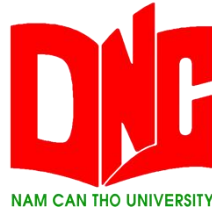


**TRƯỜNG ĐẠI HỌC NAM CẦN THƠ**  
**KHOA CÔNG NGHỆ - KỸ THUẬT**



**LÊ NGUYỄN NHẬT DŨY**  
**CHUNG VINH HIỀN**

**XÂY DỰNG PHẦN MỀM THI TRẮC NGHIỆM  
QUA MẠNG LAN CHO SINH VIÊN KHOA  
KỸ THUẬT - CÔNG NGHỆ**

**ĐỒ ÁN CHUYÊN NGÀNH**

**Ngành Công nghệ Thông tin**

**Mã số Ngành: 7480201**

**Tháng 11-2020**

**TRƯỜNG ĐẠI HỌC NAM CẦN THƠ**  
**KHOA CÔNG NGHỆ - KỸ THUẬT**

**LÊ NGUYỄN NHẬT DŨY**

**MSSV: 176874**

**CHUNG VINH HIỂN**

**MSSV: 177089**

**XÂY DỰNG PHẦN MỀM THI TRẮC NGHIỆM  
QUA MẠNG LAN CHO SINH VIÊN KHOA  
KỸ THUẬT - CÔNG NGHỆ**

**ĐỒ ÁN CHUYÊN NGÀNH**

**Ngành Công nghệ Thông tin**

**Mã số Ngành: 7480201**

**GIẢNG VIÊN HƯỚNG DẪN**

**THS. PHAN THỊ XUÂN TRANG**

**Tháng 11-2020**

## CHẤP THUẬN CỦA HỘI ĐỒNG

Đề án “Xây dựng phần mềm thi trắc nghiệm qua mạng LAN cho sinh viên khoa kỹ thuật - công nghệ”, do sinh viên Lê Nguyễn Nhật Dũy và sinh viên Chung Vinh Hiền thực hiện dưới sự hướng dẫn của Ths. Phan Thị Xuân Trang. Đề án đã báo cáo và được Hội đồng chấm khóa luận thông qua ngày .....

**Ủy viên**

(Ký tên)

**Thư ký**

(Ký Tên)

-----  
**GHI CHÚC DANH, HỌ, TÊN**

-----  
**GHI CHÚC DANH, HỌ, TÊN**

**Phản biện 1**

(Ký tên)

**Phản biện 2**

(Ký tên)

-----  
**GHI CHÚC DANH, HỌ, TÊN**

-----  
**GHI CHÚC DANH, HỌ, TÊN**

**Cán bộ hướng dẫn**

(Ký tên)

**Chủ tịch Hội đồng**

(Ký tên)

-----  
**GHI CHÚC DANH, HỌ, TÊN**

-----  
**GHI CHÚC DANH, HỌ, TÊN**

## **LỜI CẢM TẠ**

Lời đầu tiên chúng em xin chân thành cảm ơn đến ThS. Phan Thị Xuân Trang. Trong thời gian qua, cô đã rất tận tình hướng dẫn để chúng em hoàn thành đồ án này.

Chúng em xin cảm ơn quý thầy cô Khoa Kỹ thuật – Công nghệ Trường Đại học Nam Cần Thơ đã tận tâm giảng dạy và truyền đạt cho chúng em những kiến thức quý báu để chúng em hoàn thiện chương trình học tại trường. Đồng thời, chúng em cũng rất cảm ơn Chủ tịch Hội đồng quản trị, Ban Giám hiệu Trường Đại học Nam Cần Thơ đã tạo điều kiện về thời gian, không gian để chúng em có thể hoàn thành nhiệm vụ học tập của mình.

Mặc dù, chúng em đã có rất nhiều cố gắng nỗ lực để hoàn thành đồ án nhưng vẫn không tránh khỏi những thiếu sót. Kính mong nhận được sự đóng góp ý kiến của quý thầy cô.

Chúng em xin chân thành cảm ơn!

Cần thơ, ngày ..... tháng ..... năm 2020.

**Người thực hiện 1**

**Người thực hiện 2**

## **LỜI CAM ĐOAN**

Chúng em xin cam kết đồ án này được hoàn thành dựa trên các kết quả nghiên cứu của chúng em và các kết quả nghiên cứu này chưa được dùng cho bất cứ đồ án nào khác.

Cần thơ, ngày ..... tháng ..... năm 2020

**Người thực hiện 1**

**Người thực hiện 2**

## This image shows a full page of white paper with horizontal dotted lines. The lines are evenly spaced and run across the entire width of the page, providing a guide for handwriting or typing. There are no margins, text, or other markings on the page.

**Giảng viên hướng dẫn**

## This image shows a full page of a document template designed for handwritten notes or essays. It features approximately 28 evenly spaced, thin grey horizontal lines across the entire width of the page. The margins are consistent on all sides, providing ample space for writing. There are no pre-printed questions, headings, or other markings on the page.

**Giảng viên phản biện**

## MỤC LỤC

|  |    |
|--|----|
| CHƯƠNG 1: GIỚI THIỆU.....  | 1  |
| 1.1 Lý do chọn đề tài.....   | 1  |
| 1.2 Mục tiêu nghiên cứu.....   | 1  |
| 1.3 Đối tượng, phạm vi nghiên cứu.....                               | 2  |
| 1.4 Phương pháp nghiên cứu.....                                      | 2  |
| 1.5 Nội dung.....  | 3  |
| CHƯƠNG 2: CƠ SỞ LÝ LUẬN VÀ PHƯƠNG PHÁP NGHIÊN CỨU.....               | 4  |
| 2.1 Tổng quan về trắc nghiệm.....                                    | 4  |
| 2.2 Tổng quan về mạng LAN.....                                       | 9  |
| 2.3 Tổng quan về hệ thống thông tin.....                             | 12 |
| 2.4 Tổng quan về hệ quản trị cơ sở dữ liệu Microsoft SQL Server..... | 14 |
| 2.5 Tổng quan về môi trường phát triển Visual Studio Code.....       | 15 |
| 2.6 Tổng quan về ASP.Net Core.....                                   | 17 |
| 2.7 Tổng quan về Entity Framework.....                               | 18 |
| 2.8 Tổng quan về Electron.....                                       | 21 |
| 2.9 Tổng quan về ReactJS.....  | 24 |
| 2.10 RESTful API.....  | 26 |
| 2.11 Tổng quan về PowerDesigner.....                                 | 27 |
| 2.12 Tổng quan về ngôn ngữ lập trình C#.....                         | 28 |
| 2.13 Tổng quan về HTML, CSS, JavaScript.....                         | 29 |
| CHƯƠNG 3: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG.....                        | 34 |
| 3.1 Phân tích hệ thống.....  | 34 |
| 3.2 Thiết kế cơ sở dữ liệu.....                                      | 37 |
| 3.3 Thiết kế giao diện.....  | 47 |
| CHƯƠNG 4: KẾT LUẬN.....  | 56 |
| TÀI LIỆU THAM KHẢO.....  | 57 |



## DANH SÁCH BẢNG

|   |    |
|---|----|
| Bảng 2.1 Bảng so sánh tự luận và trắc nghiệm.....             | 5  |
| Bảng 2.2 Bảng thể hiện mức độ ưu tiên của các kiểu nguồn..... | 31 |
| Bảng 3.1 Mô tả chi tiết bảng TaiKhoan.....                    | 41 |
| Bảng 3.2 Mô tả chi tiết bảng LopHoc.....                      | 42 |
| Bảng 3.3 Mô tả chi tiết bảng HocPhan.....                     | 42 |
| Bảng 3.4 Mô tả chi tiết bảng DeThi.....                       | 42 |
| Bảng 3.5 Mô tả chi tiết bảng KhoCauHoi.....                   | 43 |
| Bảng 3.6 Mô tả chi tiết bảng CauHoi.....                      | 43 |
| Bảng 3.7 Mô tả chi tiết bảng DapAn.....                       | 44 |
| Bảng 3.8 Mô tả chi tiết bảng SinhVien.....                    | 44 |
| Bảng 3.9 Mô tả chi tiết bảng CaThi.....                       | 44 |
| Bảng 3.10 Mô tả chi tiết bảng ChiTietDeThi.....               | 45 |
| Bảng 3.11 Mô tả chi tiết bảng BaiLam.....                     | 45 |
| Bảng 3.12 Mô tả chi tiết bảng ThiSinh.....                    | 46 |

## DANH SÁCH HÌNH

|  |    |
|--|----|
| Hình 2.1 Cấu trúc của Entity Framework.....                          | 20 |
| Hình 3. 1 Sơ đồ phân cấp chức năng BFD.....                          | 34 |
| Hình 3.2 Mô hình luồng dữ liệu mức ngữ cảnh.....                     | 34 |
| Hình 3.3 Mô hình luồng dữ liệu mức 1.....                            | 35 |
| Hình 3.4 Mô hình luồng dữ liệu mức 2 của quản lý hệ thống.....       | 35 |
| Hình 3.5 Mô hình luồng dữ liệu mức 2 của quản lý đề thi.....         | 36 |
| Hình 3.6 Mô hình luồng dữ liệu mức 2 của quản lý thi.....            | 36 |
| Hình 3.7 Mô hình thực thể kết hợp ERD.....                           | 37 |
| Hình 3.8 Mô hình dữ liệu mức vật lý.....                             | 41 |
| Hình 3.9 Giao diện đăng nhập.....                                    | 47 |
| Hình 3.10 Giao diện quản lý tài khoản.....                           | 48 |
| Hình 3.11 Giao diện quản lý lớp học.....                             | 48 |
| Hình 3.12 Giao diện quản lý học viên của lớp học.....                | 49 |
| Hình 3.13 Giao diện quản lý học phần.....                            | 50 |
| Hình 3.14 Giao diện quản lý kho câu hỏi của học phần.....            | 50 |
| Hình 3.15 Giao diện quản lý câu hỏi của kho câu hỏi.....             | 51 |
| Hình 3.16 Giao diện quản lý đề thi của học phần.....                 | 52 |
| Hình 3.17 Giao diện quản lý ca thi của học phần.....                 | 52 |
| Hình 3.18 Giao diện quản lý thi.....                                 | 53 |
| Hình 3.19 Giao diện danh sách sinh viên tham gia thi của kỳ thi..... | 53 |
| Hình 3.20 Giao diện đăng nhập.....                                   | 54 |
| Hình 3.21 Giao diện chọn kỳ thi.....                                 | 54 |
| Hình 3.22 Giao diện thi cho sinh viên.....                           | 55 |

## **DANH MỤC TỪ VIẾT TẮT**

|       |                                    |
|-------|------------------------------------|
| GV    | : Giảng Viên                       |
| LAN   | : Mạng Cục Bộ - Local Area Network |
| VD    | : Ví Dụ                            |
| KT-CN | : Kỹ Thuật - Công Nghệ             |

# CHƯƠNG 1: GIỚI THIỆU

## 1.1 Lý do chọn đề tài

Hiện nay thi trắc nghiệm là một hình thức thi rất phổ biến, nhờ vào đặc điểm luôn có sự rõ ràng của đáp án, tính khách quan trong cách đánh giá kết quả của người học, chính xác và thuận tiện cho cả người ra đề thi và người thi, hình thức thi này đã được áp dụng ở hầu khắp hệ thống giáo dục trên thế giới.

Tại Việt Nam, trong những năm gần đây, song song với việc đổi mới phương pháp dạy và học, việc đổi mới hình thức thi cử cũng trở thành việc làm cấp thiết. Trong các hình thức thi cử, trắc nghiệm là hình thức được nhiều người chú ý nhất do có nhiều ưu điểm trong việc kiểm tra, đánh giá trình độ người dự thi dựa trên các tiêu chí khách quan, trung thực, kiểm tra được nhiều kiến thức. Do đó trắc nghiệm đang là khuynh hướng của hầu hết các kỳ thi.

Làm bài thi trắc nghiệm trên giấy hiện nay đang rất phổ biến, tuy nhiên lại không thuận tiện trong các vấn đề như tổ chức thi (không gian, thời gian), in ấn, phân phối đề thi, thu bài, chấm bài... Hiện nay các hệ thống thi trắc nghiệm trên máy tính đang phát triển rất mạnh cùng với sự phát triển của cơ sở hạ tầng công nghệ thông tin. Có thể dự đoán rằng trong tương lai không xa, các kỳ thi sẽ được tổ chức thi trực tuyến, đặc biệt với quy mô nhà trường.

Nhận thấy tầm quan trọng và hữu ích của hình thức thi trắc nghiệm, đồng thời muốn tạo ra một chương trình hỗ trợ thi trắc nghiệm có thể xử lý tự động khâu tạo đề, trộn đề đến việc tổ chức thi và trả kết quả, cũng như đánh giá câu hỏi trong ngân hàng đề và giúp cho GV có thể thuận tiện hơn trong quá trình tạo đề thi (có thể tạo offline hoặc online), chúng em đã lựa chọn đề tài “Xây dựng phần mềm thi trắc nghiệm qua mạng LAN cho sinh viên khoa kỹ thuật - công nghệ” làm đề tài đồ án chuyên ngành của chúng em.

## 1.2 Mục tiêu nghiên cứu

Nghiên cứu tổng quan về lý thuyết trắc nghiệm và các phương pháp đánh giá câu hỏi, đề thi trắc nghiệm, cùng với việc tìm hiểu, khảo sát những hệ thống thi trắc nghiệm khác để thực hiện xây dựng hệ thống thi trắc nghiệm với đề thi trắc nghiệm đã có, hỗ trợ đánh giá câu hỏi trắc nghiệm để nâng cao chất lượng câu hỏi trắc nghiệm trong ngân hàng. Nghiên cứu các công cụ, nền tảng lập trình được sử dụng để xây dựng hệ thống: ADO.NET, C#, MS SQLServer2008, T-SQL, DevExpress. Cài đặt triển khai hệ thống qua mạng, kiểm lỗi và hoàn thiện các tính năng theo yêu cầu của đề tài đặt ra

### **1.3 Đối tượng, phạm vi nghiên cứu**

#### **1.3.1 Không gian**

Hệ thống này được áp dụng đối với sinh viên khoa KT-CN Trường Đại học Nam Cần Thơ.

#### **1.3.2 Thời gian**

Thời gian thực hiện đề nghiên cứu là học kì 1 của năm thứ 4 đại học, tại trường Đại học Nam Cần Thơ.

#### **1.3.3 Đối tượng nghiên cứu**

Đối tượng nghiên cứu của đề tài này là quy trình tạo đề, trộn đề và quy trình thi trắc nghiệm trên máy tính. Từ đó, xây dựng chương trình ứng dụng.

#### **1.3.4 Phạm vi nghiên cứu**

Hệ thống thi trắc nghiệm được xây dựng theo hình thức thi trên máy tính có nối mạng LAN. Hệ thống được xây dựng để sử dụng trong các kỳ thi trắc nghiệm của Khoa KT-CN Trường Đại học Nam Cần Thơ.

### **1.4 Phương pháp nghiên cứu**

#### **1.4.1 Phương pháp nghiên cứu lý luận**

Phương pháp quan sát: trực tiếp quan sát và sử dụng thử một số phần mềm thi trắc nghiệm hiện hành, để phân tích ưu nhược điểm, các quy trình cơ bản đối với phần mềm thi trắc nghiệm khách quan.

Phương pháp thu thập tài liệu: thông qua quá trình thu thập tài liệu từ các nguồn như sách báo, mạng internet, tổng hợp các tài liệu, nghiên cứu tài liệu, ngôn ngữ và công nghệ liên quan.

Phương pháp phân tích, thiết kế hệ thống: Dựa trên những thông tin thu được để tiến hành phân tích, thiết kế hệ thống bao gồm những chức năng chính của phần mềm sẽ xây dựng thông qua quá trình mô hình hóa hệ thống, chuẩn hóa cơ sở dữ liệu một cách phù hợp

#### **1.4.2 Phương pháp chuyên gia**

Tham khảo ý kiến của giảng viên hướng dẫn để có thể thiết kế chương trình phù hợp với yêu cầu thực tiễn đáp ứng yêu cầu của người dùng.

### 1.4.3 Phương pháp thực nghiệm

Phân tích yêu cầu thực tế và xây dựng các bước phân tích hệ thống để hỗ trợ lập trình, xây dựng ứng dụng. Đánh giá kết quả đạt được, triển khai, bảo trì.

## 1.5 Nội dung

Bố cục của đề án gồm 4 chương như sau:

**Chương 1: Giới thiệu** – Nêu lên lý do chọn đề tài, mục tiêu nghiên cứu, đối tượng nghiên cứu, phạm vi nghiên cứu và phương pháp nghiên cứu.

**Chương 2: Cơ sở lý luận và phương pháp nghiên cứu** – Trình bày tổng quát về lý thuyết trắc nghiệm và giới thiệu các ngôn ngữ, công cụ, nền tảng, công nghệ để xây dựng hệ thống.

**Chương 3: Phân tích và thiết kế hệ thống** – Trình bày các giai đoạn từ khảo sát phân mềm, mô phỏng hệ thống, phân tích hệ thống thông tin, thiết kế cơ sở dữ liệu đến việc thiết kế giao diện cho hệ thống.

**Chương 4: Kết luận** – Nêu lên kết quả của quá trình nghiên cứu, xây dựng hệ thống và ưu nhược điểm của hệ thống. Từ đó đưa các giải pháp, hướng phát triển để hệ thống hoàn thiện hơn

## CHƯƠNG 2: CƠ SỞ LÝ LUẬN VÀ PHƯƠNG PHÁP NGHIÊN CỨU

### 2.1 Tổng quan về trắc nghiệm

#### 2.1.1 Một số khái niệm cơ bản

Trắc nghiệm theo nghĩa rộng là một hoạt động được thực hiện để đo lường năng lực của các đối tượng nào đó nhằm những mục đích xác định.

Trong giáo dục, trắc nghiệm được tiến hành thường xuyên ở các kì thi, kiểm tra để đánh giá kết quả học tập, đối với một phần của môn học, toàn bộ môn học, đối với cả một cấp học, hoặc để tuyển chọn một số người có năng lực nhất vào một khoá học.

Người ta có thể phân chia các phương pháp trắc nghiệm ra làm ba loại:

- + Loại Quan sát: Giúp xác định những thái độ, những phản ứng vô thức, những kỹ năng thực hành và một số kỹ năng về nhận thức, chẳng hạn cách giải quyết vấn đề trong một tình huống đang được nghiên cứu.

- + Loại Vấn đáp: Có tác dụng tốt khi nêu lên các câu hỏi phát sinh trong một tình huống cần kiểm tra. Trắc nghiệm vấn đáp thường được dùng khi tương tác giữa người chấm và người học là quan trọng, chẳng hạn cần xác định thái độ phản ứng khi phỏng vấn,...

- + Loại Viết: Thường được sử dụng nhiều nhất vì nó có những ưu điểm sau: Cho phép kiểm tra cùng lúc nhiều học sinh; Cho phép học sinh cân nhắc nhiều hơn khi trả lời; Đánh giá được một vài loại tư duy ở mức độ cao; Cung bản ghi rõ ràng các câu trả lời của học sinh để dùng khi chấm; Dễ quản lý hơn vì bản thân người chấm không tham gia vào bối cảnh kiểm tra.

Trắc nghiệm viết thường được chia thành hai nhóm: Trắc nghiệm tự luận (Essay) và trắc nghiệm khách quan (Objective test).

Trắc nghiệm khách quan là phương pháp kiểm tra, đánh giá kết quả học tập của học sinh bằng hệ thống câu hỏi trắc nghiệm khách quan. Gọi là khách quan vì cách cho điểm (đánh giá) hoàn toàn không phụ thuộc vào người chấm.

Để thuận tiện trong cách gọi, nên trong nội dung của đồ án này từ “tự luận” sẽ được dùng thay cho trắc nghiệm tự luận và từ “trắc nghiệm” thay cho trắc nghiệm khách quan. Bảng so sánh dưới đây chúng ta sẽ có một cái nhìn rõ ràng hơn giữa tự luận và trắc nghiệm.

Bảng 2.1 Bảng so sánh tự luận và trắc nghiệm

| Tự luận  | Trắc nghiệm  |
|--|--|
| Một câu hỏi thuộc loại tự luận đòi hỏi thí sinh phải tự mình soạn câu trả lời và diễn tả câu trả lời bằng ngôn ngữ của mình  | Một câu hỏi trắc nghiệm buộc thí sinh phải lựa chọn câu trả lời đúng nhất trong một số câu đã cho sẵn  |
| Một bài tự luận gồm số câu hỏi tương đối ít và có tính cách tổng quát, đòi hỏi thí sinh phải triển khai câu trả lời bằng lời lẽ dài dòng                           | Một bài trắc nghiệm thường gồm nhiều câu hỏi có tính cách chuyên biệt chỉ đòi hỏi những câu trả lời ngắn gọn   |
| Trong khi làm một bài tự luận, thí sinh phải bỏ ra phần lớn thời gian để suy nghĩ và viết  | Trong khi làm một bài trắc nghiệm, thí sinh dùng nhiều thời gian để đọc và suy nghĩ  |
| Chất lượng của một bài tự luận tùy thuộc chủ yếu vào kỹ năng của người chấm bài  | Chất lượng của một bài trắc nghiệm được xác định một phần lớn do kỹ năng của người soạn thảo bài trắc nghiệm   |
| Một bài thi theo lối tự luận tương đối dễ soạn, nhưng khó chấm và khó cho điểm chính xác   | Một bài thi trắc nghiệm khó soạn, nhưng việc chấm và cho điểm tương đối dễ dàng và chính xác.  |
| Thí sinh có nhiều tự do bộc lộ cá tính của mình trong câu trả lời, và người chấm bài cũng có tự do cho điểm các câu trả lời theo xu hướng riêng của mình           | Người soạn thảo trắc nghiệm có nhiều tự do bộc lộ kiến thức và các giá trị của mình qua việc đặt các câu hỏi, nhưng chỉ cho thí sinh quyền tự do chứng tỏ mức độ hiểu biết của mình qua tỉ lệ câu trả lời đúng |
| Trong các câu hỏi tự luận, nhiệm vụ học tập của người học và trên cơ sở đó giám khảo thẩm định mức độ hoàn thành nhiệm vụ ấy không được phát biểu một cách rõ ràng | Trong các câu hỏi trắc nghiệm, nhiệm vụ học tập của người học và trên cơ sở đó giám khảo thẩm định mức độ hoàn thành các nhiệm vụ ấy được phát biểu một cách rõ ràng.  |
| Một bài tự luận cho phép và đôi khi  | Một bài trắc nghiệm cho phép và đôi  |



|  |   |
|--|---|
| khuyến khích sự “lừa phỉnh” (chẳng hạn như bằng những ngôn từ hoa mỹ hay bằng cách đưa ra những bằng chứng khó có thể xác định được) | khi khuyến khích sự phỏng đoán.   |
| Sự phân bố điểm số của một bài thi tự luận có thể được kiểm soát một phần lớn do người chấm  | Phân bố điểm số của thí sinh hầu như hoàn toàn được quyết định do bài trắc nghiệm |

### 2.1.2 Các dạng câu hỏi trắc nghiệm

#### 2.1.2.1 Câu trắc nghiệm đa lựa chọn (multiple choice question)

Cấu trúc của câu trắc nghiệm dạng đa lựa chọn

+ Phần câu hỏi: là một câu hỏi (kết thúc là dấu chấm hỏi) hay câu chưa hoàn tất. Trong phần này, người soạn trắc nghiệm đặt ra một vấn đề hay đưa ra một ý tưởng rõ ràng giúp cho người trả lời hiểu rõ câu trắc nghiệm ấy muốn hỏi điều gì để lựa chọn câu trả lời thích hợp.

+ Phần lựa chọn: có thể có 3,4,5 lựa chọn hoặc nhiều hơn. Mỗi lựa chọn là một câu trả lời. Trong tất cả các lựa chọn chỉ có một lựa chọn được xác định là đúng nhất, gọi là “đáp án”. Những lựa chọn còn lại đều phải sai, là câu gây nhiễu. Điều quan trọng người soạn thảo cần lưu ý là phải làm sao cho các câu nhiễu này đều hấp dẫn ngang nhau đối với những học sinh chưa nắm vững kiến thức, thúc đẩy học sinh ấy chọn vào những câu nhiễu này.

Câu trắc nghiệm dạng này có ưu điểm:

- Độ may rủi thấp: nếu câu trắc nghiệm có N lựa chọn thì độ may rủi là  $1/N$ .
- Nếu soạn đúng quy cách, kết quả có tính tin cậy và tính giá trị cao.
- Có thể khảo sát thành quả học tập của một số đông học sinh, chấm bài nhanh, kết quả chính xác.
- Có thể được sử dụng để kiểm tra các kỹ năng nhận thức bậc cao.

Bên cạnh đó, câu trắc nghiệm đa lựa chọn vẫn có một số nhược điểm:

- Tuy độ may rủi thấp nhưng người trả lời vẫn có thể đoán mò.
- Vì có nhiều phương án lựa chọn nên khó xây dựng được các câu hỏi có chất lượng cao

- Để có được một bài trắc nghiệm có tính tin cậy và tính giá trị cao, người soạn thảo trắc nghiệm phải đầu tư nhiều thời gian và phải tuân thủ đầy đủ các bước soạn thảo câu trắc nghiệm

#### **2.1.2.2 Câu trắc nghiệm đúng – sai (true – false question)**

Cấu trúc của câu trắc nghiệm dạng đúng sai: gồm một câu phát biểu và phần trả lời là lựa chọn Đúng hoặc Sai

Ưu điểm của dạng câu này:

- Dễ xây dựng
- Có thể đặt được nhiều câu hỏi trong một bài trắc nghiệm với thời gian cho trước, điều này làm tăng tính tin cậy của bài trắc nghiệm nếu như các câu trắc nghiệm Đúng – Sai được soạn thảo theo đúng cách.
- Trong khoảng thời gian ngắn có thể soạn được nhiều câu trắc nghiệm Đúng– Sai vì người soạn trắc nghiệm không cần phải tìm ra phần trả lời cho học sinh lựa chọn.

Nhược điểm của dạng câu này:

- Độ may rủi cao (50%) do đó dễ khuyến khích sự đoán mò.
- Thường chỉ được dùng để kiểm tra mức độ nhận biết, hiểu

#### **2.1.2.3 Câu trắc nghiệm ghép đôi (matching question)**

Cấu trúc của câu trắc nghiệm dạng ghép đôi: gồm có 3 phần.

+ Phần chỉ dẫn cách trả lời: chỉ ra cho người trả lời biết cách ghép hai cột với nhau cho đúng, có ý nghĩa, hợp logic.

+ Phần gốc (cột 1, về trái): gồm những câu ngắn, đoạn, chữ, ...

+ Phần lựa chọn (cột 2, về phải): cũng gồm những câu ngắn, đoạn, chữ, ...

Ưu điểm của dạng câu này:

- Dễ xây dựng.
- Có thể hạn chế sự đoán mò bằng cách tăng số lượng lựa chọn.

Nhược điểm của dạng câu này:

- Chỉ chủ yếu dùng để kiểm tra khả năng nhận biết.
- Thông tin có tính dàn trải, không nhấn mạnh được những điều quan trọng

#### **2.1.2.4 Câu trắc nghiệm điền khuyết (filling question)**

Cấu trúc của câu trắc nghiệm dạng điền khuyết có 2 dạng:

- + Dạng 1 gồm những câu hỏi với lời giải đáp ngắn
- + Dạng 2 gồm những câu phát biểu với một hay nhiều chỗ để trống mà người trả lời phải điền vào bằng một từ hay một nhóm từ ngắn

Ưu điểm của dạng câu này:

- Dễ xây dựng.
- Người trả lời không thể đoán mò.

Nhược điểm của dạng câu này:

- Thường chỉ được dùng để kiểm tra trình độ mức độ nhận biết, hiểu.
- Khó đánh giá nội dung trả lời

#### **2.1.2.5 Câu trắc nghiệm hình ảnh (image question)**

Cấu trúc của câu trắc nghiệm dạng hình ảnh: gồm phần chỉ dẫn trả lời hướng dẫn cho người làm bài cách để trả lời và hình ảnh kèm theo.

Ưu điểm của dạng câu này:

- Dễ xây dựng.
- Xác suất trả lời đúng thấp vì chỉ có một vùng lựa chọn đúng trong vô số vùng lựa chọn.

Nhược điểm của dạng câu này: thường chỉ được dùng để kiểm tra trình độ mức độ nhận biết và hiểu

#### **2.1.3 Những trường hợp dùng trắc nghiệm**

Chúng ta nên sử dụng trắc nghiệm để khảo sát thành quả học tập trong những trường hợp sau:

- Khi ta cần khảo sát thành quả học tập của một số đông học sinh, hay muốn rằng bài khảo sát ấy có thể được sử dụng lại vào một lúc khác.
- Khi ta muốn có những điểm số đáng tin cậy, không phụ thuộc vào chủ quan của người chấm bài.
- Khi các yếu tố công bằng, vô tư, chính xác được coi là những yếu tố quan trọng nhất của việc thi cử.

- Khi ta có nhiều câu trắc nghiệm tốt đã được lưu trữ sẵn để có thể lựa chọn và cấu trúc lại một bài thi trắc nghiệm mới. Đặc biệt là muốn chấm nhanh và công bố kết quả sớm.
- Khi muốn ngăn ngừa nạn học tủ, học vẹt, gian lận trong thi cử của thí sinh.

## **2.2 Tổng quan về mạng LAN**

### **2.2.1 Mạng Lan là gì?**

LAN là viết tắt của Local Area Network là mạng cục bộ dùng để kết nối các máy tính với nhau trong 1 khu vực. Kết nối được thực hiện thông qua môi trường truyền thông tốc độ cao như dây cáp. Các LAN cũng có thể kết nối với nhau thành WAN (Wide Area Network).

LAN thường bao gồm một máy chủ (server, host) còn gọi là máy phục vụ. Máy chủ thường là máy có bộ xử lý (CPU) tốc độ cao, bộ nhớ (RAM) và đĩa cứng (HD) lớn. Khi lắp đặt mạng LAN sẽ cần các thiết bị kết nối khác giữa các máy tính với nhau.

### **2.2.2 Cơ chế hoạt động của mạng LAN**

+ Việc kết nối các máy tính với một dây cáp được dùng như một phương tiện truyền tin chung cho tất cả các máy tính. Công việc kết nối vật lý vào mạng được thực hiện bằng cách cắm một card giao tiếp mạng NIC (Network Interface Card) vào trong máy tính và nối nó với cáp mạng. Sau khi kết nối vật lý đã hoàn tất, quản lý việc truyền tin giữa các trạm trên mạng tùy thuộc vào phần mềm mạng.

+ Khi một máy muốn gửi một thông điệp cho máy khác thì nó sẽ dùng một phần mềm trong máy nào đó đặt thông điệp vào một gói tin (packet) bao gồm dữ liệu thông điệp được bao bọc giữa tín hiệu đầu và tín hiệu cuối, và dùng phần mềm mạng để gửi gói tin đó đến máy đích.

+ NIC sẽ chuyển gói tín hiệu vào mạng LAN, gói tín hiệu được truyền đi như một dòng các bit dữ liệu. Khi nó chạy trong cáp chung mọi máy đều nhận được tín hiệu này

+ NIC ở mỗi trạm sẽ kiểm tra địa chỉ đích trong tín hiệu đầu của gói để xác định đúng địa chỉ đến, khi gói tín hiệu đi tới máy có địa chỉ cần đến, đích ở máy đó sẽ sao gói tín hiệu rồi lấy dữ liệu ra khỏi gói tin và đưa vào máy tính.

Với cơ chế hoạt động như trên mạng LAN thường được sử dụng trong một phạm vi giới hạn nhỏ để chia sẻ các tài nguyên dùng chung đồng thời cho phép nhiều máy tính truy cập Internet thông qua một cổng truy cập Internet.

Việc Quản lý thi cử của sinh viên các khoa tại trường Cao đẳng, Đại học là vấn đề được chú trọng, đặc biệt thi cử trực tiếp trên máy tính của các học phần yêu cầu bắt buộc thi trên máy tính để nhấn mạnh khả năng thực hành tốt. Phòng máy tính của trường đã được thiết lập hệ thống mạng LAN căn bản có thể áp dụng và thiết kế theo mô hình Client-Server. Bởi vì, mô hình Client-Server là mô hình tập trung dữ liệu tại máy Server sau khi các máy Client gửi thông tin, mô hình này có ưu điểm chỉ quản lý dữ liệu tại một điểm chính là Server qua việc thực hiện lập trình kết nối tự động từ module cài đặt trên các máy Client đến module cài đặt trên một Server. Từ đó, Nhóm tác giả đưa ra mục tiêu là ứng dụng công nghệ thông tin để xây dựng phần mềm để quản lý thi cử trên máy tính dựa trên mô hình Client-Server đáp ứng nhu cầu cần thiết về quản lý việc thi cử sinh viên thi trên máy tính nhằm đảm bảo, an toàn, và đặc biệt phù hợp với quy trình thi cử

### **2.2.3 Ưu điểm và nhược điểm của việc tổ chức, quản lý thi trắc nghiệm trên mạng LAN**

\* Ưu điểm:

- Lợi ích cho cá nhân sinh viên:
- Nâng cao kỹ năng sử dụng máy tính và hiểu biết về công nghệ thông tin và truyền thông.
- Đánh giá sự hiểu biết và am hiểu kỹ càng về mặt lý thuyết mà giảng viên giảng dạy trên lớp từ đó áp dụng làm thực hành trên máy với các bài tập theo từng nội dung từng chương của chương trình học phần đó.
- Nắm rõ các trường hợp cá biệt không tham gia học lý thuyết thì giảng viên sẽ nắm rõ các trường hợp sinh viên đó đi học hay không qua việc làm bài thi thực hành trên máy để đánh giá thực lực của sinh viên được kỹ càng hơn.
- Qua việc thi trên máy sẽ tránh được sinh viên trao đổi, giúp đỡ làm bài với nhau (thi tự luận có thời gian chép bài nguyên văn, hỏi bài, sử dụng tài liệu, đánh dấu bài, ...).
- Qua việc thi trên máy sẽ đánh giá được cách sử dụng máy tính chủ động, sáng tạo, nhanh nhẹn khi làm bài thi của sinh viên.
- Lợi ích cho nơi đào tạo:
- Thi trên máy mang lại thước đo chuẩn mực và đáng tin cậy về kỹ năng sử dụng máy tính.
- Thi trên máy bao hàm các kiến thức cơ bản và cập nhật về công nghệ thông tin và truyền thông trên thế giới. Là cơ sở tốt cho các trường đại học đào tạo nguồn nhân lực cho tỉnh, nhà nước.

- Sinh viên làm bài trực tiếp trên máy tính, giúp đánh giá chính xác kỹ năng, kiến thức tin học của sinh viên.
- Lợi ích cho xã hội:
- Là cửa ngõ cho xã hội Internet và công nghệ thông tin.
- Cầu nối khuyến khích phổ cập kỹ thuật số.
- Tạo điều kiện thuận lợi cho tính linh động trong công việc và tái đào tạo.

**\* Nhược điểm**

**- Bất lợi cho cán bộ coi thi:**

- Luôn theo dõi, kiểm tra danh sách sinh viên vào phòng thi, chỉ dẫn ngồi vào máy thi, lý do tránh ngồi một số máy không sử dụng được.
- Sau khi sinh viên thi xong, điều khó khăn nhất là kiểm tra bài thi đã nộp của sinh viên có đủ và đúng như danh sách ký sinh viên vào phòng thi.
- Trường hợp trùng bài thi lý do sinh viên trao đổi và sao chép (copy) bài thi cho nhau bởi việc sử dụng phương tiện công nghệ (USB, hoặc thẻ nhớ, điện thoại smartphone, ...) mà chưa chỉnh sửa tên bài thi theo quy định định dạng của cán bộ coi thi thông báo trước khi lưu và gửi bài thi cho cán bộ coi thi.
- Nếu trong trường hợp mạng máy tính theo mô hình LAN nội bộ không kết nối các máy tính với nhau và chia sẻ một thư mục chung để gom bài thi về thư mục này diễn ra không tốt. Khi đó việc thu gom bài thi bắt buộc cán bộ coi thi phải dùng USB để thu gom bài thi, bên cạnh đó, sẽ gây trường hợp lỗi bài thi do lỗi máy tính hoặc chương trình virus gây ra làm mất bài thi. Quan trọng cuối cùng là mất bài thi có thể do lỗi cán bộ coi thi hoặc lỗi sinh viên lưu bài thi chưa hoàn chỉnh.

**- Bất lợi cho sinh viên:**

- Trước khi gửi bài thi, sinh viên phải thực hiện theo quy định yêu cầu lưu bài thi đúng theo quy định của cán bộ coi thi thông báo. Nhưng một số trường hợp không chú ý sẽ tự động lưu bài theo cách riêng của mình hoặc lưu theo định dạng mặc nhiên của chương trình máy tính mà sinh viên sử dụng để làm bài thi là lấy dòng đầu tiên để đặt tên cho tập tin bài thi đó. Dẫn đến việc cán bộ coi thi khó khăn trong quá trình kiểm tra bài thi so với danh sách sinh viên thi trong phòng thi.
- Sinh viên không biết chắc chắn mình đã lưu bài thi hay gửi bài thi theo quy định của định dạng và đúng đường dẫn vào nơi lưu bài thi của

mình đề cán bộ coi thi thu gom bài thi. Từ đó đưa ra nhận xét về phía sinh viên: sinh viên chưa tự tin hoặc quy về trách nhiệm cho sinh viên việc hoàn thành lưu bài thi theo quy định.

- Sinh viên kiện tụng hoặc thắc mắc điểm sau khi thi với lý do tại cột chú ý của danh sách điểm thi là bài thi không có, hoặc lỗi bài thi khi giảng viên chấm bài không thể mở được tập tin bài thi của sinh viên hoặc mở ra không có nội dung trong tập tin bài thi. Điều này dẫn đến ảnh hưởng cho giảng viên giảng dạy và chấm thi phải trả lời thắc mắc sinh viên một cách phù hợp nếu không sẽ gây kiện tụng ảnh hưởng đến khoa nói riêng, nhà trường nói chung

## **2.3 Tổng quan về hệ thống thông tin**

### **2.3.1 Một số khái niệm cơ bản**

Hệ thống là một tập hợp có tổ chức của nhiều phần tử thường xuyên tương tác với nhau, có những mối quan hệ ràng buộc lẫn nhau và cùng nhau hoạt động chung cho một mục đích nào đó.

Thông tin là một hay tập hợp những phần tử mà ta thường gọi là các tín hiệu phản ánh ý nghĩa về một đối tượng, một hiện tượng hay một quá trình nào đó của sự vật thông qua quá trình nhận thức. Trong tin học, thông tin là sự tinh lọc từ việc xử lý dữ liệu. Chính vì vậy mà hai thành phần quan trọng của hệ thống thông tin là thành phần dữ liệu và thành phần xử lý.

Hệ thống thông tin, là tập hợp người, thủ tục và các nguồn lực để thu thập, xử lý, truyền và phát thông tin trong một tổ chức. Hệ thống thông tin có thể là thủ công nếu dựa vào các công cụ như giấy, bút. Hệ thống thông tin hiện đại là hệ thống tự động hóa dựa vào máy tính (phần cứng, phần mềm) và các công nghệ thông tin khác..

### **2.3.2 Nhiệm vụ và vai trò của hệ thống thông tin**

Chức năng chính của hệ thống thông tin là xử lý thông tin của hệ thống nghiệp vụ. Quá trình xử lý thông tin như một mô hình hộp đen bao gồm: Bộ xử lý, thông tin đầu vào, thông tin đầu ra và thông tin phản hồi cần thiết của hệ thống.

Nhiệm vụ của hệ thống thông tin:

Về đối ngoại: Hệ thống thông tin thu nhận thông tin từ môi trường bên ngoài và đưa thông tin ra môi trường bên ngoài.

Về đối nội: Hệ thống thông tin là cầu nối liên lạc giữa các bộ phận của hệ thống nghiệp vụ.

Vai trò của hệ thống thông tin: Đóng vai trò trung gian giữa hệ thống nghiệp vụ và môi trường, giữa hệ thống con quyết định và hệ thống con tác nghiệp. Ngoài ra hệ thống thông tin còn cung cấp thông tin cho các hệ thống quyết định và tác nghiệp.

### **2.3.3 Các thành phần cơ bản của hệ thống thông tin**

Con người: Hệ thống thông tin cung cấp thông tin cho mọi người bao gồm người sử dụng bên trong (người quản lý, kỹ thuật viên) và người sử dụng bên ngoài (khách hàng, nhà cung cấp).

Thủ tục: Bao gồm các dữ liệu mô tả công việc của tất cả mọi người, cả người sử dụng cuối và nhân viên trong hệ thống thông tin.

Phần cứng: Bao gồm tất cả các thiết bị vật lý sử dụng trong hệ thống thông tin.

Phần mềm: Bao gồm 2 loại.

- Phần mềm hệ thống: điều khiển phần cứng và các phần mềm khác. VD: Hệ điều hành, phần mềm điều khiển thiết bị (driver).
- Phần mềm ứng dụng: các chương trình xử lý dữ liệu để tạo ra thông tin. VD: Ứng dụng do các công ty tạo ra, phần mềm mua của nhà cung cấp.

Tệp (File) dữ liệu: Hầu hết dữ liệu được xử lý trong hệ thống thông tin phải được giữ lại vì lý do pháp luật hoặc vì sự cần thiết được xử lý trong tương lai. Những file này là thành phần của hệ thống thông tin, được tạo ra trực tiếp hoặc lưu trữ trong file.

Các dữ liệu là các thông tin được cấu trúc hóa.

Luồng thông tin vào: Các thông tin cần thiết cho quá trình xử lý, có thể là các thông tin phản ánh cấu trúc doanh nghiệp và các thông tin phản ánh hoạt động của doanh nghiệp.

Luồng thông tin ra: Thông tin ra được tổng hợp từ các thông tin đầu vào và phụ thuộc vào nhu cầu quản lý trong từng trường hợp, từng đơn vị cụ thể. Thông tin ra là kết quả của việc tra cứu nhanh về một đối tượng cần quan tâm, đồng thời phải đảm bảo sự chính xác và kịp thời.

Các xử lý là các quy trình, các phương pháp, chức năng xử lý thông tin và biến đổi thông tin. Các xử lý nhằm sản sinh ra các thông tin có cấu trúc theo thể thức quy định như các chứng từ giao dịch, các sổ sách báo cáo thống kê. Cung cấp các thông tin trợ giúp quyết định.



### 2.3.4 Vòng đời phát triển của hệ thống thông tin

Vòng đời phát triển hệ thống – SDLC (System Development Life Cycle): bao gồm nhiều giai đoạn từ khi bắt đầu dự án hệ thống cho đến khi kết thúc khai thác hệ thống.

Quá trình phân tích và thiết kế hệ thống bao gồm các công việc cần hoàn thành theo trình tự nhất định có thể bao gồm các bước sau đây:

- Xác định vấn đề, các yêu cầu quản lý hệ thống
- Xác định mục tiêu, ưu tiên, giải pháp sơ bộ và chứng minh tính khả thi
- Phân tích các chức năng và dữ liệu của hệ thống
- Thiết kế logic: Trả lời câu hỏi làm gì? là gì? phân tích sâu hơn các chức năng, các dữ liệu của hoạt động cũ để đưa ra mô hình hoạt động mới
- Thiết kế vật lý: Đưa ra những biện pháp, phương tiện thực hiện, nhằm trả lời câu hỏi làm thế nào?
- Cài đặt hệ thống: Lựa chọn ngôn ngữ, hệ quản trị cơ sở dữ liệu và lập trình
- Khai thác và bảo trì.

## 2.4 Tổng quan về hệ quản trị cơ sở dữ liệu Microsoft SQL Server

### 2.4.1 Một số khái niệm cơ bản

Cơ sở dữ liệu là một tập hợp các dữ liệu được tổ chức và lưu trữ theo một cấu trúc chặt chẽ nhằm phục vụ nhiều đối tượng sử dụng với nhiều mục đích khác nhau

Hệ quản trị cơ sở dữ liệu (Database Management System – DBMS) là một phần mềm tổng quát nhằm hỗ trợ việc lưu trữ, truy xuất và quản trị cơ sở dữ liệu. Nó cung cấp cho người dùng và ứng dụng một môi trường thuận tiện và sử dụng hiệu quả tài nguyên dữ liệu. Nói cách khác, Hệ quản trị CSDL là phần mềm chuyên dụng để giải quyết tốt các tình huống như bảo mật, cạnh tranh trong truy xuất.

Ưu điểm:

- Quản lý được dữ liệu dư thừa.
- Đảm bảo tính nhất quán cho dữ liệu.
- Tạo khả năng chia sẻ dữ liệu nhiều hơn.
- Cải tiến tính toàn vẹn cho dữ liệu.

Nhược điểm:

- Hệ quản trị CSDL tốt thì khá phức tạp.

- Hệ quản trị CSDL tốt thường rất lớn chiếm nhiều dung lượng bộ nhớ.
- Giá cả khác nhau tùy theo môi trường và chức năng.
- Hệ quản trị CSDL được viết tổng quát cho nhiều người dùng thì thường chậm.

#### **2.4.2 Hệ quản trị cơ sở dữ liệu Microsoft Sql Server**

SQL viết tắt của Structured Query Language (Ngôn ngữ truy vấn có cấu trúc), là công cụ để sử dụng để tổ chức, quản lý và truy xuất dữ liệu được lưu trữ trong các CSDL quan hệ.

Ngôn ngữ truy vấn có cấu trúc và các hệ quản trị CSDL quan hệ là một trong những nền tảng kỹ thuật quan trọng trong công nghiệp máy tính. Hiện nay SQL được xem là ngôn ngữ chuẩn trong CSDL. Các hệ quản trị CSDL quan hệ thương mại hiện nay có như: Oracle, SQL Server, Informix, DB2,... đều chọn SQL làm ngôn ngữ cho sản phẩm của mình.

SQL Server là một hệ quản trị CSDL nhiều người dùng kiểu Client/Server. Đây là hệ thống cơ bản dùng lưu trữ dữ liệu cho hầu hết các ứng dụng lớn hiện nay.

### **2.5 Tổng quan về môi trường phát triển Visual Studio Code**

#### **2.5.1 Giới thiệu tổng quát**

Visual Studio Code là một trình biên tập mã được phát triển bởi Microsoft dành cho Windows, Linux và macOS. Nó hỗ trợ chức năng debug, đi kèm với Git, có syntax highlighting, tự hoàn thành mã thông minh, snippets, và cải tiến mã nguồn. Nó cũng cho phép tùy chỉnh, do đó, người dùng có thể thay đổi theme, phím tắt, và cá tùy chọn khác. Nó miễn phí và là phần mềm mã nguồn mở, mặc dù gói tải xuống chính thì là có giấy phép.

Visual Studio Code được công bố, ra mắt phiên bản thử nghiệm vào ngày 29 tháng 4 năm 2015 bởi Microsoft tại hội nghị Build 2015.

Vào ngày 18 tháng 11 năm 2015, Visual Studio Code được phát hành dưới giấy phép MIT và mã nguồn của nó được đưa lên GitHub. Chức năng Extension (tiện ích mở rộng) cũng được công bố.

Vào ngày 14 tháng 4 năm 2016, Visual Studio Code hoàn tất thử nghiệm và được phát hành chính thức.

Visual Studio Code được dựa trên Electron, một nền tảng được sử dụng để triển khai các ứng dụng Nodejs máy tính cá nhân chạy trên động cơ bố trí Blink. Mặc dù nó sử dụng nền tảng Electron nhưng phần mềm này không phải là một bản

khác của Atom, nó thực ra được dựa trên trình biên tập của Visual Studio Online (tên mã là "Monaco").

### **2.5.2 Tính năng của Visual Studio Code**

#### **Hỗ trợ nhiều ngôn ngữ lập trình**

Visual Studio Code hỗ trợ nhiều ngôn ngữ lập trình như C/C++, C#, F#, Visual Basic, HTML, CSS, JavaScript, ... Vì vậy, nó dễ dàng phát hiện và đưa ra thông báo nếu chương trình có lỗi.

#### **Hỗ trợ đa nền tảng**

Các trình viết code thông thường chỉ được sử dụng hoặc cho Windows hoặc Linux hoặc Mac Systems. Nhưng Visual Studio Code có thể hoạt động tốt trên cả ba nền tảng trên.

#### **Cung cấp kho tiện ích mở rộng**

Trong trường hợp lập trình viên muốn sử dụng một ngôn ngữ lập trình không nằm trong số các ngôn ngữ Visual Studio hỗ trợ, họ có thể tải xuống tiện ích mở rộng. Điều này vẫn sẽ không làm giảm hiệu năng của phần mềm, bởi vì phần mở rộng này hoạt động như một chương trình độc lập.

#### **Kho lưu trữ an toàn**

Đi kèm với sự phát triển của lập trình là nhu cầu về lưu trữ an toàn. Với Visual Studio Code, người dùng có thể hoàn toàn yên tâm vì nó dễ dàng kết nối với Git hoặc bất kỳ kho lưu trữ hiện có nào.

#### **Hỗ trợ web**

Visual Studio Code hỗ trợ nhiều ứng dụng web. Ngoài ra, nó cũng có một trình soạn thảo và thiết kế website.

#### **Lưu trữ dữ liệu dạng phân cấp**

Phần lớn tệp lưu trữ đoạn mã đều được đặt trong các thư mục tương tự nhau. Ngoài ra, Visual Studio Code còn cung cấp các thư mục cho một số tệp đặc biệt quan trọng.

#### **Hỗ trợ viết Code**

Một số đoạn code có thể thay đổi chút ít để thuận tiện cho người dùng. Visual Studio Code sẽ đề xuất cho lập trình viên các tùy chọn thay thế nếu có.

#### **Hỗ trợ thiết bị đầu cuối**

Visual Studio Code có tích hợp thiết bị đầu cuối, giúp người dùng khỏi phải chuyển đổi giữa hai màn hình hoặc trở về thư mục gốc khi thực hiện các thao tác.

### **Màn hình đa nhiệm**

Người dùng Visual Studio Code có thể mở cùng lúc nhiều tệp tin và thư mục – mặc dù chúng không hề liên quan với nhau.

### **Intellisense**

Hầu hết các trình viết mã đều có tính năng nhắc mã Intellisense, nhưng ít chương trình nào chuyên nghiệp bằng Visual Studio Code. Nó có thể phát hiện nếu bất kỳ đoạn mã nào không đầy đủ. Thậm chí, khi lập trình viên quên không khai báo biến, Intellisense sẽ tự động giúp họ bổ sung các cú pháp còn thiếu.

### **Hỗ trợ Git**

Visual Studio Code hỗ trợ kéo hoặc sao chép mã trực tiếp từ GitHub. Mã này sau đó có thể được thay đổi và lưu lại trên phần mềm.

### **Bình luận**

Việc để lại nhận xét giúp người dùng dễ dàng nhớ công việc cần hoàn thành.

## **2.6 Tổng quan về ASP.Net Core**

### **2.6.1 Giới thiệu tổng quát**

ASP.NET đã được sử dụng từ nhiều năm để phát triển các ứng dụng web. Kể từ đó, framework này đã trải qua một sự thay đổi tiến hóa đều đặn và cuối cùng đã đưa chúng ta đến với ASP.NET Core 1.0 hậu duệ mới gần đây nhất của nó.

- ASP.NET Core không phải là phiên bản tiếp theo của ASP.NET 4.6
- Đó là một framework hoàn toàn mới, may mắn rằng nó một dự án side-by-side tương tự với mọi thứ mà chúng ta biết.
- Nó thực ra được viết lại trên framework ASP.NET 4.6 hiện tại những kích thước nhỏ hơn và nhiều modular hơn.

ASP.NET Core là một web framework mã nguồn và được tối ưu hóa cho cloud để phát triển các ứng dụng web chạy trên nhiều nền tảng như Windows, Linux và Mac. Hiện tại, nó bao gồm MVC framework được kết hợp các tính năng của MVC và Web API thành một web framework duy nhất.

- Các ứng dụng ASP.NET Core có thể chạy trên .NET Core hoặc trên .NET Framework hoàn chỉnh.

- Nó đã được thiết kế để cung cấp một framework tối ưu cho các ứng dụng để triển khai tới cloud hoặc chạy on-premises.
- Nó bao gồm những modular với các thành phần tối thiểu, do đó bạn giữ được tính linh hoạt trong quá trình xây dựng các giải pháp của mình.
- Bạn có thể phát triển và chạy các ứng dụng đa nền tảng từ ASP.NET Core trên Windows, Mac và Linux.

### 2.6.2 Ưu điểm của ASP.NET Core

ASP.NET Core đi kèm với những ưu điểm sau:

- ASP.NET Core có một số thay đổi kiến trúc dẫn đến modular framework nhỏ hơn.
- ASP.NET Core không còn dựa trên System.Web.dll. Nó dựa trên một tập hợp nhiều yếu tố của Nuget packages.
- Điều này cho phép bạn tối ưu ứng dụng của mình chỉ cần những NuGet packages cần thiết.
- Lợi ích của diện tích bề mặt ứng dụng nhỏ hơn thì bảo mật chặt chẽ hơn, giảm dịch vụ, cải thiện hiệu suất và giảm chi phí.

Với ASP.NET Core, bạn có thể nhận được các cải tiến sau:

- Xây dựng và chạy các ứng dụng ASP.NET Core đa nền tảng trên Windows, Mac và Linux.
- Được xây dựng trên .NET Core, hỗ trợ side-by-side app versioning.
- Công cụ mới giúp đơn giản hóa việc phát triển web hiện đại.
- Liên kết đơn các web stack như Web UI và API Web.
- Cấu hình dựa trên môi trường đám mây sẵn có.
- Được xây dựng dựa trên cho DI (Dependency Injection).
- Tag Helpers làm cho các Razor makeup trở nên tự nhiên hơn với HTML.
- Có khả năng host trên IIS hoặc self-host.

## 2.7 Tổng quan về Entity Framework

### 2.7.1 Giới thiệu tổng quát

Entity Framework ra đời từ 2008 cùng với .NET 3.5. Phiên bản hiện nay là 6.0. Có một số mốc đặc biệt trong lịch sử framework này.

Phiên bản đầu tiên chỉ hỗ trợ hướng tiếp cận Database-first. Nghĩa là EF lúc đó chỉ có thể làm việc với một cơ sở dữ liệu sẵn có. EF giúp sinh code các lớp model bằng C# hoặc VB.NET.

Đến phiên bản 4 xuất hiện hướng tiếp cận Model-first, giúp thiết kế các lớp thực thể trước sử dụng giao diện đồ họa.

Phiên bản 4.1 đưa thêm hướng tiếp cận Code-first, nghĩa là từ giờ mọi thứ được viết bằng code VB.NET hoặc C# ngay từ đầu, một hướng tiếp cận được lập trình viên hoan nghênh đặc biệt.

Entity Framework là một ánh xạ quan hệ đối tượng (ORM), là một loại công cụ giúp đơn giản hóa ánh xạ giữa các đối tượng trong phần mềm của bạn với các bảng và cột của cơ sở dữ liệu quan hệ.

- Entity Framework (EF) là một framework ORM mã nguồn mở độc lập với .NET Framework.
- Entity Framework đảm nhiệm việc tạo các kết nối cơ sở dữ liệu và thực thi các lệnh, cũng như lấy kết quả truy vấn và tự động ánh xạ các kết quả đó thành các đối tượng trong ứng dụng của bạn.
- Entity Framework giúp theo dõi các thay đổi của đối tượng và cập nhật các thay đổi đó trở lại cơ sở dữ liệu cho bạn.

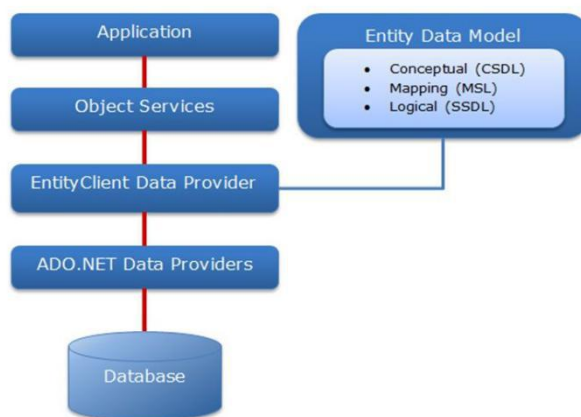
### **2.7.2 Tính năng của Entity Framework**

Sau đây là các tính năng cơ bản của Entity Framework. Danh sách này được tạo dựa trên các tính năng đáng chú ý nhất và cũng từ các câu hỏi thường gặp về Entity Framework.

- Entity Framework là một công cụ của Microsoft.
- Entity Framework đang được phát triển như một sản phẩm mã nguồn mở.
- Entity Framework không còn bị ràng buộc hoặc phụ thuộc vào chu kỳ phát hành của .NET Framework.
- Làm việc với bất kỳ cơ sở dữ liệu quan hệ nào có Entity Framework provider hợp lệ.
- Tạo truy vấn SQL từ LINQ to Entities.
- Entity Framework sẽ tạo các truy vấn có tham số.
- Theo dõi các thay đổi của các đối tượng trong bộ nhớ.
- Cho phép tạo các câu lệnh thêm, cập nhật và xóa.
- Entity Framework hỗ trợ stored procedure.

## 2.7.4 Kiến trúc của Entity Framework

Kiến trúc của Entity Framework được minh họa trong hình sau:



Hình 2.1 Cấu trúc của Entity Framework

### Application

Application (ứng dụng) là tầng chứa giao diện tương tác với người dùng, có thể là ứng dụng web hoặc ứng dụng Windows Forms. Tầng này thường sử dụng Repository pattern để làm việc với cơ sở dữ liệu thông qua Entity Framework.

### Object Services

Object Services là tầng chứa quá trình tương tác giữa ứng dụng và database, hay nói cách khác nó là nơi chủ yếu để truy cập dữ liệu từ database và trả ngược kết quả về giao diện. Object Services cung cấp các tiện ích để truy vết các thay đổi và cập nhật thay đổi trở lại ở database.

### EntityClient Data Provider

Đây là tầng cung cấp các kết nối, diễn dịch các truy vấn thực thể thành truy vấn nguồn dữ liệu, trả về datareader để EF dùng chuyển dữ liệu thực thể thành các đối tượng. Phần này kết nối ADO.NET Data Providers để gửi hoặc lấy dữ liệu từ database.

### ADO.NET Data Providers

Đây là tầng thấp nhất để dịch các truy vấn LINQ to Entity thành các câu lệnh SQL và thực thi các câu lệnh trong hệ thống DBMS. Tầng này kết với database sử dụng ADO.NET.

### Entity Data Model (EDM)

Entity Data Model gồm 3 phần chính: mô hình khái niệm (CSDL – Conceptual schema definition language), mô hình ánh xạ (MSL – mapping

specification language) và mô hình lưu trữ (SSDL – store schema definition language). Entity Data Model khác với EntityClient Data Provider ở chỗ nó sử dụng LINQ là ngôn ngữ truy vấn tương tác với database.

### **Mô hình khái niệm (CSDL)**

Mô hình khái niệm chứa các lớp mô hình và mối quan hệ giữa các lớp này. Nó độc lập với mô hình quan hệ các bảng trong database.

### **Mô hình lưu trữ (SSDL)**

Mô hình lưu trữ là 1 mô hình thiết kế database bao gồm các table, view, stored procedure, mối quan hệ giữa chúng các bảng và khóa. Mô hình này thể hiện gần giống mô hình quan hệ các bảng trong database.

### **Mô hình ánh xạ (MSL)**

Mô hình ánh xạ gồm thông tin về cách mô hình khái niệm được ánh xạ đến mô hình lưu trữ.

## **2.8 Tổng quan về Electron**

### **2.8.1 Giới thiệu tổng quát**

Electron được thành lập bởi Cheng Zhao, trong quá trình phát triển Atom, một trình soạn thảo văn bản đa nền tảng được phát hành vào tháng 7 năm 2013. Nó được tạo ra nguồn mở, được phát triển và hỗ trợ bởi GitHub bằng C ++, JavaScript, Objective C và Python. Nó được dự định để làm cho việc phát triển đa nền tảng dễ dàng hơn cho việc tạo ra Atom.

Electron là Framework (open source bởi Github) cho phép viết desktop app chạy trên mọi nền tảng (Mac, Window, Linux) dựa trên công nghệ web (Nodejs, HTML và CSS).

Cấu trúc của một ứng dụng Electron đơn giản nhất đi kèm với ba tệp chính:

- `pack.json` (siêu dữ liệu, tệp npm): Đây là tệp quan trọng nhất trong ứng dụng Electron. Nó được nhúng với thông tin về gói như tên, phiên bản, v.v.
- `main.js` : Đây là mã.
- `index.html` (GUI): Đó là tất cả về giao diện người dùng đồ họa.



### 2.8.3 Cơ chế hoạt động của Electron

Tập chính được xác định trong tệp `pack.json` sẽ được thực thi. Tệp `main.js` tạo một cửa sổ ứng dụng hoặc phiên bản cửa sổ trình duyệt để chạy các trang web. Điều này sẽ đủ mạnh để tương tác với Giao diện người dùng đồ họa gốc của HĐH.

Để chạy ứng dụng Electron, bạn sẽ cần cài đặt `npm`. Vì `npm` được phân phối cùng với `Code.js`, bạn sẽ tự động cài đặt `npm` trên máy tính của bạn khi bạn tải xuống `Node.js`.

Electron là khung chính đằng sau hai trình soạn thảo mã nguồn mở chính: Mã Visual Studio của Microsoft và Nguyên tử của GitHub. Tệp thực thi Electron (`electron.exe` trong Windows, `electron.app` trên OS X và `electron` trên Linux) cung cấp khung. Tệp thực thi Electron có thể dễ dàng được chỉnh sửa theo yêu cầu của nhà phát triển (ví dụ sẽ là: thêm nhãn hiệu, biểu tượng tùy chỉnh, v.v.).

Trang web chạy các quy trình riêng của nó được gọi là quy trình Kết xuất. Các phiên bản cửa sổ trình duyệt chạy các trang web thông qua quá trình kết xuất của chính nó. Vì vậy, khi phiên bản cửa sổ trình duyệt bị hủy, quá trình kết xuất được đính kèm với nó cũng sẽ bị hủy.

Có hai loại quy trình Electron chính:

- Quy trình chính - Đây là điểm vào của ứng dụng. Đây là tệp sẽ được thực thi khi bạn chạy ứng dụng.
- Quá trình kết xuất - Đây là bộ điều khiển cho một cửa sổ nhất định trong ứng dụng. Mỗi cửa sổ có quy trình `Renderer` riêng.

### 2.8.4 Tính năng của Electron

#### Bảo vệ

Bạn không cần phải suy nghĩ nhiều khi di chuyển ứng dụng hiện tại của mình sang Electron vì ứng dụng chúng tôi đang tạo là một ứng dụng máy tính để bàn và dữ liệu nằm trong hệ thống. Vì điều này, bạn có thể đảm bảo an toàn dữ liệu. Trong trường hợp bạn cần lưu trữ dữ liệu trên đám mây, hãy kiểm tra xem mạng đám mây của bạn có đủ chức năng bảo mật trước đó để tránh mọi bất ngờ không mong muốn hay không.

#### Khả năng tiếp cận cấp thấp

Trước khi bắt đầu, bạn cũng phải kiểm tra xem tất cả các chức năng bạn đang cung cấp cho ứng dụng máy tính để bàn của bạn cũng có sẵn trong Electron hay không. Theo kinh nghiệm của tôi, Electron cung cấp đủ quyền kiểm soát để có các tính năng tương tác mở rộng trong ứng dụng của bạn, chẳng hạn như Phím tắt. Nó

cũng cung cấp khả năng truy cập cấp thấp cho các thành phần hệ thống phần cứng và hoạt động.

### **Khả năng truy cập phần cứng**

Các nhà phát triển có thể có quyền truy cập đầy đủ vào tất cả các API truy cập cấp phần cứng được hiển thị qua JavaScript / Plugin. Không cần phải thỏa hiệp cho tính năng này nếu bạn muốn chuyển sang Electron.

### **Hiệu suất**

Electron phát triển mạnh ở khía cạnh này. Nếu được chăm sóc đúng cách trong khi phát triển (chỉ tải những gì bạn cần), Electron có thể hiển thị một số lợi ích lớn về mặt hiệu suất khi so sánh với các ứng dụng gốc. Electron tiết kiệm rất nhiều thời gian và cung cấp nhiều tùy chọn hơn để chơi hoặc phát triển bằng cách có một cơ sở mã duy nhất cho tất cả các nền tảng chính. Đây là những vấn đề chính khi chúng ta đối phó với sự phát triển tự nhiên của các ứng dụng, tất cả đều được Electron giải quyết một cách hiệu quả.

Tuy nhiên, rất nhiều người nói rằng Electron rất đói bộ nhớ. Tôi đồng ý với tuyên bố này, nhưng chỉ khi bạn phát triển các ứng dụng mà không được chăm sóc đúng cách.

### **Quản lý mã và ứng dụng**

Là Chủ sở hữu sản phẩm, bạn không cần duy trì các nhóm khác nhau cho mỗi nền tảng và bạn sẽ được giải tỏa khỏi những rắc rối khi giải thích lại các yêu cầu của mình với các nhóm khác nhau. Nó cũng sẽ giảm công việc kiểm toán để đảm bảo sản phẩm có chức năng tương tự trên các nền tảng.

Là một nhà phát triển, bạn không cần phải lo lắng về các cơ sở mã khác nhau. Nếu bạn gặp lỗi trên bất kỳ nền tảng nào, bạn có thể sửa nó tại cơ sở mã. Các lỗi sẽ không bao giờ hiển thị trên các nền tảng khác. Tuy nhiên, bạn vẫn nên theo dõi các chức năng ở cấp độ hệ điều hành.

### **Tái sử dụng**

Bởi vì chúng tôi đang sử dụng một cơ sở mã duy nhất, điều đó có nghĩa là chúng tôi có thể sử dụng điều này cho cả ứng dụng web và ứng dụng máy tính để bàn. Theo một cách nào đó, chúng tôi cũng đang sử dụng lại mã cơ sở trên các nền tảng khác nhau kể từ khi chúng tôi mã hóa một lần, phân phối ở mọi nơi.

### **Sản xuất**

Chúng ta càng sử dụng nhiều khung công tác, chúng ta càng nhận được nhiều sự hỗ trợ hơn. Điều này, đến lượt nó, cung cấp cho chúng tôi nhiều thư viện nguồn mở hơn, chúng tôi có thể sử dụng lại và giảm thời gian sản xuất nhưng với nhiều tính năng hơn.

### **Triển khai / Xây dựng**

Đây là một trong những khía cạnh thú vị của Electron. Có một mô-đun đóng gói điện tử có sẵn giúp chúng ta bó toàn bộ cơ sở mã vào các gói tương ứng. Mặc dù mọi người tranh luận rằng Electron chiếm rất nhiều bộ nhớ, nhưng như tôi đã nói, Electron cần một chút quan tâm trong quá trình phát triển để tránh điều này.

### **UI / UX**

Với các công nghệ web, bạn có thể mở ra nhiều công nghệ cung cấp Giao diện người dùng (UI) và Trải nghiệm người dùng (UX) tuyệt vời cho tất cả người dùng của bạn một cách thoải mái. Ngoài ra, bạn có thể chắc chắn rằng bạn đang cung cấp trải nghiệm giống nhau cho tất cả người dùng của bạn trên các nền tảng khác nhau.

### **Chi phí và thời gian**

Bạn sẽ tiết kiệm rất nhiều thời gian và tiền bạc phát triển bởi vì, đối với gần xếp công nghệ mà chúng tôi sử dụng, có rất nhiều nhà phát triển có thể làm điều đó với chi phí thấp hơn và đạt được kết quả tốt. Bạn có thể tiết kiệm rất nhiều thời gian bằng cách sử dụng cơ sở mã duy nhất và bất kỳ nhà phát triển nào cũng có thể làm việc trên mọi thứ.

## **2.9 Tổng quan về ReactJS**

### **2.9.1 Giới thiệu tổng quát**

React (Hay ReactJS, React.js) là một thư viện Javascript mã nguồn mở để xây dựng các thành phần giao diện có thể tái sử dụng. Nó được tạo ra bởi Jordan Walke, một kỹ sư phần mềm tại Facebook. Người bị ảnh hưởng bởi XHP (Một nền tảng thành phần HTML cho PHP). React lần đầu tiên được triển khai cho ứng dụng Newsfeed của Facebook năm 2011, sau đó được triển khai cho Instagram.com năm 2012. Nó được mở mã nguồn (open-sourced) tại JSConf US tháng 5 năm 2013.

Nó cho phép những nhà phát triển web tạo ra giao diện người dùng nhanh chóng. Phần Views của Reactjs thường được hiển thị bằng việc chủ yếu dùng các component mà chứa các component cụ thể hoặc các thẻ HTML.

### **2.9.2 Tính năng của React**

#### **JSX**

JSX là phần mở rộng cú pháp JavaScript. Không nhất thiết phải sử dụng JSX trong phát triển React, nhưng nó được khuyến khích.

## **Component**

React bao gồm tất cả các component. Là một đoạn mã ngắn có ý nghĩa thể hiện một chức năng hay đối tượng nhất định, có thể tái sử dụng. Một ứng dụng React có thể có hàng chục, hoặc hàng trăm các component liên kết, tương tác với nhau. Bạn cần nghĩ mọi thứ như một component. Điều này sẽ giúp bạn duy trì mã khi làm việc trên các dự án quy mô lớn hơn.

## **Luồng dữ liệu một chiều và Flux**

React triển khai luồng dữ liệu một chiều giúp bạn dễ dàng quản lý về ứng dụng của mình. Flux là một kiến trúc mà Facebook sử dụng trong khi làm việc với React. Flux không phải là một framework hay một thư viện (library). Nó đơn giản chỉ là một kiểu kiến trúc mới hỗ trợ thêm cho React, đồng thời xây dựng ý tưởng về luồng dữ liệu một chiều (Unidirectional Data Flow).

## **Giấy phép**

React được cấp phép theo Facebook Inc. Tài liệu được cấp phép theo CC BY 4.0.

### **2.9.3 Ưu điểm và nhược điểm của ReactJS**

\* Nhược điểm

#### **Tài liệu phức tạp:**

Reactjs chỉ phục vụ cho tầng View. React chỉ là View Library nó không phải là một MVC framework như những framework khác. Đây chỉ là thư viện của Facebook giúp render ra phần view. Vì thế React sẽ không có phần Model và Controller, mà phải kết hợp với các thư viện khác. Do đó bạn vẫn cần chọn các công nghệ khác để có được bộ công cụ hoàn chỉnh để phát triển.

Khó tiếp cận cho người mới học Web

Không phải tất cả các trình duyệt tiêu chuẩn đều hỗ trợ React

Để giải quyết vấn đề này nên sử dụng các trình cắm bổ sung - ví dụ: thư viện S5-shim để hỗ trợ IE8. Có tồn tại các tiện ích mở rộng khác cho React.js, tuy nhiên, khi xem xét trọng số cơ bản của bộ khung, việc sử dụng chúng nên được giảm thiểu.

#### **Rất ít tiện ích có sẵn:**

React.js là một khung công tác tương đối trẻ, vì vậy tất cả, ngay cả các tiêu chuẩn, các widget sẽ phải được viết gần như từ đầu. Cần một hộp thả xuống hoặc hộp đèn tùy chỉnh? Bạn sẽ phải viết một số lượng lớn mã ngay cả đối với các tác vụ đơn giản như vậy.

**\* Ưu điểm**

Phát triển giao diện người dùng dựa trên các thành phần riêng biệt - và đó là tương lai của sự phát triển.

Reactjs cực kì hiệu quả: Reactjs tạo ra cho chính nó DOM ảo – nơi mà các component thực sự tồn tại trên đó. Điều này sẽ giúp cải thiện hiệu suất rất nhiều, vì DOM ảo JavaScript nhanh hơn DOM thông thường.

Có thể được sử dụng trên máy khách và máy chủ cũng như với các khuôn khổ khác.

Các mẫu thành phần và dữ liệu cải thiện khả năng đọc, giúp duy trì các ứng dụng lớn hơn.

Dễ tiếp cận với những người có kiến thức về JavaScript.

## **2.10 RESTful API**

API (Application Programming Interface) là giao diện lập trình ứng dụng giúp tạo ra các phương thức kết nối với các thư viện và ứng dụng khác nhau. Vậy RESTful API là API đi theo cấu trúc REST. RESTful API không được xem là một công nghệ, nó là một giải pháp để tạo ra các ứng dụng Web Services thay thế cho các kiểu khác như SOAP, WSDL (Web Service Definition Language),...

Một số HTTP method theo chuẩn RESTful API như sau:

- GET : truy xuất tài nguyên (READ)
- POST : tạo tài nguyên mới ( CREATE)
- PUT / PATCH: cập nhật, sửa đổi tài nguyên (UPDATE)
- DELETE : xóa tài nguyên ( DELETE)

Các nguyên lý của REST:

- Client-Server: bằng cách tách phần giao diện người dùng ra khỏi phần lưu trữ, xử lý dữ liệu. Ta có thể làm gọn các thành phần của máy chủ và được tách một máy chủ ra thành những máy chủ riêng biệt (máy chủ cho phần backend và máy chủ cho frontend). Khả năng mở rộng hệ thống cũng được cải thiện (VD: máy chủ backend có thể cung cấp cho cả giao diện web và di động thay vì dính liền với một trong hai).

- Không lưu trữ trạng thái (Stateless): mỗi request khi gửi đến máy chủ đều đính kèm các thông tin để máy chủ có thể hiểu được request đó (chẳng hạn như từ đâu, của ai, cần cái gì).
- Khả năng cache: nếu một phản hồi của máy chủ cho một yêu cầu được đánh dấu là có thể cache thì sau một khoảng thời gian nhất định, những yêu cầu giống như vậy sẽ sử dụng lại dữ liệu từ phản hồi đó. Điều này làm giảm áp lực đến máy chủ và tăng hiệu năng cho phía client.
- Chuẩn hoá các interface: bằng cách tạo ra các quy ước chuẩn để giao tiếp giữa các thành phần trong hệ thống để đơn giản hóa việc client có thể tương tác với server. Các quy ước này áp dụng cho toàn bộ các service giúp cho người sử dụng hệ thống dễ hơn. Dễ hiểu hơn trên hệ thống ta đặt ra 1 chuẩn API để người dùng dù là mobile, web đều có thể kết nối vào được. Hệ thống REST có yếu điểm ở đây vì khi chuẩn hóa rồi ta không thể tối ưu từng kết nối.
- Phân lớp hệ thống : chia tách các thành phần hệ thống theo từng lớp, mỗi lớp chỉ sử dụng lớp ở dưới nó và giao tiếp với lớp ở ngay trên nó. Điều này giúp giảm độ phức tạp của hệ thống, giúp các thành phần tách biệt nhau từ đó dễ dàng mở rộng từng thành phần.

## **2.11 Tổng quan về PowerDesigner**

### **2.11.1 Giới thiệu tổng quát**

PowerDesigner là môi trường mô hình hóa tổng thể dưới dạng đồ họa và dễ dàng sử dụng. Công dụng PowerDesigner cung cấp:

- Trình bày các mô hình bằng các ký hiệu đồ họa phù hợp với các phương pháp mô hình hóa trong thực tế như mô hình thực thể và mối kết hợp; mô hình quan hệ...
- Tự động tạo hồ sơ mô tả các đối tượng trên mô hình.
- Tự động tạo mã phát sinh CSDL và các chức năng xử lý từ mô hình đã xây dựng.

Vai trò của Power Designer trong việc thiết kế mô hình cơ sở dữ liệu :

- Trình bày mô hình ở dạng đồ họa.
- Kiểm tra tính hợp lệ của mô hình được thiết kế.
- Phát sinh mô hình dữ liệu logic hay mô hình dữ liệu vật lý.

### 2.11.2 Sơ lược về thực thể, thuộc tính, liên kết

Thực thể: là khái niệm để chỉ một lớp các đối tượng có cùng đặc tính chung mà người ta muốn quản lý thông tin về nó. Ví dụ, sinh viên, hàng hóa, vật tư.. Một đối tượng cụ thể trong thực thể được gọi là một cá thể (còn gọi là một thể hiện của thực thể).

Thuộc tính: Để mô tả thông tin về một thực thể người ta thường dựa vào các đặc trưng riêng của thực thể đó. Các đặc trưng đó được gọi là thuộc tính của thực thể.

Liên kết: là một sự ghép nối giữa hai hay nhiều thực thể. Phản ánh sự liên hệ giữa các thực thể. Có 3 kiểu liên kết:

- Quan hệ một một (1-1) : Hai thực thể A và B có mối quan hệ 1-1 nếu một thực thể kiểu A tương ứng với một thực thể kiểu B và ngược lại.
- Quan hệ một nhiều (1-n) : Hai thực thể A và B có mối quan hệ 1- n nếu một thực thể kiểu A tương ứng với nhiều thực thể kiểu B và một thực thể của B chỉ tương ứng với một thực thể kiểu A.
- Quan hệ nhiều nhiều (n-n) : Hai thực thể A và B có mối quan hệ n-n nếu một thực thể kiểu A tương ứng với nhiều thực thể kiểu B và ngược lại.

### 2.12 Tổng quan về ngôn ngữ lập trình C#

Ngôn ngữ C# được phát triển bởi đội ngũ kỹ sư của Microsoft, trong đó người dẫn đầu là Anders Hejlsberg và Scott Wiltamuth. Cả hai người này đều là những người nổi tiếng, trong đó Anders Hejlsberg được biết đến là tác giả của Turbo Pascal, một ngôn ngữ lập trình PC phổ biến. Và ông đứng đầu nhóm thiết kế Borland Delphi, một trong những thành công đầu tiên của việc xây dựng môi trường phát triển tích hợp (IDE) cho lập trình client/server.

Ngôn ngữ C# là một ngôn ngữ được dẫn xuất từ C, C++ và Java, nhưng nó được tạo từ nền tảng phát triển hơn. Microsoft bắt đầu với công việc trong C, C++ và Java và thêm vào những đặc tính mới để làm cho ngôn ngữ này dễ sử dụng hơn. Nhiều trong số những đặc tính này khá giống với những đặc tính có trong ngôn ngữ Java. Không dừng lại ở đó, Microsoft đưa ra một số mục đích khi xây dựng ngôn ngữ này. Những mục đích này được tóm tắt như sau

#### **C# là ngôn ngữ đơn giản:**

Như ta đã biết thì ngôn ngữ C# dựng trên nền tảng C++ và Java nên ngôn ngữ C# khá đơn giản. Nếu chúng ta thân thiện với C và C++ hoặc thậm chí là Java, chúng ta sẽ thấy C# khá giống về diện mạo, cú pháp, biểu thức, toán tử và những

chức năng khác được lấy trực tiếp từ ngôn ngữ C và C++, nhưng nó đã được cải tiến để làm cho ngôn ngữ đơn giản hơn. Một vài trong các sự cải tiến là loại bỏ các dư thừa, hay là thêm vào những cú pháp thay đổi.

### **C# là ngôn ngữ hiện đại:**

Một vài khái niệm khá mới mẻ khá mơ hồ với các bạn vừa mới học lập trình, như xử lý ngoại lệ, những kiểu dữ liệu mở rộng, bảo mật mã nguồn..v..v... Đây là những đặc tính được cho là của một ngôn ngữ hiện đại cần có. Và C# chứa tất cả các đặc tính ta vừa nêu trên. Các bạn sẽ dần tìm hiểu được các đặc tính trên qua các bài học trong series này.

### **C# là một ngôn ngữ lập trình thuần hướng đối tượng:**

Lập trình hướng đối tượng(tiếng Anh: Object-oriented programming, viết tắt: OOP) là một phương pháp lập trình có 4 tính chất. Đó là tính trừu tượng (abstraction), tính đóng gói (encapsulation), tính đa hình (polymorphism) và tính kế thừa (inheritance). C# hỗ trợ cho chúng ta tất cả những đặc tính trên. Và để hiểu rõ hơn thì chúng ta sẽ có một chương trình bày về phần này.

### **C# là một ngôn ngữ ít từ khóa:**

C được sử dụng để mô tả ngôn ngữ# là ngôn ngữ sử dụng giới hạn những từ khóa (gồm khoảng 80 từ khóa và mười mấy kiểu dữ liệu xây dựng sẵn). Nếu bạn nghĩ rằng ngôn ngữ có càng nhiều từ khóa thì sẽ càng mạnh mẽ hơn. Điều này không phải sự thật, lấy ví dụ ngôn ngữ C# làm điển hình nhé. Nếu bạn học sâu về C# bạn sẽ thấy rằng ngôn ngữ này có thể được sử dụng để làm bất cứ nhiệm vụ nào.

## **2.13 Tổng quan về HTML, CSS, JavaScript**

### **2.13.1 HTML**

HTML (HyperText Markup Language) là một ngôn ngữ đánh dấu được thiết kế ra để tạo nên các trang web với các mẫu thông tin được trình bày trên World Wide Web. Cùng với CSS và JavaScript, HTML tạo ra bộ ba nền tảng kỹ thuật cho World Wide Web. HTML được định nghĩa như là một ứng dụng đơn giản của SGML và được sử dụng trong các tổ chức cần đến các yêu cầu xuất bản phức tạp. HTML đã trở thành một chuẩn Internet do tổ chức World Wide Web Consortium (W3C) duy trì. Phiên bản chính thức mới nhất của HTML là HTML 4.01 (1999). Sau đó, các nhà phát triển đã thay thế nó bằng XHTML. Hiện nay, HTML đang được phát triển tiếp với phiên bản HTML5 hứa hẹn mang lại diện mạo mới cho Web.



Bằng cách dùng HTML động hoặc Ajax, lập trình viên có thể được tạo ra và xử lý bởi số lượng lớn các công cụ, từ một chương trình soạn thảo văn bản đơn giản – có thể gõ vào ngay từ những dòng đầu tiên – cho đến những công cụ xuất bản WYSIWYG phức tạp. Hypertext là cách mà các trang Web (các tài liệu HTML) được kết nối với nhau. Và như thế, đường link có trên trang Web được gọi là Hypertext. Như tên gọi đã gợi ý, HTML là ngôn ngữ đánh dấu bằng thẻ (Markup Language), nghĩa là bạn sử dụng HTML để đánh dấu một tài liệu text bằng các thẻ (tag) để nói cho trình duyệt Web cách để cấu trúc nó để hiển thị ra màn hình.

Có bốn loại phần tử đánh dấu trong HTML:

- Đánh dấu Có cấu trúc miêu tả mục đích của phần văn bản
- Đánh dấu trình bày miêu tả phần hiển hình trực quan của phần văn bản bất kể chức năng của nó là gì. Cách dùng đánh dấu trình bày này bây giờ không còn được khuyến dùng mà nó được thay thế bằng cách dùng CSS.
- Đánh dấu liên kết ngoài chứa phần liên kết từ trang này đến trang kia.
- Các phần tử thành phần điều khiển giúp tạo ra các đối tượng (VD: các nút và các danh sách).

### 2.13.2 CSS

CSS (Cascading Style Sheets) được dùng để miêu tả cách trình bày các tài liệu viết bằng ngôn ngữ HTML và XHTML. Ngoài ra ngôn ngữ định kiểu theo tầng cũng có thể dùng cho XML, SVG, XUL. Các đặc điểm kỹ thuật của CSS được duy trì bởi World Wide Web Consortium (W3C). Thay vì đặt các thẻ quy định kiểu dáng cho văn bản HTML (hoặc XHTML) ngay trong nội dung của nó thì nên sử dụng CSS.

#### Tác dụng của CSS:

Hạn chế tối thiểu việc làm rối mã HTML của trang Web bằng các thẻ quy định kiểu dáng (chữ đậm, chữ in nghiêng, chữ có gạch chân, chữ màu), khiến mã nguồn của trang Web được gọn gàng hơn, tách nội dung của trang Web và định dạng hiển thị, dễ dàng cho việc cập nhật nội dung.

Tạo ra các kiểu dáng có thể áp dụng cho nhiều trang Web, giúp tránh phải lặp lại việc định dạng cho các trang Web giống nhau.

Có 3 cách để sử dụng CSS:

- Inline CSS: Áp dụng trực tiếp trên một đối tượng nhất định bằng thuộc tính style.

- Internal CSS: Đặt CSS ở đầu trang Web để áp dụng kiểu dáng cho toàn bộ trang ấy, khi đó chỉ cần đặt đoạn CSS vào trong cặp thẻ <style> rồi đặt vào trong phần header của Web (giữa <head> và </head>).
- External CSS: Đặt các thuộc tính CSS vào một tệp tin riêng biệt (\*.css), khi đó có thể tham chiếu đến từ nhiều trang Web khác nhau.

Việc áp dụng các định dạng sẽ được ưu tiên cho các kiểu nguồn có mức độ ưu tiên cao hơn.

Bảng 2.2 Bảng thể hiện mức độ ưu tiên của các kiểu nguồn

| Mức độ ưu tiên | Kiểu nguồn                              | Mô tả   |
|----------------|---|---|
| 1              | Quan trọng (important)                  | Thẻ “!important” cho phép ghi đè lên các kiểu dáng trước  |
| 2              | Trực tiếp (inline)                      | Kiểu dáng xác định trực tiếp cho một đối tượng HTML bằng thuộc tính style   |
| 3              | Phương tiện (media)                     | Kiểu dáng chung xác định cho tất cả phương tiện trong văn bản, cho đến khi một kiểu dáng khác được xác định cho một loại phương tiện cụ thể |
| 4              | Người dùng chỉ định (user defines)      | Một số trình duyệt hiện đại cho phép người dùng quyết định kiểu dáng cho nội dung hiển thị  |
| 5              | Bộ chọn cụ thể (specific selector)      | Kiểu dáng xác định bởi bộ chọn cụ thể trong từng tình huống (VD: #heading p) sẽ đè lên các kiểu dáng chung                                  |
| 6              | Thứ tự thuộc tính (rule order)          | Kiểu dáng ở cuối tệp tin sẽ có mức ưu tiên cao hơn kiểu dáng ở đầu tệp.   |
| 7              | Thuộc tính kế thừa (parent inheritance) | Một thành phần không có kiểu dáng cụ thể nào được gán cho nó sẽ kế thừa kiểu dáng của thuộc tính mẹ   |
| 8              | Kiểu dáng ở đầu trang HTML              | Kiểu dáng được đặt ở trong cặp thẻ <style> trong phần <head> của một trang HTML có  |

|   |                          |  |
|---|--------------------------|--|
|   |                          | mức độ ưu tiên cao hơn kiểu dáng mặc định của trình duyệt                                      |
| 9 | Mặc định của trình duyệt | Nếu một thành phần không có bất cứ kiểu dáng nào sẽ sử dụng kiểu dáng mặc định của trình duyệt |

### 2.13.3 JavaScript

JavaScript, theo phiên bản hiện hành, là một ngôn ngữ lập trình thông dịch được phát triển từ các ý niệm nguyên mẫu. Ngôn ngữ này được dùng rộng rãi cho các trang web (phía người dùng) cũng như phía máy chủ (với Nodejs). Nó vốn được phát triển bởi Brendan Eich tại Hãng truyền thông Netscape với cái tên đầu tiên Mocha, rồi sau đó đổi tên thành LiveScript, và cuối cùng thành JavaScript. Giống Java, JavaScript có cú pháp tương tự C, nhưng nó gần với Self hơn Java. “.js” là phần mở rộng thường được dùng cho tập tin mã nguồn JavaScript.

Phiên bản mới nhất của JavaScript là ECMAScript 11 (2020). ECMAScript là phiên bản chuẩn hóa của JavaScript.

JavaScript là một ngôn ngữ lập trình dựa trên nguyên mẫu với cú pháp phát triển từ C. Giống như C, JavaScript có khái niệm từ khóa, do đó, JavaScript gần như không thể được mở rộng.

Cũng giống như C, JavaScript không có bộ xử lý xuất/nhập (input/output) riêng. Trong khi C sử dụng thư viện xuất/nhập chuẩn, JavaScript dựa vào phần mềm ngôn ngữ được gắn vào để thực hiện xuất/nhập.

Trên trình duyệt, rất nhiều trang web sử dụng JavaScript để thiết kế trang web động và một số hiệu ứng hình ảnh thông qua DOM. JavaScript được dùng để thực hiện một số tác vụ không thể thực hiện được với chỉ HTML như kiểm tra thông tin nhập vào, tự động thay đổi hình ảnh,... Ở Việt Nam, JavaScript còn được ứng dụng để làm bộ gõ tiếng Việt giống như bộ gõ hiện đang sử dụng trên trang Wikipedia tiếng Việt. Tuy nhiên, mỗi trình duyệt áp dụng JavaScript khác nhau và không tuân theo chuẩn W3C DOM, do đó trong rất nhiều trường hợp lập trình viên phải viết nhiều phiên bản của cùng một đoạn mã nguồn để có thể hoạt động trên nhiều trình duyệt. Một số công nghệ nổi bật dùng JavaScript để tương tác với DOM bao gồm DHTML, Ajax và SPA.

Bên ngoài trình duyệt, JavaScript có thể được sử dụng trong tập tin PDF của Adobe Acrobat và Adobe Reader. Điều khiển Dashboard trên hệ điều hành Mac OS X phiên bản 10.4 cũng có sử dụng JavaScript. Công nghệ kịch bản linh động (active

scripting) của Microsoft có hỗ trợ ngôn ngữ JScript làm một ngôn ngữ kịch bản dùng cho hệ điều hành. JScript.NET là một ngôn ngữ tương thích với CLI gần giống JScript nhưng có thêm nhiều tính năng lập trình hướng đối tượng.

Từ khi Nodejs ra đời vào năm 2009, Javascript được biết đến nhiều hơn là một ngôn ngữ đa nền khi có thể chạy trên cả môi trường máy chủ cũng như môi trường nhúng.

Mỗi ứng dụng này đều cung cấp mô hình đối tượng riêng cho phép tương tác với môi trường chủ, với phần lõi là ngôn ngữ lập trình JavaScript gần như giống nhau.

### Cú pháp trong JavaScript

- Dấu cách, tab và ký tự dòng mới sử dụng bên ngoài một chuỗi ký tự được gọi là khoảng trắng. Khác với C, khoảng trắng trong JavaScript có thể ảnh hưởng trực tiếp tới ý nghĩa của câu lệnh. Sử dụng phương pháp "tự động thêm dấu chấm phẩy", bất cứ một dòng JavaScript nào thích hợp sẽ được coi là một câu lệnh hợp lệ (giống như có dấu chấm phẩy trước ký tự dòng mới).

- Cú pháp chú thích của JavaScript giống với C++. Lập trình viên có thể chú thích trên nhiều dòng bằng cách bao bọc chú thích với /\* và \*/ hoặc sử dụng // để chú thích từ vị trí // đến hết dòng.

- Có 3 cách để định nghĩa biến trong JavaScript: var, let, const (let và const chỉ áp dụng được với các phiên bản ECMAScript 6 trở đi).

- Toán tử số học: + (cộng), - (trừ), \* (nhân), / (chia), % (chia lấy dư).

- Toán tử tăng, giảm: ++(tăng), --(giảm)

- Toán tử quan hệ: == (bằng nhau), != (khác nhau), === (bằng nhau và cùng loại), !== (khác nhau và khác loại), > (lớn hơn), < (nhỏ hơn), >= (lớn hơn hoặc bằng), <= (nhỏ hơn hoặc bằng).

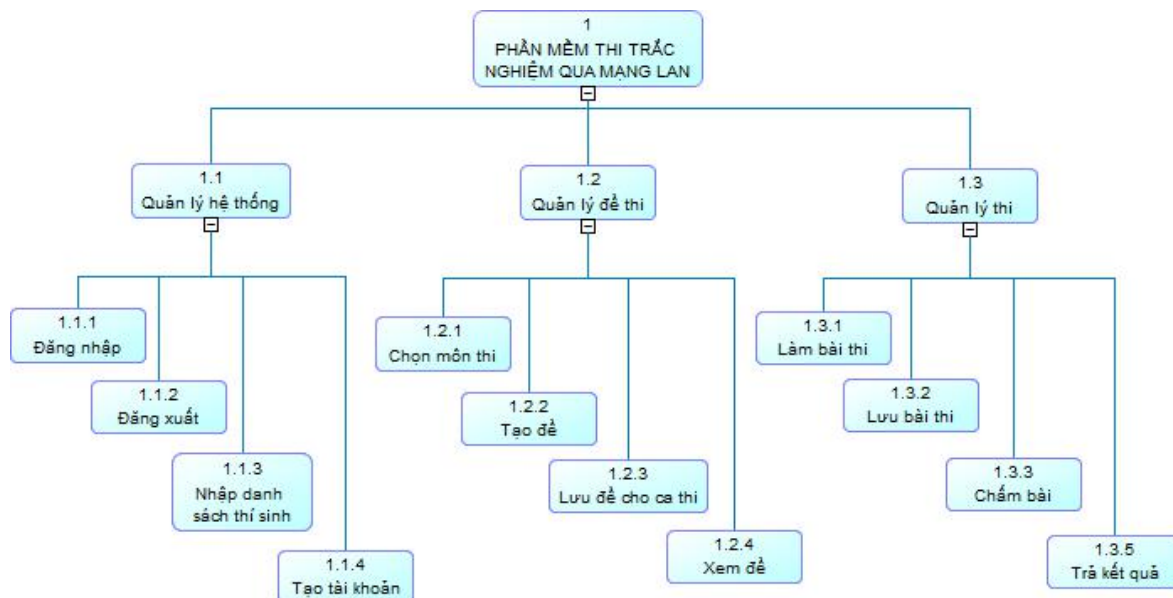
- Toán tử logic: && (và), || (hoặc), ! (phủ định).

- Toán tử tam phân (ternary): cú pháp: điều\_kiện ? biểu\_thức\_đúng: biểu\_thức\_sai; Toán tử này sẽ trả lại giá trị là kết quả của biểu\_thức\_đúng nếu điều\_kiện có giá trị bool bằng true, ngược lại nó sẽ trả lại giá trị bằng biểu\_thức\_sai.

## CHƯƠNG 3: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

### 3.1 Phân tích hệ thống

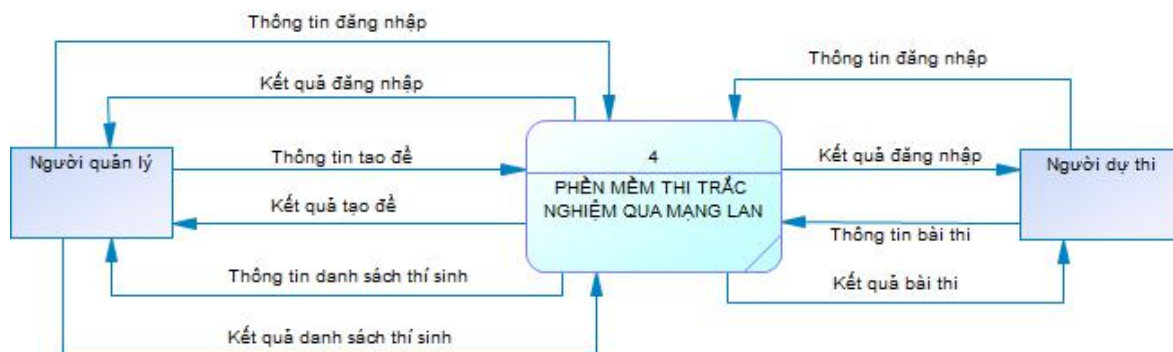
#### 3.1.1 Xây dựng sơ đồ phân rã chức năng BFD



Hình 3. 1 Sơ đồ phân cấp chức năng BFD

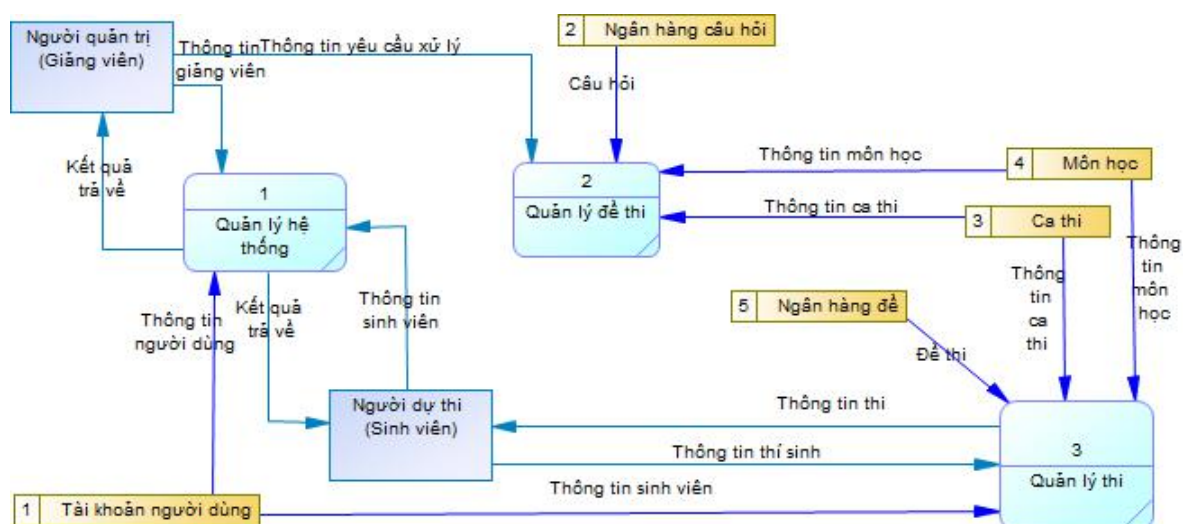
#### 3.1.2 Mô hình luồng dữ liệu DFD

##### 3.1.2.1 Mức ngữ cảnh



Hình 3.2 Mô hình luồng dữ liệu mức ngữ cảnh

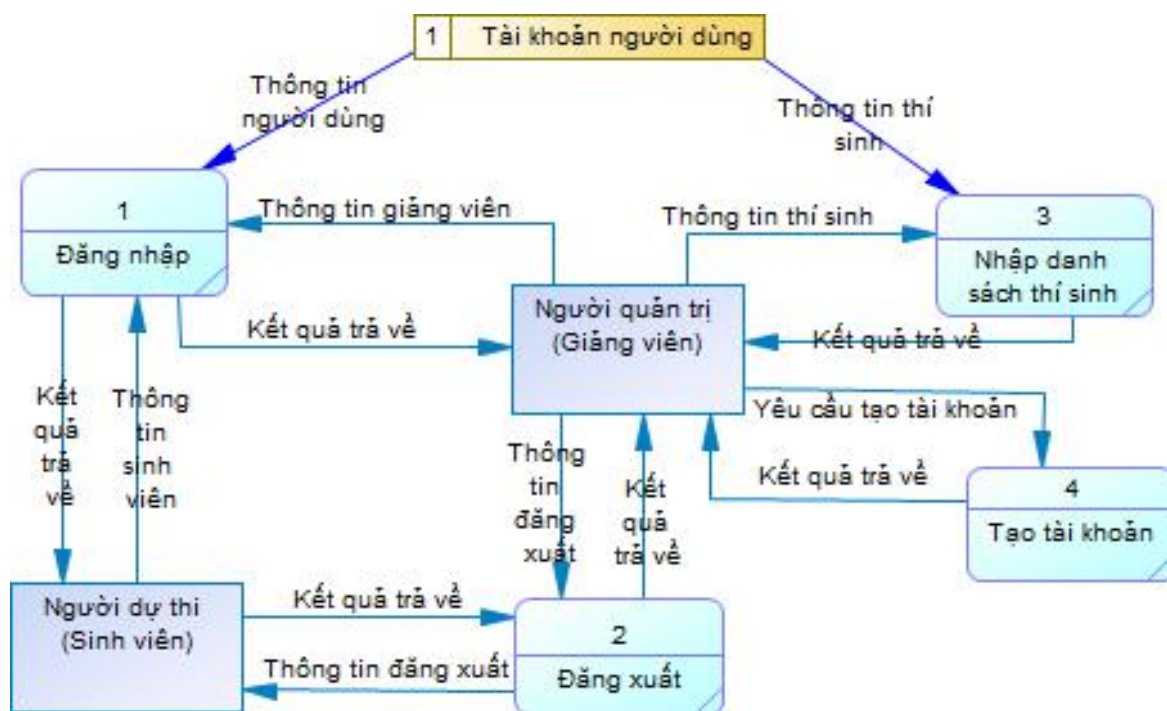
##### 3.1.2.2 Mức 1



Hình 3.3 Mô hình luồng dữ liệu mức 1

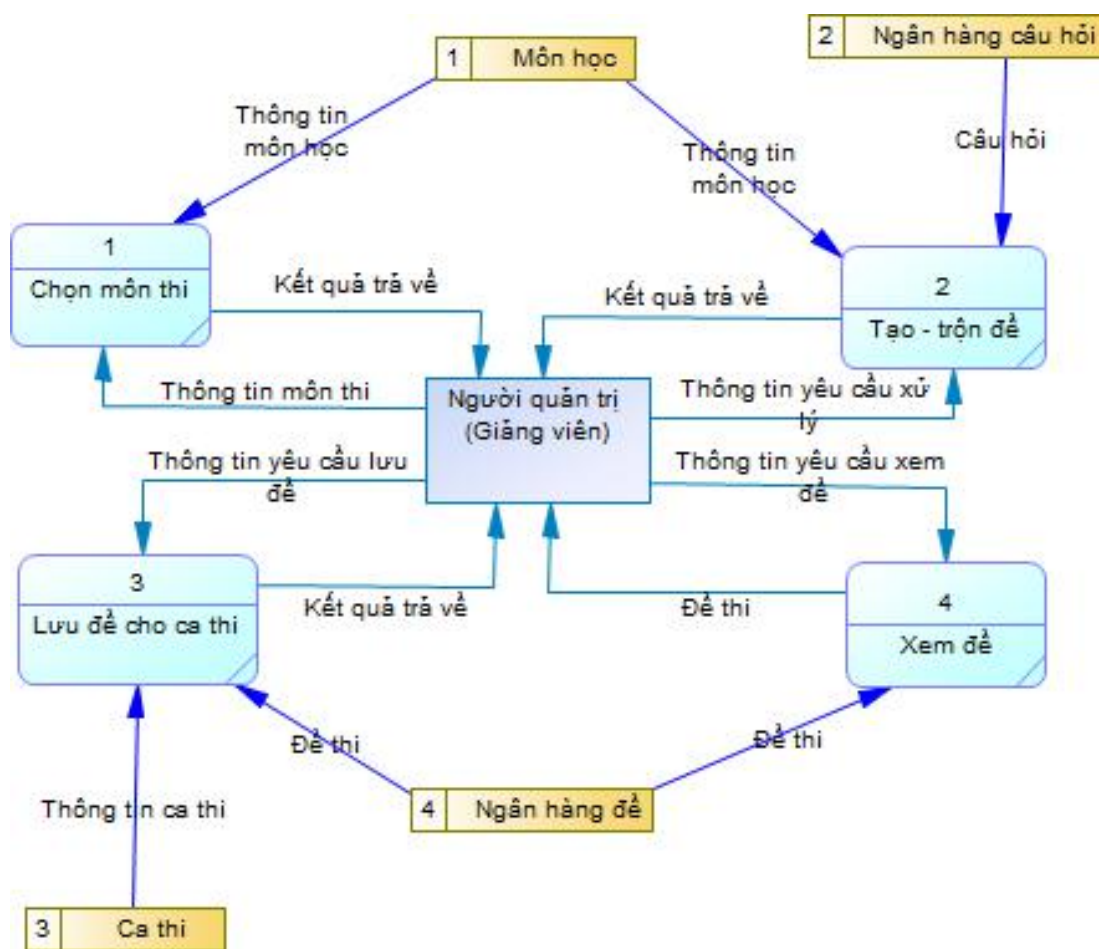
#### 4.1.2.3 Mức 2

\* Quản lý hệ thống



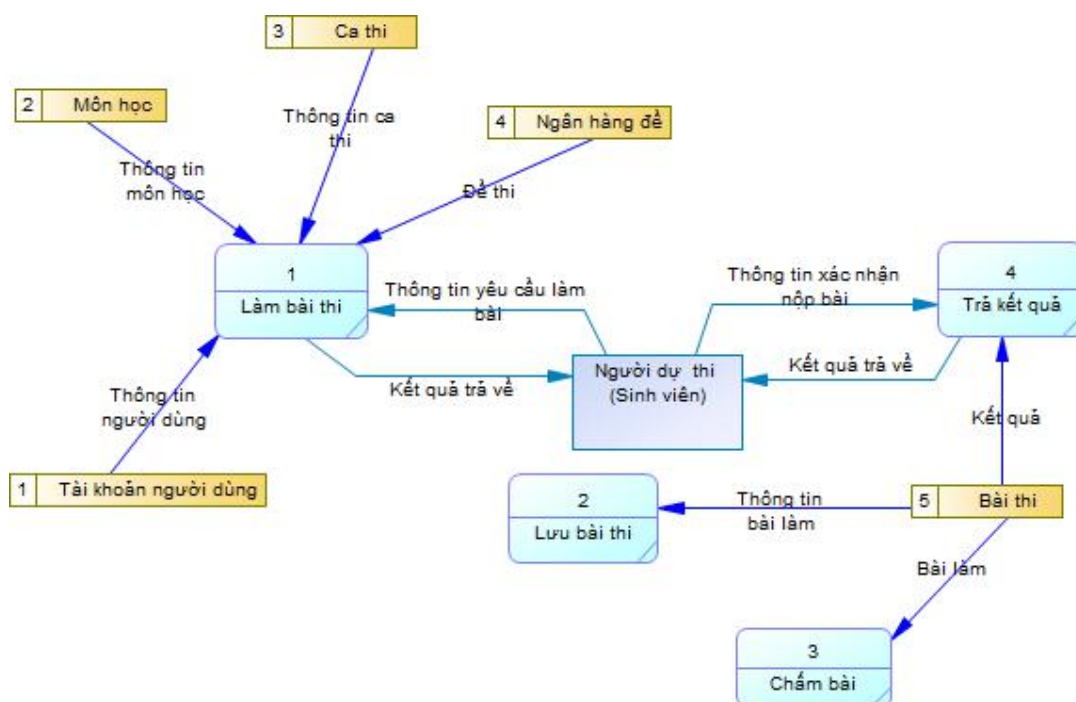
Hình 3.4 Mô hình luồng dữ liệu mức 2 của quản lý hệ thống

\* Quản lý đề thi



Hình 3.5 Mô hình luồng dữ liệu mức 2 của quản lý đề thi

\*Quản lý thi

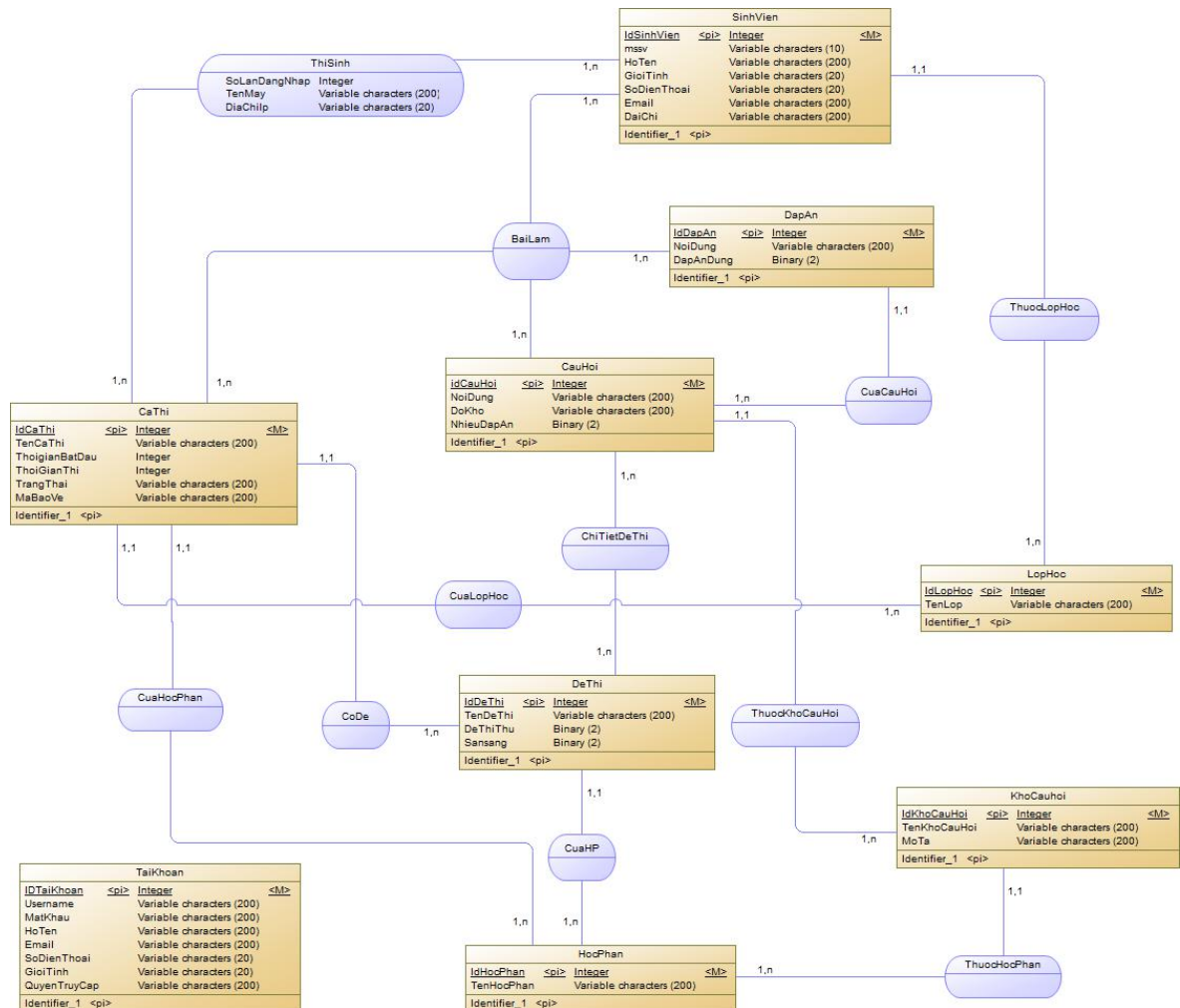


Hình 3.6 Mô hình luồng dữ liệu mức 2 của quản lý thi

## 3.2 Thiết kế cơ sở dữ liệu

### 3.2.1 Xây dựng mô hình thực thể kết hợp ERD

#### 3.2.1.1 Mô hình



Hình 3.7 Mô hình thực thể kết hợp ERD

#### 3.2.1.2 Mô tả chi tiết các thực thể

##### \* Thực thể TaiKhoan (Tài khoản)

- Là tập hợp các tài khoản của người dùng.
- Mỗi tài khoản có một mã duy nhất là IdTaiKhoan (Id tài khoản), username (Tên người dùng), MatKhau (Mật khẩu), hoTen (Họ tên), Email (Địa chỉ email), SoDienThoai (Số điện thoại), GioiTinh (Giới tính), QuyenTruyCap (Quyền truy cập).

##### \* Thực thể LopHoc (Lớp học)



- Là tập hợp các lớp của khoa KT\_CN trong trường.
- Mỗi lớp học có một mã duy nhất là IdLopHoc (Id lớp học), TenLop (Tên lớp học).

\* Thực thể HocPhan (Học phần)

- Là tập hợp các học phần của sinh viên trong trường.
- Mỗi học phần có một mã duy nhất là IdHocPhan (Id học phần), TenHocPhan (Tên học phần).

\* Thực thể DeThi (Đề thi)

- Là tập hợp các đề thi của một học phần.
- Mỗi đề thi của từng học phần có một mã duy nhất là IdDeThi (Id đề thi), TenDeThi (Tên đề thi), DeThiThu (Đề thi thứ), SangSang (Sẵn sàng).

\* Thực thể KhoCauHoi (Kho câu hỏi)

- Là tập hợp các câu hỏi của một học phần.
- Mỗi kho câu hỏi của từng học phần có một mã duy nhất là IdKhoCauHoi (Id kho câu hỏi), TenKhoCauHoi (Tên kho câu hỏi), MoTa (Mô tả).

\* Thực thể CauHoi (Câu hỏi)

- Là tập hợp các câu hỏi trong kho câu hỏi của một đề thi.
- Mỗi câu hỏi có một mã duy nhất là IdCauHoi (Id câu hỏi), NoiDung (Nội dung), NhiencauHoi (Nhiều câu hỏi), DoKho (Độ khó).

\* Thực thể DapAn (Đáp án)

- Là tập hợp các đáp án trong các câu hỏi của một bài thi.
- Mỗi đáp án có một mã duy nhất là IdDapAn (Id đáp án), NoiDung (Nội dung), DapAnDung (Đáp án đúng).

\* Thực thể SinhVien (Sinh viên)

- Là tập hợp các sinh viên của một lớp làm bài thi.
- Mỗi sinh viên có một mã duy nhất là IdSinhVien (Id sinh viên), mssv (mã số sinh viên), hoten (Họ tên), GioiTinh (Giới tính của sinh viên), SoDienThoai (Số điện thoại liên hệ của sinh viên), Email (địa chỉ Email của sinh viên), DiaChi (Địa chỉ của sinh viên).

\* Thực thể CaThi (Ca thi)

- Là tập hợp các ca thi của các lớp trong khoa.
- Mỗi ca thi có một mã duy nhất là IdCaThi (Id ca thi), TenCaThi (Tên ca thi), ThoiGianBatDau (Thời gian bắt đầu), ThoiGianThi (Thời gian thi), TrangThai (Trạng thái), MaBaoVe (Mã bảo vệ).

### **3.2.1.3 Mô tả chi tiết các mối kết hợp**

#### **\* Mối kết hợp ThuocLopHoc (Thuộc lớp học)**

- Là mối kết hợp giữa thực thể SINHVIEN và thực thể LopHoc.
- Mỗi sinh viên chỉ học 1 lớp, ta có mối quan hệ (1,1).
- Mỗi lớp có nhiều sinh viên, ta có mối quan hệ (1,n).

#### **\* Mối quan hệ CuaLopHoc (Của lớp học)**

- Là mối kết hợp giữa thực thể CaThi và thực thể LopHoc.
- Mỗi ca thi chỉ là của 1 lớp học, ta có mối quan hệ (1,1).
- Mỗi lớp học có nhiều ca thi, ta có mối quan hệ (1,n).

#### **\* Mối quan hệ ThuocKhoCauHoi (Thuộc kho câu hỏi)**

- Là mối kết hợp giữa thực thể CauHoi và thực thể KhoCauHoi.
- Mỗi câu hỏi chỉ thuộc 1 kho câu hỏi, ta có mối quan hệ (1,1).
- Mỗi kho câu hỏi có nhiều câu hỏi, ta có mối quan hệ (1,n).

#### **\* Mối quan hệ CuaCauHoi (Của kho câu hỏi)**

- Là mối kết hợp giữa thực thể DapAn và thực thể CauHoi.
- Mỗi đáp án của chỉ 1 câu hỏi, ta có mối quan hệ (1,1).
- Mỗi câu hỏi có thể có nhiều đáp án, ta có mối quan hệ (1,n).

#### **\* Mối quan hệ ThuocHocPhan (Thuộc học phần)**

- Là mối kết hợp giữa thực thể KhoCauHoi và thực thể HocPhan.
- Mỗi kho câu hỏi chỉ thuộc một học phần, ta có mối quan hệ (1,1).
- Mỗi học phần có thể có nhiều kho câu hỏi, ta có mối quan hệ (1,n).

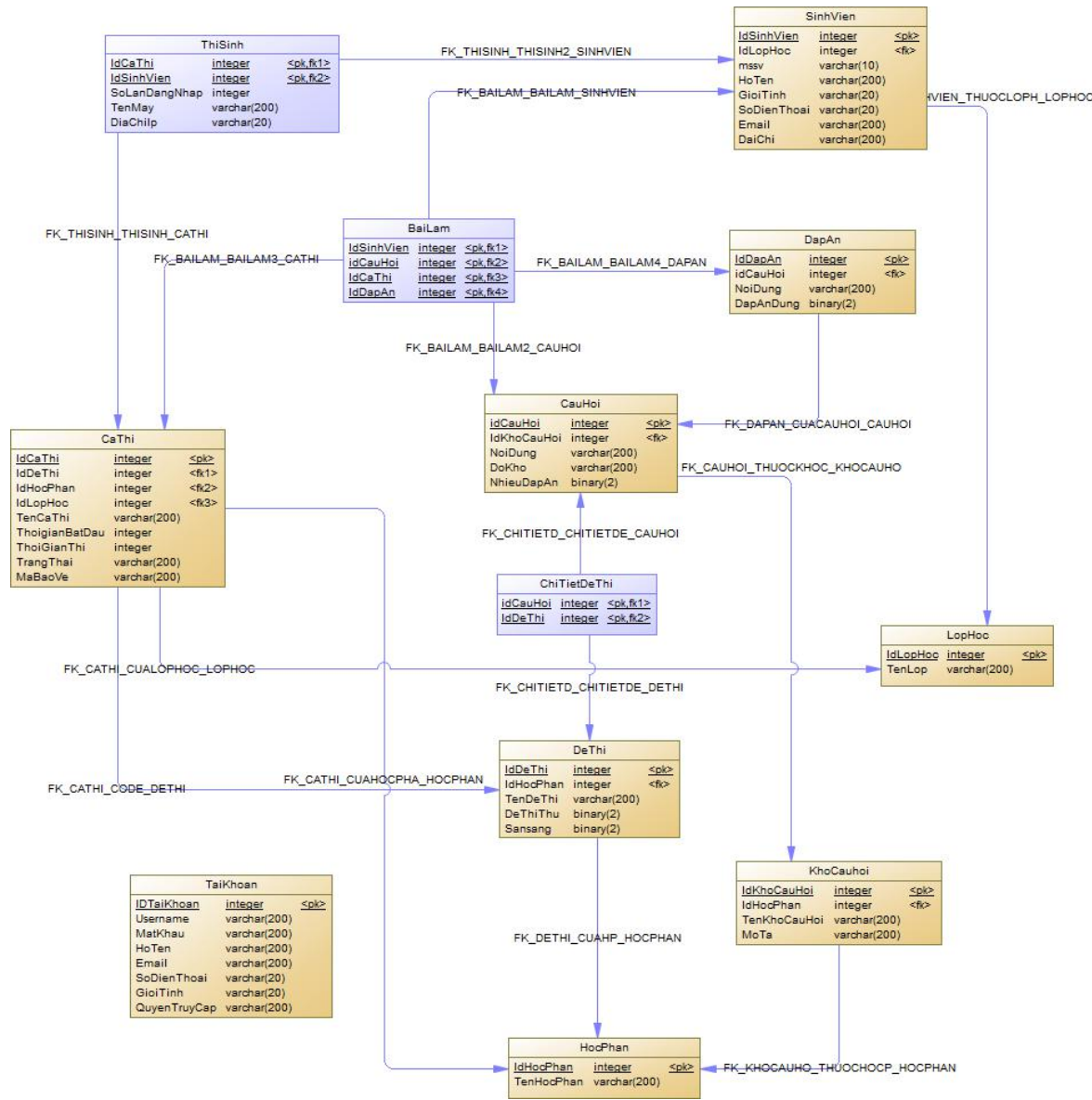
#### **\* Mối quan hệ CuaHocPhan (Của học phần)**

- Là mối kết hợp giữa thực thể CaThi và thực thể HocPhan.
- Mỗi ca thi của chỉ 1 học phần, ta có mối quan hệ (1,1).

- Mỗi học phần có nhiều ca thi, ta có mỗi quan hệ (1,n).
- \* Mỗi quan hệ CuaHP (Của học phần)
  - Là mỗi kết hợp giữa thực thể DeThi và thực thể HocPhan.
  - Mỗi đề thi của chỉ 1 học phần, ta có mỗi quan hệ (1,1).
  - Mỗi học phần có nhiều đề thi, ta có mỗi quan hệ (1,n).
- \* Mỗi quan hệ CoDe (Có đề)
  - Là mỗi kết hợp giữa thực thể CaThi và thực thể DeThi.
  - Mỗi ca thi chỉ có 1 đề thi, ta có mỗi quan hệ (1,1).
  - Mỗi đề thi có nhiều ca thi, ta có mỗi quan hệ (1,n).
- \* Mỗi quan hệ ChiTietDeThi (Chi tiết đề thi)
  - Là mỗi kết hợp giữa thực thể CauHoi và thực thể DeThi.
  - Mỗi câu hỏi có nhiều đề thi, ta có mỗi quan hệ (1,n).
  - Mỗi đề thi có nhiều câu hỏi, ta có mỗi quan hệ (1,n).
- \* Mỗi quan hệ ThiSinh (Thí sinh)
  - Là mỗi kết hợp giữa thực thể CaThi và thực thể SinhVien. Mỗi thí sinh có SoLanDangNhap (Số lần đăng nhập), SoMay (Số máy), DiaChiIp (Địa chỉ Ip).
  - Mỗi ca thi có nhiều sinh viên, ta có mỗi quan hệ (1,n).
  - Mỗi sinh viên có nhiều ca thi, ta có mỗi quan hệ (1,n).
- \* Mỗi quan hệ BaiLam (Bài làm)
  - Là mỗi kết hợp giữa thực thể CaThi và thực thể SinhVien và thực thể CauHoi và thực thể DapAn, cho biết đáp án các câu hỏi của sinh viên trong ca thi.
  - Mỗi ca thi có bài làm của nhiều sinh viên, ta có mỗi quan hệ (1,n).
  - Mỗi sinh viên có bài làm của nhiều ca thi, ta có mỗi quan hệ (1,n).
  - Mỗi câu hỏi của bài làm có nhiều đáp án, ta có mỗi quan hệ (1,n).
  - Mỗi đáp án bài làm của nhiều câu hỏi, ta có mỗi quan hệ (1,n).

3.2.2 Xây dựng mô hình dữ liệu vật lý

3.2.2.1 Mô hình



Hình 3.8 Mô hình dữ liệu mức vật lý

3.2.2.2 Mô tả chi tiết các bảng

Bảng 3.1 Mô tả chi tiết bảng TaiKhoan

| STT | Tên thuộc tính    | Diễn giải     | Kiểu dữ liệu | Ràng buộc | Mô tả      |
|-----|-------------------|---------------|--------------|-----------|------------|
| 1   | <u>IdTaiKhoan</u> | Id tài khoản  | Bigint       | Not null  | Khóa chính |
| 2   | Username          | Tên đăng nhập | Varchar(max) | Null      |            |

|   |              |                |               |      |  |
|---|--------------|----------------|---------------|------|--|
| 3 | MatKhau      | Mật khẩu       | Nvarchar(max) | Null |  |
| 4 | HoTen        | Họ tên         | Nvarchar(max) | Null |  |
| 5 | Email        | Email          | Nvarchar(max) | Null |  |
| 6 | SoDienThoai  | Số điện thoại  | Nvarchar(max) | Null |  |
| 7 | GioiTinh     | Giới tính      | Nvarchar(max) | Null |  |
| 8 | QuyenTruyCap | Quyền truy cập | Nvarchar(max) | Null |  |

Bảng 3.2 Mô tả chi tiết bảng LopHoc

| STT | Tên thuộc tính  | Diễn giải  | Kiểu dữ liệu  | Ràng buộc | Mô tả      |
|-----|-----------------|------------|---------------|-----------|------------|
| 1   | <u>IdLopHoc</u> | Id lớp học | Bigint        | Not null  | Khóa chính |
| 2   | TenLop          | Tên lớp    | Nvarchar(max) | Null      |            |

Bảng 3.3 Mô tả chi tiết bảng HocPhan

| STT | Tên thuộc tính   | Diễn giải    | Kiểu dữ liệu  | Ràng buộc | Mô tả      |
|-----|------------------|--------------|---------------|-----------|------------|
| 1   | <u>IdHocPhan</u> | Id học phần  | Bigint        | Not null  | Khóa chính |
| 2   | TenHocPhan       | Tên học phần | Nvarchar(max) | Null      |            |

Bảng 3.4 Mô tả chi tiết bảng DeThi

| STT | Tên thuộc tính | Diễn giải | Kiểu dữ liệu | Ràng buộc | Mô tả      |
|-----|----------------|-----------|--------------|-----------|------------|
| 1   | <u>IdDeThi</u> | Id đề thi | Bigint       | Not null  | Khóa chính |

|   |           |             |               |          |            |
|---|-----------|-------------|---------------|----------|------------|
| 2 | TenDeThi  | Tên đề thi  | Nvarchar(max) | Null     |            |
| 3 | DeThiThu  | Đề thi thử  | Bit           | Null     |            |
| 4 | SanSang   | Sẵn sàng    | Bit           | Null     |            |
| 5 | IdHocPhan | Id học phần | Bigint        | Not null | Khóa ngoại |

Bảng 3.5 Mô tả chi tiết bảng KhoCauHoi

| STT | Tên thuộc tính     | Diễn giải       | Kiểu dữ liệu  | Ràng buộc | Mô tả      |
|-----|--------------------|-----------------|---------------|-----------|------------|
| 1   | <u>IdKhoCauHoi</u> | Id kho câu hỏi  | Bigint        | Not null  | Khóa chính |
| 2   | TenKhoCauHoi       | Tên kho câu hỏi | Nvarchar(max) | Null      |            |
| 3   | MoTa               | Mô tả           | Nvarchar(max) | Null      |            |
| 4   | IdHocPhan          | Id học phần     | Bigint        | Not null  | Khóa ngoại |

Bảng 3.6 Mô tả chi tiết bảng CauHoi

| STT | Tên thuộc tính  | Diễn giải      | Kiểu dữ liệu  | Ràng buộc | Mô tả      |
|-----|-----------------|----------------|---------------|-----------|------------|
| 1   | <u>IdCauHoi</u> | Id câu hỏi     | Int           | Not null  | Khóa chính |
| 2   | NoiDung         | Nội dung       | Nvarchar(max) | Null      |            |
| 3   | NhieuDapAn      | Bit            | Int           | Null      |            |
| 4   | DoKho           | Độ khó         | Nvarchar(max) | Null      |            |
| 5   | IdKhoCauHoi     | Id kho câu hỏi | Bigint        | Not null  | Khóa ngoại |

Bảng 3.7 Mô tả chi tiết bảng DapAn

| STT | Tên thuộc tính | Diễn giải   | Kiểu dữ liệu  | Ràng buộc | Mô tả      |
|-----|----------------|-------------|---------------|-----------|------------|
| 1   | <u>IdDapAn</u> | Id đáp án   | Bigint        | Not null  | Khóa chính |
| 2   | NoiDung        | Nội dung    | Nvarchar(max) | Null      |            |
| 3   | DapAnDung      | Đáp án đúng | Bit           | Null      |            |
| 4   | IdCauHoi       | Id câu hỏi  | Bigint        | Not null  | Khóa ngoại |

Bảng 3.8 Mô tả chi tiết bảng SinhVien

| STT | Tên thuộc tính    | Diễn giải       | Kiểu dữ liệu  | Ràng buộc | Mô tả      |
|-----|-------------------|-----------------|---------------|-----------|------------|
| 1   | <u>IdSinhVien</u> | Id sinh viên    | Bigint        | Not null  | Khóa chính |
| 2   | MSSV              | Mã số sinh viên | Nvarchar(max) | Null      |            |
| 3   | HoTen             | Họ tên          | Nvarchar(max) | Null      |            |
| 4   | GioiTinh          | Giới tính       | Nvarchar(max) | Null      |            |
| 5   | SoDienThoai       | Số điện thoại   | Nvarchar(max) | Null      |            |
| 6   | Email             | Email           | Nvarchar(max) | Null      |            |
| 7   | DiaChi            | Địa chỉ         | Nvarchar(max) | Null      |            |
| 8   | ID_LOP            | Mã lớp          | Bigint        | Not null  | Khóa ngoại |

Bảng 3.9 Mô tả chi tiết bảng CaThi

| STT | Tên thuộc tính | Diễn giải | Kiểu dữ liệu | Ràng | Mô tả |
|-----|----------------|-----------|--------------|------|-------|
|-----|----------------|-----------|--------------|------|-------|

|    |                |                   |               | buộc     |            |
|----|----------------|-------------------|---------------|----------|------------|
| 1  | <u>IdCaThi</u> | Mã lớp            | Bigint        | Not null | Khóa chính |
| 2  | TenCaThi       | Tên lớp           | Nvarchar(max) | Null     |            |
| 3  | ThoiGianBatDau | Thời gian bắt đầu | Bigint        | Not null |            |
| 4  | ThoiGianThi    | Thời gian thi     | Bigint        | Not null |            |
| 5  | TrangThai      | Trạng thái        | Nvarchar(max) | Null     |            |
| 6  | MaBaoVe        | Mã bảo vệ         | Nvarchar(max) | Null     |            |
| 7  | IdHocPhan      | Id học phần       | Bigint        | Not null | Khóa ngoại |
| 8  | IdLopHoc       | Id lớp học        | Bigint        | Not null | Khóa ngoại |
| 9  | IdGiamThi      | Id giám thị       | Bigint        | Not null | Khóa ngoại |
| 10 | IdDeThi        | Id đề thi         | Bigint        | Not null | Khóa ngoại |

Bảng 3.10 Mô tả chi tiết bảng ChiTietDeThi

| STT | Tên thuộc tính  | Diễn giải  | Kiểu dữ liệu | Ràng buộc | Mô tả                  |
|-----|-----------------|------------|--------------|-----------|------------------------|
| 1   | <u>IdDeThi</u>  | Id đề thi  | Bigint       | Not null  | Khóa chính, khóa ngoại |
| 2   | <u>IdCauHoi</u> | Id câu hỏi | Bigint       | Not null  | Khóa chính, khóa ngoại |

Bảng 3.11 Mô tả chi tiết bảng BaiLam

| STT | Tên thuộc tính | Diễn giải | Kiểu dữ liệu | Ràng buộc | Mô tả |
|-----|----------------|-----------|--------------|-----------|-------|
|-----|----------------|-----------|--------------|-----------|-------|



|   |                   |              |        |          |                           |
|---|-------------------|--------------|--------|----------|---------------------------|
| 1 | <u>IdCaThi</u>    | Id ca thi    | Bigint | Not null | Khóa chính,<br>Khóa ngoại |
| 2 | <u>IdSinhVien</u> | Id sinh viên | Bigint | Not null | Khóa chính,<br>Khóa ngoại |
| 3 | <u>IdCauHoi</u>   | Id câu hỏi   | Bigint | Not null | Khóa chính,<br>Khóa ngoại |
| 4 | <u>IdDapAn</u>    | Id đáp án    | Bigint | Not null | Khóa chính,<br>Khóa ngoại |

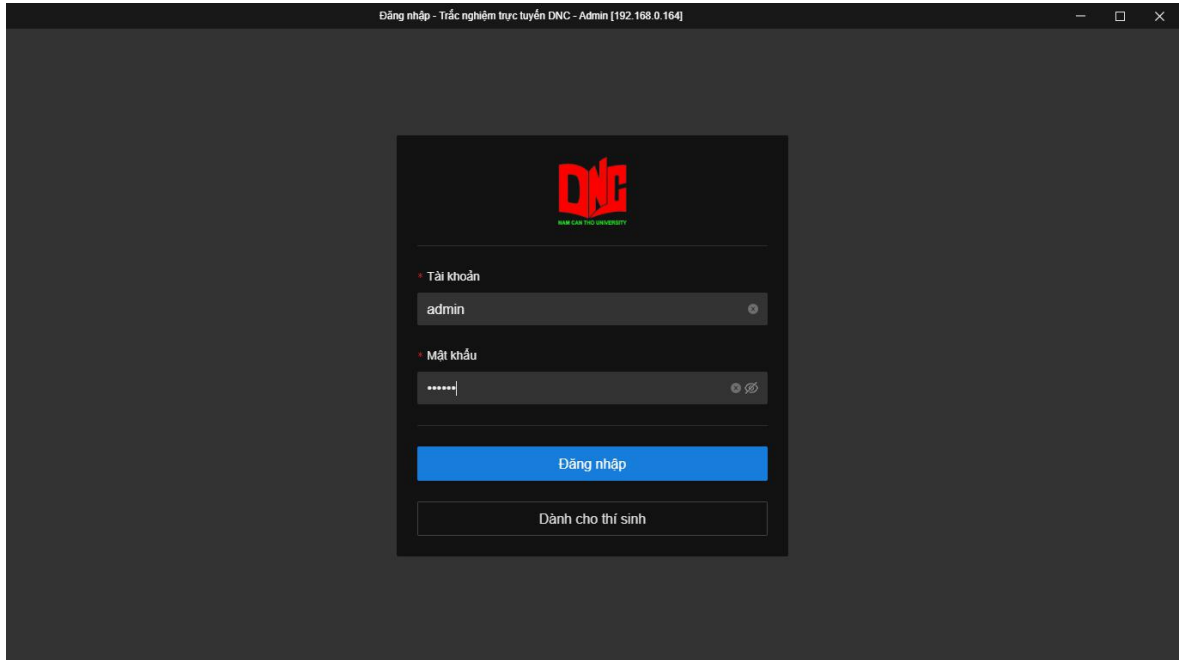
Bảng 3.12 Mô tả chi tiết bảng ThiSinh

| STT | Tên thuộc tính    | Diễn giải        | Kiểu dữ liệu  | Ràng buộc | Mô tả                     |
|-----|-------------------|------------------|---------------|-----------|---------------------------|
| 1   | <u>IdSinhVien</u> | Id sinh viên     | Bigint        | Not null  | Khóa chính,<br>Khóa ngoại |
| 2   | <u>IdCaThi</u>    | Id ca thi        | Bigint        | Not null  | Khóa chính,<br>Khóa ngoại |
| 3   | SoLanDangNhap     | Số lần đăng nhập | Int           | Null      |                           |
| 4   | TenMay            | Tên máy          | Nvarchar(max) | Null      |                           |
| 5   | DiaChiIp          | Địa chỉ ip       | Nvarchar(max) | Null      |                           |

### 3.3 Thiết kế giao diện

#### 3.3.1 Giao diện dành cho người quản trị

##### 3.3.1.1 Giao diện đăng nhập

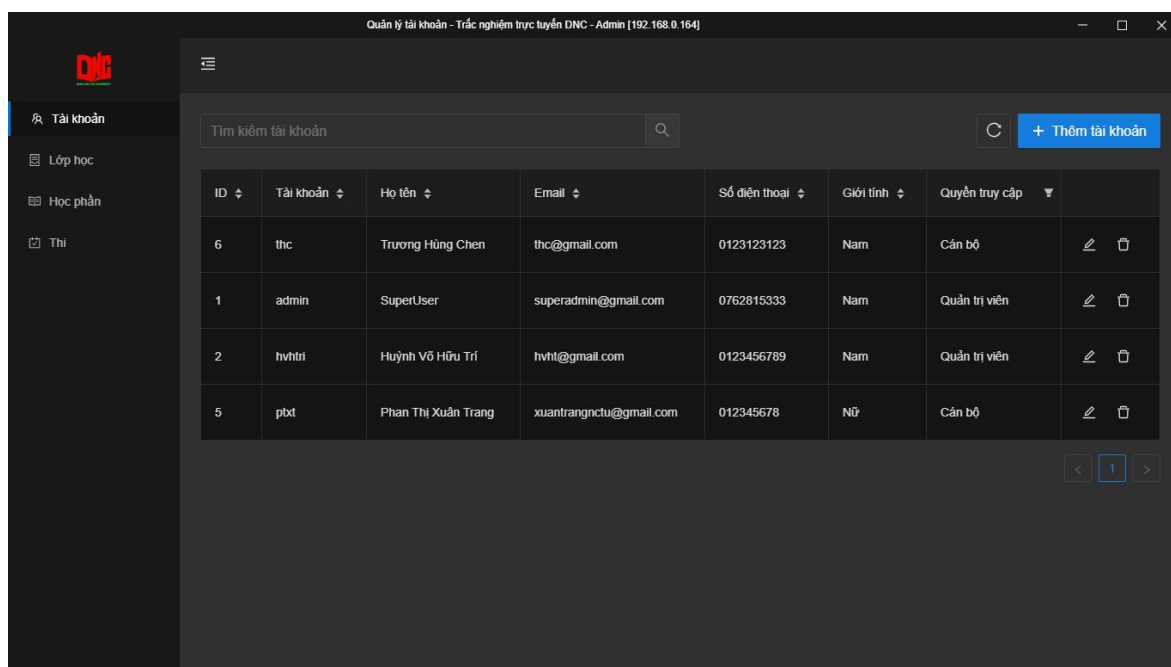


Hình 3.9 Giao diện đăng nhập

##### 3.3.1.2 Giao diện quản lý tài khoản

Các chức năng:

- Thêm tài khoản.
- Sửa và xóa tài khoản.
- Tìm kiếm tài khoản.
- Sắp xếp và lọc dữ liệu.

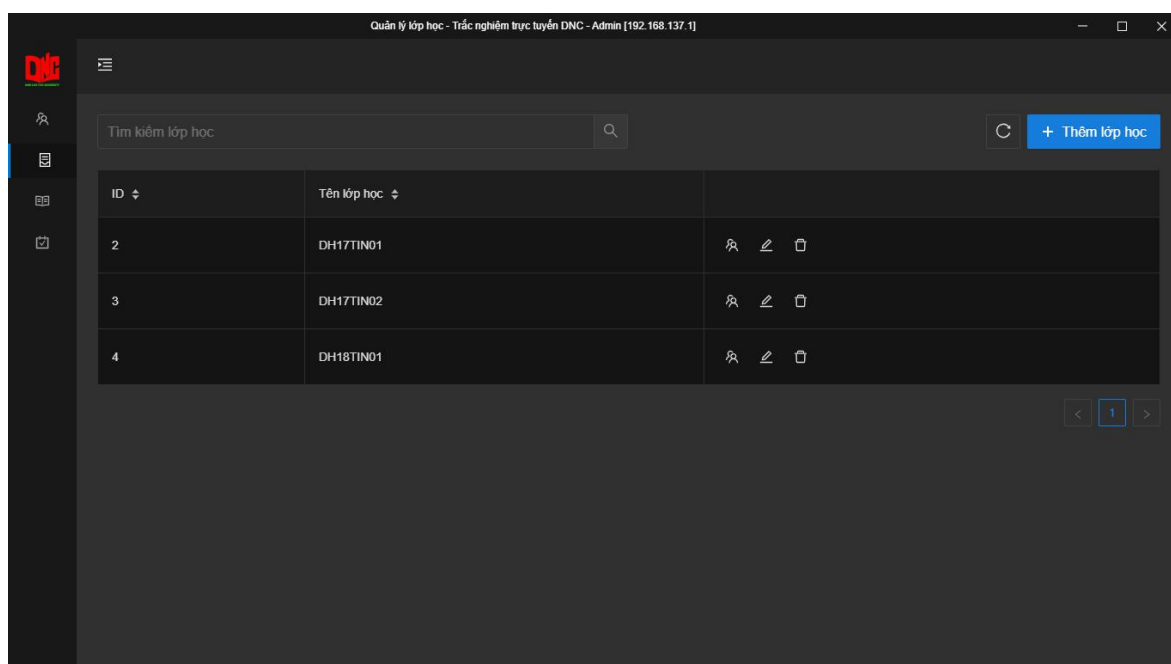


Hình 3.10 Giao diện quản lý tài khoản

### 3.3.1.3 Giao diện quản lý lớp học

Các chức năng:

- Quản lý sinh viên từng lớp
- Thêm lớp học
- Sửa và xóa lớp học
- Tìm kiếm và lọc dữ liệu

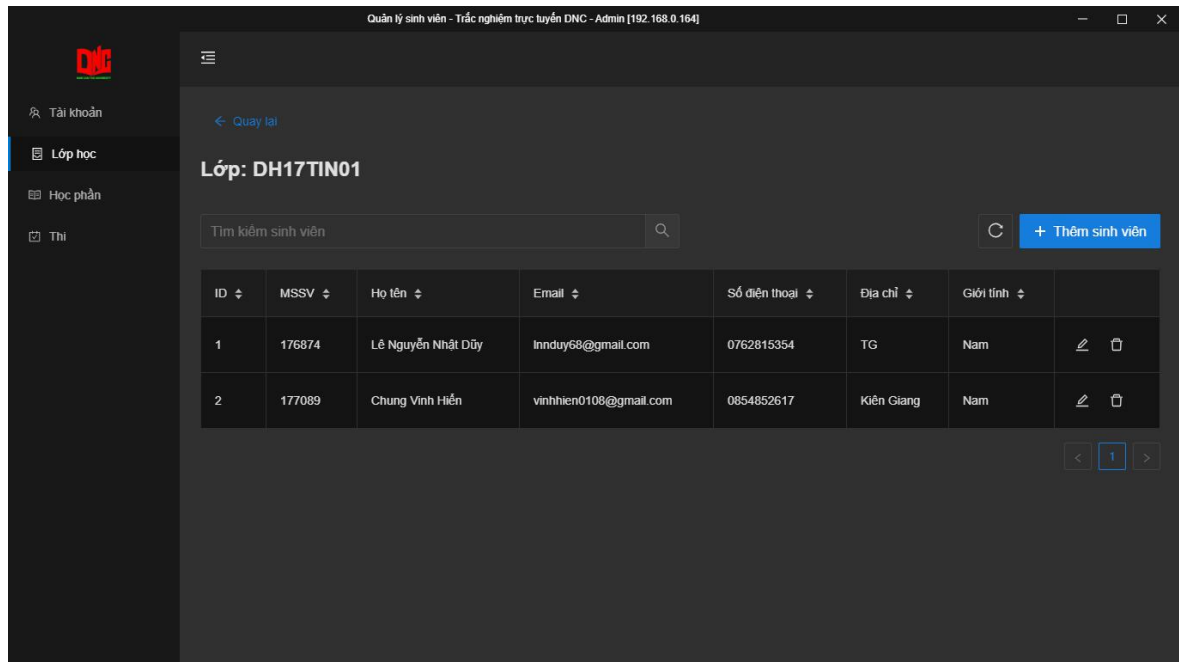


Hình 3.11 Giao diện quản lý lớp học

### \* Quản lý học viên của lớp học

Các chức năng:

- Thêm sinh viên
- Sửa và xóa sinh viên
- Tìm kiếm và lọc dữ liệu

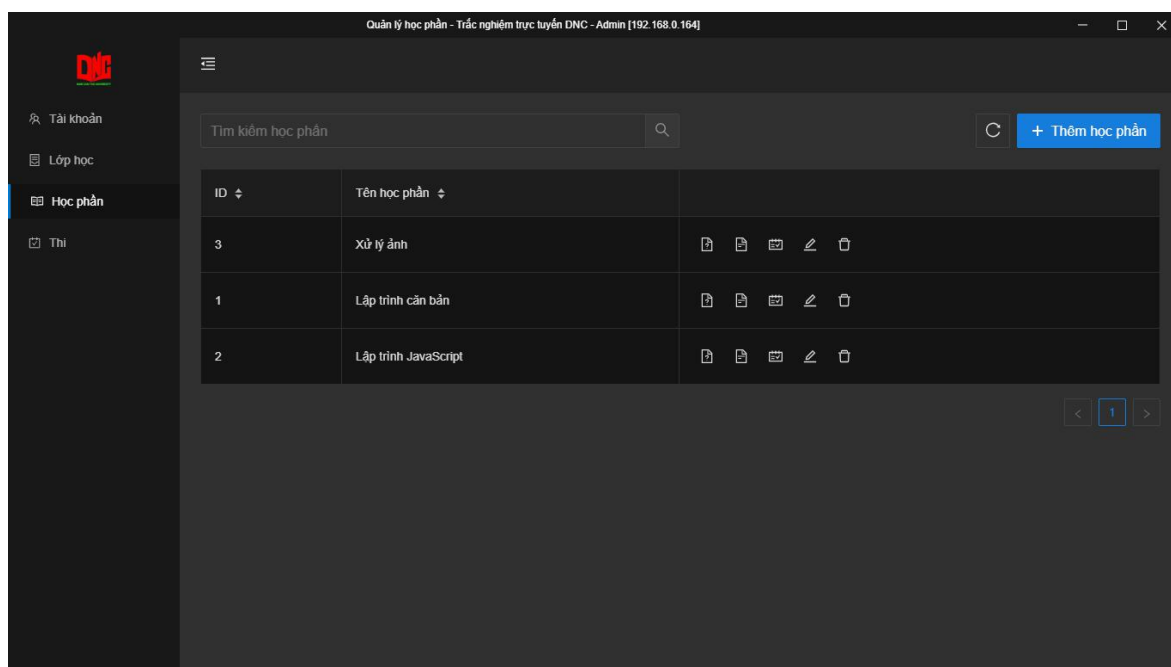


Hình 3.12 Giao diện quản lý học viên của lớp học

### 3.3.1.4 Giao diện quản lý học phần

Các chức năng:

- Quản lý kho câu hỏi
- Quản lý đề thi
- Quản lý ca thi
- Thêm học phần
- Sửa và xóa học phần
- Tìm kiếm và lọc dữ liệu

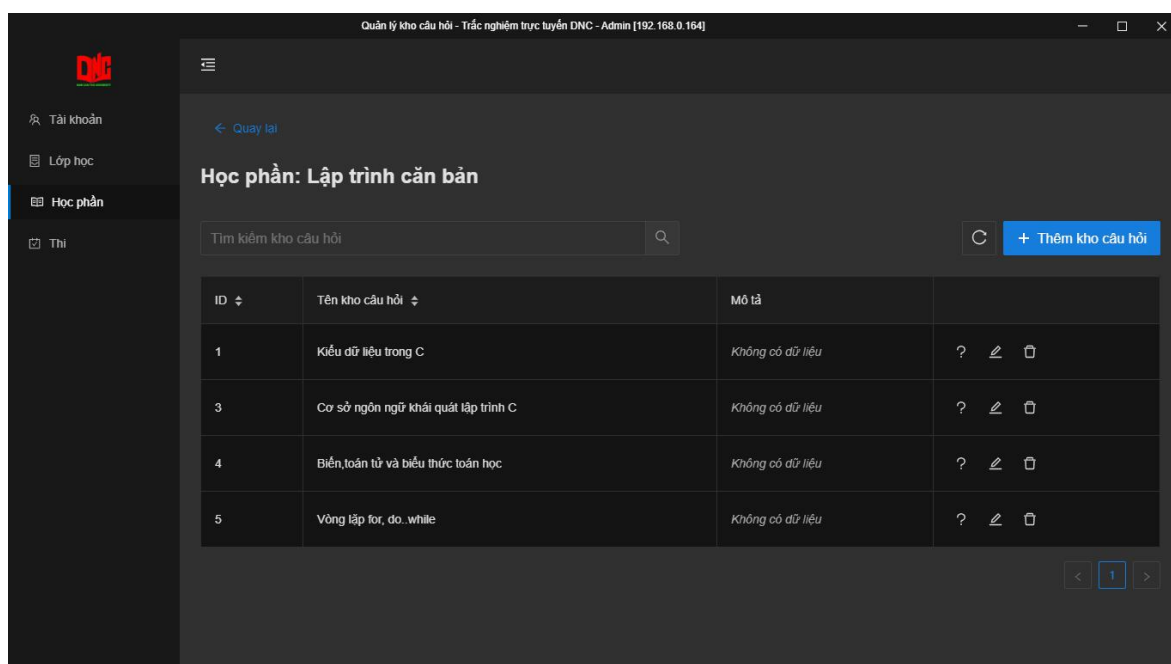


Hình 3.13 Giao diện quản lý học phần

a. Giao diện quản lý kho câu hỏi của học phần

Các chức năng:

- Quản lý câu hỏi
- Thêm câu hỏi
- Sửa và xóa kho câu hỏi
- Tìm kiếm và lọc dữ liệu

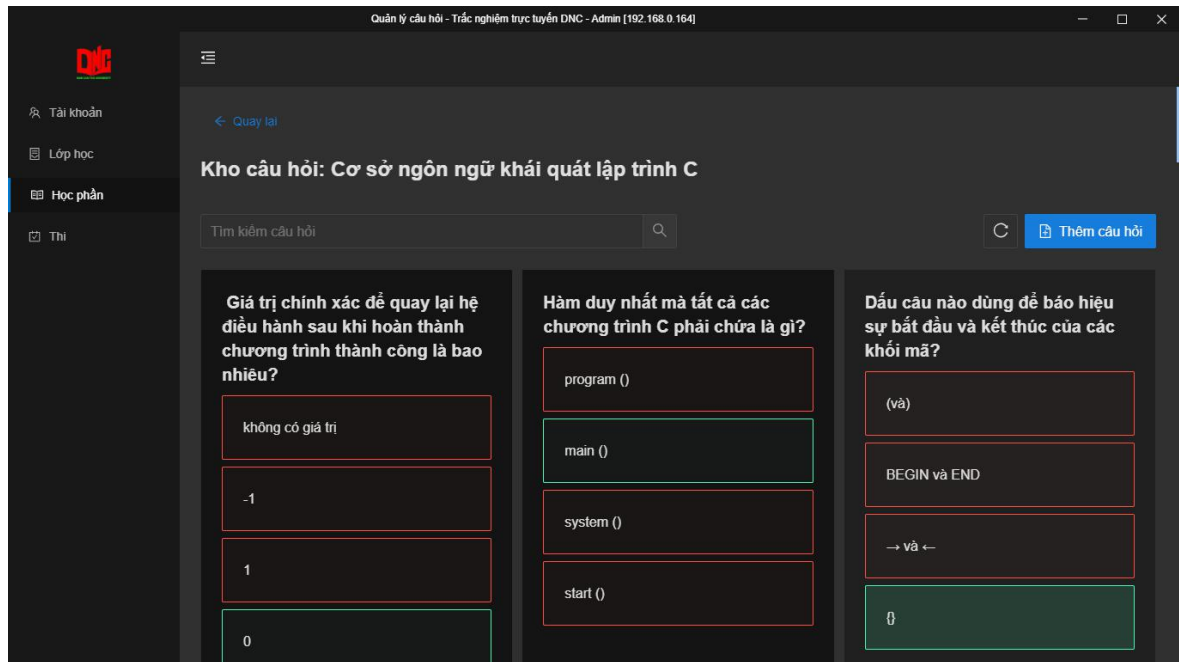


Hình 3.14 Giao diện quản lý kho câu hỏi của học phần

### \* Giao diện quản lý câu hỏi của kho câu hỏi

#### Các chức năng

- Thêm câu hỏi
- Sửa và xóa câu hỏi
- Tìm kiếm và lọc dữ liệu

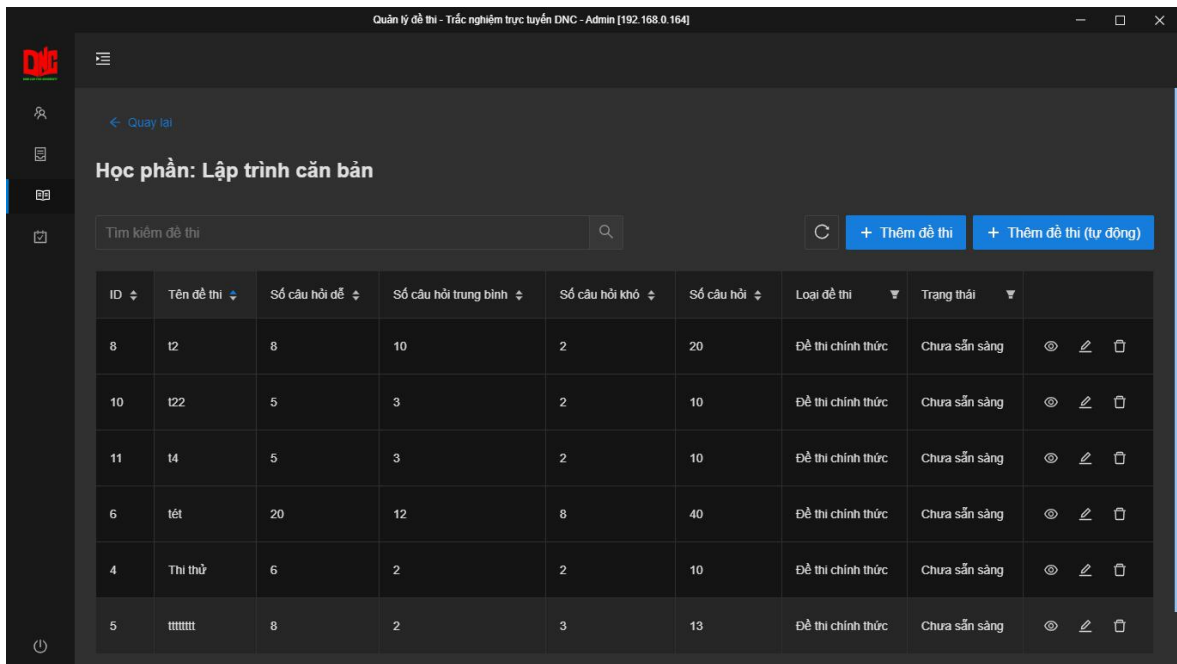


Hình 3.15 Giao diện quản lý câu hỏi của kho câu hỏi

#### b. Giao diện quản lý đề thi của học phần

#### Các chức năng:

- Xem đề thi
- Thêm đề thi
- Sửa và xóa đề thi
- Tìm kiếm và lọc dữ liệu

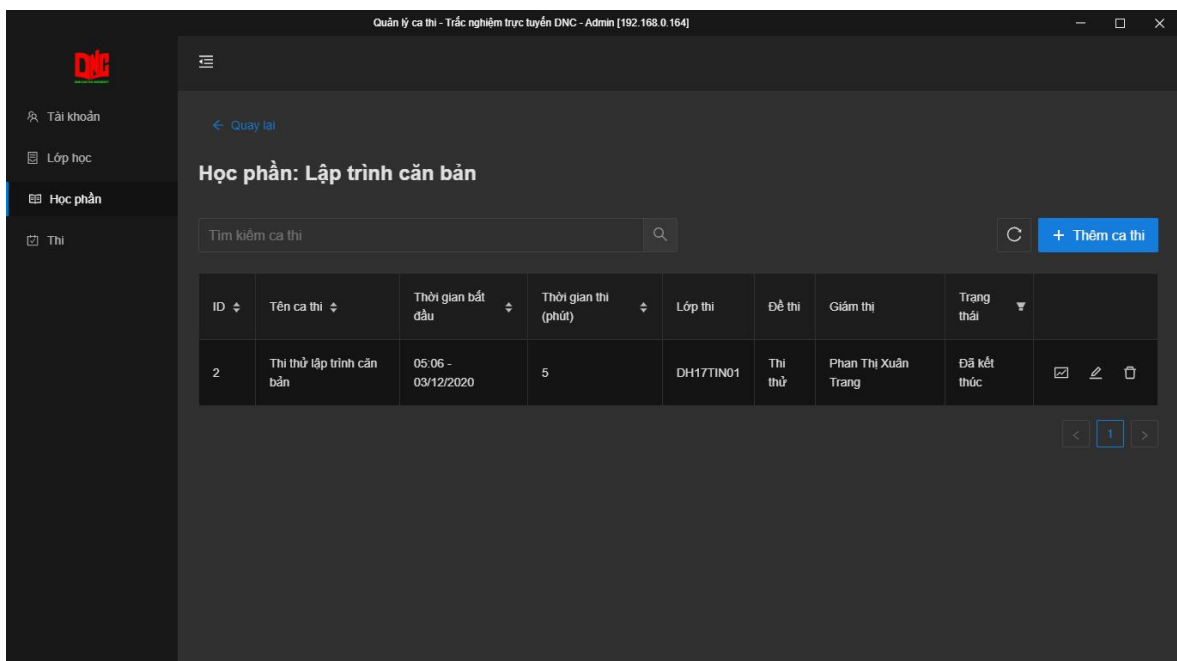


Hình 3.16 Giao diện quản lý đề thi của học phần

### c. Giao diện quản lý ca thi của học phần

Các chức năng:

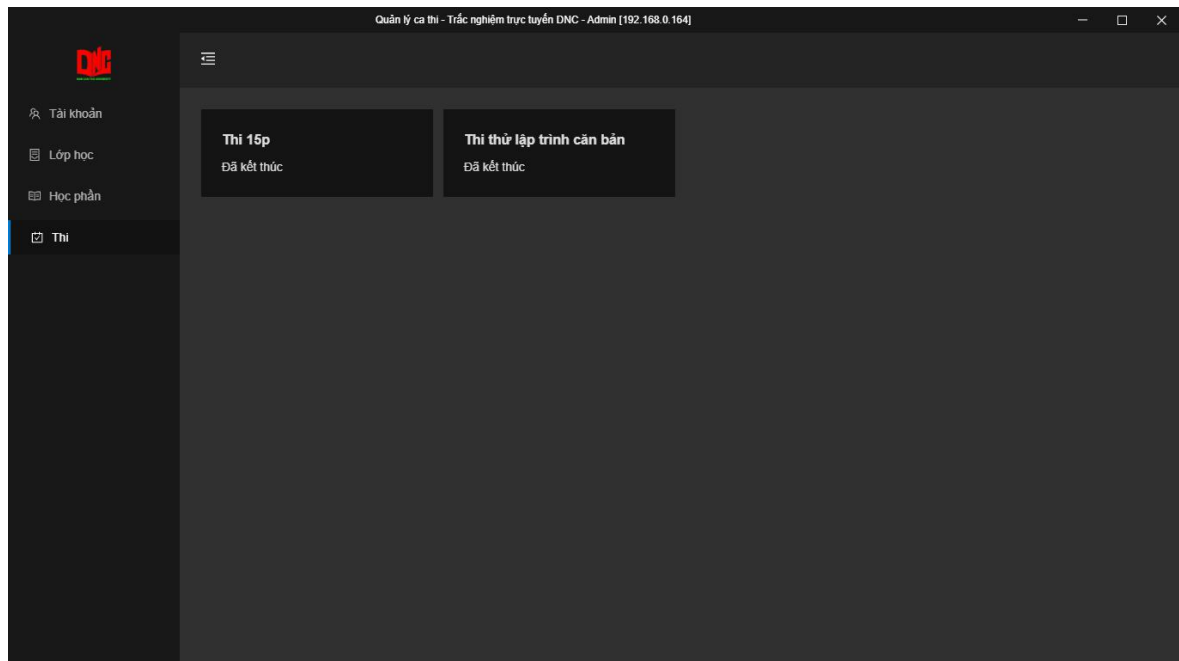
- Xem kết quả (Khi quá trình khi kết thúc)
- Thêm ca thi
- Sửa và xóa ca thi
- Tìm kiếm và lọc dữ liệu



Hình 3.17 Giao diện quản lý ca thi của học phần

### 3.3.1.5 Giao diện quản lý thi

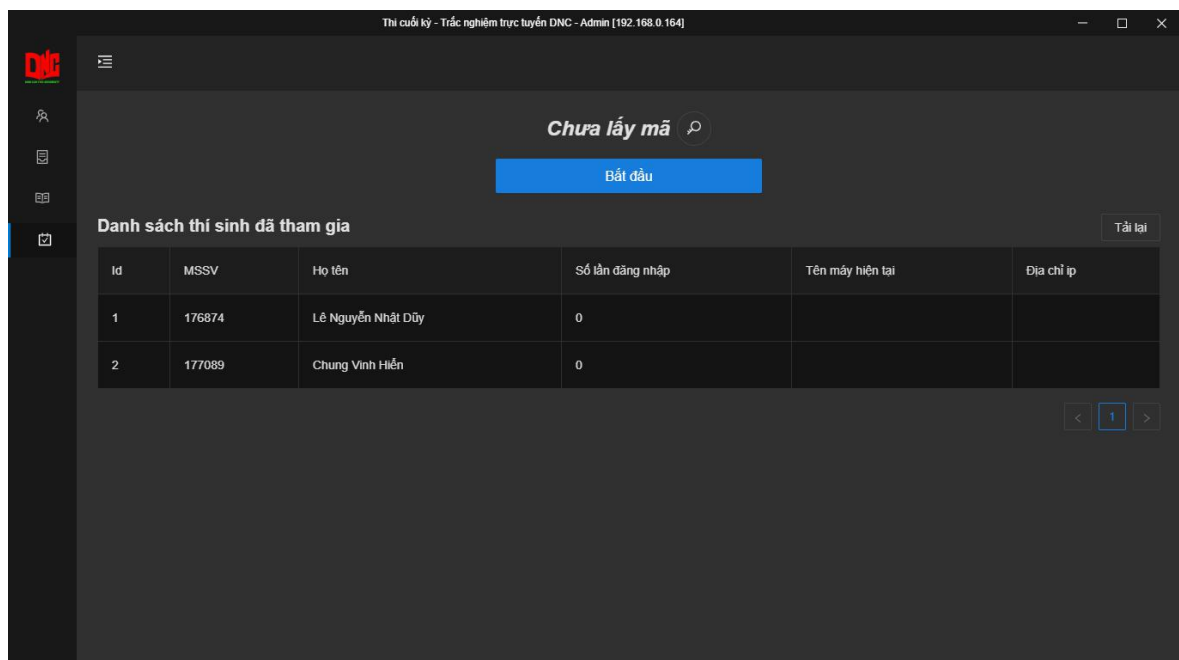
Gồm các kỳ thi



Hình 3.18 Giao diện quản lý thi

\* Giao diện danh sách sinh viên tham gia thi của kỳ thi

Bắt đầu thi giảng viên cung cấp mã cho các sinh viên dự thi

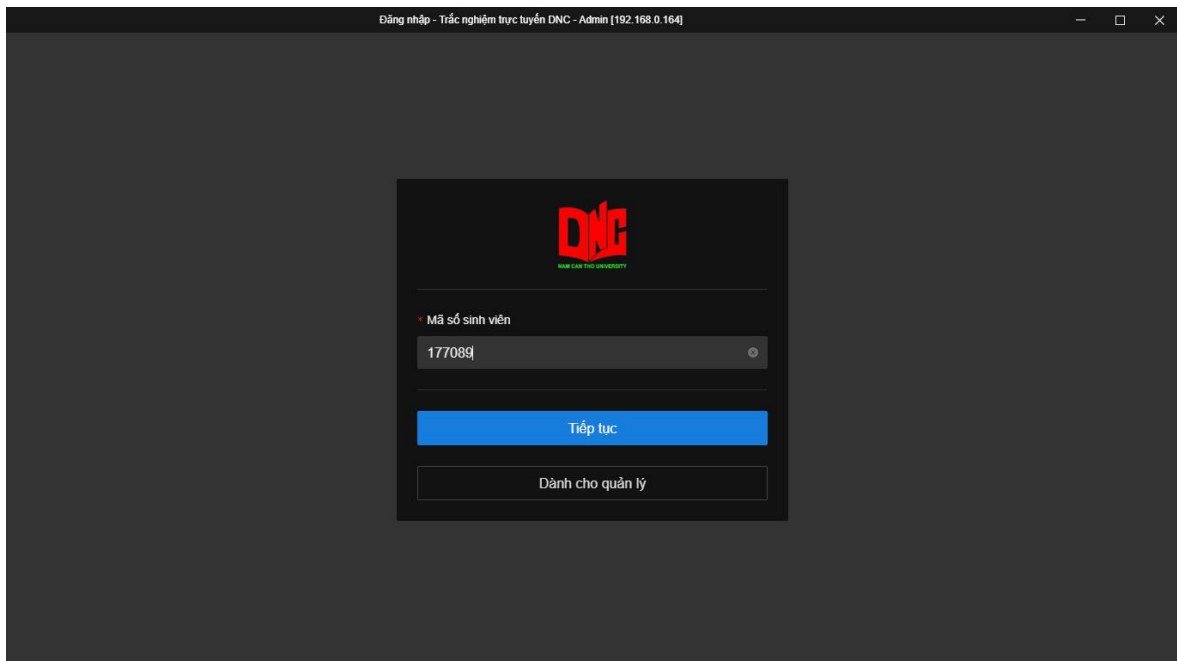


Hình 3.19 Giao diện danh sách sinh viên tham gia thi của kỳ thi



### 3.3.2 Giao diện người dự thi (Sinh viên)

#### 3.3.2.1 Giao diện đăng nhập

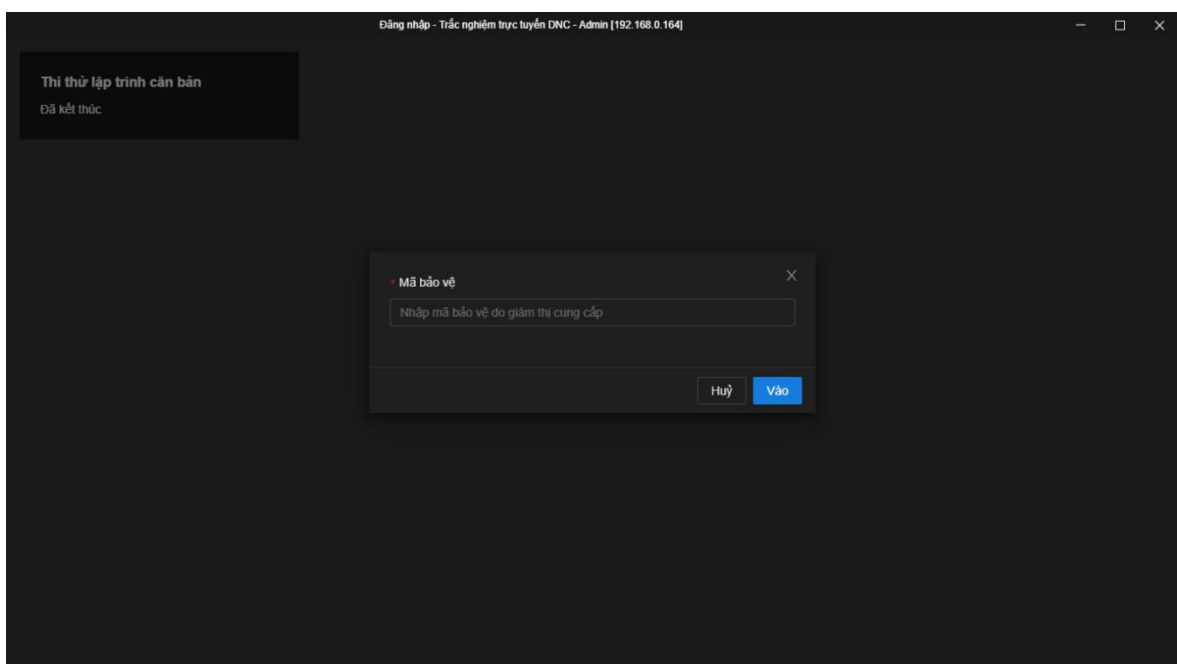


Hình 3.20 Giao diện đăng nhập

#### 3.3.2.2 Giao diện chọn kỳ thi

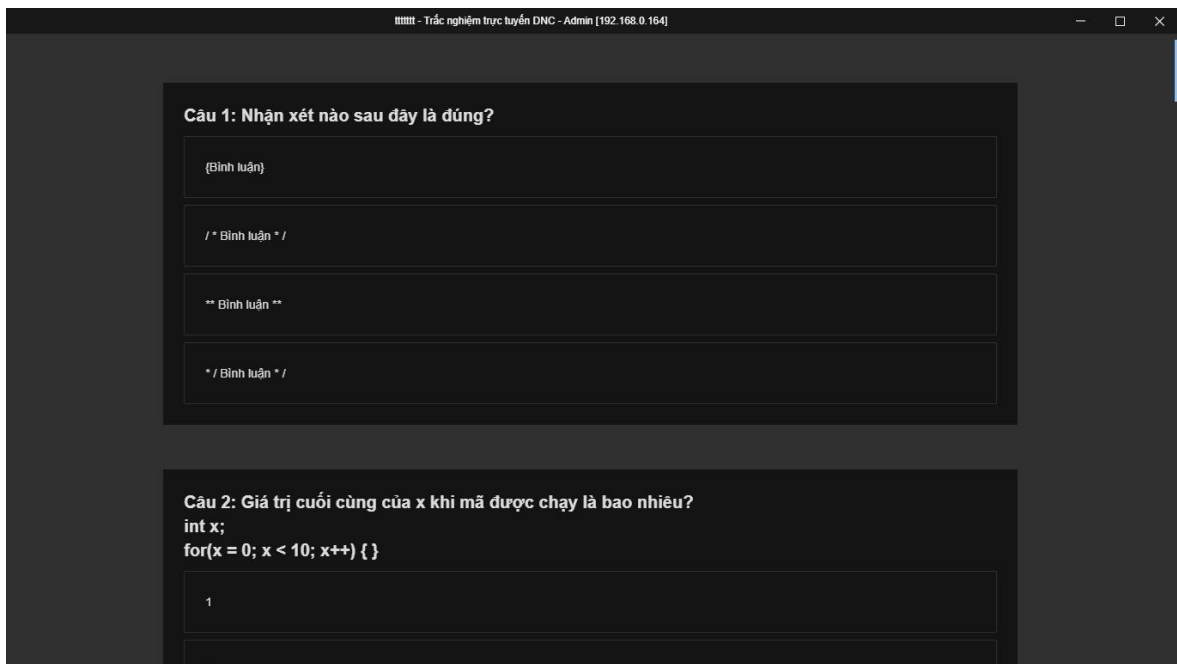
Sinh viên lấy mã từ giảng viên cung cấp để bắt đầu thi

Sau khi kết thúc kết quả trả về cho giảng viên trong quản lý ca thi



Hình 3.21 Giao diện chọn kỳ thi

### 3.3.2.3 Giao diện thi của sinh viên



The screenshot shows a web application window titled "Trắc nghiệm trực tuyến DNC - Admin [192.168.0.164]". The interface is dark-themed. It contains two question blocks. The first block, "Câu 1: Nhận xét nào sau đây là đúng?", has four text input fields with placeholder text: "(Bình luận)", "/\* Bình luận \*/", "\*\* Bình luận \*\*", and "\*/ Bình luận \*/". The second block, "Câu 2: Giá trị cuối cùng của x khi mã được chạy là bao nhiêu?", contains a code snippet: 

```
int x;  
for(x = 0; x < 10; x++) { }
```

 followed by a text input field containing the number "1".

Hình 3.22 Giao diện thi cho sinh viên

## **CHƯƠNG 4: KẾT LUẬN**

Đề tài đã bước đầu thu được những thành quả khả quan về mục tiêu tin học hóa của Khoa Công nghệ và kỹ thuật Trường Đại học Nam Cần Thơ. Những chức năng được thiết kế trực quan và hợp lý giúp cho người dùng không bị khó khăn trong quá trình thi cũng như quản lý so với quy trình thủ công.

Ngoài ra, đề tài cũng thật sự hữu ích và có tính thực tế cao đối với chuyên ngành công nghệ thông tin. Trong tương lai, nếu điều kiện khách quan cho phép, chúng em sẽ tiếp tục phát triển để triển khai hệ thống vào thực tế, giúp cho nhà trường tiết kiệm chi phí, và giúp cho giáo viên, sinh viên dễ dàng trong việc quản lý và thi trắc nghiệm. Hệ thống có thể được phát triển và mở rộng để tích hợp với hệ thống quản lý điểm toàn trường, toàn khóa cho sinh viên.

## TÀI LIỆU THAM KHẢO

- [1]. Ths. Trần Văn Nhuộm (2019), *Giáo trình Phân tích và thiết kế hệ thống thông tin*. Khoa Kỹ thuật - Công nghệ, Trường Đại học Nam Cần Thơ.
- [2]. Ths. Trần Xuân Trang (2019), *Giáo trình Hệ quản trị cơ sở dữ liệu*. Khoa Kỹ thuật - Công nghệ, Trường Đại học Nam Cần Thơ.
- [3]. Vũ Hoàng Mạnh Hải và Nguyễn Đình Ngọc Duy (2005), *Xây dựng phần mềm quản lý ngân hàng câu hỏi trắc nghiệm tuyển sinh đại học*. Khoa công nghệ thông tin, Đại học khoa học tự nhiên.
- [4]. Hà Trọng Nhân và Hà Nhật Tâm (2005), *Xây dựng hệ thống hỗ trợ thi trắc nghiệm*. Khoa công nghệ thông tin, Đại học khoa học tự nhiên.
- [5]. Lê Thành Nam, Ngô Thị Thùy Hương và Nguyễn Viết Chinh (2016), *Xây Dựng Phần Mềm Quản Lý Thi Trắc Nghiệm Trên Hệ Thống Mạng Lan Ứng Dụng Tại Trung Tâm Ngoại Ngữ - Tin Học & Kỹ Năng Nghề Nghiệp Trường CĐSP Hòa Bình*. Nghiên cứu khoa học, Trường Cao đẳng Sư phạm Hòa Bình.
- [6]. Quách Chí Cường (2020). *Visual Studio Code là gì? Tính năng nổi bật của Visual Studio Code*. [truy cập ngày 8/12/2020]. <https://cuongquach.com/visual-studio-code-la-gi.html>.
- [7]. Trung Nguyen (2020), *Giới thiệu Entity Framework*. [truy cập ngày 8/12/2020]. <https://comdy.vn/entity-framework/gioi-thieu-entity-framework/>.
- [8]. Huỳnh Hoàng Anh (2018), *Sử dụng Electron để phát triển ứng dụng máy tính để bàn đa nền tảng*. [truy cập ngày 8/12/2020]. <https://helpex.vn/article/su-dung-electron-de-phat-trien-ung-dung-may-tinh-de-ban-da-nen-tang-5c54d031507419248c9af821>.
- [9]. Phan Khánh Đoàn (2018), *Tổng quan về React.js*. [truy cập ngày 8/12/2020]. <https://helpex.vn/article/tong-quan-ve-react-js-5c54d1ca507419248c9af954>.
- [10]. HVIT CLAN (2020), *Bài 2 - Tổng quan về ReactJS*. [truy cập ngày 8/12/2020]. <https://hvitclan.vn/blog/bai-2-tong-quan-ve-reactjs-234>.