|  |
| --- |
| **BỘ GIÁO DỤC VÀ ĐÀO TẠO**  NGUYỄN VĂN ĐỨC KHẢO SÁT ĐÁNH GIÁ PHỤ HUYNH HAPPY MONTESSORI 20201  **TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT HƯNG YÊN**  **NGUYỄN VĂN ĐỨC**  **XÂY DỰNG WEBSITE KHẢO SÁT ĐÁNH GIÁ Ý KIẾN PHỤ HUYNH TRƯỜNG HAPPY MONTESSORI**  **ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC**  **HƯNG YÊN - 2021** |

|  |
| --- |
| **BỘ GIÁO DỤC VÀ ĐÀO TẠO**  **TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT HƯNG YÊN**  **NGUYỄN VĂN ĐỨC**  **XÂY DỰNG WEBSITE KHẢO SÁT ĐÁNH GIÁ Ý KIẾN PHỤ HUYNH TRƯỜNG HAPPY MONTESSORI**  NGÀNH: CÔNG NGHỆ THÔNG TIN  CHUYÊN NGÀNH: KỸ THUẬT PHẦN MỀM  **ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC**  **NGƯỜI HƯỚNG DẪN**  **TS. NGUYỄN VĂN QUYẾT**  **HƯNG YÊN - 2021** |

**NHẬN XÉT**

**Nhận xét của giảng viên hướng dẫn:**

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

**GIẢNG VIÊN HƯỚNG DẪN**

*(Ký và ghi rõ họ tên)*

**LỜI CAM ĐOAN**

Em xin cam đoan đồ án tốt nghiệp “Website khảo sát đánh giá ý kiến phụ huynh Trường Happy Montesspri” là công trình nghiên cứu của bản thân. Những phần sử dụng tài liệu tham khảo trong đồ án đã được nêu rõ trong phần tài liệu tham khảo. Các số liệu, kết quả trình bày trong đồ án là hoàn toàn trung thực, nếu sai em xin chịu hoàn toàn trách nhiệm và chịu mọi kỷ luật của bộ môn và nhà trường đề ra.

*Hưng Yên, ngày … tháng … năm…..*

Sinh viên

Nguyễn Văn Đức

MỤC LỤC

[MỤC LỤC 5](#_Toc71502439)

[DANH SÁCH CÁC THUẬT NGỮ 7](#_Toc71502440)

[DANH SÁCH BẢNG BIỂU 8](#_Toc71502441)

[DANH SÁCH HÌNH VẼ 9](#_Toc71502442)

[CHƯƠNG 1: MỞ ĐẦU 10](#_Toc71502443)

[1.1 Lý do chọn đồ án 10](#_Toc71502444)

[1.2 Mục tiêu của đồ án 10](#_Toc71502445)

[1.2.1 Mục tiêu tổng quát 10](#_Toc71502446)

[1.2.2 Mục tiêu cụ thể 10](#_Toc71502447)

[1.3 Giới hạn và phạm vi của đồ án 11](#_Toc71502448)

[1.3.1 Đối tượng nghiên cứu 11](#_Toc71502449)

[1.3.2 Phạm vi nghiên cứu 11](#_Toc71502450)

[1.4 Nội dung thực hiện 11](#_Toc71502451)

[1.5 Phương pháp tiếp cận 12](#_Toc71502452)

[CHƯƠNG 2: CƠ SỞ LÝ THUYẾT 13](#_Toc71502453)

[**2.1 Phương pháp phát triển phần mềm hướng đối tượng** 13](#_Toc71502454)

[CHƯƠNG 3: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG 17](#_Toc71502455)

[3.1 Đặc tả yêu cầu phần mềm 17](#_Toc71502456)

[3.1.1 Các yêu cầu chức năng 17](#_Toc71502457)

[3.1.2 Biểu đồ lớp thực thể 19](#_Toc71502458)

[3.1.3 Các yêu cầu phi chức năng 23](#_Toc71502459)

[3.2 Thiết kế hệ thống 24](#_Toc71502460)

[3.2.1 Thiết kế kiến trúc 24](#_Toc71502461)

[3.2.2 Thiết kê cơ sở dữ liệu 26](#_Toc71502462)

[3.2.3 Thiết kế lớp đối tượng 31](#_Toc71502463)

[3.2.4 Thiết kế giao diện 34](#_Toc71502464)

[CHƯƠNG 4: TRIỂN KHAI WEBSITE 35](#_Toc71502465)

[4.1 Xây dựng Web API 35](#_Toc71502466)

[4.2 Xây dựng các chức năng 35](#_Toc71502467)

[4.2.1 Các chức năng nghiệp vụ phân hệ người dùng 35](#_Toc71502468)

[4.2.2 Các chức hệ thống 35](#_Toc71502469)

[4.3 Kiểm thử và triển khai ứng dụng 35](#_Toc71502470)

[KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 36](#_Toc71502471)

[Kết quả đạt được 36](#_Toc71502472)

[Hạn chế của đề tài 36](#_Toc71502473)

[Hướng phát triển của đề tài 36](#_Toc71502474)

[TÀI LIỆU THAM KHẢO 37](#_Toc71502475)

[PHỤ LỤC 38](#_Toc71502476)

DANH SÁCH CÁC THUẬT NGỮ

|  |  |  |
| --- | --- | --- |
| Từ viết tắt | Từ đầy đủ | Giải thích |
| KSDG | Khảo sát đánh giá |  |
| API | **Application Programming Interface** |  |
|  |  |  |
|  |  |  |

DANH SÁCH BẢNG BIỂU

[Bảng 3.1: Mô tả thuộc tính đối tượng Question 20](#_Toc71502852)

[Bảng 3.2: Mô tả thuộc tính đối tượng QuestionType 20](#_Toc71502853)

[Bảng 3.3: Mô tả thuộc tính đối tượng Children 21](#_Toc71502854)

[Bảng 3.4: Mô tả thuộc tính đối tượng ClassStudent 21](#_Toc71502855)

[Bảng 3.5: Mô tả thuộc tính đối tượng Evalution 22](#_Toc71502856)

[Bảng 3.6: Mô tả thuộc tính đối tượng Parents 22](#_Toc71502857)

[Bảng 3.7: Mô tả thuộc tính đối tượng Account 23](#_Toc71502858)

[Bảng 3.8: Mô tả cấu trúc bảng Children 27](#_Toc71502859)

[Bảng 3.9: Mô tả cấu trúc bảng ClassStudent 28](#_Toc71502860)

[Bảng 3.10: Mô tả cấu trúc bảng Evalution 28](#_Toc71502861)

[Bảng 3.11: Mô tả cấu trúc bảng Parents 29](#_Toc71502862)

[Bảng 3.12: Mô tả cấu trúc bảng Question 30](#_Toc71502863)

[Bảng 3.13: Mô tả cấu trúc bảng QuestionType 31](#_Toc71502864)

DANH SÁCH HÌNH VẼ

[Hình 3.1: UseCase Tổng quát 17](#_Toc71502920)

[Hình 3.2: Mô tả chức năng thêm mới câu hỏi 18](#_Toc71502921)

[Hình 3.3: Biểu đồ lớp VOPC thêm câu hỏi 31](#_Toc71502922)

[Hình 3.4: Biểu đồ lớp VOPC Cập nhật câu hỏi 32](#_Toc71502923)

[Hình 3.5: Biểu đồ lớp VOPC Xóa câu hỏi 32](#_Toc71502924)

[Hình 3.6: Biểu đồ lớp VOPC Đăng nhập 33](#_Toc71502925)

[Hình 3.7: Biểu đồ tuần tự Đăng ký 33](#_Toc71502926)

[Hình 3.8: Biểu đồ thêm câu hỏi 34](#_Toc71502927)

[Hình 3.9: Biểu đồ lớp chi tiết Quản lý câu hỏi 34](#_Toc71502928)

# MỞ ĐẦU

1.1 Lý do chọn đồ án

Ngày này, các trường Mầm non vẫn chưa cải tiến được về việc thu thập các ý kiến của phụ huynh về vấn đề con em học ở trường. Nhà trường vẫn phải theo phương pháp cũ là hỏi han phụ huynh qua truyền miệng về việc đóng góp ý kiến đánh giá. Thời đại nay 4.0 chúng ta luôn cải tiến theo công nghệ, để giảm bớt sự bất lợi đó, em đã lên ý tưởng về việc xây dựng một website “Đánh giá ý kiến phụ huynh”. Ứng dụng website nhằm giảm bớt sự bất lợi từ nhà trường và phụ huynh hơn, nhà trường có thể quản lý các vấn đề qua hệ thống quản lý, phụ huynh có thể đánh giá trên website online của nhà trường.

1.2 Mục tiêu của đồ án

1.2.1 Mục tiêu tổng quát

Website “Đánh giá khảo sát ý kiến phụ huynh Trường Happy Montessori” nhằm giúp nhà trường và các phụ huynh có một trải nghiệm tốt nhất về phần mềm. Giúp nhà trường có thể quản lý các vấn đề từ phụ huynh và con em một cách hiệu quả nhất mà không tốn nhiều thời gian. Phần mềm còn mang lại các cải tiến mới cho ngành Giáo Dục của Việt Nam ta, một trải nghiệm tốt nhất đến với tay người tiêu dùng.

1.2.2 Mục tiêu cụ thể

Website “Đánh giá khảo sát ý kiến phụ huynh Trường Happy Montessori” bao gồm các mục tiêu chính như sau:

Nhà Trường:

* Quản lý nội dung câu hỏi đánh giá.
* Quản lý các phiếu đánh giá.
* Thống kê các chỉ tiêu, số lần đánh giá,…
* Quản lý tài khoản người truy cập.

Phụ huynh:

* Đánh giá trên website của nhà trường.

1.3 Giới hạn và phạm vi của đồ án

1.3.1 Đối tượng nghiên cứu

Đối tượng nghiên cứu: Trường mầm non Happy Montessori và Phụ huynh học sinh

*Khách thể nghiên cứu:*

1.3.2 Phạm vi nghiên cứu

*Phạm vi không gian*: Tại Trường Mầm Non Happy Montessori*.*

*Phạm vi thời gian:* Tháng 4/2020, các dữ liệu về câu hỏi của nhà trường và nguồn khách hàng do Trường mầm non Happy Montessori cung cấp.

*Ý nghĩa khoa học và thực tiễn của đề tài:* Giúp ngành giáo dục cải tiến về mặt công nghệ số một cách tiện ích nhất.

1.4 Nội dung thực hiện

Website “Đánh giá khảo sát ý kiến phụ huynh Trường Mầm non Happy Montessori” chia làm 2 phía thực hiện:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Nội dung thực hiện | Mô tả | | Ý nghĩa |
| **Nhà Trường** | | | |  |
| 1 | Quản lý nội dung câu hỏi | | | |
|  | Thêm nội câu hỏi | | Người dùng thêm các nội dung câu hỏi trên Form hiện thị của giao diện | Giúp người dùng có thể thêm nhanh các nội dung câu hỏi khi cần thiết |
|  | Sửa nội dung câu hỏi | | Người dùng sửa các nội dung câu hỏi khi chọn câu hỏi và hiển thị trên Form giao diện | Giúp người dùng khi sửa thông tin có thể thay đổi các thông tin cần thiết |
|  | Xóa nội dung câu hỏi | | Người dùng chọn Xóa bản ghi và hiển thị Popup thông báo | Người dùng không muốn nội dung câu hỏi đó tồn tại nữa có thể xóa |
|  | Refresh nội dung câu hỏi | | Người dùng chọn Refesh sẽ load lại grid của nội dung câu hỏi | Khi muốn cập nhật lại các câu hỏi khi muốn |
|  | Tìm kiếm nội dung câu hỏi | | Nhập các tên trên filter để tìm ra thông tin cần tìm | Giúp người dùng có thể tìm kiến nhanh các câu hỏi khi cần |
|  | Xem chi tiết nội dung câu hỏi | | Người dùng chọn xem chi tiết nội dùng | Giúp người dùng xem chi tiết một câu hỏi nào đó.` |
| 2 | Quản lý phiếu đánh giá | | | |
|  | Thêm phiếu đánh giá | | Người dùng thêm phiếu đánh giá trên Form hiện thị của giao diện | Giúp người dùng có thể thêm nhanh phiếu đánh giá khi có ai cần. |
|  | In phiếu đánh giá | | Người dùng in các phiếu đánh giá qua kết nối với máy in | Giúp người dùng in phiếu đánh giá khi cần thiết |
| 3 | Thống kê chỉ tiêu, số lần đánh giá | | | |

1.5 Phương pháp tiếp cận

* Nghiên cứu tài liệu.
* Nghiên cứu công nghệ phát triển website: VueJS, ASP.NET Core API, SQL Server
* Nghiên cứu các hệ thống website của các trường khác.
* Khảo sát các mô hình dự án thực tế.

# CƠ SỞ LÝ THUYẾT

**2.1 Phương pháp phát triển phần mềm hướng đối tượng**

Để phát triển được một phần mềm hướng đối tượng ta cần qua các bước:

* Xác định yêu cầu
* Yêu cầu phần mềm là tất cả các yêu cầu về phần mềm do khách hàng, người sử dụng phần mềm – nêu ra, bao gồm: các chứ năng của phần mềm, hiệu năng của phần mềm, các yêu cầu về thiết kế và giao diện, các yêu cầu đặc biệt khác
* Yêu cầu phần mềm được phân loại theo 4 thành phần của phần mềm:

- Các yêu cầu về phần mềm (Software).

- Các yêu cầu về phần cứng (Hardware).

- Các yêu cầu về dữ liệu (Data).

- Các yêu cầu về con người (People, Users).

* Yêu cầu chức năng

 − Lưu trữ

− Tra cứu

− Tính toán

− Kết xuất

* Yêu cầu phi chức năng
* Phân tích yêu cầu hướng đối tượng (OOA)
* Phân tích được xem như là cây cầu giữa kĩ nghệ hệ thống với thiết kế phần mềm.
* Mục đích của OOA:
* Định nghĩa các lớp (và các mối quan hệ hành vi liên kết với chúng) phù hợp với vấn đề cần giải quyết.
* Phát triển một tập các mô hình mô tả hệ thống phần mềm máy tính như nó hoạt động để thỏa mãn các yêu cầu của người dùng đã được xác định.
* Thiết kế hướng đối tượng

Với các hệ thống hướng đối tượng chúng ta có thể xác định ra 4 tầng thiết kế như sau

• Tầng hệ thống con: Chứa mô tả của mỗi hệ thống con cho phép hệ thống đạt được các yêu cầu của khách hàng và thực hiện các hạ tầng kĩ thuật hỗ trợ yêu cầu khách hàng.

• Tầng lớp và đối tượng: Chứa phân lớp cho phép hệ thống được tạo ra bằng  tổng quát hóa và tính tăng dần đặc tả đích

• Tầng thông điệp: Chứa các chi tiết cho phép các đối tượng giao tiếp.

 • Tầng trách nhiệm: Chứa các cấu trúc dữ liệu và các thuật toán thiết kế cho mọi thuộc tính và phương thức của lớp.

* Cài đặt

Cài đặt là trọng tâm của các bước lặp trong giai đoạn Xây dựng. Nó cũng được tiến hành trong giai đoạn Chi tiết để tạo ra khung kiến trúc khả thi. Cài đặt cũng được tiến hành trong giai đoạn chuyển giao để xử lí các lỗi được tìm thấy khi phát hành phiên bản beta.

* Kiểm thử

Sau khi một phần mềm được hoàn tất. Công việc cuối cùng là kiểm thử nhằm phát hiện lỗi bằng việc xem xét lại đặc tả, thiết kế và mã hoá.

• Là khâu mấu chốt đảm bảo chất lượng phần mềm.

• Nguyên tắc kiểm thử:

− Mọi hoạt động kiểm thử phải tuân theo yêu cầu của khách hàng.

− Kiểm thử phải được lập kế hoạch trước khi được tiến hành.

− Kiểm thử ban đầu ít, sau đó quá trình lớn dần.

− Không thể kiểm thử mọi khía cạnh.

− Để kiểm thử đạt kết quả cao nhất, quá trình kiểm thử phải có sự tham gia của bên thứ 3.

• Mục đích của kiểm thử là để đảm bảo chất lượng. Luồng công việc này liên quan đến:

− Xét duyệt sự tương tác giữa các thành phần trong hệ thống

− Xét duyệt sự tích hợp đúng đắn các thành phần

− Xét duyệt tất cả các yêu cầu đã được cài đặt

− Đảm bảo rằng phát hiện các lỗi trước khi triển khai hệ thống

• Các bước kiểm thử:

− Kiểm thử đơn vị

− Kiểm thử tích hợp

− Kiểm thử hệ thống

− Kiểm thử sự chấp nhận

**2.2 Công nghệ áp dụng**

2.2.1 Công nghệ Fontend – VueJS

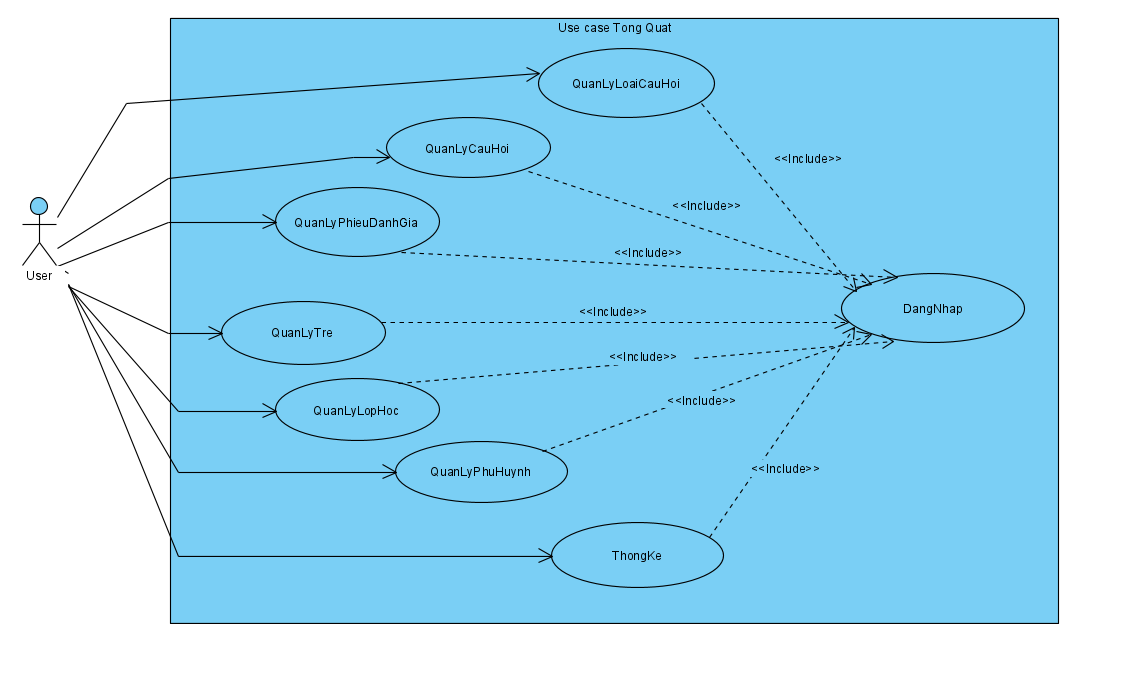
* VueJS là một JS Framework dùng để xây dựng giao diện người dùng. Vue được thiết kế từ đầu theo hướng cho phép và khuyến khích việc phát triển ứng dụng theo từng bước... ... Khác với các framework nguyên khối (monolithic), Vue được thiết kế từ đầu theo hướng cho phép và khuyến khích việc phát triển ứng dụng theo từng bước
* Khác với các framework nguyên khối (monolithic), Vue được thiết kế từ đầu theo hướng cho phép và khuyến khích việc phát triển ứng dụng theo từng bước. Khi phát triển lớp giao diện (view layer), người dùng chỉ cần dùng thư viện lõi (core library) của Vue, vốn rất dễ học và tích hợp với các thư viện hoặc dự án có sẵn. Cùng lúc đó, nếu kết hợp với những kĩ thuật hiện đại như SFC (single file components) và các thư viện hỗ trợ, Vue cũng đáp ứng được dễ dàng nhu cầu xây dựng những ứng dụng một trang (SPA - Single-Page Applications) với độ phức tạp cao hơn nhiều.
* Vuejs là JS Frameworkmọi hoạt động của Vuejsđều được Rendervà xử lý bằng Javascript, Chính vì thế điểu này làm cho website của bạn trở nên nhẹ hơn, và tốc độ cũng trở nên nhanh hơn
* Vậy Nhược điểm của Vuejs là gì? Đó chính là liên quan đến SEO đấy bạn ạ, Vuejs có hỗ trợ Server Side Rendering hay còn gọi SSRtuy nhiên việc không đơn giản như thế, để Render được HTML từ Vuejs phí server bạn cần phải tuân thủ sử dụng thuần Vuejs và các Component của bên thứ 3 nhúng vào cũng phải tuần thủ quy định về SSR. Mà không phải đơn vị nào cũng hỗ trợ điều này ở thời điểm hiện tại. Vì thế việc SEO trên Vuejs trở nên khó khăn hơn.

2.2.2 Công nghệ Backend - .Net Core API

* [ASP.NET Core](https://netcore.vn/muc-bai-viet/aspnet-core) là một framework mới của Microsoft. Nó được thiết kế lại từ đầu để trở nên nhanh chóng, linh hoạt và chạy trên được nhiều nền tảng khác nhau. Trong bài viết này, ASP.NET Core là một framework có thể được sử dụng để phát triển web với .NET. Nếu bạn có kinh nghiệm làm việc với MVC hoặc Web API trong một vài năm thì bạn sẽ nhận thấy những tính năng quen thuộc. Kết thúc khóa học ASP.NET Core của NET CORE VN. Bạn sẽ có những thứ cần thiết để bắt đầu làm việc được với ASP.NET Core và viết 1 ứng dụng có thể thêm, sửa hay xem dữ liệu từ database như các môn lập trình khác bạn đã từng học.
* 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 4.6Đó 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.
* Ta có thể phát triển và chạy các ứng dụng đa nền tảng từ ASP.NET Core trên Windows, Mac và Linux.
* 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.
* Xây dựng và chạy các ứng dụng ASP.NET Core đa nền tảng trên Windows, Mac và Linux.
* Được xây dựng trên [.NET Core](https://netcore.vn/), 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 makup trở nên tự nhiên hơn với HTML.
* Có khả năng host trên IIS hoặc self-host.

# PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

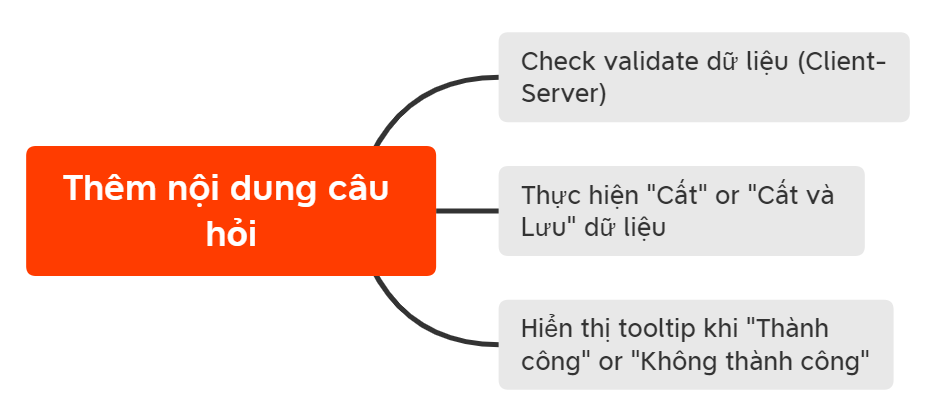
* 1. Đặc tả yêu cầu phần mềm
     1. Các yêu cầu chức năng
        1. Chức năng của phân hệ quản trị nội dung (nếu có)



Hình 3.1: UseCase Tổng quát

Mô tả:

* Người quản trị đăng nhập hệ thống
* Quản lý các UseCase tổng quát như hình 3.1.
* Quản lý câu hỏi gồm 6 chức năng:
* Mô tả: Chức năng Thêm nội dung câu hỏi – Giúp thực thi được các yêu cầu phía người dùng khi muốn thêm mới nội dung câu hỏi.
* Biểu đồ sử dụng:

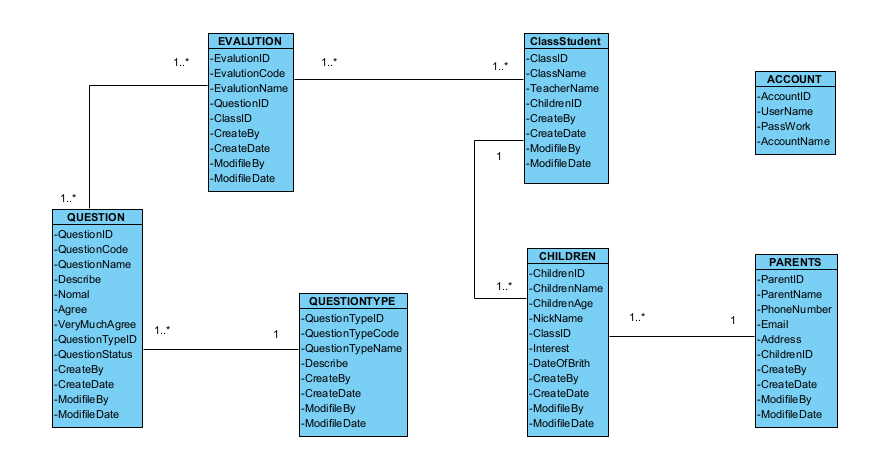


Hình 3.2: Mô tả chức năng thêm mới câu hỏi

* Luồng sự kiện:

|  |
| --- |
| **Usecase name:** *Thêm mới bản ghi*  **Brief description:** Quản trị thêm mới bản ghi trên Form giao diện  **Actors:**  **Precondition:** Hiển thị Form thêm nội dung câu hỏi  **Post – condition:**  Form được reset các ô text và option  **Trigger:** |
| **Basic flow** |
| 1.Admin chọn button [Thêm]  2. SYSTEM Hệ thống hiển thị Form giao diện đã được reset.  3. Admin nhập dữ liệu vào các ô input và option  4. Admin khi nhấn nút “Lưu”  5. SYSTEM Hệ thống kiểm tra validate dữ liệu  5.1 If “True” return hàm Lưu() và trả về tooltip “Thêm thành công”.  5.2 Else “False” return tooltip “Dữ liệu không hợp lệ” Quay lại bước 3 |

* + - 1. Chức năng của phân hệ người dung (nếu có)
    1. Biểu đồ lớp thực thể



* Mô tả các lớp của hệ thống

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên thuộc tính** | **Mô tả** |
| 1 | QuestionID | Khóa chính |
| 2 | QuestionCode | Mã câu hỏi |
| 3 | QuestionName | Tên câu hỏi |
| 4 | Describe | Diễn tả |
| 5 | Satisfied | Hài lòng |
| 6 | Unsatisfied | Không hài lòng |
| 7 | VerySatisfied | Rất hài lòng |
| 8 | VeryUnsatisfied | Rất không hài lóng |
| 9 | Normal | Bình thường |
| 10 | QuestionTypeID | Mã loại câu hỏi |
| 11 | QuestionStatus | Trạng thái câu hỏi |
| 12 | CreateBy | Người tạo |
| 13 | CreateDate | Ngày tạo |
| 14 | ModifileBy | Người sửa |
| 15 | ModifileDate | Ngày sửa |

Bảng 3.1: Mô tả thuộc tính đối tượng Question

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên thuộc tính** | **Mô tả** |
| 1 | QuestionTypeID | Khóa chính |
| 2 | QuestionTypeCode | Mã loại câu hỏi |
| 3 | QuestionTypeName | Tên loại câu hỏi |
| 4 | Describe | Diễn tả |
| 12 | CreateBy | Người tạo |
| 13 | CreateDate | Ngày tạo |
| 14 | ModifileBy | Người sửa |
| 15 | ModifileDate | Ngày sửa |

Bảng 3.2: Mô tả thuộc tính đối tượng QuestionType

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên thuộc tính** | **Mô tả** |
| 1 | ChildrenID | Khóa chính |
| 2 | ChildrenName | Tên học sinh |
| 3 | ChildrenAge | Tuổi |
| 4 | NickName | Tên phụ |
| 12 | ClassID | Mã lớp |
| 13 | Interest | Sở thích |
| 14 | DateOfBrith | Sinh nhật |
| 12 | CreateBy | Người tạo |
| 13 | CreateDate | Ngày tạo |
| 14 | ModifileBy | Người sửa |
| 15 | ModifileDate | Ngày sửa |

Bảng 3.3: Mô tả thuộc tính đối tượng Children

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên thuộc tính** | **Mô tả** |
| 1 | ClassID | Khóa chính |
| 2 | ClassName | Tên lớp |
| 3 | TeacherName | Tên giáo viên |
| 4 | ChildrenID | Mã học sinh |
| 12 | CreateBy | Người tạo |
| 13 | CreateDate | Ngày tạo |
| 14 | ModifileBy | Người sửa |
| 15 | ModifileDate | Ngày sửa |

Bảng 3.4: Mô tả thuộc tính đối tượng ClassStudent

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên thuộc tính** | **Mô tả** |
| 1 | EvalutionID | Khóa chính |
| 2 | EvalutionCode | Mã phiếu đánh giá |
| 3 | EvalutionName | Tên phiếu đánh giá |
| 4 | QuestionID | Mã câu hỏi |
| 12 | ClassID | Mã lớp |
| 12 | CreateBy | Người tạo |
| 13 | CreateDate | Ngày tạo |
| 14 | ModifileBy | Người sửa |
| 15 | ModifileDate | Ngày sửa |

Bảng 3.5: Mô tả thuộc tính đối tượng Evalution

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên thuộc tính** | **Mô tả** |
| 1 | EvalutionID | Khóa chính |
| 2 | EvalutionCode | Mã phiếu đánh giá |
| 3 | EvalutionName | Tên phiếu đánh giá |
| 4 | QuestionID | Mã câu hỏi |
| 12 | ClassID | Mã lớp |
| 12 | CreateBy | Người tạo |
| 13 | CreateDate | Ngày tạo |
| 14 | ModifileBy | Người sửa |
| 15 | ModifileDate | Ngày sửa |

Bảng 3.6: Mô tả thuộc tính đối tượng Parents

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên thuộc tính** | **Mô tả** |
| 1 | AccountID | Khóa chính |
| 2 | UserName | Tên tài khoản |
| 3 | PassWork | Mật khẩu |
| 4 | AccountName | Tên người dùng |
| 5 | Rule | Quyền |

Bảng 3.7: Mô tả thuộc tính đối tượng Account

* + 1. Các yêu cầu phi chức năng

Các yêu cầu phi chức năng là những ràng buộc và điều kiện đối với các yêu cầu chức năng của hệ thống như: ràng buộc về thời gian, ràng buộc về hiệu năng, các tiêu chuẩn được sử dụng...

Những yêu cầu này ảnh hưởng đến chất lượng, khả năng sử dụng của hệ thống và có tác động trực tiếp đến sự hài lòng của người sử dụng, do đó, quyết định sự thành công của hệ thống. Các yêu cầu phi chức năng chung cho các hệ thống thông tin cung cấp dịch vụ công trực tuyến, gồm 7 chức năng cơ bản, đó là:

(1) Hiệu năng hoạt động:

* Yêu cầu về thời gian;
* Tài nguyên sử dụng;
* Công suất tối đa;

(2) Tương thích:

* Cùng tồn tại;
* Tương tác liên thông;

(3) Tính khả dụng: là mức độ sử dụng được và làm hài lòng người sử dụng như:

* Phù hợp với nhu cầu;
* Dễ dàng học cách sử dụng;
* Giao diện người sử dụng;
* Khả năng truy cập, khai thác;

(4) Tính tin cậy:

* Trưởng thành;
* Sẵn sàng;
* Khả năng chịu lỗi;
* Khả năng phục hồi;
* Thời gian giữa các lần xảy ra sự cố gián đoạn hoạt động của hệ thống;

(5) An toàn thông tin:

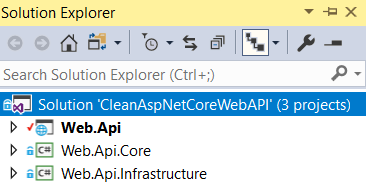
* Bảo mật;
* Toàn vẹn;
* Xác thực.

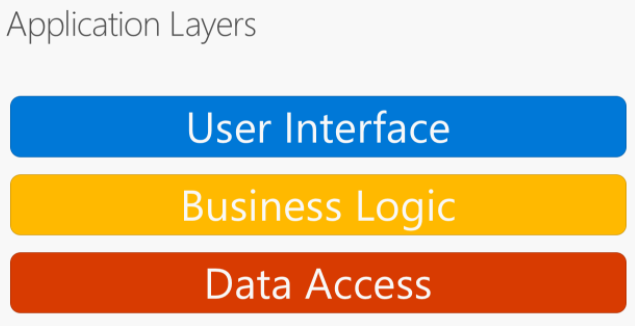
(6) Duy trì được là Phân tích được; Hiệu chỉnh được và Khả chuyển là Mức độ hiệu suất và hiệu quả của việc dịch chuyển một hệ thống từ một nền tảng phần cứng, phần mềm, hệ điều hành từ môi trường sử dụng này sang môi trường sử dụng khác; hỗ trợ khả năng dịch chuyển từ nền tảng chạy trên Unix sang nền tảng chạy trên Windows Server một cách dễ dàng, không phát sinh thêm chi phí hoặc kinh phí không đáng kể; Thích ứng: là hỗ trợ và sử dụng các trình duyệt thông dụng hiện nay như Micrsoft Internet Explorer, Google Crome, Mozila Firefox…; Cài đặt được; Vận hành; khai thác; Khả năng thay thế được là mức độ một sản phẩm phần mềm của hệ thống có thể được thay thế bởi một sản phẩm phần mềm khác có cùng mục đích sử dụng trong cùng một môi trường.

(7) Các yêu cầu phi chức năng khác: Yêu cầu phi chức năng cho công cụ tiềm kiếm, thống kê, xây dựng báo cáo; Lưu trữ: là tính đáp ứng của hệ thống đối với danh mục tiêu chuẩn, quy chuẩn kỹ thuật về ứng dụng công nghệ thông tin trong cơ quan nhà nước; Nền tảng công nghệ nền, phát triển ứng dụng là yêu cầu, định hướng lựa chọn công nghệ nền tảng dùng để xây dựng, phát triển hệ thống trong các dịch vụ công trực tuyến cấp huyện được triển khai độc lập, ưu tiên lựa chọn hệ quản trị cơ sở dữ liệu miễn phí đối với các phần mềm ứng dụng nội bộ đang có trên thị trường, ưu tiên lựa chọn các phần mềm miễn phí mã nguồn mở.

* 1. Thiết kế hệ thống
     1. Thiết kế kiến trúc

a) Kiến trúc ứng dụng Backend - Clean Architecture ASP.NET Core WebApi.



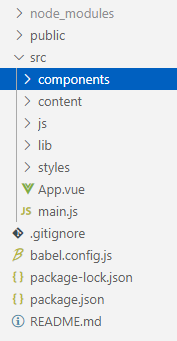


*Hình 3.16 Các lớp ứng dụng điển hình trong Clean Architecture*

Các lớp này thường được viết tắt là UI, BLL (Business Logic Layer) và DAL (Data Access Layer). Sử dụng kiến trúc này, người dùng đưa ra các yêu cầu thông qua lớp giao diện người dùng, lớp này chỉ tương tác với BLL. Đến lượt nó, BLL có thể gọi DAL cho các yêu cầu truy cập dữ liệu. Lớp giao diện người dùng không nên trực tiếp đưa ra bất kỳ yêu cầu nào tới DAL, cũng như không nên tương tác trực tiếp với sự bền bỉ thông qua các phương tiện khác. Tương tự như vậy, BLL chỉ nên tương tác với sự bền bỉ bằng cách thông qua DAL. Theo cách này, mỗi lớp có trách nhiệm nổi tiếng của riêng nó.

Một nhược điểm của cách tiếp cận phân lớp truyền thống này là các phụ thuộc thời gian biên dịch chạy từ trên xuống dưới. Nghĩa là, lớp giao diện người dùng phụ thuộc vào BLL, lớp này phụ thuộc vào DAL. Điều này có nghĩa là BLL, thường giữ logic quan trọng nhất trong ứng dụng, phụ thuộc vào chi tiết triển khai truy cập dữ liệu (và thường là sự tồn tại của cơ sở dữ liệu). Kiểm tra logic nghiệp vụ trong một kiến trúc như vậy thường rất khó, đòi hỏi một cơ sở dữ liệu kiểm tra. Nguyên tắc đảo ngược phụ thuộc có thể được sử dụng để giải quyết vấn đề này, như bạn sẽ thấy trong phần tiếp theo.

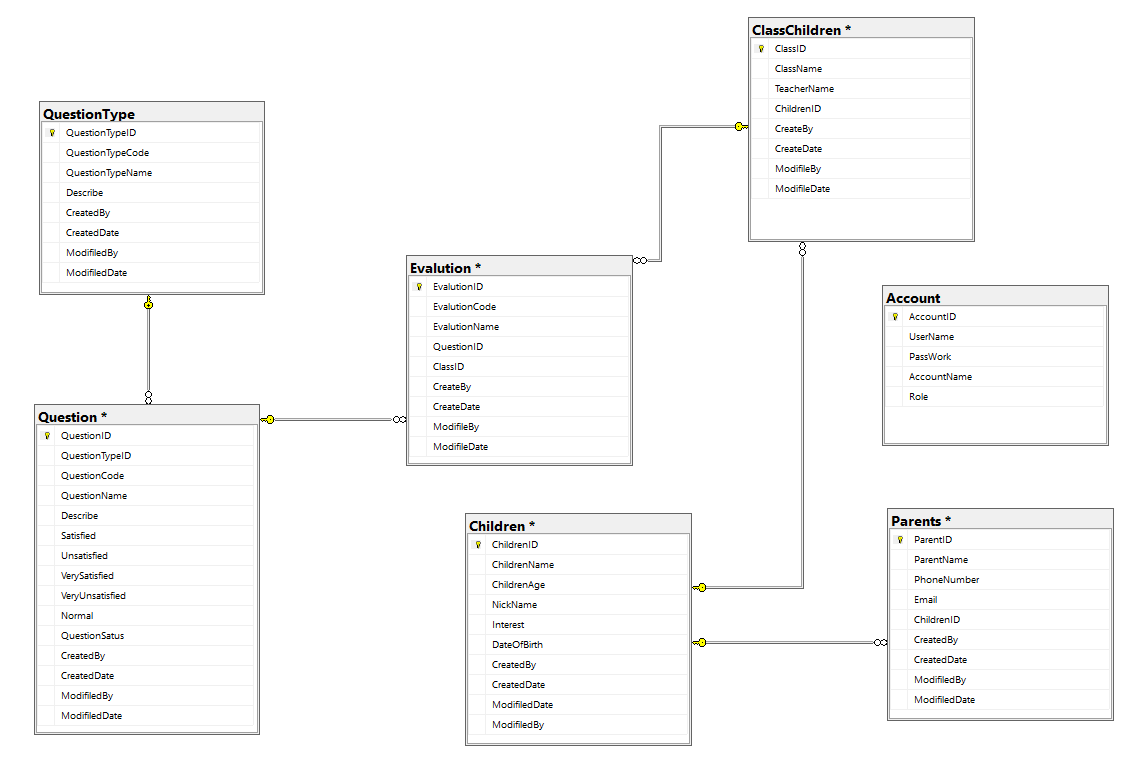
b) Kiến trúc ứng dụng Fontend – VueJS CLI Architecture



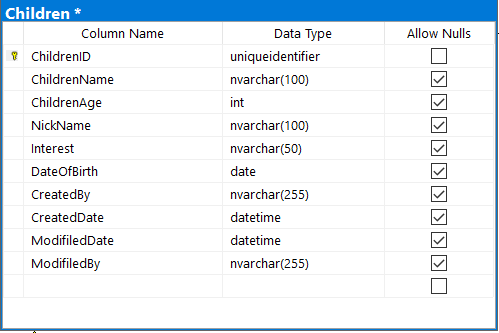
*Hình 3.17 Kiến trúc dự án VueJS*

* **src**: đây là thư mục chưa mã nguồn dự án. Trong thư mục này lại phân chia tiếp.
* **assets**: Module assets nơi mà bạn sẽ làm việc với Webpack
* **components**: Tất cả UI components sẽ nằm ở đây.
* **router**: đây là nơi bạn sẽ viết routes và kết nối chúng với UI components.
* **vue**: Đây là entry point component. Là nơi sẽ khởi tạo tất cả các component khác. Hiểu nôm na là tệp chính của dự án
* **js**: Entry point file để mount App.vue.
* **assets**: pure assets ( assets của riêng dự án), không liên quan tới webpack.
* **html**: Bạn có nhớ là ứng dụng SPA (Single Page Application) thì có 1 trang duy nhất. Sau đó, nội dung của trang bị thay đổi mà không phải tải lại trang. Và đây chính là trang duy nhất đó.
* Một template: Là phần hiển thị của một component. Hiểu nôm na là UI của component
* Một script: là nơi thực hiện logic cho component
* Và một style: định dạng trang, mục đích để trang trí “sắc đẹp” cho component.
  + 1. Thiết kê cơ sở dữ liệu

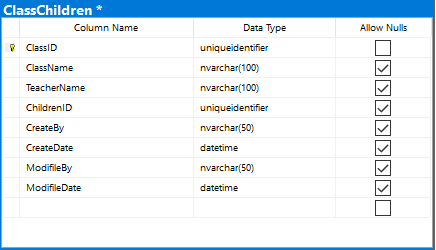
- Mô hình cơ sở dữ liệu quan hệ



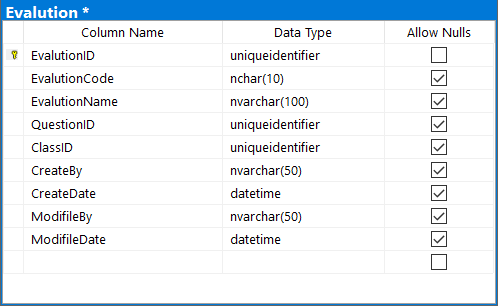
- Mô tả cấu trúc bảng



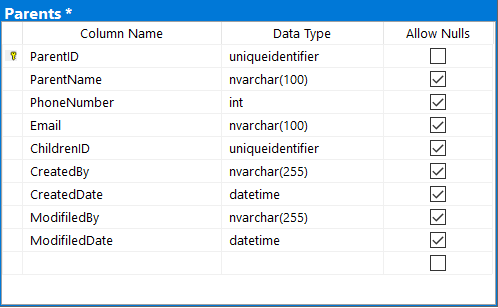
Bảng 3.8: Mô tả cấu trúc bảng Children



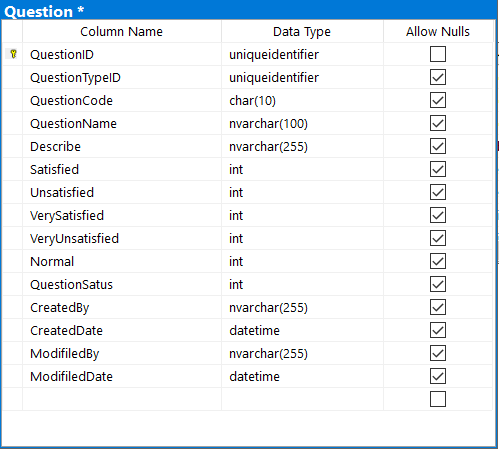
Bảng 3.9: Mô tả cấu trúc bảng ClassStudent



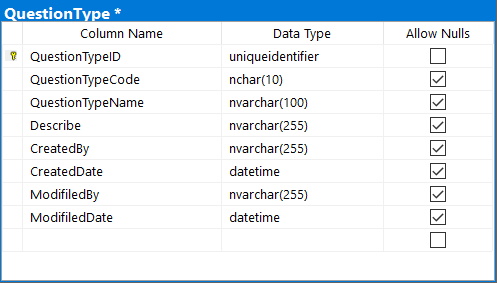
Bảng 3.10: Mô tả cấu trúc bảng Evalution



Bảng 3.11: Mô tả cấu trúc bảng Parents

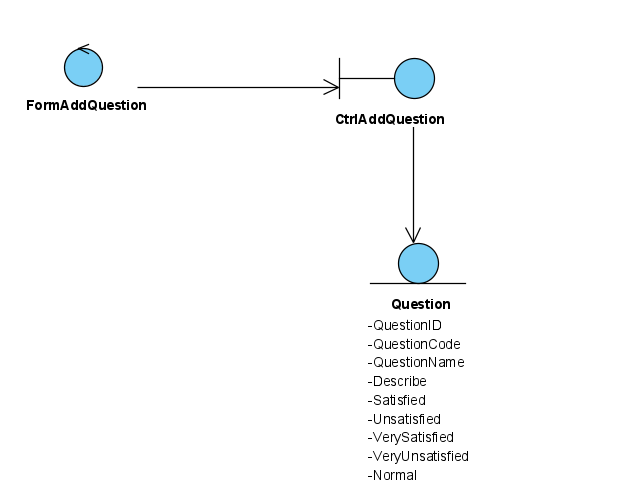


Bảng 3.12: Mô tả cấu trúc bảng Question



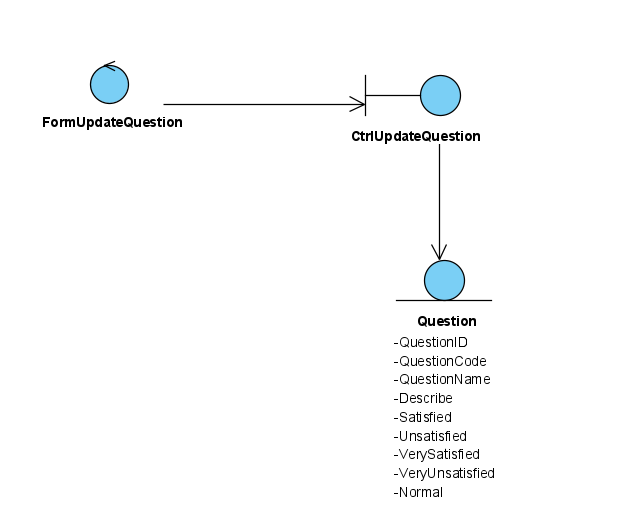
Bảng 3.13: Mô tả cấu trúc bảng QuestionType

* + 1. Thiết kế lớp đối tượng
       1. Biểu đồ lớp VOPC của các ca sử dụng
* Biểu đồ lớp VOPC của Use Case Quản lý Question
  + Thêm câu hỏi



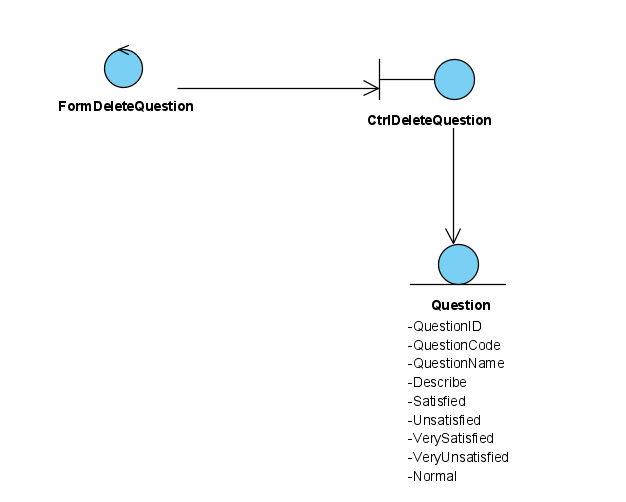
Hình 3.3: Biểu đồ lớp VOPC thêm câu hỏi

* + Cập nhật câu hỏi



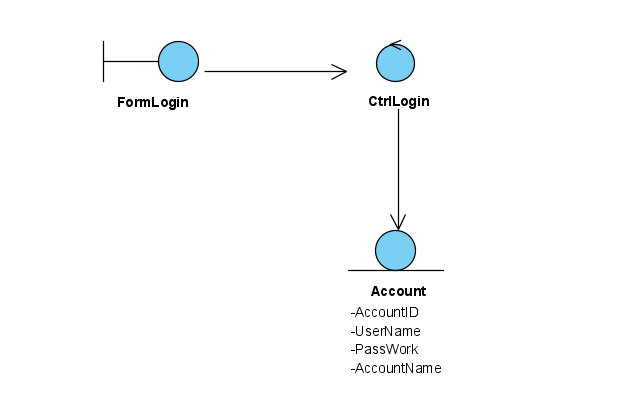
Hình 3.4: Biểu đồ lớp VOPC Cập nhật câu hỏi

* + Xóa câu hỏi



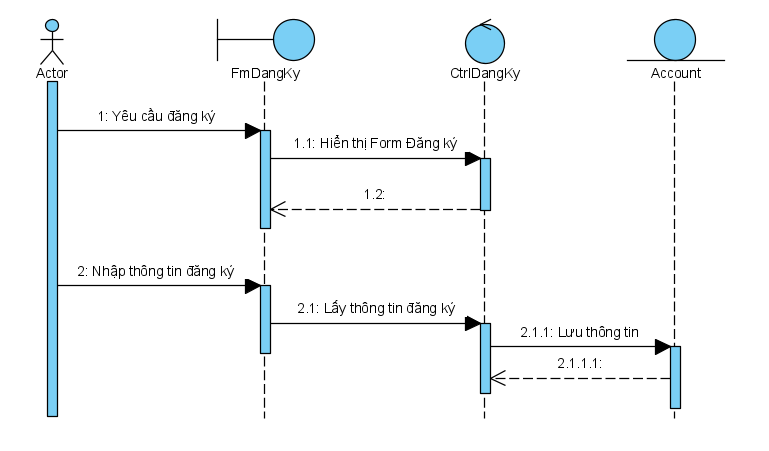
Hình 3.5: Biểu đồ lớp VOPC Xóa câu hỏi

* Biểu đồ lớp VOPC của Use case Đăng nhập
  + Đăng nhập



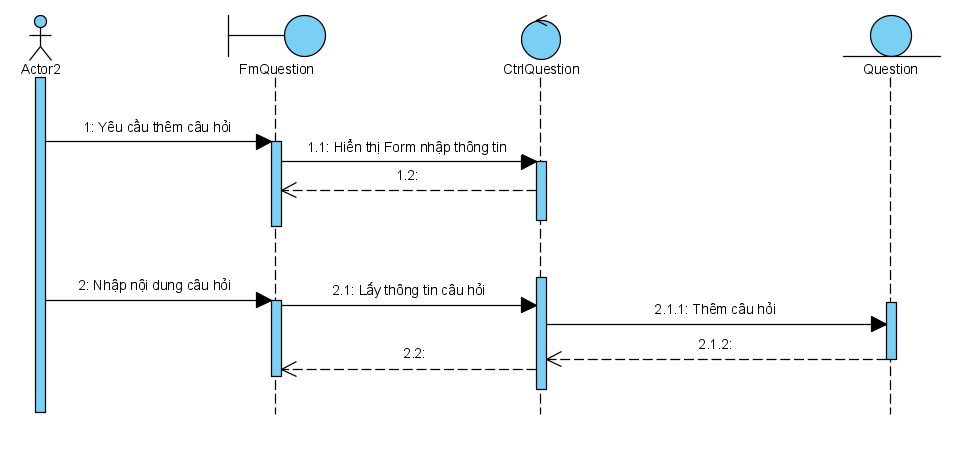
Hình 3.6: Biểu đồ lớp VOPC Đăng nhập

* + - 1. Biểu đồ tuần tự
* Biểu đồ tuần tự của Use case Đăng ký



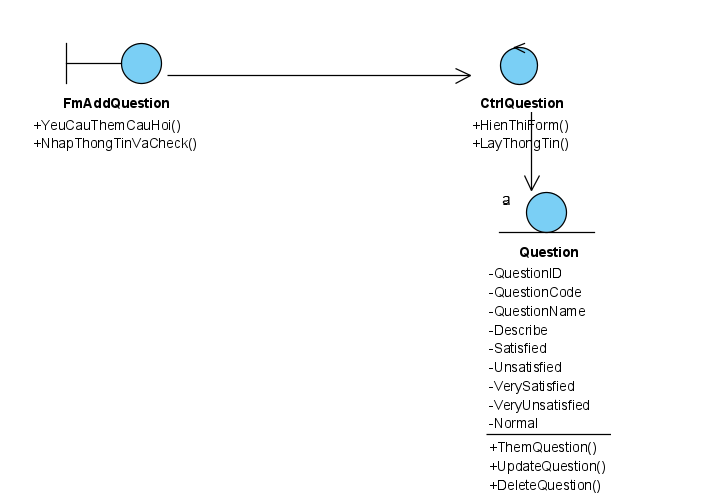
Hình 3.7: Biểu đồ tuần tự Đăng ký

* Biểu đồ tuần tự của Use case Quản lý câu hỏi
  + Thêm câu hỏi



Hình 3.8: Biểu đồ thêm câu hỏi

* + - 1. Biểu đồ lớp chi tiết
* Biểu đồ lớp chi tiết cho Use case Quản lý câu hỏi



Hình 3.9: Biểu đồ lớp chi tiết Quản lý câu hỏi

* + 1. Thiết kế giao diện

# TRIỂN KHAI WEBSITE

* 1. Xây dựng Web API

<Trình bày phương pháp xây dựng API và đặc tả các API>

* 1. Xây dựng các chức năng
     1. Các chức năng nghiệp vụ phân hệ người dùng

<Trình bày ý tưởng xây dựng các chức năng nghiệp vụ như: Hiển thị, tìm kiếm, tính toán, thống kê báo cáo>

* + 1. Các chức hệ thống

<Trình bày ý tưởng xây dựng các chức chức năng hệ thống như đăng nhập, đăng ký, quản lý người dùng>

* + 1. **Các chức năng phân hệ quản trị (nếu có)**

<Trình bày ý tưởng xây dựng các chức năng phía trang quản trị >

* 1. Kiểm thử và triển khai ứng dụng

# KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Kết quả đạt được

*<Phần này trình bày kết quả đạt được của đề tài>*

Hạn chế của đề tài

*<Phần này trình bày hạn chế của đề tài>*

Hướng phát triển của đề tài

<*Phần này trình bày hướng phát triển tiếp theo của đề tài*>

TÀI LIỆU THAM KHẢO

x

|  |  |
| --- | --- |
| [1] | Tên tác giả (năm XB), Tên sách, NXB, Nơi XB |
| [2] | ……….. |

PHỤ LỤC

1. <Tiêu đề phụ lục 1 (nếu có) >
2. <Tiêu đề phụ lục 1 (nếu có) >

…..