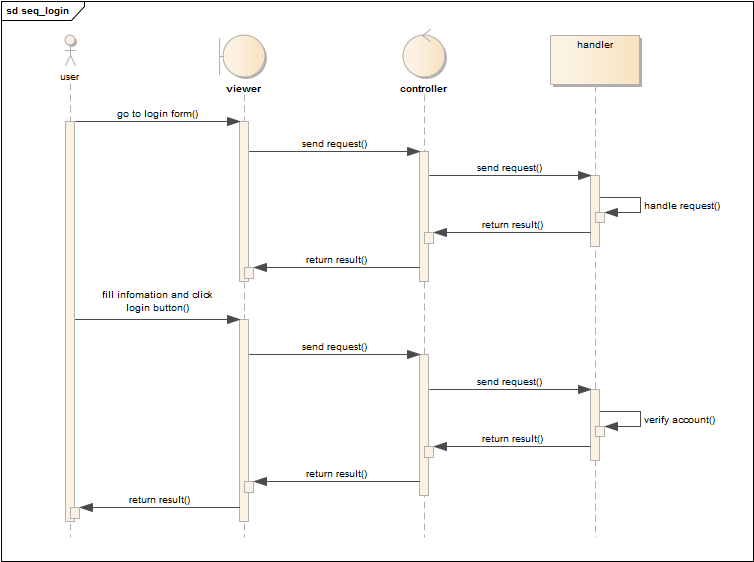
* *Biểu đồ hoạt động cho use case làm bài thi*

****

HÌNH 3. Biểu đồ hoạt động cho use case làm bài thi

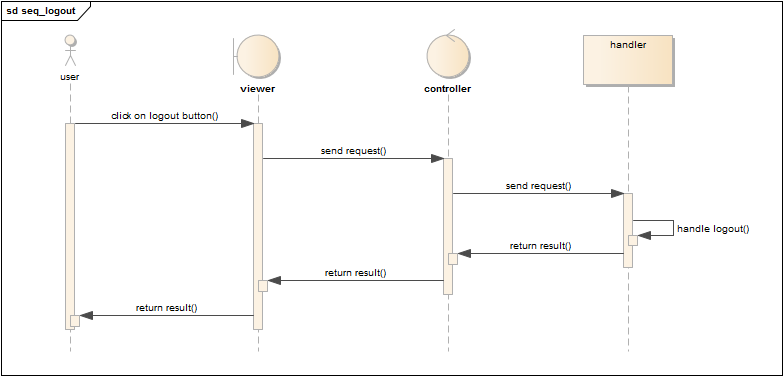
1. ***Biểu đồ tuần tự***

* *Biểu đồ tuần tự cho use case login*

****

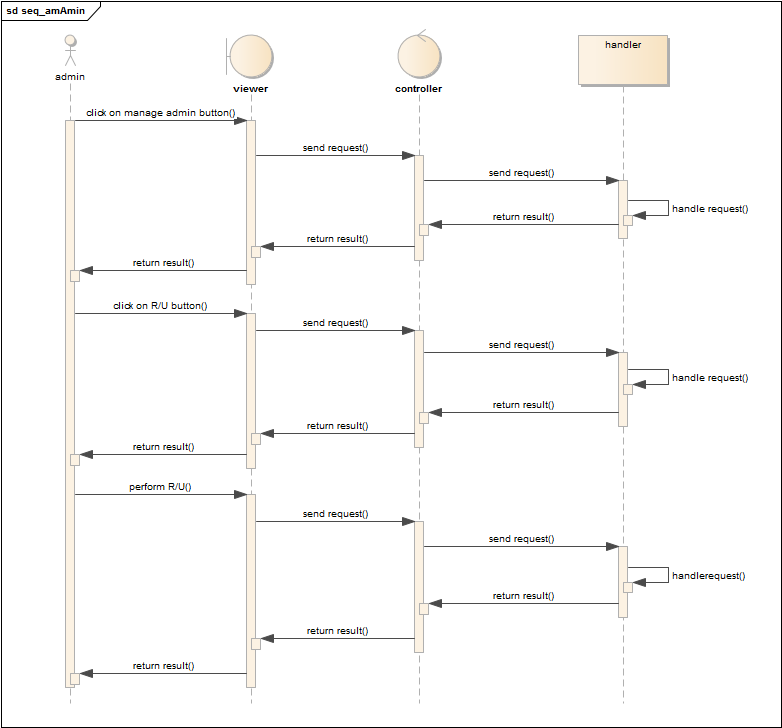
HÌNH 3. Biểu đồ tuần tự cho use case login

* *Biểu đồ tuần tự cho use case logout*

****

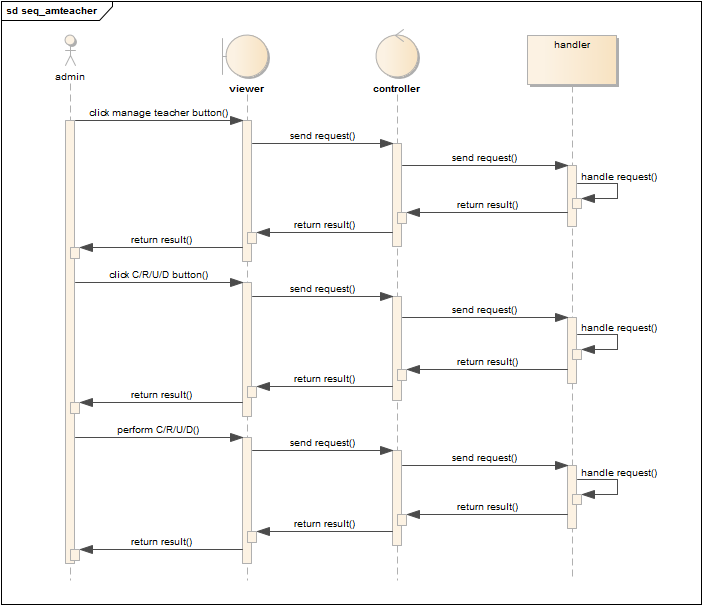
HÌNH 3. Biểu đồ tuần tự cho use case logout

* *Biểu đồ tuần tự cho use case quản lý admin*

****

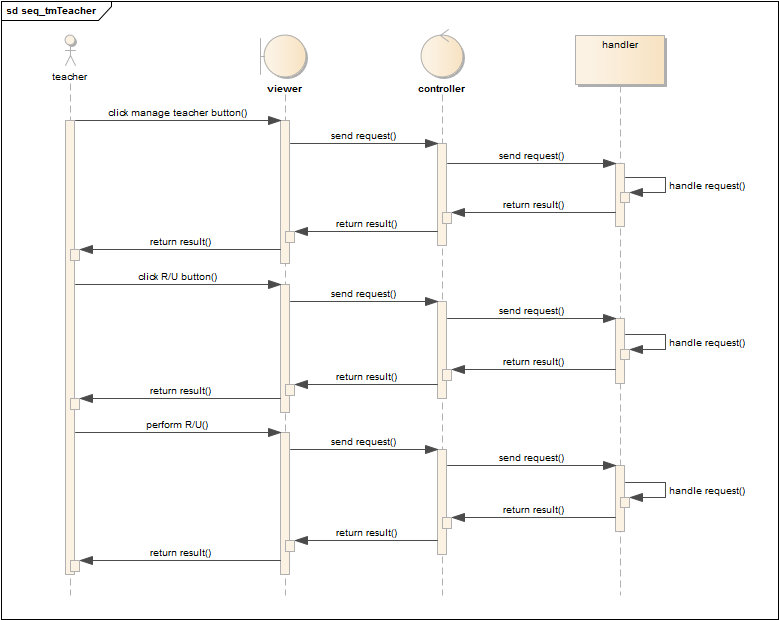
HÌNH 3. Biểu đồ tuần tự cho use case quản lý admin

* *Biểu đồ tuần tự cho use case admin – quản lý teacher*

****

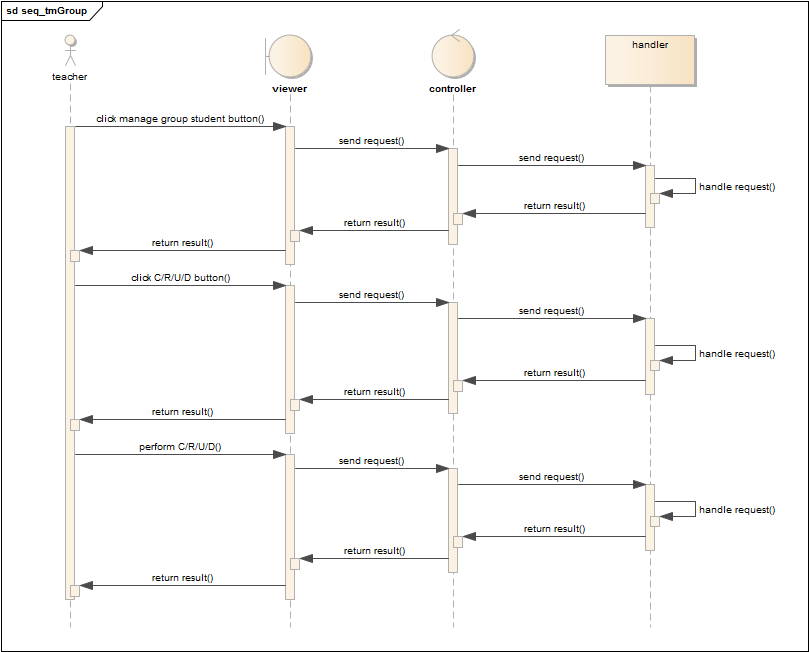
HÌNH 3. Biểu đồ tuần tự cho use case admin – quản lý teacher

* *Biểu đồ tuần tự cho use case teacher – quản lý teacher*

****

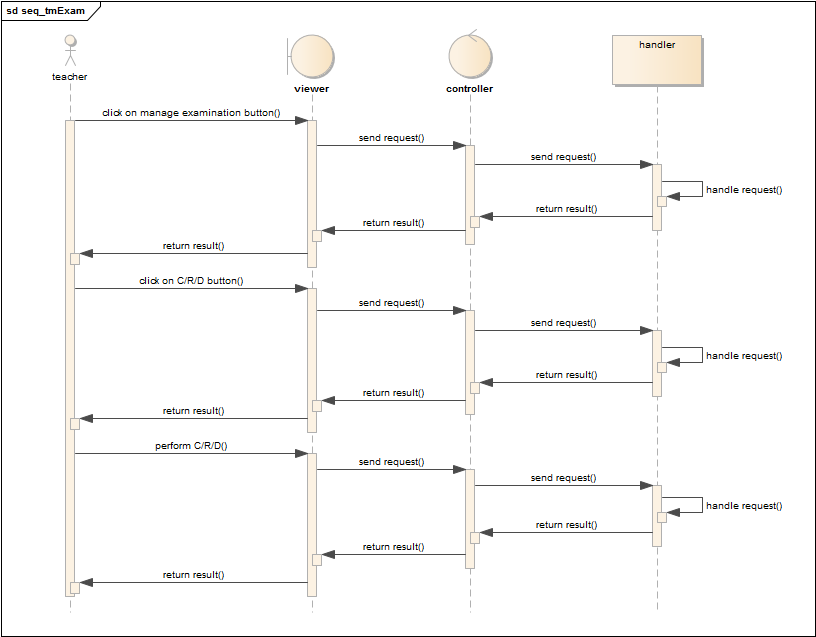
HÌNH 3. Biểu đồ tuần tự cho use case teacher – quản lý teacher

* *Biểu đồ tuần tự cho use case quản lý nhóm sinh viên*

****

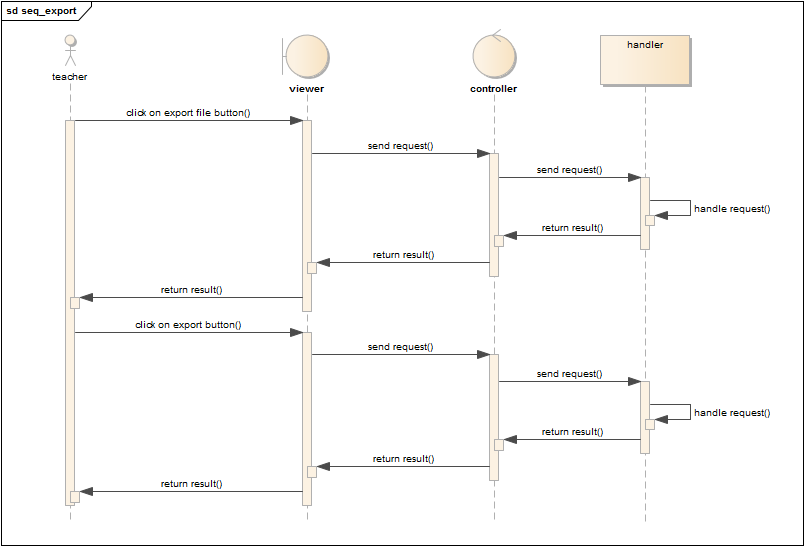
HÌNH 3. Biểu đồ tuần tự cho use case quản lý nhóm sinh viên

* *Biểu đồ tuần tự cho use case quản lý đề thi*

****

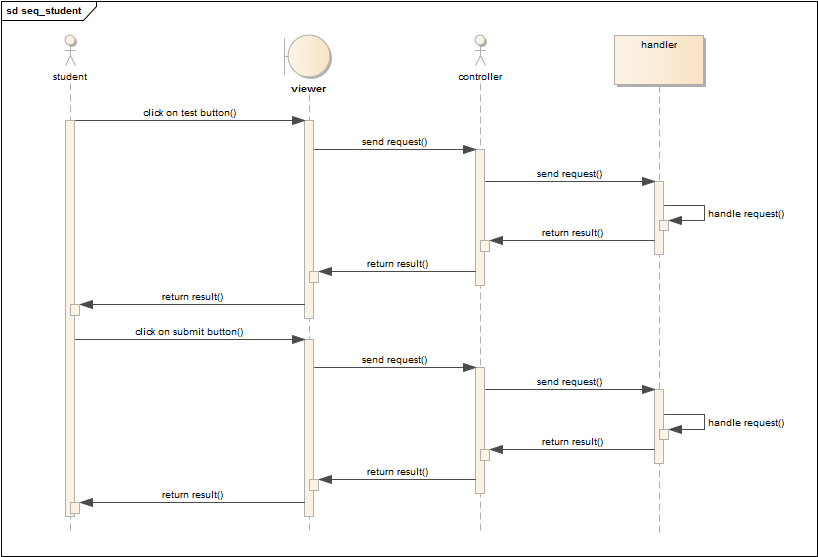
HÌNH 3. Biểu đồ tuần tự cho use case quản lý đề thi

* *Biểu đồ tuần tự cho use case xuất file*

****

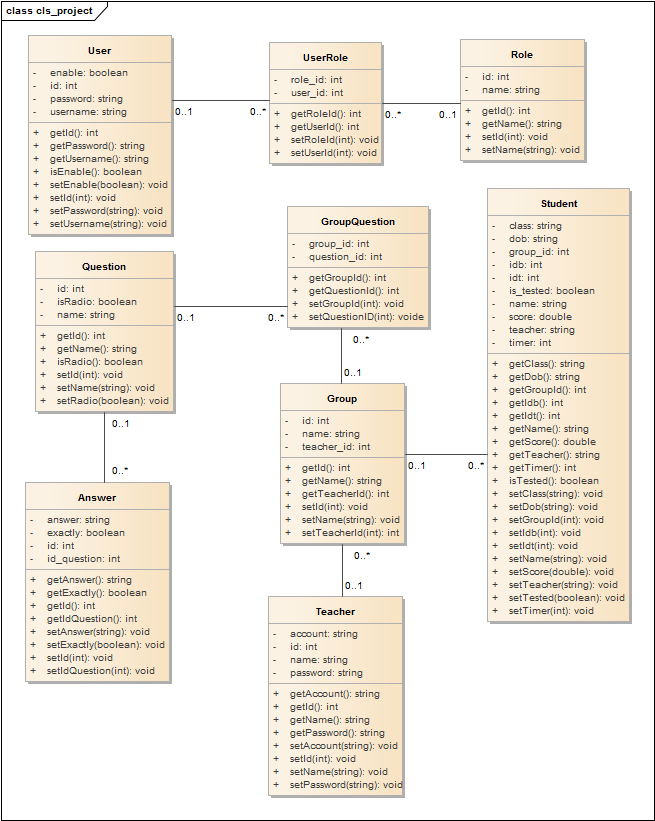
HÌNH 3. Biểu đồ tuần tự cho use case xuất file

* *Biểu đồ tuần tự cho use case làm bài thi*

****

HÌNH 3. Biểu đồ tuần tự cho use case làm bài thi

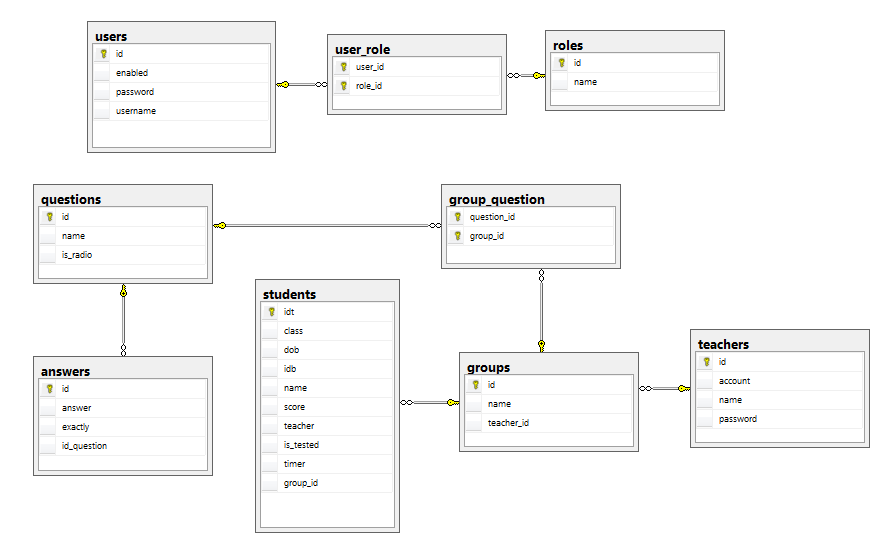
1. ***Biểu đồ lớp***

****

HÌNH 3. Biểu đồ lớp thiết kế

1. **Thiết kế cơ sở dữ liệu**

Từ biểu đồ lớp, ta có được mối liên kết giữa các bảng trong CSDL như hình vẽ sau:



HÌNH 3. Mối liên kết các bảng trong CSDL

Cụ thể ta có chi tiết các bảng như sau:

* *Bảng users*

BẢNG 3. Bảng users

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên trường | Kiểu dữ liệu | Kích thước | Ràng buộc | Diễn giải |
| Id | Integer | 5 | Khóa chính |  |
| Enable | Boolean | 1 | Not null |  |
| Username | String | 30 | Not null |  |
| Password | String | 20 | Not null |  |

* *Bảng user\_role*

BẢNG 3. Bảng user\_role

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên trường | Kiểu dữ liệu | Kích thước | Ràng buộc | Diễn giải |
| User\_id | Integer | 5 | Khóa chính |  |
| Role\_id | Integer | 5 | Khóa chính |  |

* *Bảng roles*

BẢNG 3. Bảng roles

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên trường | Kiểu dữ liệu | Kích thước | Ràng buộc | Diễn giải |
| Id | Integer | 5 | Khóa chính |  |
| Name | String | 30 | Not null |  |

* *Bảng questions*

BẢNG 3. Bảng questions

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên trường | Kiểu dữ liệu | Kích thước | Ràng buộc | Diễn giải |
| Id | Integer | 5 | Khóa chính |  |
| Name | String | 1024 | Not null |  |
| Is\_radio | Boolean | 1 | Not null |  |

* *Bảng answers*

BẢNG 3. Bảng answers

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên trường | Kiểu dữ liệu | Kích thước | Ràng buộc | Diễn giải |
| Id | Integer | 5 | Khóa chính |  |
| Answer | String | 1024 | Not null |  |
| Exactly | Boolean | 1 | Not null |  |
| Id\_question | Integer | 5 | Khóa ngoại |  |

* *Bảng group\_question*

BẢNG 3. Bảng group\_question

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên trường | Kiểu dữ liệu | Kích thước | Ràng buộc | Diễn giải |
| Group\_id | Integer | 5 | Khóa chính |  |
| Question\_id | Integer | 5 | Khóa chính |  |

* *Bảng groups*

BẢNG 3. Bảng groups

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên trường | Kiểu dữ liệu | Kích thước | Ràng buộc | Diễn giải |
| Id | Integer | 5 | Khóa chính |  |
| Name | String | 20 | Not null |  |
| Teacher\_id | Integer | 5 | Khóa ngoại |  |

* *Bảng students*

BẢNG 3. Bảng students

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên trường | Kiểu dữ liệu | Kích thước | Ràng buộc | Diễn giải |
| Idt | Integer | 5 | Khóa chính |  |
| Idb | Integer | 5 | Not null |  |
| Dob | String | 15 | Not null |  |
| Name | String | 30 | Not null |  |
| Class | String | 10 | Not null |  |
| Group\_id | Integer | 5 | Khóa ngoại |  |
| Score | Double | 5 | Not null |  |
| Timer | Integer | 5 | Not null |  |
| Is\_tested | Boolean | 1 | Not null |  |
| Teacher | String | 30 | Not null |  |

* *Bảng teachers*

BẢNG 3. Bảng teachers

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên trường | Kiểu dữ liệu | Kích thước | Ràng buộc | Diễn giải |
| Id | Integer | 5 | Khóa chính |  |
| Name | String | 20 | Not null |  |
| Account | String | 30 | Not null |  |
| Password | String | 20 | Not null |  |

# Chương 4: TRIỂN KHAI VÀ ĐÁNH GIÁ KẾT QUẢ

*Triển khai và đánh giá kết quả chính là thước đo sự thành công của một ứng dụng, nó giúp cho người dùng hiểu được cách thức triển khai, cài đặt và sử dụng ứng dụng đó như thế nào. Sự thành công của ứng dụng chính là quá trình cài đặt và kiểm duyệt kết quả.*

1. **Các thuật toán triển khai trong ứng dụng**

Các thuật toán dùng trong ứng dụng đa phần được xây dựng dựa trên những cấu trúc dữ liệu của package “java.util.\*” trong ngôn ngữ Java. Trong đó sử dụng các cấu trúc dữ liệu:

* List: Arraylist, LinkedList
* Set: HashSet
* Map: HashMap

1. ***Thuật toán xáo trộn đề thi***

* B1: Lấy toàn bộ các câu hỏi trong DB lưu vào một đối tượng List (LinkedList)
* B2: Dùng vòng lặp for duyệt list này theo theo thứ tự kích thước giảm dần:

for i = list.size; i > 0; i--

* B2.1: Tạo một số ngẫu nhiên trong khoảng 0 – i
* B2.2: Lấy đối tượng Question theo số ngẫu nhiên này – là thứ tự trong list
* B2.3: Dùng vòng lặp for duyệt list – là đối tượng List<Answer> được lấy ra từ đối tượng Question đã lấy ở B2.2 (List<Answer> là thuộc tính của Question) theo thứ tự kích thước giảm dần:

for j = list.size; j > 0; j--

* B2.3.1: Tạo một số ngẫu nhiên trong khoảng 0 – j
* B2.3.2: Lấy đối tượng Answer theo số ngẫu nhiên này – là thứ tự trong list chứa các answer.
* B2.3.3: Xóa đối tượng Answer này trong list chứa nó
* B2.4: Xóa đối tượng Question này trong list chứa nó

1. ***Thuật toán tính điểm thi sinh viên***

Có 2 loại câu hỏi:

* Câu hỏi 1 đáp án
* Câu hỏi nhiều đáp án

Đối với loại câu hỏi nhiều đáp án, việc tính điểm cho câu hỏi ấy có tính bù trừ

* B1: Khởi tạo

diem\_tren\_mot\_cau = 10/so\_cau\_hoi

diem\_thi = 0

* B2: Duyệt qua đề thi mà sinh viên đã làm
* B2.1: Nếu đề thi là loại một đáp án

- B2.1.1: Nếu đáp án sinh viên chọn là đáp án đúng thì diem\_thi = diem\_thi + diem\_tren\_mot\_cau

* B2.2: Nếu đề thi là loại nhiều đáp án

- B2.1.1: Xem xét các đáp án sinh viên chọn có bao nhiêu đáp án đã chọn là đúng, bao nhiêu đáp án đã chọn là sai, sau đó tính điểm bằng cách:

diem\_thi = diem\_thi + (so\_dap\_an\_chon\_dung – so\_dap\_an\_chon\_sai)/ tong\_cac\_dap\_an\_dung\_ban\_dau \* diem\_tren\_mot\_cau

1. **Yêu cầu hệ thống**

*Một ứng dụng được thực thi yêu cầu phải có các môi trường làm việc và yêu cầu hệ thống để ứng dụng đó có thể thực thi được nên môi trường cài đặt ứng dụng được chú trọng.*

1. ***Các công cụ sử dụng***

* Jdk > 1.7
* SQL Server
* Gradle > 2.3

1. ***Yêu cầu cấu hình máy tính***

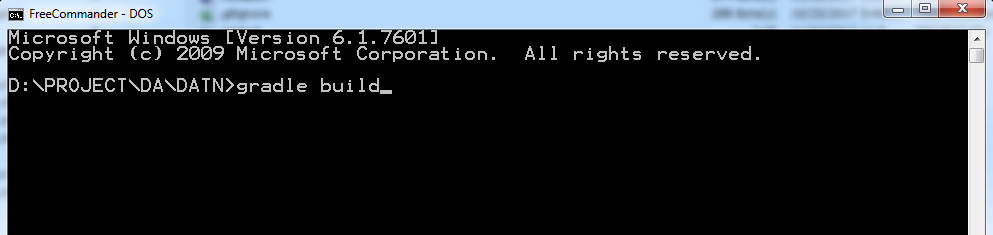
Cấu hình tối thiểu:

* CPU: Intel Core i3.
* Tốc độ máy: 1.80GHz (3MB L3 cache).
* Dung lượng Memory: 2GB

1. **Minh họa quy trình build và run ứng dụng**

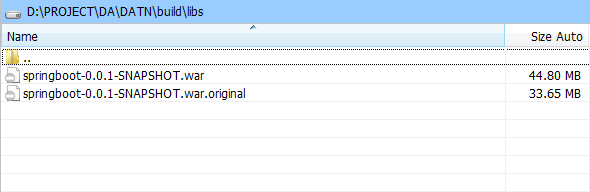
Giả sử thư mục gốc – nơi chứa source code của chúng là “D:\PROJECT\DATN”

* Ta sẽ dùng lệnh *gradle build* để build source



HÌNH 4. Hướng dẫn build source code

* Kết quả ta sẽ có được một file với định dạng là *\*.war*, ví dụ:



HÌNH 4. Kết quả sau khi build source code

Ở trên hình chính là file *springboot-0.0.1-SNAPSHOT.war*

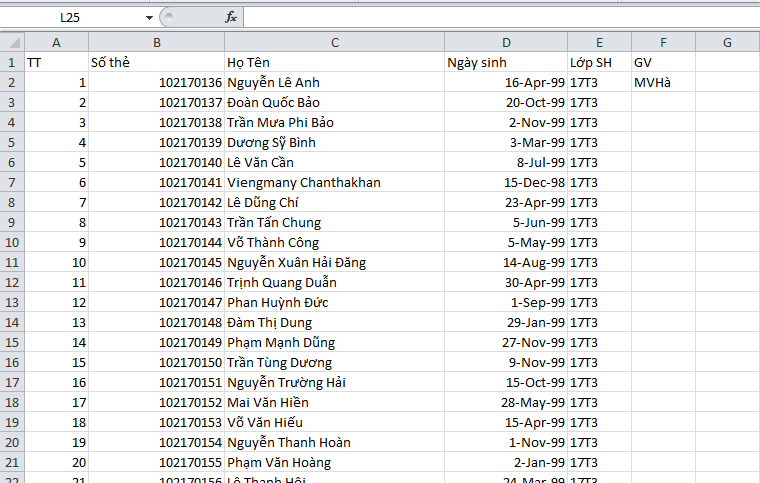
* Tiếp theo chúng ta sẽ run file *.war* này bằng lệnh java như sau



HÌNH 4. Hướng dẫn run ứng dụng

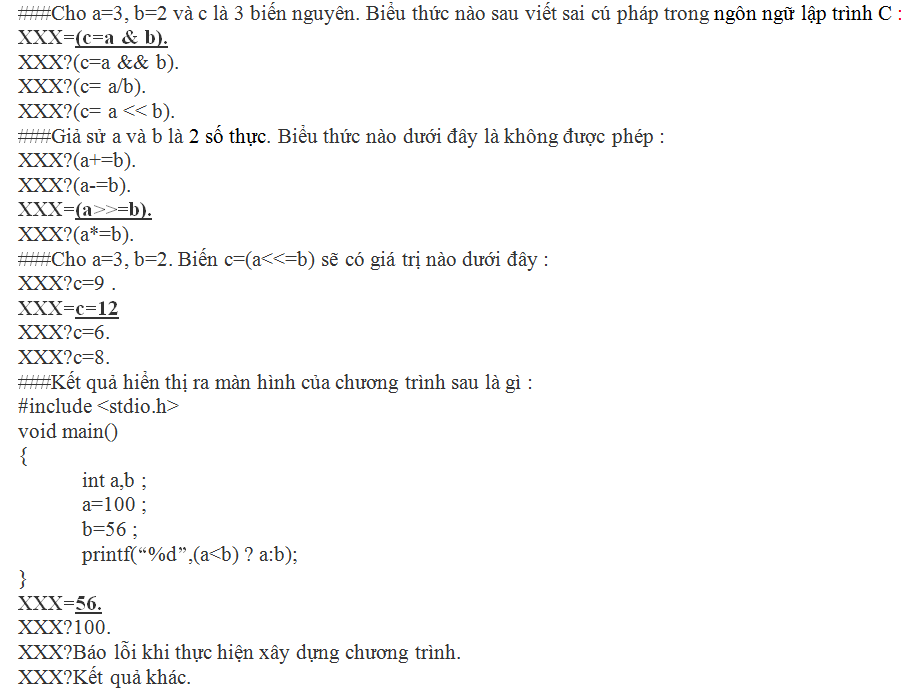
1. **Đánh giá kết quả**

* Đầu tiên ta chuẩn bị 2 file đầu vào:
* File excel chứa thông tin các nhóm thi có định dạng như sau:



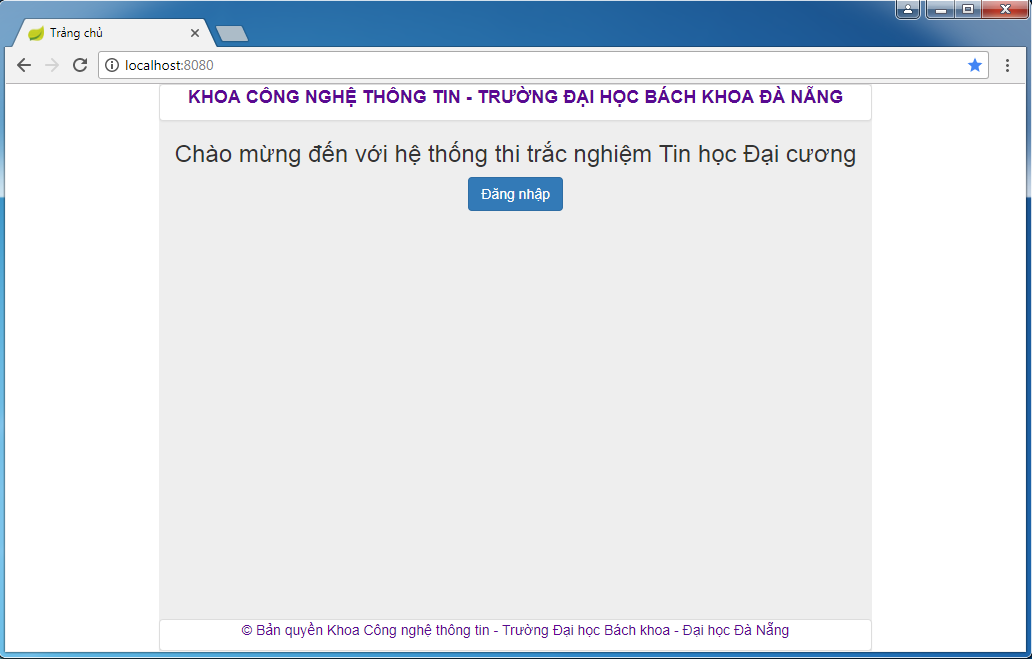
HÌNH 4 Một phần của file excel chứa thông tin nhóm sinh viên

* File word chứa các câu hỏi của đề thi có định dạng như sau:



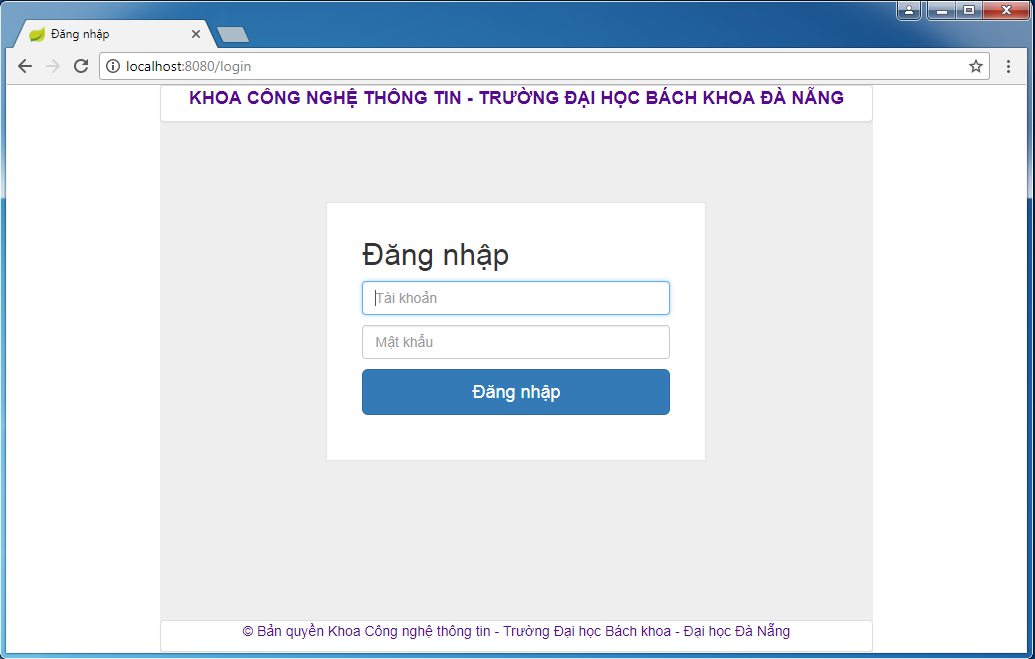
HÌNH 4. Một phần của file word chứa đề thi

* Ta vào trình duyệt và gõ link: “*localhost:8080*”, giao diện trang chủ sẽ xuất hiện như sau:



HÌNH 4. Màn hình trang chủ của ứng dụng

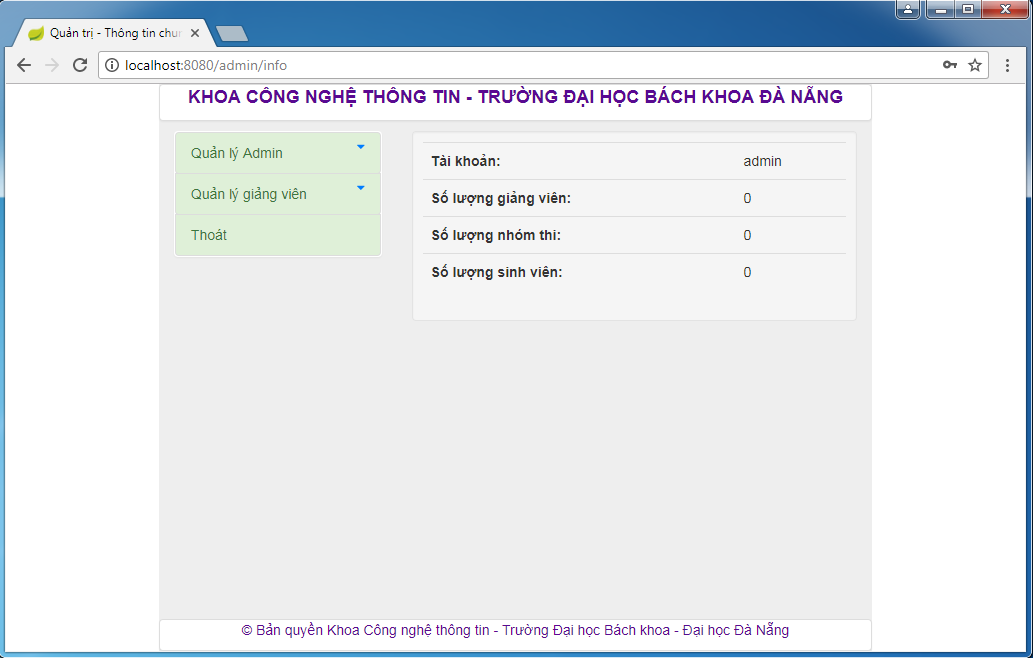
* Sau đó ta click vào button đăng nhập, được giao diện như sau:



HÌNH 4. Màn hình đăng nhập của ứng dụng

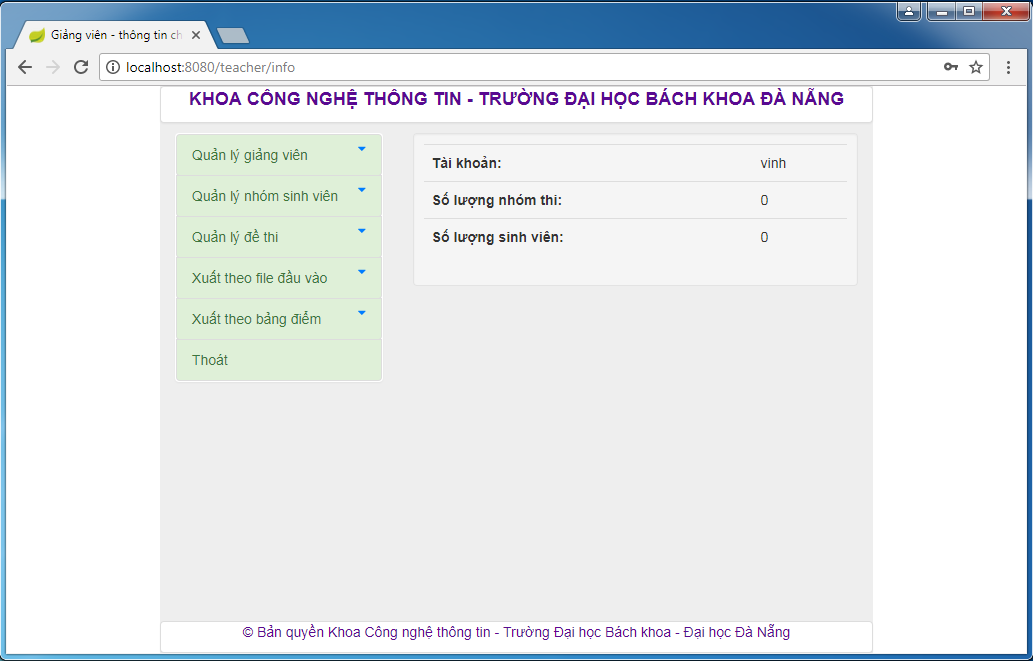
Ở đây, tùy vào tài khoản mà người dùng sẽ có các quyền: admin, teacher, student.

* Hình sau là giao diện chính của người dùng admin



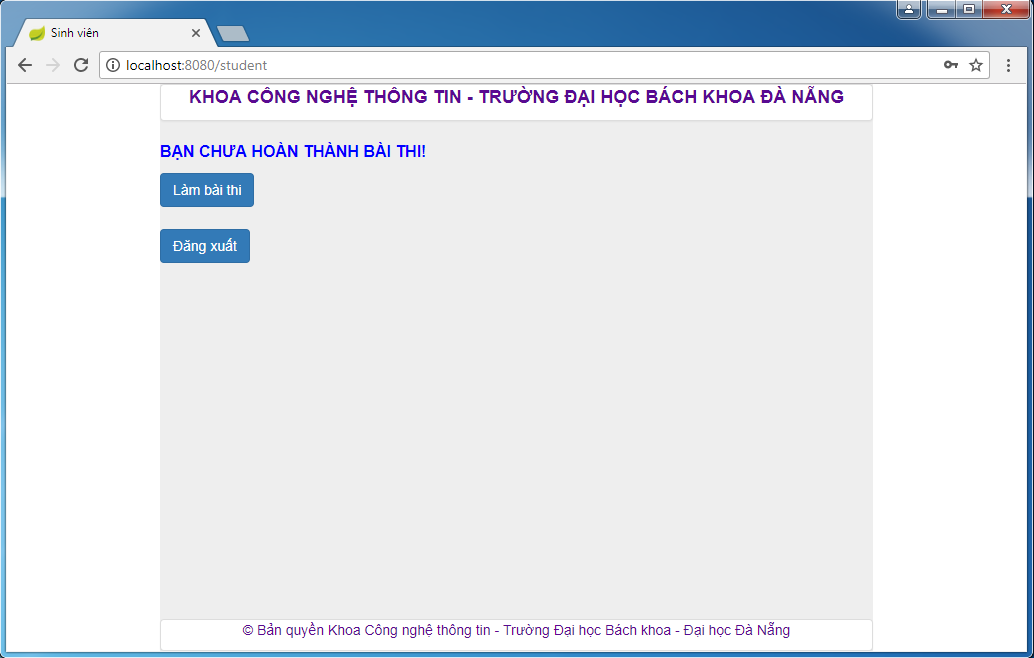
HÌNH 4. Giao diện chính của người dùng admin

* Đây là hình ảnh giao diện chính của người dùng teacher



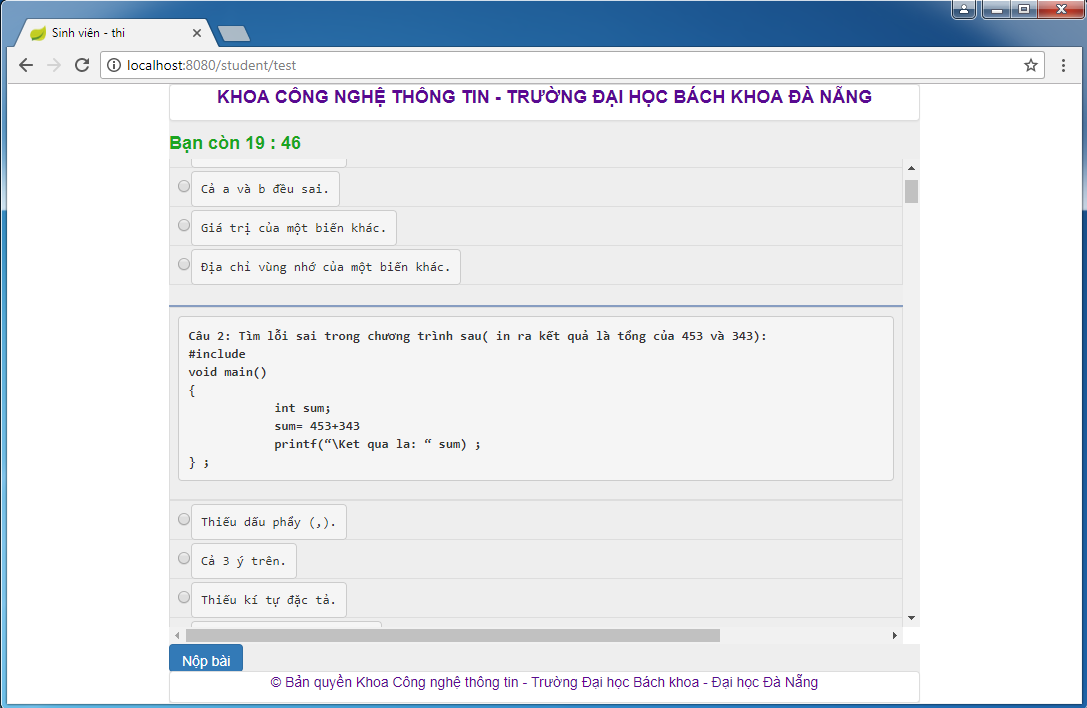
HÌNH 4. Giao diện chính của người dùng teacher

* Đây là hình ảnh giao diện chính của người dùng student



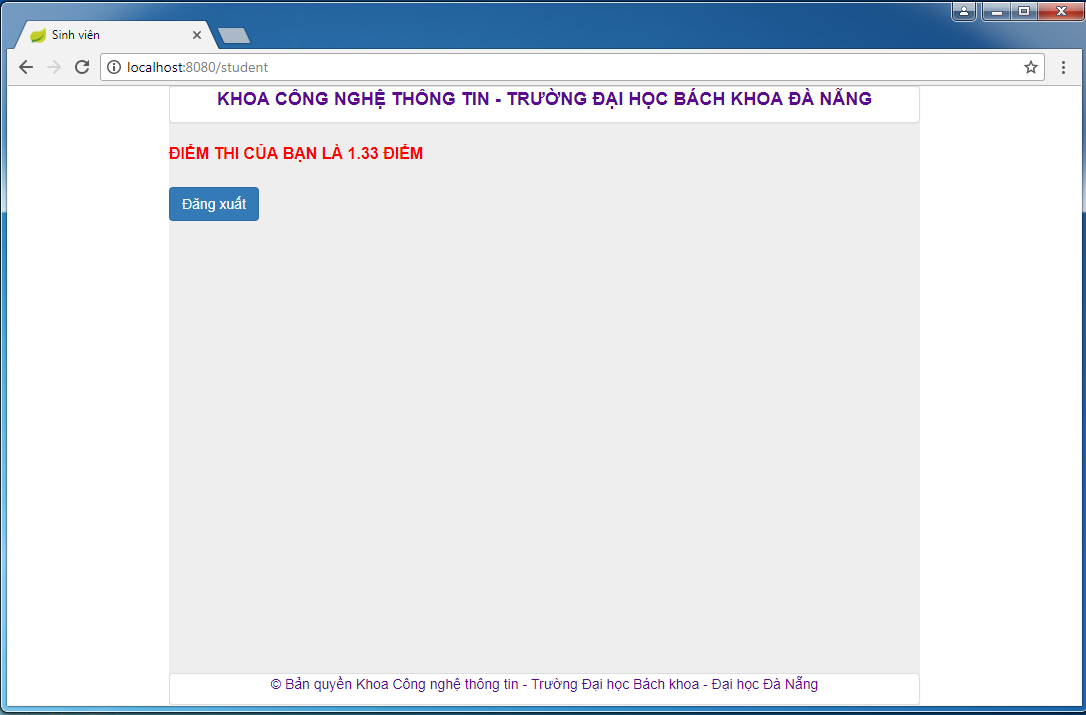
HÌNH 4. Giao diện chính của người dùng student

* Đây là giao diện khi sinh viên làm bài thi



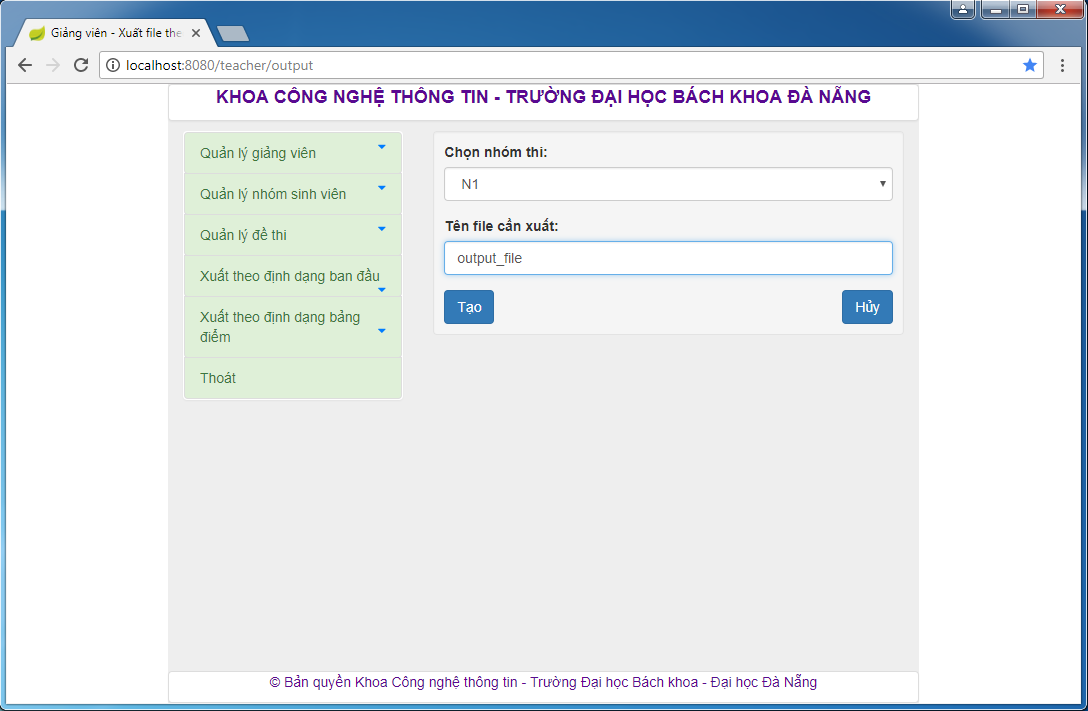
HÌNH 4. Màn hình sinh viên làm bài thi

* Đây là giao diện khi sinh viên click vào button “Nộp bài”



HÌNH 4. Màn hình sau khi sinh viên đệ trình bài thi

* Sau khi các sinh viên đã thực hiện xong bài kiểm tra, giảng viên có thể xuất kết quả theo nhóm ra file excel tùy theo định dạng (**theo file đầu vào** hoặc **theo bảng điểm**) như sau:



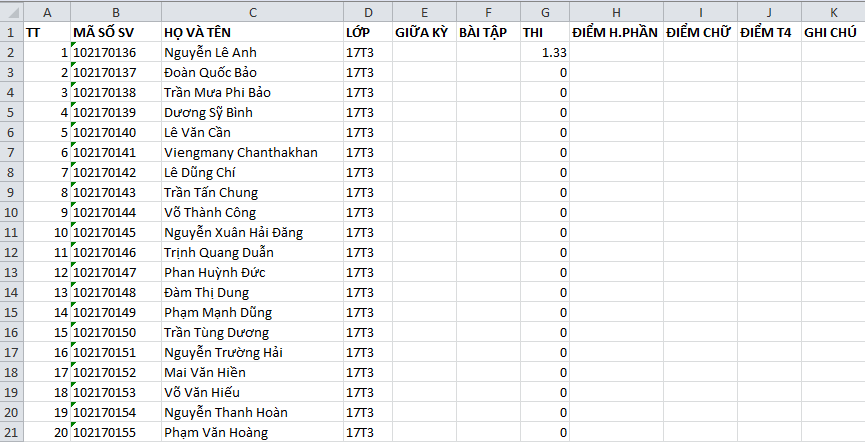
HÌNH 4. Màn hình dùng để xuất kết quả ra file excel

* Kết quả là ta có được file excel với định dạng dựa theo file đầu vào như sau:



HÌNH 4. Kết quả xuất ra file excel theo định dạng file đầu vào

* Hay ta sẽ có được file excel với định dạng theo bảng điểm như sau:



HÌNH 4. Kết quả xuất ra file excel theo định dạng bảng điểm

# KẾT LUẬN

* **Kết quả đạt được**
* **Về mặt lý thuyết**
* Hiểu được cơ chế hoạt động của các bộ giao thức: TCP/IP và HTTP
* Hiểu được kiến trúc Spring MVC cùng các module của Spring framwork: Spring JPA, Spring Security, Spring Boots…
* **Về mặt ứng dụng**
* Xây dựng thành công website quản lý thi và chấm trắc nghiệm Tin học đại cương cho trường Đại học Bách khoa Đà Nẵng
* Giúp giảm thiểu tối đa các thao tác thủ công trong việc tổ chức thi và chấm thi theo hình thức trắc nghiệm
* Xây dựng được ứng dụng với độ chính xác gần như tuyệt đối
* Hệ thống hoạt động khá ổn định
* **Hướng phát triển**
* Hiện tại ứng dụng mới chỉ xử lý các file excel ở định dạng \*.xls và các file word ở định dạng .docx. Do đó, cần phát triển thêm các tính năng để có thể xử lý cho tất cả các định dạng của file excel và file word.
* Cần phát triển thêm nhiều tính năng để thuận tiện trong việc quản lý sinh viên.

# TÀI LIỆU THAM KHẢO

1. http://www.springboottutorial.com/
2. http://www.baeldung.com/spring-mvc-excel-files/
3. https://docs.spring.io/spring/docs/current/spring-framework-reference/