­Trường Ngoại Ngữ - Tin Học TPHCM

Khoa Công nghệ Thông Tin

Đồ án Tốt nghiệp Đại học

Bộ môn: Công nghệ phần mềm

Farm Hub

|  |  |
| --- | --- |
| Sinh viên thực hiện | Trần Hoàng Long – 15DH110018  Phùng Ngô Trọng Nghĩa – 15DH110364 |
| Người hướng dẫn | TS.Bùi Thị Thanh Tú |

­­

TPHCM, 6/2019

# Lời cam kết

Họ và tên sinh viên: Trần Hoàng Long

Điện thoại liên lạc: 0522936344 Email: tranhoanglong51198@gmail.com

Lớp15TH1501 Hệ đào tạo: Đại Học chính qui

Tôi – Trần Hoàng Long – cam kết Đồ án Tốt nghiệp (ĐATN) là công trình nghiên cứu của bản thân tôi dưới sự hướng dẫn của *Thạc sĩ Bùi Thị Thanh Tú*. Các kết quả nêu trong ĐATN là trung thực, là thành quả của riêng tôi, không sao chép theo bất kỳ công trình nào khác. Tất cả những tham khảo trong ĐATN – bao gồm hình ảnh, bảng biểu, số liệu, và các câu từ trích dẫn – đều được ghi rõ ràng và đầy đủ nguồn gốc trong danh mục tài liệu tham khảo. Tôi xin hoàn toàn chịu trách nhiệm với dù chỉ một sao chép vi phạm quy chế của nhà trường.

|  |  |
| --- | --- |
|  | *Hồ Chí Minh, ngày 5 tháng 6 năm 2019*  Tác giả ĐATN  *Họ và tên sinh viên* |

# Lời cảm ơn

Lời cảm ơn của sinh viên (SV) tới người yêu, gia đình, bạn bè, thầy cô, và chính bản thân mình vì đã chăm chỉ và quyết tâm thực hiện ĐATN để đạt kết quả tốt nhất ☺

# Tóm tắt

~~Sinh viên viết tóm tắt ĐATN của mình trong mục này, với 200 đến 350 từ. Theo trình tự, các nội dung tóm tắt cần có: (i) Giới thiệu vấn đề (tại sao có vấn đề đó, hiện tại được giải quyết chưa, có những hướng tiếp cận nào, các hướng này giải quyết như thế nào, hạn chế là gì), (ii) Hướng tiếp cận sinh viên lựa chọn là gì, vì sao chọn hướng đó, (iii) Tổng quan giải pháp của sinh viên theo hướng tiếp cận đã chọn, và (iv) Đóng góp chính của ĐATN là gì, kết quả đạt được sau cùng là gì. Sinh viên cần viết thành đoạn văn, không được viết ý hoặc gạch đầu dòng.~~

# Abstract

~~Mục này khuyến khích sinh viên viết lại mục “Tóm tắt” đồ án tốt nghiệp ở trang trước bằng tiếng Anh. Phần này phải có đầy đủ các nội dung như trong phần tóm tắt bằng tiếng Việt.~~

~~Sinh viên không nhất thiết phải trình bày mục này. Nhưng nếu lựa chọn trình bày, sinh viên cần đảm bảo câu từ và ngữ pháp chuẩn tắc, nếu không sẽ có tác dụng ngược, gây phản cảm.~~

# Mục lục

[Lời cam kết iii](#_Toc529131830)

[Lời cảm ơn iv](#_Toc529131831)

[Tóm tắt v](#_Toc529131832)

[Abstract vi](#_Toc529131833)

[Mục lục vii](#_Toc529131834)

[Danh mục hình vẽ xi](#_Toc529131835)

[Danh mục bảng xii](#_Toc529131836)

[Danh mục công thức xiii](#_Toc529131837)

[Danh mục các từ viết tắt xiv](#_Toc529131838)

[Danh mục thuật ngữ xv](#_Toc529131839)

[Chương 1 Giới thiệu đề tài 1](#_Toc529131840)

[1.1 Đặt vấn đề 1](#_Toc529131841)

[1.2 Mục tiêu và phạm vi đề tài 1](#_Toc529131842)

[1.3 Định hướng giải pháp 2](#_Toc529131843)

[1.4 Bố cục đồ án 2](#_Toc529131844)

[Chương 2 Khảo sát và phân tích yêu cầu 4](#_Toc529131845)

[2.1 Khảo sát hiện trạng 4](#_Toc529131846)

[2.2 Tổng quan chức năng 4](#_Toc529131847)

[2.2.1 Biểu đồ use case tổng quan 5](#_Toc529131848)

[2.2.2 Biểu đồ use case phân rã XYZ 5](#_Toc529131849)

[2.2.3 Quy trình nghiệp vụ 5](#_Toc529131850)

[2.3 Đặc tả chức năng 5](#_Toc529131851)

[2.3.1 Đặc tả use case A 5](#_Toc529131852)

[2.3.2 Đặc tả use case B 5](#_Toc529131853)

[2.4 Yêu cầu phi chức năng 6](#_Toc529131854)

[Chương 3 Công nghệ sử dụng 7](#_Toc529131855)

[Chương 4 Phát triển và triển khai ứng dụng 8](#_Toc529131856)

[4.1 Thiết kế kiến trúc 8](#_Toc529131857)

[4.1.1 Lựa chọn kiến trúc phần mềm 8](#_Toc529131858)

[4.1.2 Thiết kế tổng quan 8](#_Toc529131859)

[4.1.3 Thiết kế chi tiết gói 9](#_Toc529131860)

[4.2 Thiết kế chi tiết 10](#_Toc529131861)

[4.2.1 Thiết kế giao diện 10](#_Toc529131862)

[4.2.2 Thiết kế lớp 10](#_Toc529131863)

[4.2.3 Thiết kế cơ sở dữ liệu 11](#_Toc529131864)

[4.3 Xây dựng ứng dụng 11](#_Toc529131865)

[4.3.1 Thư viện và công cụ sử dụng 11](#_Toc529131866)

[4.3.2 Kết quả đạt được 11](#_Toc529131867)

[4.3.3 Minh hoạ các chức năng chính 12](#_Toc529131868)

[4.4 Kiểm thử 12](#_Toc529131869)

[4.5 Triển khai 12](#_Toc529131870)

[Chương 5 Các giải pháp và đóng góp nổi bật 13](#_Toc529131871)

[Chương 6 Kết luận và hướng phát triển 14](#_Toc529131872)

[6.1 Kết luận 14](#_Toc529131873)

[6.2 Hướng phát triển 14](#_Toc529131874)

[Tài liệu tham khảo 15](#_Toc529131875)

[Phụ lục A-1](#_Toc529131876)

[A Hướng dẫn viết đồ án tốt nghiệp A-1](#_Toc529131877)

[A.1 Quy định chung A-1](#_Toc529131878)

[A.2 Tạo đề mục A-2](#_Toc529131879)

[A.3 Bảng biểu A-2](#_Toc529131880)

[A.4 Hình vẽ A-3](#_Toc529131881)

[A.5 Tài liệu tham khảo A-4](#_Toc529131882)

[A.6 Công thức toán học A-5](#_Toc529131883)

[A.7 Tham chiếu chéo A-5](#_Toc529131884)

[A.8 Cập nhật mục lục và tham chiếu chéo A-6](#_Toc529131885)

[A.9 In quyển đồ án tốt nghiệp A-6](#_Toc529131886)

[B Đặc tả use case B-6](#_Toc529131887)

[B.1 Đặc tả use case “Thống kê tình hình mượn sách” B-6](#_Toc529131888)

[B.2 Đặc tả use case “Đăng ký làm thẻ mượn” B-6](#_Toc529131889)

[C Công nghệ sử dụng C-7](#_Toc529131890)

[C.1 Công nghệ bảo mật dữ liệu C-7](#_Toc529131891)

[C.2 Công nghệ blockchain C-7](#_Toc529131892)

[D Thiết kế gói D-7](#_Toc529131893)

[D.1 Thiết kế gói cho kiến trúc tổng quan D-7](#_Toc529131894)

[D.2 Thiết kế gói cho chức năng “Trả sách” D-7](#_Toc529131895)

[E Thiết kế lớp E-7](#_Toc529131896)

# Danh mục hình vẽ

[**Hình 1** Ví dụ biểu đồ phụ thuộc gói 18](#_Toc10657929)

[**Hình 2** Ví dụ thiết kế gói 19](#_Toc10657930)

[**Hình 3** Ví dụ hình vẽ A-4](#_Toc10657931)

# Danh mục bảng

[**Bảng 1** Danh sách thư viện và công cụ sử dụng 11](#_Toc529131900)

[**Bảng 2** Ví dụ sử dụng bảng A-3](#_Toc529131901)

# Danh mục công thức

[**Công thức 1** Khai triển Newton A-5](#_Toc529131902)

Lưu ý: Nếu ĐATN có ít hơn ba công thức toán học, sinh viên có thể xóa bỏ mục này.

# Danh mục các từ viết tắt

|  |  |
| --- | --- |
| API | Application Programming Interface  Giao diện lập trình ứng dụng |
| **EUD** | End-User Development  Phát triển ứng dụng người dùng cuối |
| **GWT** | Google Web Toolkit  Công cụ lập trình Javascript bằng Java của Google |
| **HTML** | HyperText Markup Language  Ngôn ngữ đánh dấu siêu văn bản |
| **CNTT** | Công nghệ thông tin |
| **ĐATN** | Đồ án tốt nghiệp |
| **SV** | Sinh viên |

# Danh mục thuật ngữ

|  |  |
| --- | --- |
| Browser | Trình duyệt |
| **Cache memory** | Bộ nhớ đệm |
| **E-commerce** | Thương mại điện tử |
| **Bloatware** | Ứng dụng nhà sản xuất tích hợp vào thiết bị |
| **Interpreter** | Trình thông dịch |
| **Compiler** | Trình biên dịch |

# Giới thiệu đề tài

~~Lưu ý:~~ **~~Trước khi viết ĐATN, sinh viên cần đọc kỹ hướng dẫn và quy định chi tiết~~** ~~về cách viết ĐATN trong Phụ lục A. Sinh viên tuân theo mẫu tài liệu này để viết báo cáo đồ án tốt nghiệp, vì tài liệu này đã được căn chỉnh, chỉnh sửa theo đúng chuẩn báo cáo kỹ thuật đồ án tốt nghiệp (ISO 7144:1986). Sinh viên viết trực tiếp vào file word này, chỉ chỉnh sửa nội dung, và không viết trên file word mới.~~

~~Mẫu ĐATN này được thiết kế phù hợp nhất với đa số các đề tài xây dựng phần mềm ứng dụng. Với các dạng đề tài khác (giải pháp, nghiên cứu, phần mềm đặc thù, v.v.), sinh viên dựa trên cấu trúc và hướng dẫn của báo cáo này để đề xuất và trao đổi với giáo viên hướng dẫn để thiết kế khung báo cáo đồ án cho phù hợp. Sinh viên lưu ý~~ **~~trong mọi trường hợp, SV luôn phải sử dụng định dạng báo cáo này, và phải đọc kỹ toàn bộ các hướng dẫn từ đầu tới cuối~~**~~. Các hướng dẫn không chỉ áp dụng riêng cho đề tài ứng dụng, mà còn phù hợp với các dạng đề tài khác. Ngoài ra, trong mẫu ĐATN này đã được tích hợp một số hướng dẫn dành riêng cho đề tài nghiên cứu.~~

~~Chương 1 có độ dài từ 3 đến 6 trang với các nội dung sau đây~~

## Đặt vấn đề

~~Khi đặt vấn đề, sinh viên cần làm nổi bật mức độ cấp thiết, tầm quan trọng và/hoặc quy mô của bài toán của mình.~~

~~Gợi ý cách trình bày cho sinh viên: Xuất phát từ tình hình thực tế gì, dẫn đến vấn đề hoặc bài toán gì. Vấn đề hoặc bài toán đó, nếu được giải quyết, đem lại lợi ích gì, cho những ai, còn có thể được áp dụng vào các lĩnh vực khác nữa không.~~

Ở Việt Nam, nông nghiệp giữ vị trí quan trọng vì nhiều lý do: 80% dân số chủ yếu sống ở nông thôn, nguồn sống chính dựa vào nông nghiệp. Trong cơ cấu kinh tế quốc dân, GDP do nông nghiệp tạo ra vẫn giữ vị trí hàng đầu, trên 50 % giá trị xuất khẩu là nông sản, thủy sản. Sự phát triển của khu vực này có vị trí quan trọng trong việc thực hiện thành công công nghiệp hóa, hiện đại hóa đất nước.

Nhiều năm trở lại đây, “giải cứu nông sản” đã trở thành cụm từ quen thuộc khi mà đến vụ thu hoạch là hàng loạt nông sản sụt giảm giá nghiêm trọng, thậm chí phải đổ bỏ vì không thể tiêu thụ. Vì thế đã hình thành các phong trào “giải cứu” mang tính tự phát, từ giải cứu dưa hấu, thanh long, khoai tây, súp lơ, thậm chí đến giải cứu tỏi, ớt, đường... Điểm chung là hầu hết các sản phẩm này đều được trồng đại trà, không theo quy hoạch, không tìm hiểu thị trường tiêu thụ, cũng không có nhãn mác, xuất xứ hàng hóa hay các thông số tiêu chuẩn kỹ thuật liên quan. Trong quá trình “giải cứu”, việc mua - bán chủ yếu xuất phát từ ý muốn giúp đỡ của cộng đồng với người nông dân, chứ không phải vì chất lượng sản phẩm. Cùng với hiện tượng “giải cứu nông sản”, còn một hiện tượng khác nổi lên là thị trường mua bán các sản phẩm “quê” như: gạo quê, gà quê, rau quả quê,… nghĩa là trên một khía cạnh nào đó đã cố gắng gắn “mác” để người tiêu dùng an tâm về tâm lý, trong khi thực chất những sản phẩm này cũng không hề có tiêu chí chất lượng. Chưa kể vẫn còn hiện tượng kinh doanh chụp giật, gian lận khi xuất khẩu và làm thủ tục kiểm dịch thực vật, cạnh tranh không lành mạnh làm giảm uy tín hàng nông sản Việt Nam.

Vì vậy, quan tâm hàng đầu của người tiêu dùng là “Thực phẩm sạch - An toàn”. Một sản phẩm nông sản đáp ứng được tiêu chí về chất lượng phải có: nhật ký đồng ruộng (quá trình tưới tiêu, bón phân, phun thuốc trừ sâu, vv), kiểm dịch thực vật (thành phần dịch hại, biện pháp phòng trừ), giống cây trồng, truy xuất nguồn gốc,  sản lượng, vv..

Ngoài ra, một trong những nhược điểm của chủ nông trại hiện nay là làm việc manh mún. Bên cạnh đó, do tập quán làm việc theo cảm tính dẫn đến người chủ nông trại không có định hướng phát triển hoạt động nông nghiệp rõ ràng nếu như không có sự tư vấn chi tiết của các cơ quan chuyên môn, của những người có kinh nghiệm.

Về phía thương lái, cũng chính vì không có sự tương tác với chủ nông trại nên hầu như cũng không nắm rõ được tình hình nguồn cung. Điều này đưa đến một tình cảnh đó là "bên cung không nắm được cầu, và bên cầu không nắm được cung" gây hệ quả là thương lái không có đủ thông tin để dự báo và lập kế hoạch cung ứng hiệu quả.

Vậy những vấn đề đặt ra là:

* Làm sao để đảm bảo sản phẩm nông sản sạch và an toàn.
* Làm sao để giúp người chủ nông trại nắm bắt được giá cả và khối lượng nông sản cung cấp trên toàn thị trường để tránh tình trạng cung nhiều hơn cầu và ngược lại ?
* Làm sao để thương lái và chủ nông trại có thể tương tác tốt với nhau ?
* Làm sao để tránh tình trạng bán phá giá, thương lái ép giá chủ nông trại khiến giá nông sản giảm ngoài tầm kiểm soát ?

## Mục tiêu và phạm vi đề tài

~~Sinh viên trước tiên cần trình bày tổng quan các kết quả của các nghiên cứu hiện nay cho bài toán giới thiệu ở phần 1.1 (đối với đề tài nghiên cứu), hoặc về các sản phẩm hiện tại/về nhu cầu của người dùng (đối với đề tài ứng dụng). Tiếp đến, sinh viên tiến hành so sánh và đánh giá tổng quan các sản phẩm/nghiên cứu này.~~

~~Dựa trên các phân tích và đánh giá ở trên, sinh viên khái quát lại các hạn chế hiện tại đang gặp phải. Trên cơ sở đó, sinh viên sẽ hướng tới giải quyết vấn đề cụ thể gì, khắc phục hạn chế gì, phát triển phần mềm gì, tạo nên đột phá gì, v.v.~~

~~Trong phần này, sinh viên lưu ý chỉ trình bày tổng quan, không đi vào chi tiết của vấn đề hoặc giải pháp. Nội dung chi tiết sẽ được trình bày trong các chương tiếp theo, đặc biệt là trong Chương 5.~~

Hiện tại đã có Meicai - ứng dụng giúp nông dân bán rau cho các nhà hàng của Trung Quốc. Thông qua ứng dụng di động, khách hàng có thể đặt sản vật địa phương trực tiếp từ các chủ trang trại, nhờ đó giảm chi phí trung gian nhưng vấn đề về chất lượng và an toàn thực phẩm vẫn chưa được đảm bảo.

Barter BlockTM và Binkabi DEX là hai sàn giao dịch dựa trên nền tảng Blockchain cho phép giao dịch quốc tế một cách trực tiếp, an toàn, chi phí thấp. Công nghệ Blockchain cũng giúp giải quyết vấn đề về truy xuất nguồn gốc trong nông nghiệp, giúp tháo gỡ nhiều vướng mắc trong việc không minh bạch thông tin sản phẩm. Khi áp dụng, khách hàng có thể biết được sản phẩm nguồn gốc từ đâu, chăm sóc như thế nào. Nhược điểm của blockchain là lưu trữ thông tin mãi mãi và không thể sửa được, vì thế càng sử dụng lâu dung lượng lưu trữ càng lớn. Độ trễ xác minh giao dịch cũng là một vấn đề lớn đối với Blockchain. Ngoài ra còn có các nhược điểm khác như vấn đề băng thông, vấn đề xử lý thông tin, vấn đề an toàn thực phẩm vẫn chưa được giải quyết triệt để.

Vì vậy,  chúng em sẽ hướng tới việc giải quyết các vấn đề sau:

* Vấn đề về đảm bảo chất lượng và an toàn thực phẩm.
* Vấn đề về theo dõi, nắm bắt thông tin nhu cầu thị trường.
* Vấn đề về tương tác, phối hợp giữa chủ trang trại và thương lái.

Phạm vi đề tài: Tất cả thương lái và chủ nông trại trên đất nước.

## Định hướng giải pháp

~~Từ việc xác định rõ nhiệm vụ cần giải quyết ở phần 1.2, sinh viên đề xuất định hướng giải pháp của mình theo trình tự sau: (i) Sinh viên trước tiên trình bày sẽ giải quyết vấn đề theo định hướng, phương pháp, thuật toán, kỹ thuật, hay công nghệ nào; Tiếp theo, (ii) sinh viên mô tả ngắn gọn giải pháp của mình là gì (khi đi theo định hướng/phương pháp nêu trên); và sau cùng, (iii) sinh viên trình bày đóng góp chính của đồ án là gì, kết quả đạt được là gì.~~

~~Sinh viên lưu ý không giải thích hoặc phân tích chi tiết công nghệ/thuật toán trong phần này. Sinh viên chỉ cần nêu tên định hướng công nghệ/thuật toán, mô tả ngắn gọn trong một đến hai câu và giải thích nhanh lý do lựa chọn.~~

Từ việc xác định rõ nhiệm vụ cần giải quyết ở phần 1.2, nhóm chúng em đề xuất định hướng giải pháp của mình theo trình tự sau:

(i) Phương hướng thực hiện: Đầu tiên nhóm chúng em sẽ giúp chủ nông nghiệp và thương lái nắm bắt thông tin nhu cầu thị trường và tương tác lẫn nhau. Sau đó sẽ tiếp tục giải quyết vấn đề về đảm bảo chất lượng vệ sinh an toàn thực phẩm.

(ii) Giải pháp: Trước tiên, nhóm em sẽ xây dựng một website có những chức năng như sau: cung cấp các thông tin trên thị trường nông sản (sản phẩm, khối lượng, giá, ...), giúp chủ trang trại và thương lái tương tác với nhau, theo dõi giá trung bình của nhiều loại nông sản khác nhau. Sau đó, nhóm em sẽ áp dụng IoT (Internet of Things) vào khâu sản xuất nông sản, giải quyết được vấn đề về đảm bảo chất lượng vệ sinh an toàn thực phẩm.

(iii) Kết quả đạt được: Giúp các chủ nông trại nắm bắt được nhu cầu thị trường, chủ động hơn trong kế hoạch nuôi trồng, giảm thiểu vấn đề cung nhiều hơn cầu, tương tác được với các thương lái khác để đảm bảo đầu ra cho sản phẩm của mình. Giá cả được ổn định, giảm thiểu tình trạng mua ép giá. Chất lượng nông sản được nâng cao, đảm bảo vệ sinh an toàn thực phẩm.

## Bố cục đồ án

~~Phần còn lại của báo cáo đồ án tốt nghiệp này được tổ chức như sau.~~

~~Chương 2 trình bày về v.v.~~

~~Trong Chương 3, em/tôi giới thiệu về v.v.~~

**~~Chú ý:~~** ~~Sinh viên cần viết mô tả thành đoạn văn đầy đủ về nội dung chương. Tuyệt đối không viết ý hay gạch đầu dòng. Chương 1 không cần mô tả trong phần này.~~

~~Ví dụ tham khảo mô tả chương trong phần bố cục đồ án tốt nghiệp: Chương \*\*\* trình bày đóng góp chính của đồ án, đó là một nền tảng ABC cho phép khai phá và tích hợp nhiều nguồn dữ liệu, trong đó mỗi nguồn dữ liệu lại có định dạng đặc thù riêng. Nền tảng ABC được phát triển dựa trên khái niệm DEF, là các module ngữ nghĩa trợ giúp người dùng tìm kiếm, tích hợp và hiển thị trực quan dữ liệu theo mô hình cộng tác và mô hình phân tán.~~

**~~Chú ý:~~** ~~Trong phần nội dung chính, mỗi chương của đồ án nên có phần~~ *~~Tổng quan~~* ~~và~~ *~~Kết chương~~*~~. Hai phần này đều có định dạng văn bản “Normal”, sinh viên không cần tạo định dạng riêng, ví dụ như không in đậm/in nghiêng, không đóng khung, v.v.~~

~~Trong phần~~ *~~Tổng quan~~* ~~của chương N, sinh viên nên có sự liên kết với chương N-1 rồi trình bày sơ qua lý do có mặt của chương N và sự cần thiết của chương này trong đồ án. Sau đó giới thiệu những vấn đề sẽ trình bày trong chương này là gì, trong các đề mục lớn nào.~~

~~Ví dụ về phần~~ *~~Tổng quan~~*~~: Chương 3 đã thảo luận về nguồn gốc ra đời, cơ sở lý thuyết và các nhiệm vụ chính của bài toán tích hợp dữ liệu. Chương 4 này sẽ trình bày chi tiết các công cụ tích hợp dữ liệu theo hướng tiếp cận “mashup”. Với mục đích và phạm vi của đề tài, sáu nhóm công cụ tích hợp dữ liệu chính được trình bày bao gồm: (i) nhóm công cụ ABC trong phần 4.1, (ii) nhóm công cụ DEF trong phần 4.2, nhóm công cụ GHK trong phần 4.3, v.v.~~

~~Trong phần~~ *~~Kết chương~~*~~, sinh viên đưa ra một số kết luận quan trọng của chương. Những vấn đề mở ra trong~~ *~~Tổng quan~~* ~~cần được tóm tắt lại nội dung và cách giải quyết/thực hiện như thế nào. Sinh viên lưu ý không viết~~ *~~Kết chương~~* ~~giống hệt~~ *~~Tổng quan~~*~~. Sau khi đọc phần~~ *~~Kết chương~~*~~, người đọc sẽ nắm được sơ bộ nội dung và giải pháp cho các vấn đề đã trình bày trong chương. Trong~~ *~~Kết chương~~*~~, Sinh viên nên có thêm câu liên kết tới chương tiếp theo.~~

~~Ví dụ về phần~~ *~~Kết chương~~*~~: Chương này đã phân tích chi tiết sáu nhóm công cụ tích hợp dữ liệu. Nhóm công cụ ABC và DEF thích hợp với những bài toán tích hợp dữ liệu phạm vi nhỏ. Trong khi đó, nhóm công cụ GHK lại chứng tỏ thế mạnh của mình với những bài toán cần độ chính xác cao, v.v. Từ kết quả nghiên cứu và phân tích về sáu nhóm công cụ tích hợp dữ liệu này, tôi đã thực hiện phát triển phần mềm tự động bóc tách và tích hợp dữ liệu sử dụng nhóm công cụ GHK. Phần này được trình bày trong chương tiếp theo – Chương 5.~~

# ­­Khảo sát và phân tích yêu cầu

~~Chương này có độ dài từ 9 đến 11 trang. Với đồ án nghiên cứu, sinh viên đổi tên chương thành “Tình hình nghiên cứu hiện nay” (Related works – State of the art) và phân tích rõ ngữ cảnh bài toán cũng như các kết quả nghiên cứu tương tự. Nếu là đồ án nghiên cứu, các đề mục cần thay đổi cho phù hợp. Sinh viên cần trao đổi kỹ với GV hướng dẫn để đưa ra được đề mục phù hợp nhất.~~

~~Với phương pháp phân tích thiết kế hướng đối tượng, sinh viên sử dụng biểu đồ use case theo hướng dẫn của template này. Với các phương pháp khác, sinh viên trao đổi với giáo viên hướng dẫn để đổi tên và sắp xếp lại đề mục cho phù hợp. Ví dụ, thay vì sử dụng biểu đồ use case, sinh viên đi theo hướng tiếp cận Agile có thể dùng User Story.~~

## Khảo sát hiện trạng

~~Thông thường, khảo sát chi tiết về hiện trạng và yêu cầu của phần mềm sẽ được lấy từ ba nguồn chính, đó là (i) người dùng/khách hàng, (ii) các hệ thống đã có, (iii) và các ứng dụng tương tự.~~

~~Sinh viên cần tiến hành phân tích, so sánh, đánh giá chi tiết ưu nhược điểm của các sản phẩm/nghiên cứu hiện có. Sinh viên có thể lập bảng so sánh nếu cần thiết. Kết hợp với khảo sát người dùng/khách hàng (nếu có), sinh viên nêu và mô tả sơ lược các tính năng phần mềm quan trọng cần phát triển.~~

Hiện trạng đang có những nghịch lý trong thu mua, phân phối nông sản.

**Nông sản: mua rẻ, bán đắt.** Không chỉ ở đô thị mà ngay tại các chợ nông thôn, người tiêu dùng hiện nay đang đối mặt với giá cả đắt đỏ như thời kỳ căng thẳng, lạm phát. Thông tin giá lúa tại ĐBSCL xuống thấp, thông tin trái cây xuống giá, heo hơi xuống giá là điều thật. Thế nhưng, người tiêu dùng vẫn phải gồng mình mua hàng hoá, nông sản trái cây với giá cao. Đơn cử như bưởi Năm Roi tại Mỹ Hoà, Bình Minh - Vĩnh Long, thương lái mua bưởi tại vườn giá 2.500 đồng/kg đến 3.000 đồng/kg, tuỳ theo bưởi loại 1 và loại 2. Điểm bất hợp lý nhất là giá bưởi bán tại vườn quá thấp, trong khi giá bưởi bán tại các siêu thị, của hàng trái cây giá cao gấp 2,3 lần giá bưởi tại vườn. Ngay tại Bình Minh, nơi cách xã bưởi Năm Roi Mỹ Hoà có 5-7 cây số, vậy mà cửa hàng bưởi năm roi Hoàng Gia bán bưởi loại 1 cao hơn giá bưởi tại vườn rất xa: 8.000 đồng/kg bưởi tốt.

**Phân phối nông sản: Quá nhiều trung gian.** Hệ thống phân phối lúa gạo tại nông thôn và đô thị ĐBSCL hiện nay thường tổ chức như sau: thương lái mua lúa trong vùng nông thôn, lúa này được họ cung ứng cho các đầu nậu cung ứng gạo xuất khẩu. Từ đầu mối này, các công ty xuất khẩu đặt hàng phục vụ cho những hợp đồng đã ký.

Cứ mỗi lần qua trung gian, giá gạo tăng thêm một ít, hoạt động buôn bán gạo diễn ra quanh năm theo thông lệ. Hệ thống phân phối nhiều tầng nấc này, nông dân bán hàng giá thấp, khoản lợi nhuận từ xuất khẩu đã rơi vào tay thương lái và doanh nghiệp xuất khẩu. Riêng tại các chợ ở đồng bằng sông Cửu Long, cách phân phối lúa gạo cũng theo cách tương tự: thương lái mua lúa về xay xát và gân gạo lại cho các đại lý và những chủ bán gạo lẻ, từ đâu gạo mới được bán cho người tiêu dùng. Quá trình này làm cho người sản xuất và người tiêu dùng thiệt thòi nhất.

**Vấn đề đầu ra cho sản phẩm nông nghiệp** đang là một khó khăn rất lớn cho sản xuất nông nghiệp; điều dễ nhận thấy nhất đó là hiện tượng được mùa - rớt giá. Nguyên do chính là do các hộ gia đình nông dân không có sự liên kết với nhau, vốn vay được bao nhiêu thì làm bấy nhiêu, mạnh ai nấy làm

Nông dân cần phải biết rõ sản phẩm của mình sẽ bán ở đâu, phương thức thế nào, giá cả, khối lượng bao nhiêu, yêu cầu chất lượng ra sao. Đây là vấn đề rất quan trọng, thiết thực để nông dân chủ động hoạch định và tổ chức sản xuất

Người nông dân cần phải nắm bắt được thông tin thị trường, nhu cầu thị trường cần gì, từ đó có thể chủ động chọn cây trồng, chọn giống và vụ mùa phù hợp, liên kết được với nhau trong sản xuất để khắc phục tình trạng bất cập: sản xuất nhỏ lẻ, phân tán, chuyển đổi chậm, không đồng bộ

## Tổng quan chức năng

~~Phần 2.2 này có nhiệm vụ tóm tắt các chức năng của phần mềm. Trong phần này, sinh viên lưu ý chỉ mô tả chức năng mức cao (tổng quan) mà không đặc tả chi tiết cho từng chức năng. Đặc tả chi tiết được trình bày trong phần 2.3.~~

### Biểu đồ use case tổng quan

~~Sinh viên vẽ biểu đồ use case tổng quan và giải thích các tác nhân tham gia là gì, nêu vai trò của từng tác nhân, và mô tả ngắn gọn các use case chính.~~

### Biểu đồ use case phân rã XYZ

~~Với mỗi use case mức cao trong biểu đồ use case tổng quan, sinh viên tạo một mục riêng như mục 2.2.2 và tiến hành phân rã use case đó. Lưu ý tên use case cần phân rã trong biểu đồ use case tổng quan phải khớp với tên đề mục.~~

~~Trong mỗi mục như vậy, sinh viên vẽ và giải thích ngắn gọn các use case phân rã.~~

### Quy trình nghiệp vụ

~~Nếu sản phẩm/hệ thống cần xây dựng có quy trình nghiệp vụ quan trọng/đáng chú ý, sinh viên cần mô tả và vẽ biểu đồ hoạt động minh họa quy trình nghiệp vụ đó. Sinh viên lưu ý đây không phải là luồng sự kiện của~~ *~~từng use case~~*~~, mà là luồng hoạt động kết hợp nhiều use case để thực hiện một nghiệp vụ nào đó.~~

~~Ví dụ, một hệ thống quản lý thư viện có quy trình nghiệp vụ mượn trả với mô tả sơ bộ như sau: Sinh viên làm thẻ mượn, sau đó sinh viên đăng ký mượn sách, thủ thư cho mượn, và cuối cùng sinh viên trả lại sách cho thư viện. Một hệ thống có thể có một vài quy trình nghiệp vụ quan trọng như vậy.~~

## Đặc tả chức năng

~~Sinh viên lựa chọn từ 4 đến 7 use case quan trọng nhất của đồ án để đặc tả chi tiết. Mỗi đặc tả bao gồm ít nhất các thông tin sau: (i) Tên use case, (ii) Luồng sự kiện (chính và phát sinh), (iii) Tiền điều kiện, và (iv) Hậu điều kiện. Sinh viên chỉ vẽ bổ sung biểu đồ hoạt động khi đặc tả use case phức tạp.~~

### Đặc tả use case A

### Đặc tả use case B

## Yêu cầu phi chức năng

~~Trong phần này, sinh viên đưa ra các yêu cầu khác nếu có, bao gồm các yêu cầu phi chức năng như hiệu năng, độ tin cậy, tính dễ dùng, tính dễ bảo trì, hoặc các yêu cầu về mặt kỹ thuật như về CSDL, công nghệ sử dụng, v.v.~~

Tính dễ dùng:

* Giao diện đơn giản dễ nhìn, không rối mắt
* Màu sắc tươi tắn hài hòa, bắt mắt
* Các ô nhập liệu luôn luôn có gợi ý để người dùng xác định được là sẽ nhập thông tin gì
* Trong trường hợp người dùng nhập sai, nhập thiếu dữ liệu, thông điệp cảnh báo sẽ hiện lên. Ngoài ra nội dung cảnh báo cũng sẽ mô tả chi tiết về lỗi và cách sửa lỗi giúp người dùng dễ xác định

# Công nghệ sử dụng

~~Chương này có độ dài từ 10 đến 13 trang. Nếu cần trình bày dài hơn, sinh viên đưa vào phần phụ lục. Chú ý đây là kiến thức đã có sẵn; SV sau khi tìm hiểu được thì phân tích và tóm tắt lại. Sinh viên không trình bày dài dòng, chi tiết.~~

~~Với đồ án ứng dụng, sinh viên để tên chương là “Công nghệ sử dụng”. Trong chương này, sinh viên giới thiệu về các công nghệ, nền tảng sử dụng trong đồ án. Sinh viên cũng có thể trình bày thêm nền tảng lý thuyết nào đó nếu cần dùng tới.~~

~~Với đồ án nghiên cứu, sinh viên đổi tên chương thành “Cơ sở lý thuyết”. Khi đó, nội dung cần trình bày bao gồm: Kiến thức nền tảng, cơ sở lý thuyết, các thuật toán, phương pháp nghiên cứu, v.v.~~

~~Với từng công nghệ/nền tảng/lý thuyết được trình bày, sinh viên phải phân tích rõ công nghệ/nền tảng/lý thuyết đó dùng để để giải quyết vấn đề/yêu cầu cụ thể nào ở Chương 2. Hơn nữa, với từng vấn đề/yêu cầu, sinh viên phải liệt kê danh sách các công nghệ/hướng tiếp cận tương tự có thể dùng làm lựa chọn thay thế, rồi giải thích rõ sự lựa chọn của mình.~~

~~Lưu ý: Nội dung ĐATN phải có tính chất liên kết, liền mạch, và nhất quán. Vì vậy, các công nghệ/thuật toán trình bày trong chương này phải khớp với nội dung giới thiệu của sinh viên ở phần 1.3.~~

~~Trong chương này, để tăng tính khoa học và độ tin cậy, sinh viên nên chỉ rõ nguồn kiến thức mình thu thập được ở tài liệu nào, đồng thời đưa tài liệu đó vào trong danh sách tài liệu tham khảo rồi tạo các tham chiếu chéo (xem hướng dẫn ở phụ lục A.7).~~

## Front End

### Các công nghệ được sử dụng

Có rất nhiều framework dùng để dựng giao diện trang web đẹp như PHP, React, Vue.js, Angular, vv... Nhóm chúng em chọn sử dụng các công nghệ sau:

* Ngôn ngữ đánh dấu siêu văn bản HyperText Markup Language (HTML).
* Định danh tài nguyên đồng nhất Uniform Resource Identifier (URI hay phổ biến hơn là URL).
* Phương thức giao tiếp siêu văn bản HyperText Transfer Protocol (HTTP).
* JavaScript: Được sử dụng để tạo ra những trang web tương tác. Nó được tích hợp và nhúng trong HTML. JavaScript cho phép kiểm soát các hành vi của trang web tốt hơn so với khi chỉ có một mình HTML. Với JavaScript, một website sẽ trở nên sinh động hơn với các trải nghiệm động, tăng tính tương tác, tăng thêm tính năng cho website.
* jQuery: là một thư viện JavaScript nhanh, nhỏ, đa nền tảng và giàu tính năng. Nó được thiết kế để đơn giản hóa client-side scripting của HTML. Mục đích chính của jQuery là làm việc sử dụng JavaScript trên trang web trở nên dễ dàng hơn, giúp website có tính tương tác và hấp dẫn hơn.
* Bootstrap là một framework cho phép thiết kế website responsive nhanh hơn và dễ dàng hơn. Bootstrap bao gồm các HTML templates, CSS templates và Javascript tạo ra những cái cơ bản có sẵn như: typography, forms, buttons, tables, navigation, modals, image carousels và nhiều thứ khác. Trong bootstrap có thêm các plugin JavaScript trong nó. Giúp cho việc thiết kế responsive dễ dàng hơn và nhanh chóng hơn.

### Ứng dụng vào website

Khi thiết kế layout cho website, Bootstrap trở nên cực kì hữu dụng vì nó cung cấp khả năng responsive, giúp không bị bể giao diện khi thu phóng to nhỏ màn hình. Ngoài ra Bootstrap cũng giúp rất nhiều về việc tăng cao tính thẩm mỹ cho website khi được sử dụng để thiết kế nút, slideshow, vv...

Để dựng nên một biểu đồ dựa trên database thì không thể nào thiếu ChartJS, đó là một thư viện được viết bằng JavaScript. JavaScript hỗ trợ rất nhiều tính năng khác như upload ảnh, vv... JavaScript là thứ không thể thiếu đối với hầu hết các trang web hiện nay.

## Back End

### Các công nghệ được sử dụng

Những công nghệ có thể đáp ứng được nhu cầu xử lý back-end:

* PHP:  Sở hữu một cộng đồng nguồn mở rất tích cực và có thể hoạt động liên tục trên nhiều nền tảng như UNIX, Max và Windows.
* JavaScript:  là ngôn ngữ lập trình phổ biến nhất thế giới trong 20 năm qua.
* NodeJS: là một nền tảng chạy trên môi trường V8 JavaScript runtime - một trình thông dịch JavaScript cực nhanh chạy trên trình duyệt Chrome.
* Java: từ lâu ngôn ngữ này đã được mệnh danh là loại ngôn ngữ lập trình đa năng. Java được sử dụng cho việc phát triển phần mềm trên desktop, web và android.

Nhóm chúng em lựa chọn nền tảng ASP.NET và C# làm ngôn ngữ lập trình chính cho phần xử lý back-end của mình.

**ASP.NET** là một nền tảng ứng dụng web (*web application framework*) được phát triển và cung cấp bởi [Microsoft](https://vi.wikipedia.org/wiki/Microsoft), cho phép những người lập trình tạo ra những trang web động, những [ứng dụng web](https://vi.wikipedia.org/wiki/%E1%BB%A8ng_d%E1%BB%A5ng_web) và những [dịch vụ web](https://vi.wikipedia.org/wiki/D%E1%BB%8Bch_v%E1%BB%A5_web). Lần đầu tiên được đưa ra thị trường vào tháng 1 năm 2002 cùng với phiên bản 1.0 của.NET framework, là công nghệ nối tiếp của Microsoft's Active Server Pages(ASP). ASP.NET được biên dịch dưới dạng [Common Language Runtime](https://vi.wikipedia.org/w/index.php?title=Common_Language_Runtime&action=edit&redlink=1) (CLR), cho phép những người lập trình viết mã ASP.NET với bất kỳ ngôn ngữ nào được hỗ trợ bởi.NET language. Đây là một framework rất đáng được sử dụng vì:

* **SoC:** Một trong những lợi ích chính được đưa ra bởi ASP.NET MVC đó là Separation of Concern (phân tách mối bận tâm). Framework này cung cấp sự phân tách rất rõ ràng giữa model, logic nghiệp vụ, giao diện người dùng và dữ liệu.
* **Control:** Khi sử dụng framework ASP.NET MVC, người dùng được cung cấp một bộ control rất mạnh mẽ trên JavaScript, HTML và CSS so với các control được cung cấp bởi một số hình thức truyền thống trên web.
* **Tích hợp:** Khi tích hợp MVC với jQuery, người dùng có thể viết code của mình chạy trong các trình duyệt web. Điều này sẽ giúp giảm tải cho các web server.

**Ngôn ngữ lập trình C#** là một [ngôn ngữ lập trình](https://vi.wikipedia.org/wiki/Ng%C3%B4n_ng%E1%BB%AF_l%E1%BA%ADp_tr%C3%ACnh) [hướng đối tượng](https://vi.wikipedia.org/wiki/L%E1%BA%ADp_tr%C3%ACnh_h%C6%B0%E1%BB%9Bng_%C4%91%E1%BB%91i_t%C6%B0%E1%BB%A3ng) được phát triển bởi [Microsoft](https://vi.wikipedia.org/wiki/Microsoft), là phần khởi đầu cho kế hoạch [.NET](https://vi.wikipedia.org/w/index.php?title=.NET&action=edit&redlink=1) của họ. C# được miêu tả là ngôn ngữ có được sự cân bằng giữa C++, [Visual Basic](https://vi.wikipedia.org/wiki/Visual_Basic), [Delphi](https://vi.wikipedia.org/wiki/Delphi_(ng%C3%B4n_ng%E1%BB%AF_l%E1%BA%ADp_tr%C3%ACnh)) và [Java](https://vi.wikipedia.org/wiki/Java_(ng%C3%B4n_ng%E1%BB%AF_l%E1%BA%ADp_tr%C3%ACnh)).

* Được phát triển bởi đội ngũ kỹ sư của Microsoft vào năm 2000, trong đó người dẫn đầu là Anders Hejlsberg và Scott Wiltamuth.
* C# là ngôn ngữ lập trình hiện đại, hướng đối tượng và nó được xây dựng trên nền tảng của hai ngôn ngữ mạnh nhất là C++ và Java.
* C# được thiết kế cho Common Language Infrastructure (CLI), mà gồm Executable Code và Runtime Environment, cho phép chúng ta sử dụng các ngôn ngữ high-level đa dạng trên các nền tảng và cấu trúc máy tính khác nhau.

C# có rất nhiều ưu điểm nổi bật:

* C# có cấu trúc khá gần gũi với các ngôn ngữ lập trình truyền thống, nên cũng khá dễ dàng tiếp cận và học nhanh với C#.
* C# có thể biên dịch trên nhiều nền tảng máy tính khác nhau.
* C# được xây dựng trên nền tảng của C++ và Java nên nó được thừa hưởng những ưu điểm của ngôn ngữ đó.
* C# là một phần của .NET Framework nên được sự chống lưng khá lớn đến từ bộ phận này.
* C# có IDE Visual Studio cùng nhiều plug-in vô cùng mạnh mẽ.

### Ứng dụng vào website

Được sử dụng ở cả ba lớp trong mô hình MVC, C# chính là thành phần cốt lõi khi xây dựng backend cho website

Ở hầu hết mọi website, trang chủ luôn phức tạp nhất vì phải hiển thị nhiều thông tin từ nhiều bảng khác nhau (danh sách chào mua, danh sách chào bán, top nông sản hot nhất, vv...), và biểu đồ. Vấn đề này có thể giải quyết bằng cách tạo một Model chứa tất cả những thông tin cần hiển thị, sau đó nhận dữ liệu từ Controller rồi hiển thị lên View.

Phần xử lý được chia ra làm hai luồng. Luồng thứ nhất là xử lý về dữ liệu gồm các thao tác như truy xuất, tạo mới, chỉnh sửa, xóa dữ liệu, được thực hiện ở tầng DAO (Data Access Layer). Ví dụ như truy xuất danh sách nhu cầu bán của chủ nông trại, rồi dựa vào danh sách đó truy xuất được những nhu cầu mua phù hợp từ phía thương lái, vv... Luồng thứ hai là xử lý luồng nghiệp vụ, các thao tác của người dùng, được thực hiện ở Controller

Phần View có định dạng là .cshtml, nhờ vậy có thể sử dụng C# thông qua Razor. Ở tầng View, C# dùng vòng lặp foreach để hiển thị các danh sách dữ liệu truy xuất được từ tầng DAO hiển thị lên View.

## Data Base

### Các công nghệ được sử dụng

Có rất nhiều các hệ quản trị cơ sở dữ liệu:

* MongoDB (Document Store): là cơ sở dữ liệu hướng tài liệu, một thiết kế riêng biệt cho việc lưu trữ tài liệu dạng văn kiện JSON, BSON hoặc XML. Vì là cấu trúc dữ liệu không ràng buộc khác với SQL, CSDL này không đòi hỏi người dùng tự tạo bảng nhập liệu trước khi nhập dữ liệu vào.
* MySQL: [hệ quản trị cơ sở dữ liệu](https://vi.wikipedia.org/wiki/H%E1%BB%87_qu%E1%BA%A3n_tr%E1%BB%8B_c%C6%A1_s%E1%BB%9F_d%E1%BB%AF_li%E1%BB%87u) [tự do nguồn mở](https://vi.wikipedia.org/wiki/Ph%E1%BA%A7n_m%E1%BB%81m_ngu%E1%BB%93n_m%E1%BB%9F) phổ biến nhất thế giới và được các nhà phát triển rất ưa chuộng trong quá trình phát triển ứng dụng.
* Redis (Key – Value Store): là loại lưu trữ đơn giản nhất trong các loại CSDL NoSQL đồng thời nó cũng là kiểu lưu trữ cho tất cả các hệ quản trị CSDL NoSQL.

Nhóm chúng em lựa chọn sử dụng SQL Server (hay còn gọi là Microsoft SQL Server, viết tắt là MS SQL Server**)**. Đây là một phần mềm được phát triển bởi Microsoft dùng để lưu trữ dữ liệu dựa trên chuẩn RDBMS, và nó cũng là một hệ quản trị cơ sở dữ liệu quan hệ đối tượng (ORDBMS).

SQL Server cung cấp đầy đủ công cụ để quản lý, từ giao diện GUI cho đến việc sử dụng ngôn ngữ truy vấn SQL. Ngoài ra điểm mạnh của nó là Microsoft có khá nhiều nền tảng kết hợp hoàn hảo với SQL Server như ASP.NET, C# xây dựng Winform, bởi vì nó hoạt động hoàn toàn độc lập. SQL Server có những tính năng giúp việc sử dụng hiệu quả hơn như:

* Cho phép tạo nhiều cơ sở dữ liệu
* Duy trì lưu trữ bền vững
* Bảo mật cao
* Phân tích  dữ liệu bằng SSAS - SQL Server Analysis Services
* Tạo được báo cáo bằng SSRS - SQL Server Reporting Services
* Thực hiện quá trình ETL (Extract-Transform-Load) bằng SSIS - SQL Server Integration Services

### Ứng dụng vào website

SQL Server thực sự là một trợ thủ đắc lực khi cung cấp rất nhiều tính năng tiện lợi khác ngoài việc lưu trữ dữ liệu.

Khi truy xuất nhiều bảng liên thông với nhau, SQL Server hỗ trợ tạo một Database Diagrams gồm tất cả hay một vài bảng chỉ định, giúp chúng em có thể thấy được một cách trực quan các mối quan hệ giữa các bảng, các khóa chính, khóa phụ, các trường dữ liệu, vv...

Ngoài ra, Store Procedure, Trigger, Job được sử dụng để hỗ trợ việc đảm bảo tính đúng đắn của dữ liệu, thực hiện tác vụ,  tự động xử lý một công việc tại thời điểm ấn định, vv

# Phát triển và triển khai ứng dụng

## Thiết kế kiến trúc

### Lựa chọn kiến trúc phần mềm

~~Mục này có độ dài từ một đến ba trang. Sinh viên cần lựa chọn kiến trúc phần mềm cho ứng dụng của mình như: kiến trúc ba lớp MVC, MVP, SOA, Microservice, v.v. rồi giải thích sơ bộ về kiến trúc đó (không giải thích chi tiết/dài dòng).~~

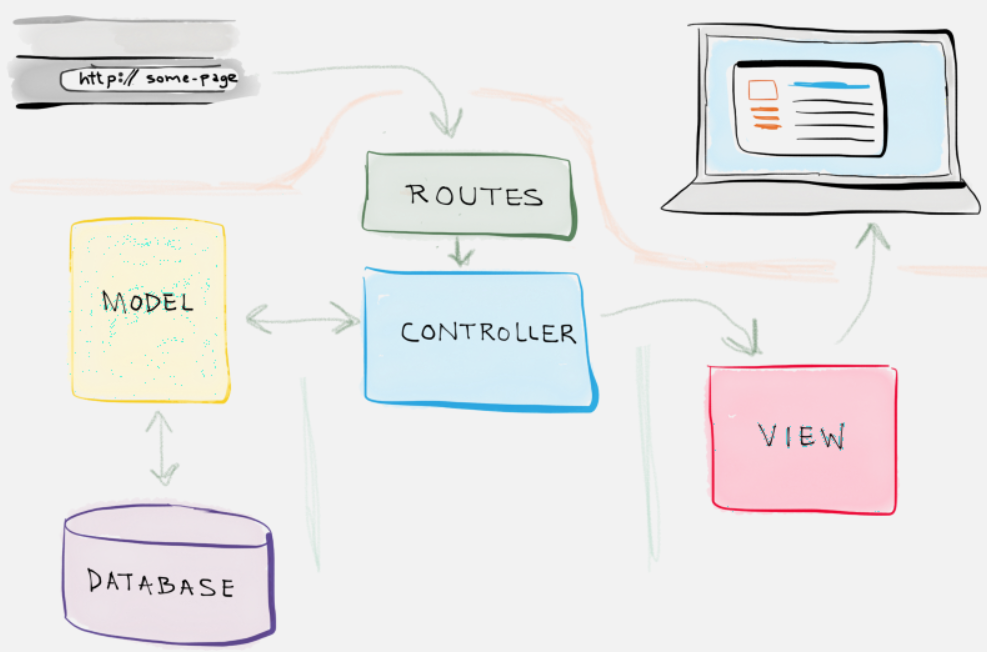
~~Sử dụng kiến trúc phần mềm đã chọn ở trên, sinh viên mô tả kiến trúc cụ thể cho ứng dụng của mình. Gợi ý: sinh viên áp dụng lý thuyết chung vào hệ thống/sản phẩm của mình như thế nào, có thay đổi, bổ sung hoặc cải tiến gì không. Ví dụ, thành phần M trong kiến trúc lý thuyết MVC sẽ là~~ *~~những thành phần cụ thể nào~~* ~~(ví dụ: là interface I + class C1 + class C2, v.v.) trong kiến trúc phần mềm của sinh viên.~~

**MVC** (*viết tắt của* **M**odel-**V**iew-**C**ontroller) là một mẫu kiến trúc phần mềm  để tạo lập giao diện người dùng trên máy tính. MVC chia một ứng dụng thành ba phần tương tác được với nhau để tách biệt giữa cách thức mà thông tin được xử lý nội hàm và phần thông tin được trình bày và tiếp nhận từ phía người dùng.

Khi sử dụng đúng cách, mẫu *MVC* giúp cho người phát triển phần mềm cô lập các nguyên tắc nghiệp vụ và giao diện người dùng một cách rõ ràng hơn. Phần mềm phát triển theo mẫu *MVC* tạo nhiều thuận lợi cho việc bảo trì vì các nguyên tắc nghề nghiệp và giao diện ít liên quan với nhau.

Trong mẫu *Model-View-Controller*:

* Mô hình (*model*) tượng trưng cho [dữ liệu](https://vi.wikipedia.org/wiki/D%E1%BB%AF_li%E1%BB%87u) của chương trình phần mềm.
* Tầm nhìn hay khung nhìn (*view*) bao gồm các thành phần của giao diện người dùng.
* Bộ kiểm tra hay bộ điều chỉnh (*controller*) quản lý sự trao đổi giữa dữ liệu và các nguyên tắc nghề nghiệp trong các thao tác liên quan đến mô hình.

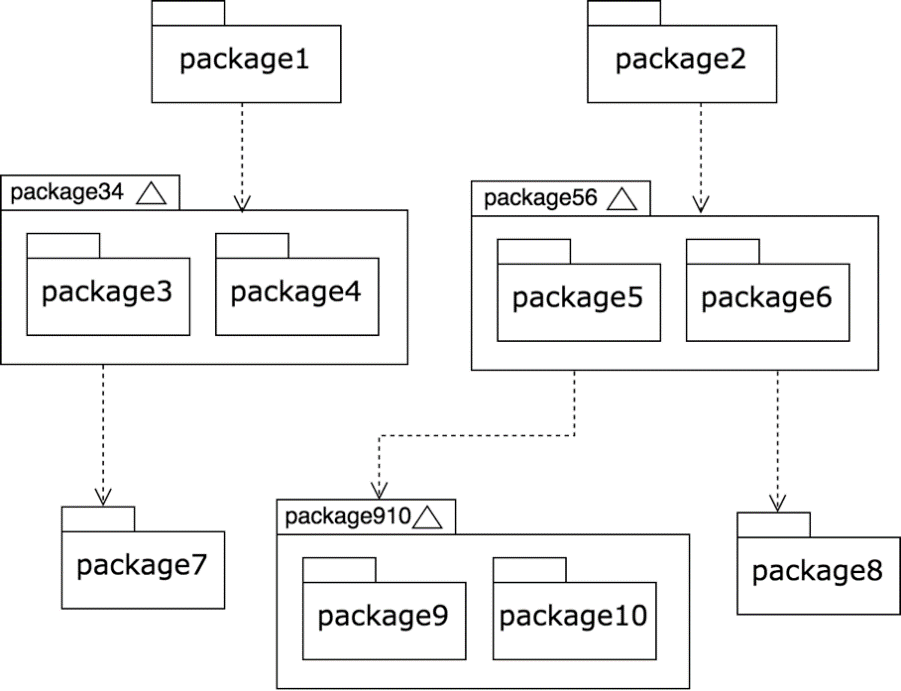


Sử dụng mẫu MVC đã chọn ở trên, kiến trúc phần mềm của bọn em chia ra làm ba vùng (area) chính là Admin, Farmer, Trader. Mỗi vùng này hoạt động dựa trên mẫu MVC và có kiến trúc như sau:

* Model: bao gồm các Model và DTO.
* View: bao gồm các khung nhìn Index, Create, Edit, Delete, Login, vv…
* Controller: chia ra làm hai phần. Phần thứ nhất là tầng DAO chỉ xử lý các yêu cầu truy xuất dữ liệu và các thao tác khác như thêm, sửa, xóa trên database. Phần thứ hai là tầng Controller quản lý các thao tác liên quan đến Model.

### Thiết kế tổng quan

~~Sinh viên vẽ biểu đồ gói UML (UML package diagram), nêu rõ sự phụ thuộc giữa các gói (package). SV cần vẽ các gói sao cho chúng được phân theo các tầng rõ ràng, không được sắp đặt package lộn xộn trong hình vẽ. Sinh viên chú ý các quy tắc thiết kế (Các gói không phụ thuộc lẫn nhau, gói tầng dưới không phụ thuộc gói tầng trên, không phụ thuộc bỏ qua tầng, v.v.) và cần giải thích sơ lược về mục đích/nhiệm vụ của từng package. SV tham khảo ví dụ minh họa trong Hình 1.~~



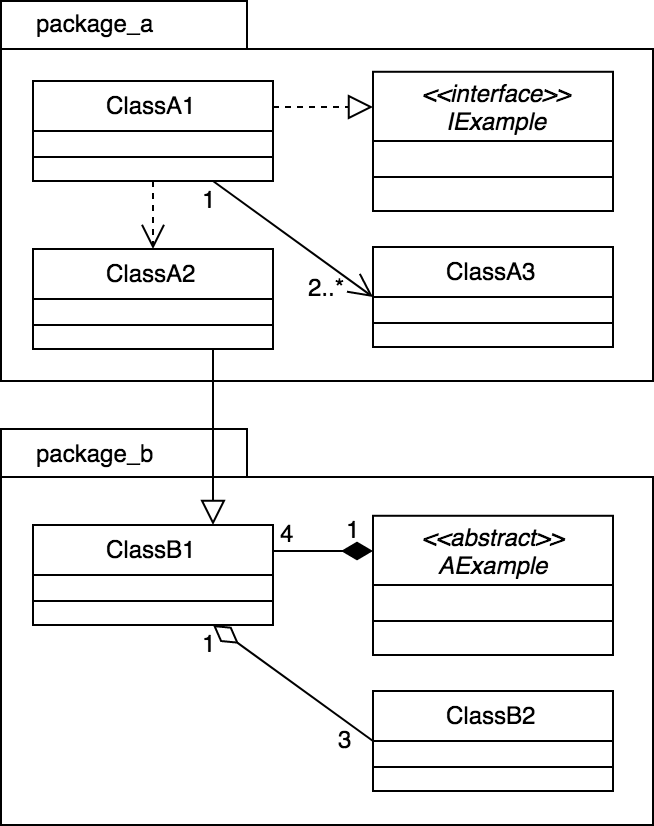
**Hình 1** Ví dụ biểu đồ phụ thuộc gói

### Thiết kế chi tiết gói

~~Sinh viên thiết kế và lần lượt vẽ biểu đồ thiết kế cho từng package, hoặc một nhóm các package liên quan để giải quyết một vấn đề gì đó. Khi vẽ thiết kế gói, sinh viên chỉ cần đưa tên lớp, không cần chỉ ra các thành viên phương thức và thuộc tính. SV tham khảo ví dụ minh họa trong Hình 2.~~

~~Sinh viên cần vẽ rõ ràng quan hệ giữa các lớp trong biểu đồ. Các quan hệ bao gồm: phụ thuộc (dependency), kết hợp (association), kết tập (aggregation), hợp thành (composition), kế thừa (inheritance), và thực thi (implementation). Các quan hệ này đều đã được minh họa trong Hình 2.~~

~~Sau khi vẽ hình minh họa, sinh viên cần giải thích ngắn gọn về thiết kế của mình.~~



**Hình 2** Ví dụ thiết kế gói

## Thiết kế chi tiết

### Thiết kế giao diện

~~Phần này có độ dài từ hai đến ba trang. Sinh viên đặc tả thông tin về màn hình mà ứng dụng của mình hướng tới, bao gồm độ phân giải màn hình, kích thước màn hình, số lượng màu sắc hỗ trợ, v.v. Tiếp đến, sinh viên đưa ra các thống nhất/chuẩn hóa của mình khi thiết kế giao diện như thiết kế nút, điều khiển, vị trí hiển thị thông điệp phản hồi, phối màu, v.v. Sau cùng sinh viên đưa ra một số hình ảnh minh họa thiết kế giao diện cho các chức năng quan trọng nhất. Lưu ý, sinh viên không nhầm lẫn giao diện thiết kế với giao diện của sản phẩm sau cùng.~~

#### Tiêu chuẩn đối với các màn hình

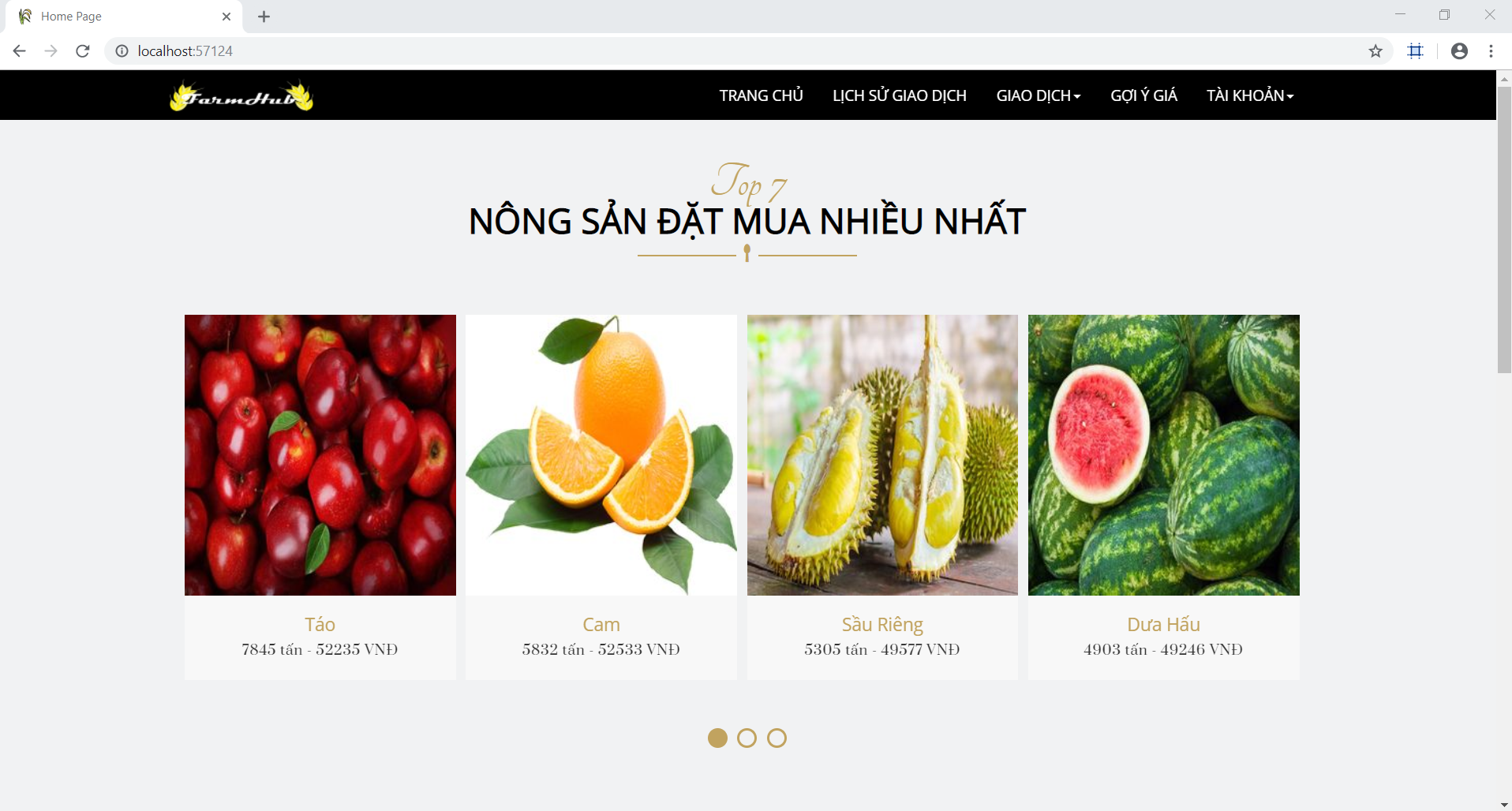
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Yếu tố | Kích thước | Canh lề | Cách tổ chức | Yêu cầu kết xuất |
| Màn hình trang chủ | 1920 x 1080 | 180px |  |  |
| Màn hình danh sách | 1920 x 1080 | 180px |  |  |
| Màn hình chi tiết | 1920 x 1080 | 180px |  |  |
| Màn hình chỉnh sửa | 1920 x 1080 | 180px |  |  |
| Màn hình tạo mới | 1920 x 1080 | 180px |  |  |

#### Tiêu chuẩn đối với các yếu tố trên màn hình

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Yếu tố | Font type | Font size | Font color | Canh lề | Kích thước | Hình dạng |
| Tiêu đề form | Tangerine | 50px | #c1a35f | center | 1140px 40px |  |
| Label | Open Sans | 35px | #fff | center | 1140px 40px |  |
| Button | Open Sans | 16px | #fff | 15px 35px | 135px 56px | Chữ nhật |
| Ô nhập liệu | Prata | 14px | #fff | 6px 12px | 455px 34px | Chữ nhật |

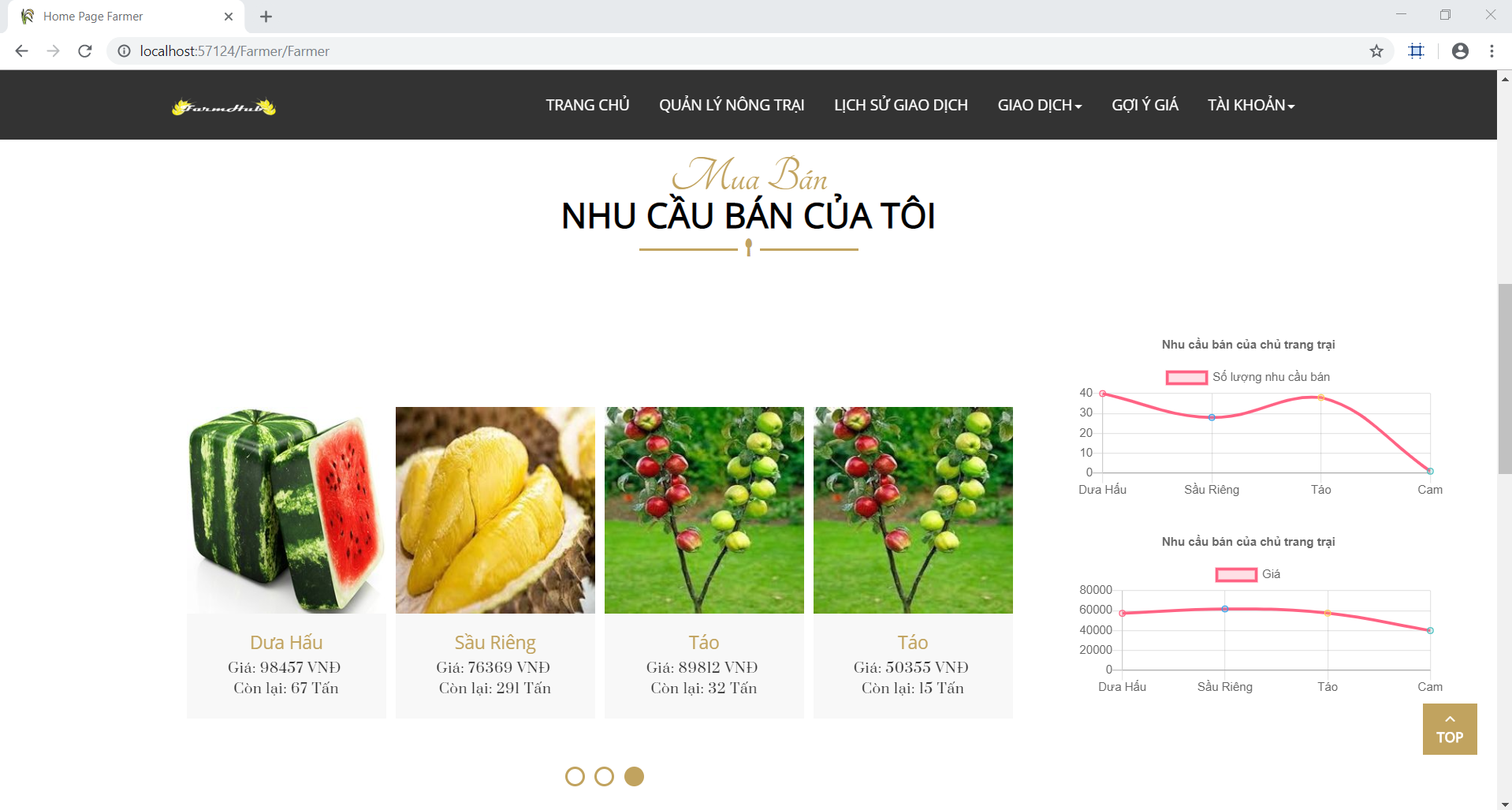
#### Giao diện chi tiết

* Màn hình trang chủ
  + Tên màn hình: Home Page
  + Ý nghĩa: Nguời dùng truy cập vào website sẽ thấy trang này đầu tiên
  + Hình ảnh:



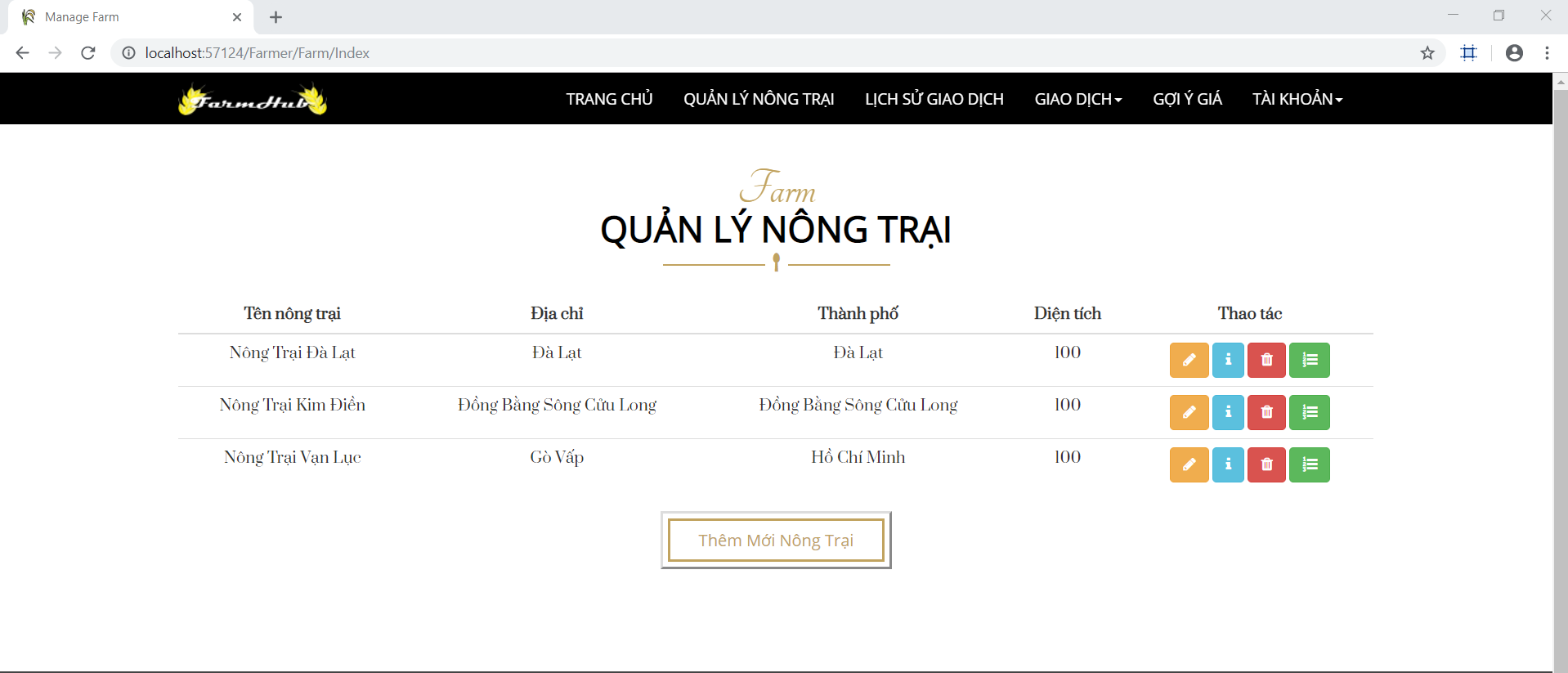
**Hình x** Home Page

* Màn hình trang chủ của Chủ Nông trại
  + Tên màn hình: Farmer Home Page
  + Ý nghĩa:
  + Hình ảnh:



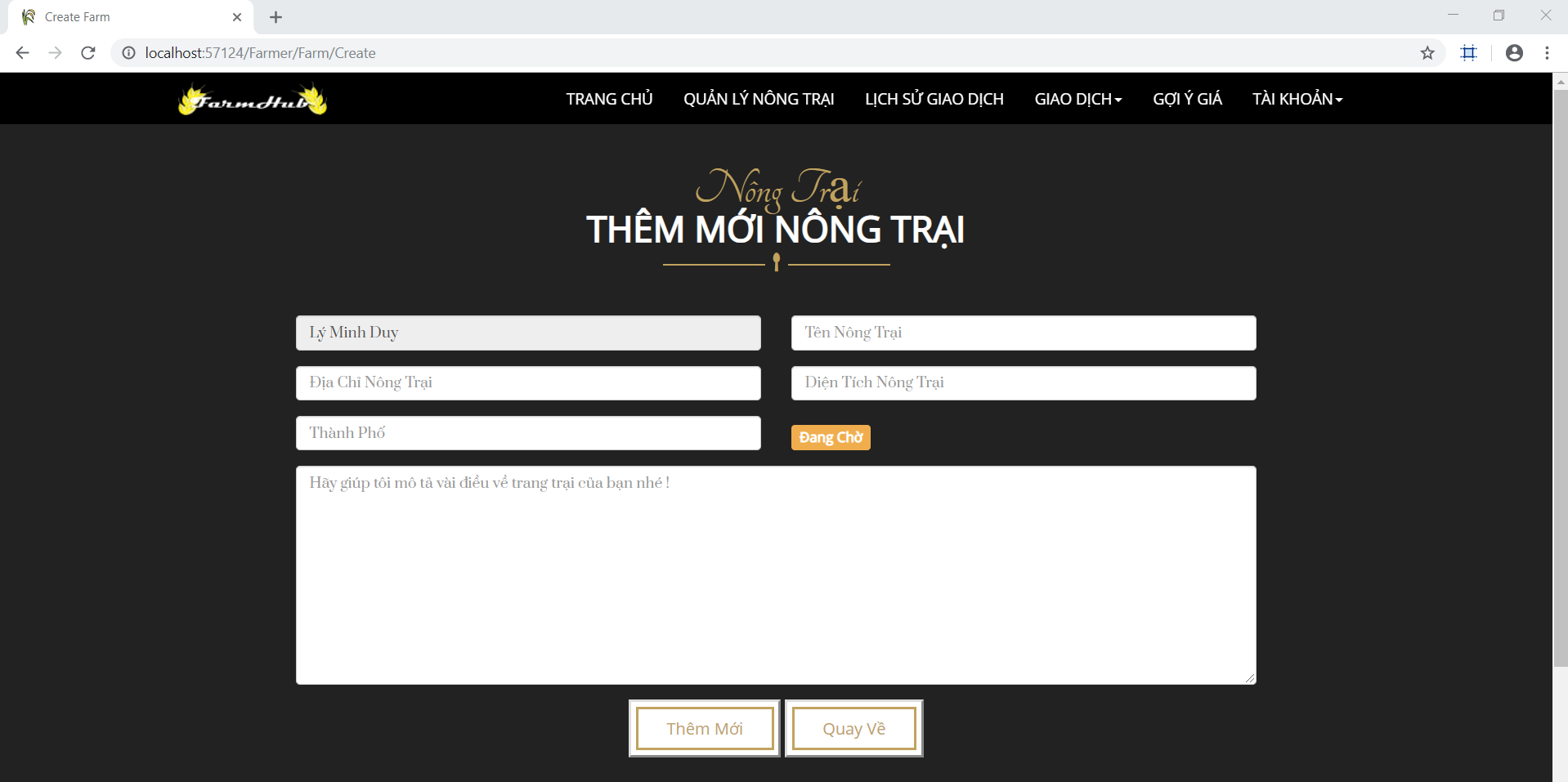
**Hình x** Farmer Home Page

* Màn hình quản lý nông trại
  + Tên màn hình: Manage Farm
  + Ý nghĩa:
  + Hình ảnh:



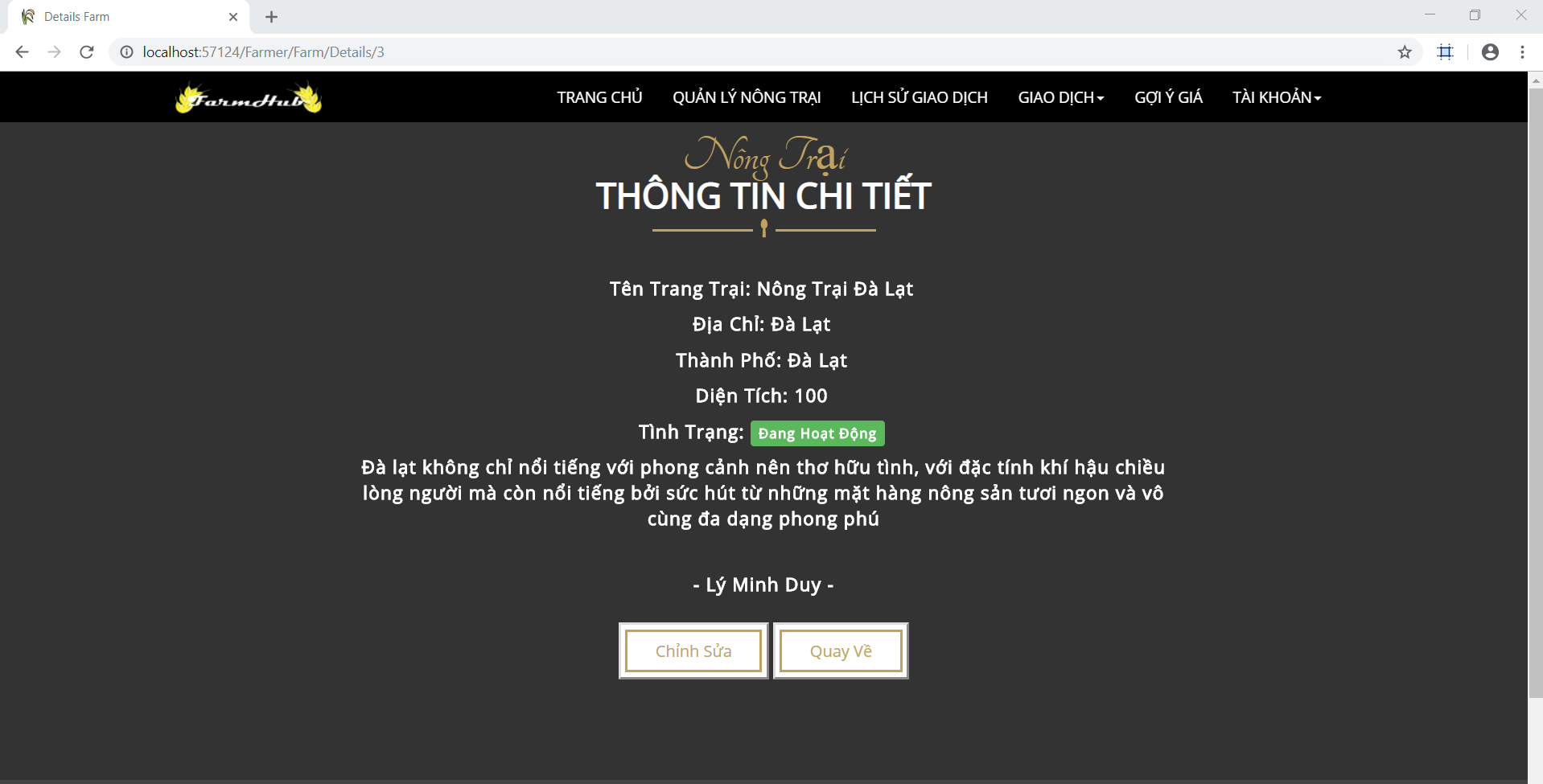
**Hình x** Manage Farm

* Màn hình thêm mới nông trại
  + Tên màn hình: Create Farm
  + Ý nghĩa: Đăng ký thông tin một nông trại mới
  + Hình ảnh:



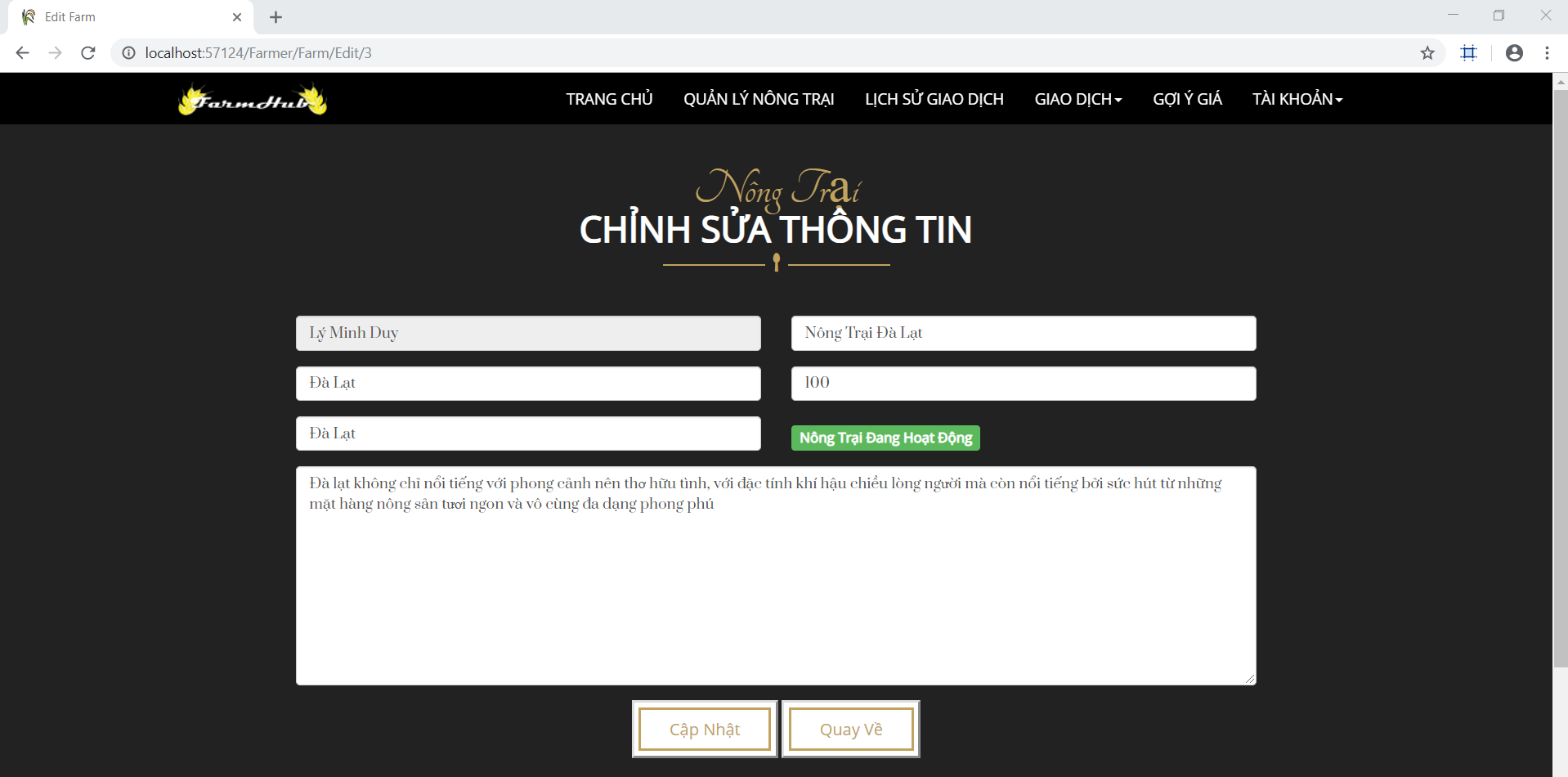
**Hình x** Create Farm

* Màn hình xem chi tiết nông trại
  + Tên màn hình: Details Farm
  + Ý nghĩa: Xem thông tin chi tiết của nông trại
  + Hình ảnh:



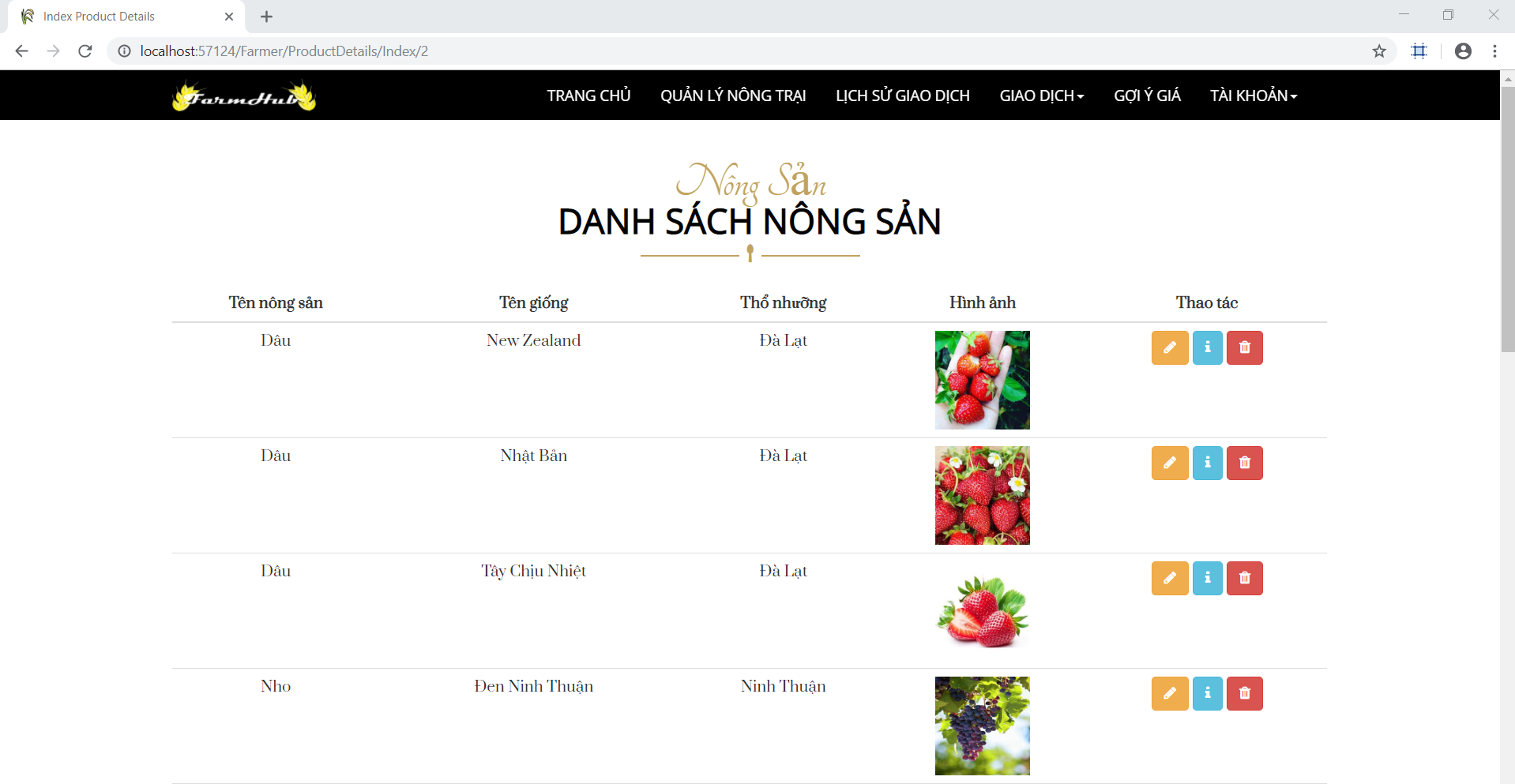
**Hình x** Details Farm

* Màn hình sửa chi tiết nông trại
  + Tên màn hình: Edit Farm
  + Ý nghĩa: Chỉnh sửa thông tin nông trại
  + Hình ảnh:



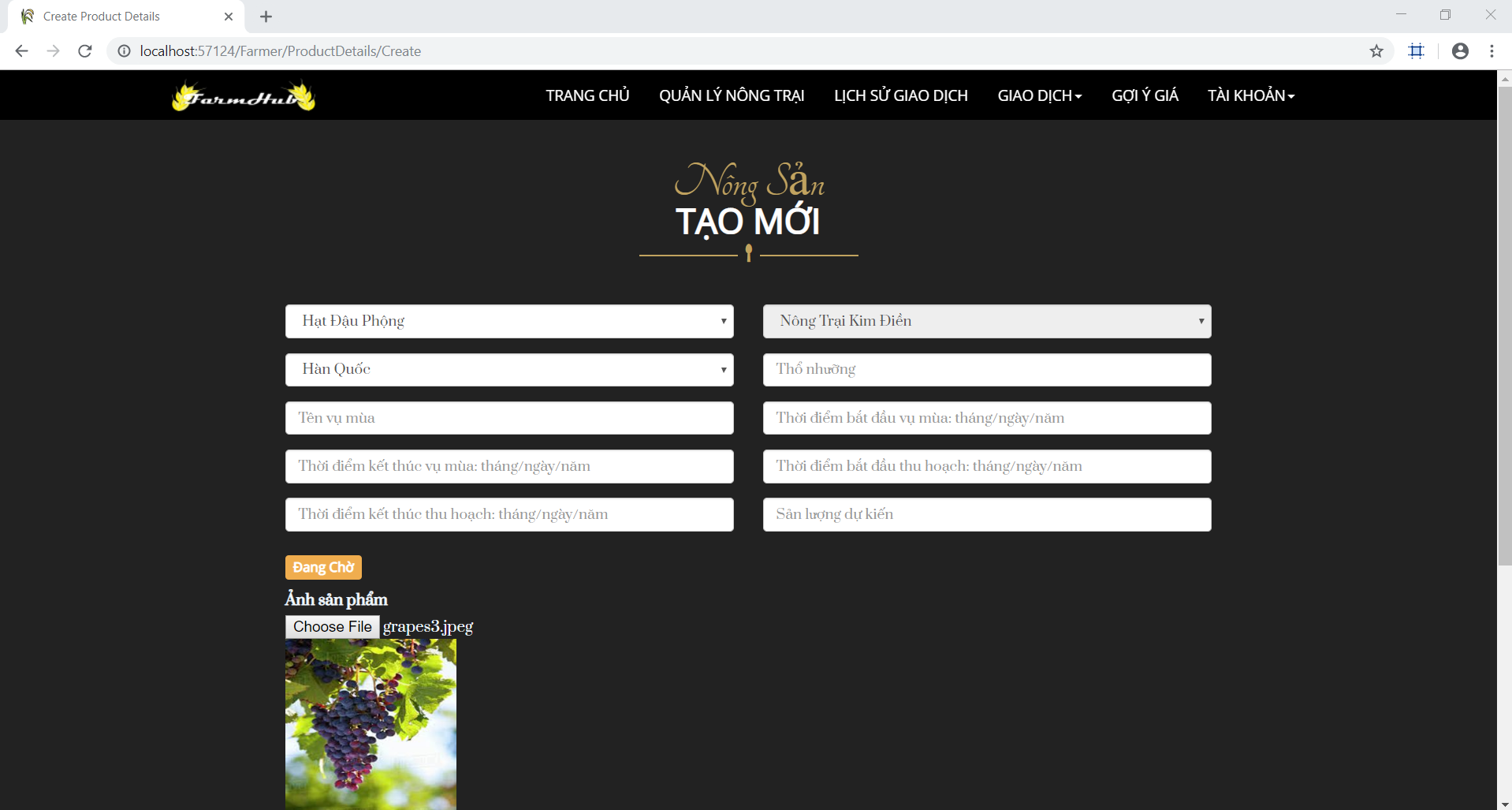
**Hình x** Edit Farm

* Màn hình quản lý nông sản
  + Tên màn hình: Index Product Details
  + Ý nghĩa: Danh sách những nông sản trồng ở một nông trại
  + Hình ảnh:



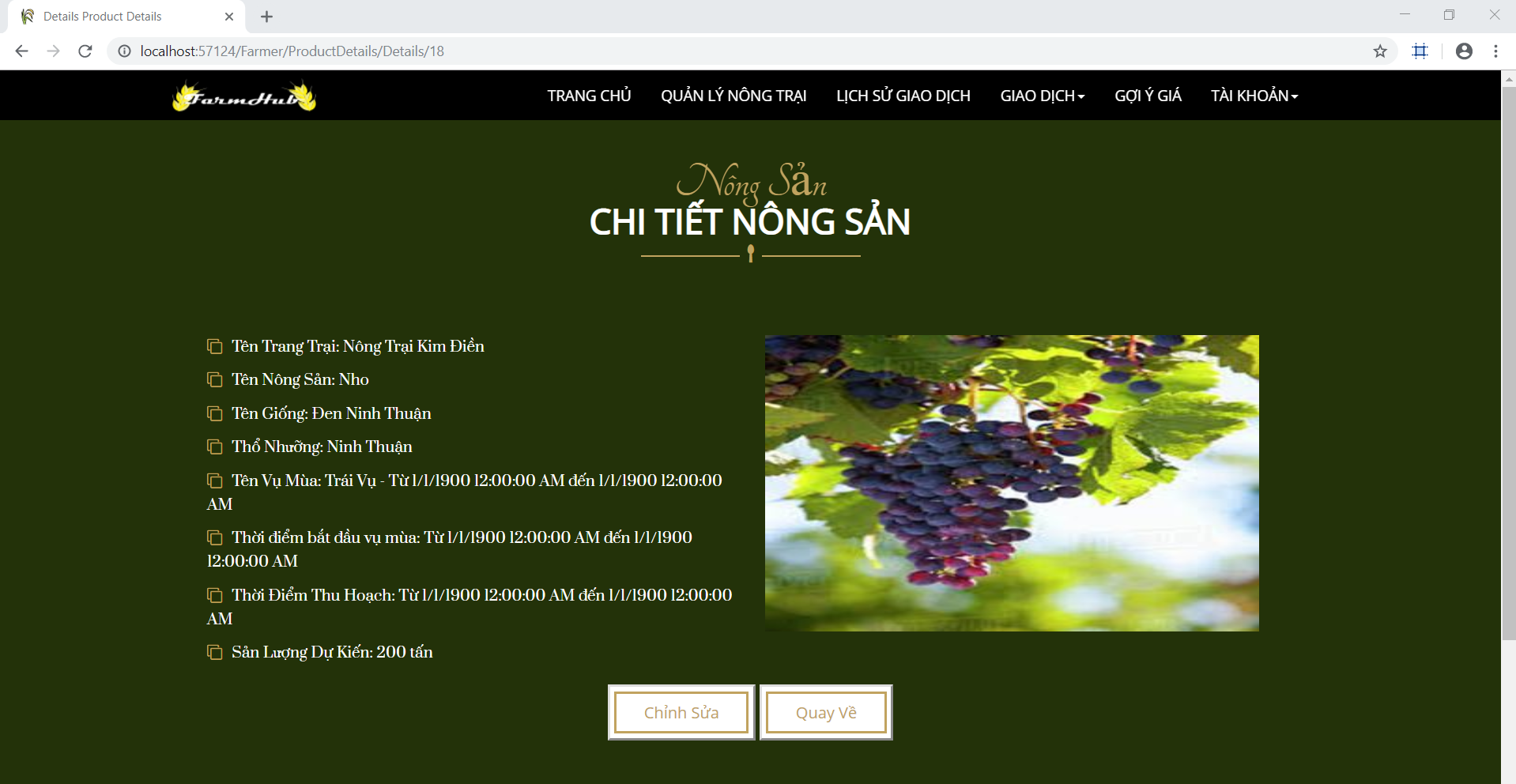
Hình x Index Product Details

* Màn hình thêm mới nông sản
  + Tên màn hình: Create Product Details
  + Ý nghĩa: Tạo mới thông tin nông sản sẽ trồng ở một nông trại
  + Hình ảnh:



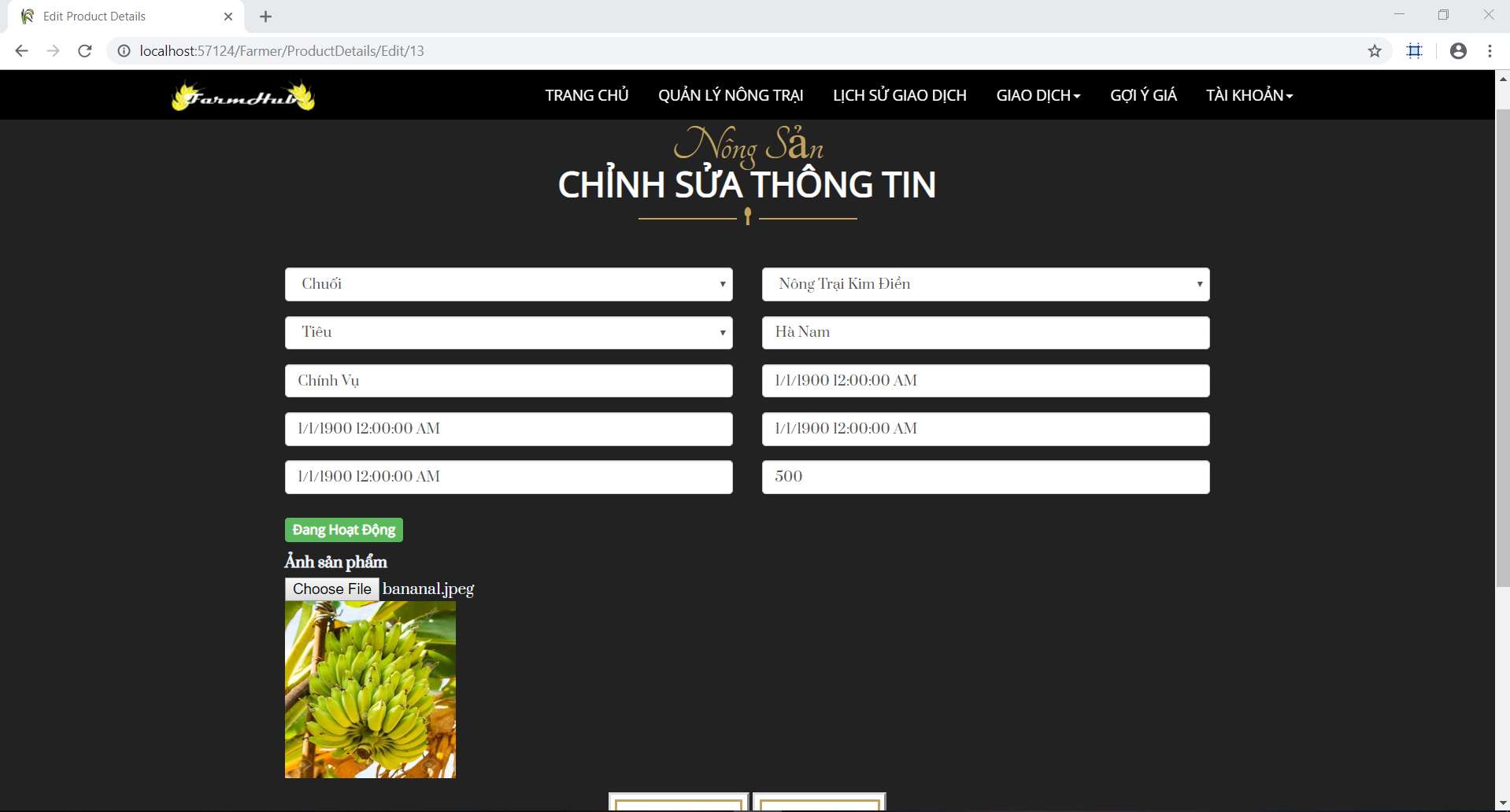
**Hình x** Create Product Details

* Màn hình xem chi tiết nông sản
  + Tên màn hình: Details Product Details
  + Ý nghĩa: Xem thông tin chi tiết của một nông sản
  + Hình ảnh:



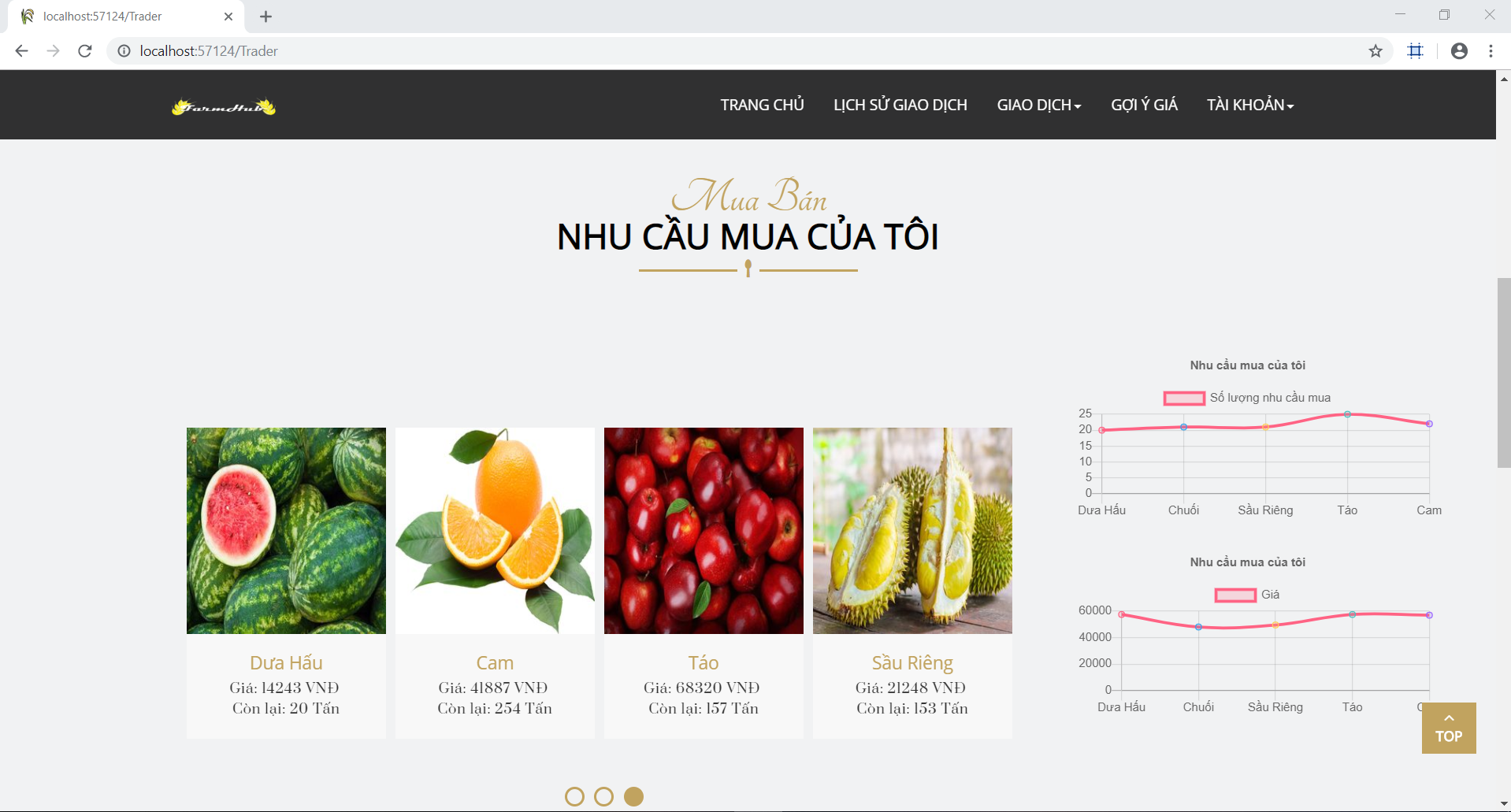
**Hình x** Details Product Details

* Màn hình sửa chi tiết nông sản
  + Tên màn hình: Edit Product Details
  + Ý nghĩa: Chỉnh sửa thông tin nông sản
  + Hình ảnh:



Hình x Edit Product Details

* Màn hình trang chủ của Chủ Doanh Nghiệp
  + Tên màn hình: Trader Home Page
  + Ý nghĩa: Trang chủ của chủ doanh nghiệp
  + Hình ảnh:



**Hình x** Trader Home Page

### Thiết kế lớp

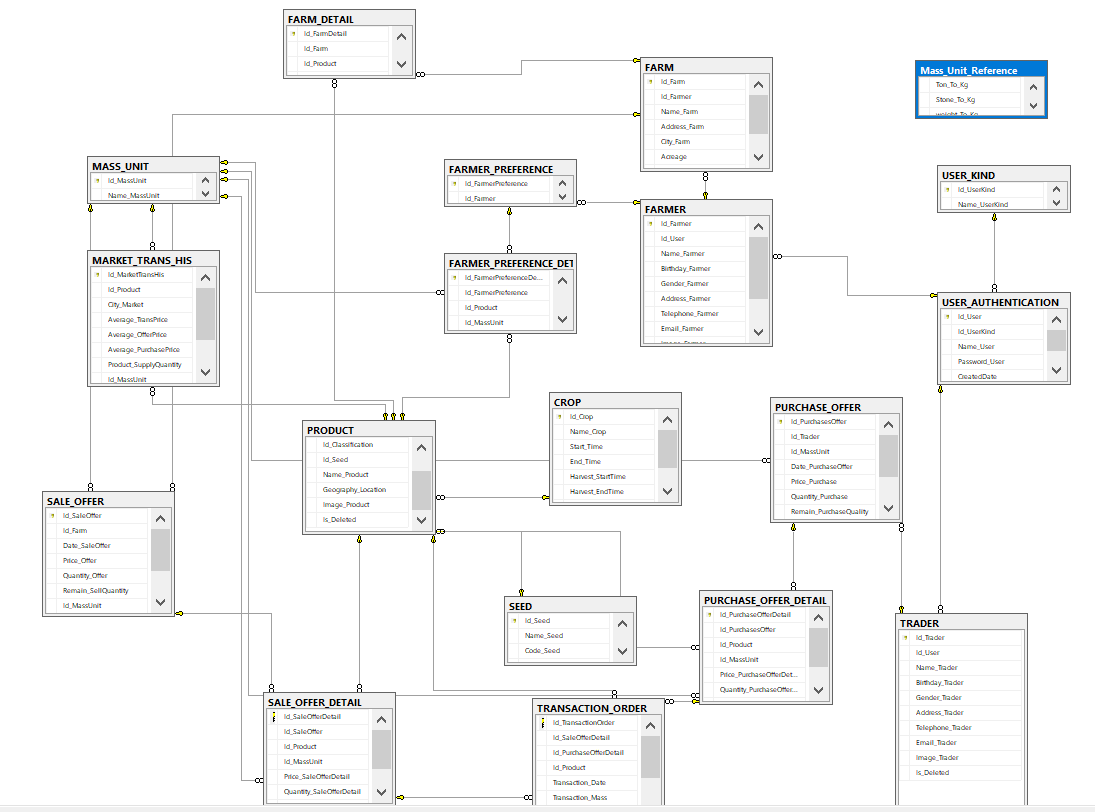
~~Phần này có độ dài từ ba đến bốn trang. Sinh viên trình bày thiết kế chi tiết các thuộc tính và phương thức cho một số lớp chủ đạo/quan trọng nhất của ứng dụng (từ 2-4 lớp). Thiết kế chi tiết cho các lớp khác, nếu muốn trình bày, sinh viên đưa vào phần phụ lục.~~

~~Để minh họa thiết kế lớp, sinh viên thiết kế luồng truyền thông điệp giữa các đối tượng tham gia cho 2 đến 3 use case quan trọng nào đó bằng biểu đồ trình tự (hoặc biểu đồ giao tiếp).~~

### Thiết kế cơ sở dữ liệu

~~Phần này có độ dài từ hai đến bốn trang. Sinh viên thiết kế, vẽ và giải thích biểu đồ thực thể liên kết (E-R diagram). Từ đó, sinh viên thiết kế cơ sở dữ liệu tùy theo hệ quản trị cơ sở dữ liệu mà mình sử dụng (SQL, NoSQL, Firebase, v.v.)~~

#### Sơ đồ logic



**Hình x** Sơ đồ logic

#### Chi tiết các bảng

##### Bảng USER\_KIND

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **TT** | **Tên thuộc tính (Field name)** | **Kiểu dữ liệu** | **Độ rộng** | **Not NULL** | **Ràng buộc / Miền giá trị** | **Mã hóa** | **Diễn giải** |
| **1** | Id\_UserKind | int |  | X | -32,768 đến 32,767 |  |  |
| 2 | Name\_UserKind | nvarchar | 50 |  |  |  |  |

##### Bảng USER\_AUTHENTICATION

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **TT** | **Tên thuộc tính (Field name)** | **Kiểu dữ liệu** | **Độ rộng** | **Not NULL** | **Ràng buộc / Miền giá trị** | **Mã hóa** | **Diễn giải** |
| **1** | Id\_User int | int |  | X | -32,768 đến 32,767 |  |  |
| 2 | Id\_UserKind | int |  |  | -32,768 đến 32,767 |  |  |
| 3 | Name\_User | varchar | 50 |  |  |  |  |
| 4 | Password\_User | varchar | 50 |  |  | X |  |
| 5 | CreatedDate | datetime |  |  | 1/1/1753 đến 31/12/9999 |  |  |
| 6 | Status\_User | tinyint |  |  | 0 đến 255. |  |  |

##### Bảng FARMER

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **TT** | **Tên thuộc tính (Field name)** | **Kiểu dữ liệu** | **Độ rộng** | **Not NULL** | **Ràng buộc / Miền giá trị** | **Mã hóa** | **Diễn giải** |
| **1** | Id\_Farmer | int |  | X | -32,768 đến 32,767 |  |  |
| 2 | Id\_User | int |  |  | -32,768 đến 32,767 |  |  |
| 3 | Name\_Farmer | nvarchar | 50 |  |  |  |  |
| 4 | Birthday\_Farmer | datetime |  |  | 1/1/1753 đến 31/12/9999 |  |  |
| 5 | Gender\_Farmer | bit |  |  | True/False |  |  |
| 6 | Address\_Farmer | nvarchar | 200 |  |  |  |  |
| 7 | Telephone\_Farmer | nvarchar | 10 |  |  |  |  |
| 8 | Email\_Farmer | nvarchar | 100 |  |  |  |  |
| 9 | Image\_Farmer | varchar | 200 |  |  |  |  |
| 10 | Farm\_Count | tinyint |  |  | 0 đến 255. |  |  |
| 11 | Is\_Deleted | bit |  |  | True/False |  |  |

##### Bảng FARM

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **TT** | **Tên thuộc tính (Field name)** | **Kiểu dữ liệu** | **Độ rộng** | **Not NULL** | **Ràng buộc / Miền giá trị** | **Mã hóa** | **Diễn giải** |
| **1** | Id\_Farm | int |  | X | -32,768 đến 32,767 |  |  |
| 2 | Id\_Farmer | int |  |  | -32,768 đến 32,767 |  |  |
| 3 | Name\_Farm | nvarchar | 50 |  |  |  |  |
| 4 | Address\_Farm | nvarchar | 200 |  |  |  |  |
| 5 | City\_Farm | nvarchar | 50 |  |  |  |  |
| 6 | Acreage | int |  |  |  |  |  |
| 7 | Description\_Farm | nvarchar | 50 |  |  |  |  |
| 8 | Is\_Deleted | bit |  |  | True/False |  |  |

##### Bảng FARM\_DETAIL

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **TT** | **Tên thuộc tính (Field name)** | **Kiểu dữ liệu** | **Độ rộng** | **Not NULL** | **Ràng buộc / Miền giá trị** | **Mã hóa** | **Diễn giải** |
| **1** | Id\_FarmDetail | int |  | X | -32,768 đến 32,767 |  |  |
| 2 | Id\_Farm | int |  |  | -32,768 đến 32,767 |  |  |
| 3 | Id\_Product | int |  |  | -32,768 đến 32,767 |  |  |
| 4 | Is\_Deleted | bit |  |  | True/False |  |  |

##### Bảng SEED

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **TT** | **Tên thuộc tính (Field name)** | **Kiểu dữ liệu** | **Độ rộng** | **Not NULL** | **Ràng buộc / Miền giá trị** | **Mã hóa** | **Diễn giải** |
| **1** | Id\_Seed | int |  | X | -32,768 đến 32,767 |  |  |
| 2 | Name\_Seed | nvarchar | 50 |  |  |  |  |
| 3 | Code\_Seed | nvarchar | 50 |  |  |  |  |
| 4 | Is\_Deleted | bit |  |  | True/False |  |  |

##### Bảng CROP

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **TT** | **Tên thuộc tính (Field name)** | **Kiểu dữ liệu** | **Độ rộng** | **Not NULL** | **Ràng buộc / Miền giá trị** | **Mã hóa** | **Diễn giải** |
| **1** | Id\_Crop | int |  | X | -32,768 đến 32,767 |  |  |
| 2 | Name\_Crop | nvarchar | 50 |  |  |  |  |
| 3 | Start\_Time | date |  |  | 1/1/1753 đến 31/12/9999 |  |  |
| 4 | End\_Time | date |  |  | 1/1/1753 đến 31/12/9999 |  |  |
| 5 | Harvest\_StartTime | date |  |  | 1/1/1753 đến 31/12/9999 |  |  |
| 6 | Harvest\_EndTime | date |  |  | 1/1/1753 đến 31/12/9999 |  |  |
| 7 | Quantity\_Expected | int |  |  | -32,768 đến 32,767 |  |  |
| 8 | Is\_Deleted | bit |  |  | True/False |  |  |

##### Bảng MASS\_UNIT

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **TT** | **Tên thuộc tính (Field name)** | **Kiểu dữ liệu** | **Độ rộng** | **Not NULL** | **Ràng buộc / Miền giá trị** | **Mã hóa** | **Diễn giải** |
| **1** | Id\_MassUnit | int |  | X | -32,768 đến 32,767 |  |  |
| 2 | Name\_MassUnit | varchar | 10 |  |  |  |  |

##### Bảng PRODUCT

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **TT** | **Tên thuộc tính (Field name)** | **Kiểu dữ liệu** | **Độ rộng** | **Not NULL** | **Ràng buộc / Miền giá trị** | **Mã hóa** | **Diễn giải** |
| **1** | Id\_Product | int |  | X | -32,768 đến 32,767 |  |  |
| 2 | Id\_Crop | int |  |  | -32,768 đến 32,767 |  |  |
| 3 | Id\_Classification | int |  |  | -32,768 đến 32,767 |  |  |
| 4 | Id\_Seed | int |  |  | -32,768 đến 32,767 |  |  |
| 5 | Name\_Product | nvarchar | 50 |  |  |  |  |
| 6 | Geography\_Location | nvarchar | 50 |  |  |  |  |
| 7 | Image\_Product | varchar | 200 |  |  |  |  |
| 8 | Description\_Product | nvarchar | 200 |  |  |  |  |
| 9 | Is\_Deleted | bit |  |  | True/False |  |  |

##### Bảng MARKET\_TRANSHIS

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **TT** | **Tên thuộc tính (Field name)** | **Kiểu dữ liệu** | **Độ rộng** | **Not NULL** | **Ràng buộc / Miền giá trị** | **Mã hóa** | **Diễn giải** |
| **1** | Id\_MarketTransHis | int |  | X | -32,768 đến 32,767 |  |  |
| 2 | Id\_Product | int |  |  | -32,768 đến 32,767 |  |  |
| 3 | City\_Market | nvarchar | 50 |  |  |  |  |
| 4 | Average\_TransPrice | int |  |  | -32,768 đến 32,767 |  |  |
| 5 | Average\_OfferPrice | int |  |  | -32,768 đến 32,767 |  |  |
| 6 | Average\_PurchasePrice | int |  |  | -32,768 đến 32,767 |  |  |
| 7 | Product\_SupplyQuantity | int |  |  | -32,768 đến 32,767 |  |  |
| 8 | Id\_MassUnit | int |  |  | -32,768 đến 32,767 |  |  |
| 9 | From\_Date | date |  |  | 1/1/1753 đến 31/12/9999 |  |  |
| 10 | To\_Date | date |  |  | 1/1/1753 đến 31/12/9999 |  |  |

##### Bảng SALE\_OFFER

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **TT** | **Tên thuộc tính (Field name)** | **Kiểu dữ liệu** | **Độ rộng** | **Not NULL** | **Ràng buộc / Miền giá trị** | **Mã hóa** | **Diễn giải** |
| **1** | Id\_SaleOffer | int |  | X | -32,768 đến 32,767 |  |  |
| 2 | Id\_Farm | int |  |  | -32,768 đến 32,767 |  |  |
| 3 | Date\_SaleOffer |  |  |  |  |  |  |
| 4 | Price\_Offer | int |  |  | -32,768 đến 32,767 |  |  |
| 5 | Quantity\_Offer | int |  |  | -32,768 đến 32,767 |  |  |
| 6 | Remain\_SellQuantity | int |  |  | -32,768 đến 32,767 |  |  |
| 7 | Id\_MassUnit | int |  |  | -32,768 đến 32,767 |  |  |
| 8 | Status\_SaleOffer | tinyint |  |  | 0 đến 255. |  |  |
| 9 | Is\_Deleted | bit |  |  | True/False |  |  |

##### Bảng SALE\_OFFER\_DETAIL

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **TT** | **Tên thuộc tính (Field name)** | **Kiểu dữ liệu** | **Độ rộng** | **Not NULL** | **Ràng buộc/ Miền giá trị** | **Mã hóa** | **Diễn giải** |
| **1** | Id\_SaleOfferDetail | int |  | X | -32,768 đến 32,767 |  |  |
| 2 | Id\_SaleOffer | int |  |  | -32,768 đến 32,767 |  |  |
| 3 | Id\_Product | int |  |  | -32,768 đến 32,767 |  |  |
| 4 | Id\_MassUnit | int |  |  | -32,768 đến 32,767 |  |  |
| 5 | Price\_SaleOfferDetail | int |  |  | -32,768 đến 32,767 |  |  |
| 6 | Quantity\_SaleOfferDetail | int |  |  | -32,768 đến 32,767 |  |  |
| 7 | Status\_SaleOfferDetail | int |  |  | -32,768 đến 32,767 |  |  |
| 8 | Is\_Deleted | int |  |  | -32,768 đến 32,767 |  |  |

##### Bảng TRADER

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **TT** | **Tên thuộc tính (Field name)** | **Kiểu dữ liệu** | **Độ rộng** | **Not NULL** | **Ràng buộc/ Miền giá trị** | **Mã hóa** | **Diễn giải** |
| **1** | Id\_Trader | int |  | X | -32,768 đến 32,767 |  |  |
| 2 | Id\_User | int |  |  | -32,768 đến 32,767 |  |  |
| 3 | Name\_Trader | nvarchar | 50 |  |  |  |  |
| 4 | Birthday\_Trader | datetime |  |  |  |  |  |
| 5 | Gender\_Trader | bit |  |  | True/False |  |  |
| 6 | Address\_Trader | nvarchar | 200 |  |  |  |  |
| 7 | Telephone\_Trader | nvarchar | 10 |  |  |  |  |
| 8 | Email\_Trader | nvarchar | 100 |  |  |  |  |
| 9 | Image\_Trader | varchar | 200 |  |  |  |  |
| 10 | Is\_Deleted | bit |  |  | True/False |  |  |

##### Bảng PURCHASE\_OFFER

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **TT** | **Tên thuộc tính (Field name)** | **Kiểu dữ liệu** | **Độ rộng** | **Not NULL** | **Ràng buộc / Miền giá trị** | **Mã hóa** | **Diễn giải** |
| **1** | Id\_PurchasesOffer | int |  | X | -32,768 đến 32,767 |  |  |
| 2 | Id\_Trader | int |  |  | -32,768 đến 32,767 |  |  |
|  | Id\_MassUnit | int |  |  | -32,768 đến 32,767 |  |  |
|  | Date\_PurchaseOffer | date |  |  |  |  |  |
|  | Price\_Purchase | int |  |  | -32,768 đến 32,767 |  |  |
|  | Quantity\_Purchase | int |  |  | -32,768 đến 32,767 |  |  |
|  | Remain\_PurchaseQuality | int |  |  | -32,768 đến 32,767 |  |  |
|  | Status\_PurchaseOffer | tinyint |  |  | 0 đến 255 |  |  |
|  | Is\_Deleted | bit |  |  | True/False |  |  |

##### Bảng PURCHASE\_OFFER\_DETAIL

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **TT** | **Tên thuộc tính (Field name)** | **Kiểu dữ liệu** | **Độ rộng** | **Not NULL** | **Ràng buộc / Miền giá trị** | **Mã hóa** | **Diễn giải** |
| **1** | Id\_PurchaseOfferDetail | int |  | X | -32,768 đến 32,767 |  |  |
| 2 | Id\_PurchasesOffer | int |  |  | -32,768 đến 32,767 |  |  |
| 3 | Id\_Product | int |  |  | -32,768 đến 32,767 |  |  |
| 4 | Id\_MassUnit | int |  |  | -32,768 đến 32,767 |  |  |
| 5 | Price\_PurchaseOfferDetail | int |  |  | -32,768 đến 32,767 |  |  |
| 6 | Quantity\_PurchaseOfferDetail | int |  |  | -32,768 đến 32,767 |  |  |
| 7 | Status\_PurchaseOfferDetail | tinyint |  |  | -32,768 đến 32,767 |  |  |
| 8 | Is\_Deleted | bit |  |  | -32,768 đến 32,767 |  |  |

##### Bảng TRANSACTION\_ORDER

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **TT** | **Tên thuộc tính (Field name)** | **Kiểu dữ liệu** | **Độ rộng** | **Not NULL** | **Ràng buộc / Miền giá trị** | **Mã hóa** | **Diễn giải** |
| **1** | Id\_TransactionOrder | int |  |  | -32,768 đến 32,767 |  |  |
| 2 | Id\_SaleOfferDetail | int |  |  | -32,768 đến 32,767 |  |  |
| 3 | Id\_PurchaseOfferDetail | int |  |  | -32,768 đến 32,767 |  |  |
| 4 | Id\_Product | int |  |  | -32,768 đến 32,767 |  |  |
| 5 | Transaction\_Date | date |  |  | 1/1/1753 đến 31/12/9999 |  |  |
| 6 | Transaction\_Mass | int |  |  | -32,768 đến 32,767 |  |  |
| 7 | Transaction\_Price | int |  |  | -32,768 đến 32,767 |  |  |
| 8 | Status\_TransactionOrder | tinyint |  |  | 0 đến 255 |  |  |
| 9 | Is\_Deleted | bit |  |  | True/False |  |  |

##### Bảng FARMER\_PREFERENCE

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **TT** | **Tên thuộc tính (Field name)** | **Kiểu dữ liệu** | **Độ rộng** | **Not NULL** | **Ràng buộc / Miền giá trị** | **Mã hóa** | **Diễn giải** |
| **1** | Id\_FarmerPreference | int |  | X | -32,768 đến 32,767 |  |  |
| 2 | Id\_Farmer | int |  |  | -32,768 đến 32,767 |  |  |

##### Bảng FARMER\_PREFERENCE\_DETAIL

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **TT** | **Tên thuộc tính (Field name)** | **Kiểu dữ liệu** | **Độ rộng** | **Not NULL** | **Ràng buộc / Miền giá trị** | **Mã hóa** | **Diễn giải** |
| **1** | Id\_FarmerPreferenceDetail | int |  |  | -32,768 đến 32,767 |  |  |
| 2 | Id\_FarmerPreference | int |  |  | -32,768 đến 32,767 |  |  |
| 3 | Id\_Product | int |  |  | -32,768 đến 32,767 |  |  |
| 4 | Id\_MassUnit | int |  |  | -32,768 đến 32,767 |  |  |
| 5 | Is\_Deleted | bit |  |  | True/False |  |  |

## Xây dựng ứng dụng

### Thư viện và công cụ sử dụng

~~Sinh viên liệt kê các công cụ, ngôn ngữ lập trình, API, thư viện, IDE, công cụ kiểm thử, v.v. mà mình sử dụng để phát triển ứng dụng. Mỗi công cụ phải được chỉ rõ phiên bản sử dụng. SV nên kẻ bảng mô tả tương tự như Bảng 1. Nếu có nhiều nội dung trình bày, sinh viên cần xoay ngang bảng.~~

**Bảng 1** Danh sách thư viện và công cụ sử dụng

|  |  |  |
| --- | --- | --- |
| Mục đích | Công cụ | Địa chỉ URL |
| IDE lập trình | Visual Studio 2017 | <https://visualstudio.microsoft.com/> |
| Hệ quản trị CSDL | SQL Server 2017 | <https://www.microsoft.com/en-us/sql-server/sql-server-downloads-free-trial> |
| Front-end | Bootstrap, JavaScripts, jQuerry | <https://getbootstrap.com/> |
| Back-end | Nền tảng ASP.NET | <https://dotnet.microsoft.com/apps/aspnet> |
| Issue Tracker | BitBucket Issue | [https://bitbucket.org](https://bitbucket.org/) |

### Kết quả đạt được

~~Sinh viên trước tiên mô tả kết quả đạt được của mình là gì, ví dụ như các sản phẩm được đóng gói là gì, bao gồm những thành phần nào, ý nghĩa, vai trò?~~

~~Sinh viên cần thống kê các thông tin về ứng dụng của mình như: số dòng code, số lớp, số gói, dung lượng toàn bộ mã nguồn, dung lượng của từng sản phẩm đóng gói, v.v. Tương tự như phần liệt kê về công cụ sử dụng, sinh viên cũng nên dùng bảng để mô tả phần thông tin thống kê này.~~

### Minh hoạ các chức năng chính

~~Sinh viên lựa chọn và đưa ra màn hình cho các chức năng chính, quan trọng, và thú vị nhất. Mỗi giao diện cần phải có lời giải thích ngắn gọn. Khi giải thích, sinh viên có thể kết hợp với các chú thích ở trong hình ảnh giao diện.~~

## Kiểm thử

~~Phần này có độ dài từ hai đến ba trang. Sinh viên thiết kế các trường hợp kiểm thử cho hai đến ba chức năng quan trọng nhất. Sinh viên cần chỉ rõ các kỹ thuật kiểm thử đã sử dụng. Chi tiết các trường hợp kiểm thử khác, nếu muốn trình bày, sinh viên đưa vào phần phụ lục.~~

~~Sinh viên sau cùng tổng kết về số lượng các trường hợp kiểm thử và kết quả kiểm thử. Sinh viên cần phân tích lý do nếu kết quả kiểm thử không đạt.~~

## Triển khai

~~Sinh viên trình bày mô hình và/hoặc cách thức triển khai thử nghiệm/thực tế. Ứng dụng của sinh viên được triển khai trên server/thiết bị gì, cấu hình như thế nào. Kết quả triển khai thử nghiệm nếu có (số lượng người dùng, số lượng truy cập, thời gian phản hồi, phản hồi người dùng, khả năng chịu tải, các thống kê, v.v.)~~

# Các giải pháp và đóng góp nổi bật

~~Chương này có độ dài tối thiểu 5 trang, tối đa không giới hạn. Sinh viên cần trình bày tất cả những nội dung đóng góp mà mình thấy tâm đắc nhất trong suốt quá trình làm ĐATN. Đó có thể là một loạt các vấn đề khó khăn mà sinh viên đã từng bước giải quyết được, là giải thuật cho một bài toán cụ thể, là giải pháp tổng quát cho một lớp bài toán, hoặc là mô hình/kiến trúc hữu hiệu nào đó được sinh viên thiết kế.~~

~~Chương này là~~ **~~cơ sở quan trọng~~** ~~để các thầy cô đánh giá sinh viên. Vì vậy, sinh viên cần phát huy tính sáng tạo, khả năng phân tích, phản biện, lập luận, tổng quát hóa vấn đề và tập trung viết cho thật tốt.~~

~~Mỗi giải pháp hoặc đóng góp của sinh viên cần được trình bày trong một mục độc lập bao gồm ba mục con: (i) dẫn dắt/giới thiệu về bài toán/vấn đề, (ii) giải pháp, và (iii) kết quả đạt được (nếu có).~~

~~Sinh viên lưu ý~~ **~~không trình bày lặp lại nội dung~~**~~. Những nội dung đã trình bày chi tiết trong các chương trước không được trình bày lại trong chương này. Vì vậy, với nội dung hay, mang tính đóng góp/giải pháp, sinh viên chỉ nên tóm lược/mô tả sơ bộ trong các chương trước, đồng thời tạo tham chiếu chéo tới đề mục tương ứng trong Chương 5 này. Chi tiết thông tin về đóng góp/giải pháp được trình bày trong mục đó.~~

~~Ví dụ, trong Chương 4, sinh viên có thiết kế được kiến trúc đáng lưu ý gì đó, là sự kết hợp của các kiến trúc MVC, MVP, SOA, v.v. Khi đó, sinh viên sẽ chỉ mô tả ngắn gọn kiến trúc đó ở Chương 4, rồi thêm các câu có dạng: “Chi tiết về kiến trúc này sẽ được trình bày trong phần 5.1”.~~

Dàn ý:

Kinh nghiệm:

* Phải có kỹ năng sử dụng tool quản lý code
* Phải nghiên cứu kỹ DB, đơn vị tiền phải để float
* Phải thống nhất trước front-end
* Phải thống nhất công nghệ làm việc với nhau JS, ViewBag, vv...

# Kết luận và hướng phát triển

## Kết luận

~~Sinh viên so sánh kết quả nghiên cứu hoặc sản phẩm của mình với các nghiên cứu hoặc sản phẩm tương tự.~~

~~Sinh viên phân tích trong suốt quá trình thực hiện ĐATN, mình đã làm được gì, chưa làm được gì, các đóng góp nổi bật là gì, và tổng hợp những bài học kinh nghiệm rút ra nếu có.~~

Ererererdfd

|  |  |
| --- | --- |
| **Binkabi - sàn giao dịch Blockchain** | **Farm Hub** |
| Bỏ được các phần trung gian, người dùng sẽ tiết kiệm được nhiều chi phí hơn khi mở rộng | Cắt bỏ được các phần trung gian, tiết kiệm chi phí |
| Tất cả nghiệp vụ trên hệ thống này có thể thực hiện thông qua “smart-contract” | Vận hành bằng Controller do chúng em viết, có thể vẫn còn sai sót |
| Có tính tự động hóa cao, do vậy giảm được chi phí server, vận hành | Các tính năng tự động hóa được thực thi bằng Job, Store Procedure, Trigger |
| Chất lượng nông sản và an toàn thực phẩm vẫn chưa được đảm bảo | Áp dụng IoT giúp đảm bảo chất lượng và vệ sinh an toàn thực phẩm, dễ dàng truy xuất nguồn gốc nông sản |
| Có nhiều vấn đề về lưu trữ, băng thông, xử lý | Có vấn đề về tốc độ xử lý server |

**Bảng x** So sánh giữa Binkabi và Farm Hub

|  |  |
| --- | --- |
| **Meicai - ứng dụng bán rau** | **Farm Hub** |
| Là một ứng dụng di động, giúp các nhà hàng có thể đặt mua trực tiếp các loại rau quả từ nông dân | Là một website giúp thương lái và chủ trang trại có thể đặt hàng trực tiếp với nhau |
| Giúp cắt bỏ các khâu trung gian, không còn bị phụ thuộc vào các nhà bán buôn truyền thống. | Cắt bỏ được các phần trung gian, tiết kiệm chi phí |
| Chỉ bán rau | Bán đa dạng các loại nông sản |
| Chất lượng nông sản và an toàn thực phẩm vẫn chưa được đảm bảo | Áp dụng IoT giúp đảm bảo chất lượng và vệ sinh an toàn thực phẩm, dễ dàng truy xuất nguồn gốc nông sản |

**Bảng x** So sánh giữa Meicai và Farm Hub

## Hướng phát triển

~~Trong phần này, sinh viên trình bày định hướng công việc trong tương lai để hoàn thiện sản phẩm hoặc nghiên cứu của mình.~~

~~Trước tiên, sinh viên trình bày các công việc cần thiết để hoàn thiện các chức năng/nhiệm vụ đã làm. Sau đó sinh viên phân tích các hướng đi mới cho phép cải thiện và nâng cấp các chức năng/nhiệm vụ đã làm.~~

Các công việc cần thiết để hoàn thiện các chức năng đã làm:

* Phát triển thêm ứng dụng trên hiện thoại
* Áp dụng IoT vào khâu sản xuất nông sản

# Tài liệu tham khảo

Lưu ý: Sinh viên không được đưa bài giảng/slide, các trang Wikipedia, hoặc các trang web thông thường làm tài liệu tham khảo.

Một trang web được phép dùng làm tài liệu tham khảo **chỉ khi** nó là công bố chính thống của cá nhân hoặc tổ chức nào đó. Ví dụ, trang web đặc tả ngôn ngữ XML của tổ chức W3C https://www.w3.org/TR/2008/REC-xml-20081126/ là TLTK hợp lệ.

Có năm loại tài liệu tham khảo mà sinh viên phải tuân thủ đúng quy định về cách thức liệt kê thông tin như sau. Lưu ý: các phần văn bản trong cặp dấu < > dưới đây chỉ là hướng dẫn khai báo cho từng loại tài liệu tham khảo; sinh viên cần xóa các phần văn bản này trong ĐATN của mình.

<**Bài báo đăng trên tạp chí khoa học**: Tên tác giả, tên bài báo, tên tạp chí, volume, từ trang đến trang (nếu có), nhà xuất bản, năm xuất bản>

1. Hovy E. H., Automated Discourse Generation Using Discourse Structure Relations, Artificial Intelligence, Elsevier Science Publishers, 63: 341-385, 1993.

<**Sách**: Tên tác giả, tên sách, volume (nếu có), lần tái bản (nếu có), nhà xuất bản, năm xuất bản>

1. Peterson L. L. and Davie B. S., Computer Networks: A Systems Approach, 2nd ed., Mogran-Kaufmann, 1999.
2. Nguyễn Thúc Hải, Mạng máy tính và các hệ thống mở, Nhà xuất bản giáo dục, 1999.

<**Tập san Báo cáo Hội nghị Khoa học**: Tên tác giả, tên báo cáo, tên hội nghị, ngày (nếu có), địa điểm hội nghị, năm xuất bản>

1. Poesio M. and Di Eugenio B., Discourse Structure and Anaphoric Accessibility, In Proc. of the ESSLLI Workshop on Information Structure, Discourse Structure and Discourse Semantics, Helsinki, 2001.

<**Đồ án tốt nghiệp, Luận văn Thạc sĩ, Tiến sĩ**: Tên tác giả, tên đồ án/luận văn, loại đồ án/luận văn, tên trường, địa điểm, năm xuất bản>

1. Knott D., A Data-Driven Methodology for Motivating a Set of Coherence Relations, Ph.D. Thesis, University of Edinburgh, UK, 1996.

<**Tài liệu tham khảo từ Internet**: Tên tác giả (nếu có), tựa đề, cơ quan (nếu có), địa chỉ trang web, thời gian lần cuối truy cập trang web>

1. Berners-Lee T., Hypertext Transfer Protocol (HTTP), CERN, ftp:/info.cern.ch/pub/www/doc/http-spec.txt.Z, last visited May 2010.
2. Princeton University, WordNet, http://www.cogsci.princeton.edu/~wn/index.shtml, last visited May 2010.

# Phụ lục

Phần phụ lục là không bắt buộc. Nếu sinh viên không có nhu cầu trình bày thêm, có thể xóa bỏ phần này. Lưu ý là phần phụ lục chỉ được đánh chỉ mục đến cấp 2, sinh viên không được phép chia nhỏ hơn nữa.

Hướng dẫn viết đồ án tốt nghiệp

Quy định chung

Dưới đây là một số quy định và hướng dẫn viết đồ án tốt nghiệp mà bắt buộc sinh viên phải **đọc kỹ và tuân thủ nghiêm ngặt**.

Sinh viên cần đảm bảo tính thống nhất toàn báo cáo (font chữ, căn dòng hai bên, hình ảnh, bảng, margin trang, đánh số trang, v.v.). Để làm được như vậy, sinh viên chỉ cần sử dụng các định dạng theo đúng template ĐATN này. Khi paste nội dung văn bản từ tài liệu khác của mình, sinh viên cần chọn kiểu Copy là “Text Only” để định dạng văn bản của template không bị phá vỡ/vi phạm.

Tuyệt đối cấm sinh viên đạo văn. Sinh viên cần ghi rõ nguồn cho tất cả những gì không tự mình viết/vẽ lên, bao gồm các câu trích dẫn, các hình ảnh, bảng biểu, v.v. Khi bị phát hiện, sinh viên sẽ **không được phép bảo vệ ĐATN**.

Tất cả các hình vẽ, bảng biểu, công thức, và tài liệu tham khảo trong ĐATN nhất thiết phải được SV giải thích và tham chiếu tới ít nhất một lần. Không chấp nhận các trường hợp sinh viên đưa ra hình ảnh, bảng biểu tùy hứng và không có lời mô tả/giải thích nào.

Sinh viên tuyệt đối không trình bày ĐATN theo kiểu viết ý hoặc gạch đầu dòng. ĐATN không phải là một slide thuyết trình; khi người đọc không hiểu sẽ không có ai giải thích hộ. Sinh viên cần viết thành các đoạn văn và phân tích, diễn giải đầy đủ, rõ ràng. Câu văn cần đúng ngữ pháp, đầy đủ chủ ngữ, vị ngữ và các thành phần câu.

Khi thực sự cần liệt kê, sinh viên nên liệt kê theo phong cách khoa học với các ký tự La Mã. Ví dụ, nhiều sinh viên luôn cảm thấy hối hận vì (i) chưa cố gắng hết mình, (ii) chưa sắp xếp thời gian học/chơi một cách hợp lý, (iii) chưa tìm được người yêu để chia sẻ quãng đời sinh viên vất vả, và (iv) viết ĐATN một cách cẩu thả.

Trong một số trường hợp nhất thiết phải dùng các bullet để liệt kê, sinh viên cần thống nhất Style cho toàn bộ các bullet các cấp mà mình sử dụng đến trong báo cáo. Nếu dùng bullet cấp 1 là hình tròn đen, toàn bộ báo cáo cần thống nhất cách dùng như vậy; ví dụ như sau:

* Đây là mục 1 – Thực sự không còn cách nào khác tôi mới dùng đến việc bullet trong báo cáo.
* Đây là mục 2 – Nghĩ lại thì tôi có thể không cần dùng bullet cũng được. Nên tôi sẽ xóa bullet và tổ chức lại hai mục này trong báo cáo của mình cho khoa học hơn. Tôi muốn thầy cô và người đọc cảm nhận được tâm huyết của tôi trong từng trang báo cáo ĐATN.

Tạo đề mục

Đề mục giúp tạo bố cục cho tài liệu. Để các tính năng tự động – ví dụ tính năng cập nhật mục lục, hoặc tính năng tham chiếu chéo – của Word hoạt động được, sinh viên cần tuân thủ theo các style đã tạo trong tài liệu này. Để hiển thị các style này, sinh viên vào tab Home trong thanh Ribbon của Word.

Để tạo đề mục cấp 1, 2, 3, 4, 5, sinh viên gõ tiêu đề cho đề mục của mình rồi chọn các style là Heading 1, 2, 3, 4, 5 tương ứng. Sinh viên hạn chế dùng tới đề mục cấp 4, và phải trong trường hợp thực sự cần thiết mới dùng đến đề mục cấp 5.

Phần phụ lục chỉ cho phép có hai cấp tiêu đề. Hai style tương ứng với hai cấp này là “Heading 7, Phụ lục cấp 1” và “Heading 8, Phụ lục cấp 2”.

Bảng biểu

Sinh viên lưu ý không để bảng tràn ra lề (margin) trên, dưới, trái hoặc phải của trang. Do không gian nhỏ hẹp, bảng nên có font là 12pt, nhỏ hơn một chút so với font thông thường (13pt) của ĐATN. Độ dãn dòng của bảng nên là 1 line. Căn lề bảng là căn giữa, nhưng nội dung văn bản trong bảng nên được căn lề trái.

Sinh viên có thể viết tắt các từ trong bảng để tiết kiệm không gian nhưng phải giải thích các từ viết tắt này ở phần Chú thích bảng. Ví dụ áp dụng được minh họa trong Bảng 2.

**Bảng 2** Ví dụ sử dụng bảng

Chú thích: Y: Year; RS: Risk Set; G: Graduated; AB: Academically Excluded;C: Censored; HRG: Hazard Ratio – Graduated

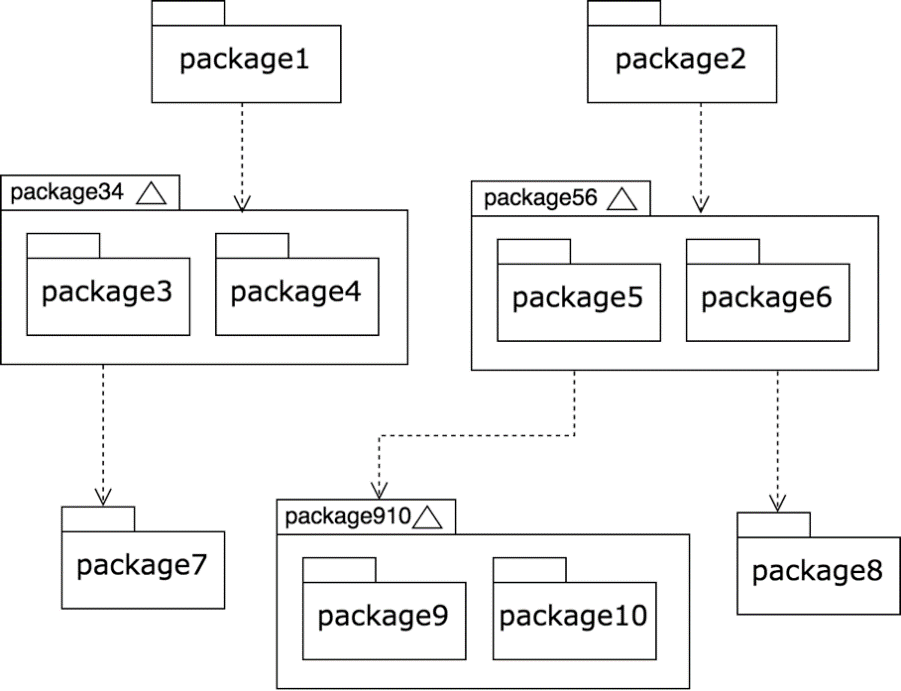
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Y** | **RS** | **G** | **AE** | **C** | **HRG**  **(%)** |
| **1** | 11.959 | 0 | 725 | 1619 | 0,0 |
| **2** | 10.457 | 0 | 474 | 1513 | 0,0 |
| **3** | 7.365 | 1213 | 335 | 966 | 16,7 |
| **4** | 900 | 599 | 145 | 405 | 55,3 |

Sinh viên được tùy ý lựa chọn định dạng (template) cho các bảng trong ĐATN của mình, nhưng phải đảm bảo tính nhất quán trên toàn ĐATN. Template bảng phải đảm bảo phần heading cột trong bảng có font in đậm và nổi bật (highlighted) hơn so với các nội dung khác trong bảng.

Vì bảng có thể kéo dài nhiều trang, tiêu đề của bảng nên để ở phần đầu của bảng. Sinh viên không thêm tiêu đề bảng bằng tay. Để thêm tiêu đề bảng tự động, sinh viên nhấn chuột phải vào bảng, chọn “Insert Caption”, chọn “Label” là “Bảng”, rồi nhấn nút “OK”. Sau đó, sinh viên nhập vào nội dung tiêu đề và căn chỉnh “Center” cho tiêu đề này. Lưu ý, sinh viên cần bôi đậm bằng tay cụm từ **Bảng n**. Kết quả thu được có dạng như sau “**Bảng 2** Ví dụ sử dụng bảng”. Sinh viên nên xoay ngang trang giấy trong trường hợp bảng có nhiều cột với nhiều nội dung văn bản.

Hình vẽ

Tương tự như bảng, sinh viên không được để hình vẽ tràn lề trang. Căn lề cho hình vẽ là căn giữa (Center). Cách thêm tiêu đề hình vẽ tương tự như cách thêm tiêu đề bảng, nhưng sinh viên chọn “Label” là “Hình” thay vì “Bảng”.



**Hình 3** Ví dụ hình vẽ

Tiêu đề hình vẽ phải đặt ở dưới hình vẽ. Nếu hình vẽ được copy từ trên mạng, sinh viên bắt buộc phải ghi rõ nguồn. Sinh viên nên thống nhất công cụ sử dụng và style cho hình vẽ trong toàn ĐATN. Các chi tiết trong hình vẽ phải được bố trí gọn gàng; chữ trong hình phải đảm bảo nhìn được rõ nét khi in báo cáo trên giấy A4. Khi resize ảnh, cần giữ nguyên tỷ lệ dài rộng, tránh làm méo hoặc vỡ hình. Ví dụ hình vẽ được minh họa trong Hình 3.

Tài liệu tham khảo

Sinh viên cần hạn chế tối đa dùng trang Web làm tài liệu tham khảo. Chỉ chấp nhận trang Web làm TLTK khi trang đó là nơi công bố chính thức của tổ chức hoặc cá nhân nào đó. Ví dụ, trang web đặc tả ngôn ngữ XML của tổ chức W3C https://www.w3.org/TR/2008/REC-xml-20081126/ là TLTK hợp lệ.

Khi giới thiệu về công cụ, API, thư viện, hoặc nền tảng nào đó, sinh viên có thể đưa ra địa chỉ URL của các tiện ích này. Sinh viên lưu ý địa chỉ URL đó không phải là tài liệu tham khảo. Trong các trường hợp tương tự như vậy, sinh viên nên tạo “Footnote”. Sinh viên tạo “Footnote” bằng cách vào mục “References”, chọn “Insert Footnote”. Ví dụ tạo Footnote như sau: TensorFlow[[1]](#footnote-1) là nền tảng học máy mã nguồn mở đang được sử dụng rộng rãi hiện nay. Lưu ý: số Footnote phải đặt sát với từ được mô tả. Như trong ví dụ trên, số 1 được đặt ngay cạnh chữ TensorFlow (không có dấu cách).

Công thức toán học

Giống như bảng, hình vẽ, và tài liệu tham khảo, công thức toán học cần được đánh số, giải thích, và tham chiếu đầy đủ.

Để thêm tiêu đề (caption) cho công thức, sinh viên đặt con trỏ văn bản vào dòng văn bản dưới công thức. Sau đó vào menu “Insert” (lưu ý là menu Insert, không phải là tab Insert trong thanh Ribbon) và chọn mục “Caption”. Từ Popup Menu hiện ra, sinh viên chọn “Label” là “Công thức”, rồi nhấn nút “OK”. Sinh viên bôi đậm chữ **Công thức n**, và căn tiêu đề ra giữa. Công thức 1 là ví dụ mẫu cho sinh viên tham khảo.

**Công thức 1** Khai triển Newton

Tham chiếu chéo

Tham chiếu chéo (Cross-reference) là tiện ích hữu hiệu cho người viết báo cáo. Nó giúp tạo các liên kết tham chiếu (hyperlink) tới các hình ảnh, bảng biểu, tài liệu tham khảo, và các đề mục một cách tự động. Ví dụ, ngay trong câu này, một tham chiếu đã được tạo ra tới mục 2.1. Người đọc dễ dàng nhấp chuột vào liên kết 2.1 để ngay lập tức chuyển đến mục đó.

Để tạo tham chiếu chéo tới các đề mục (heading), sinh viên vào tab “References”, rồi tìm và nhấn chọn “Cross-refrence”. SV chọn “Refrence type” là “Heading” và chọn “Insert Reference to” là “Heading number (no context)”. Sau đó, SV chọn phần đề mục muốn tham chiếu rồi bấm “Insert”.

Để tạo tham chiếu chéo tới các hình vẽ, bảng biểu và công thức, sinh viên cũng vào tab “References” và chọn “Cross-refrence”. SV chọn “Refrence type” là “Hình”, “Bảng”, hoặc “Công thức’. SV chọn “Insert Reference to” là “Only label and number”, trỏ đến phần muốn tham chiếu rồi bấm “Insert”. Nếu font chữ trong liên kết tham chiếu tạo ra được in đậm (bold), SV chuyển về dạng thường cho chuẩn tắc.

Thực hiện tương tự các bước mô tả ở trên, sinh viên có thể tạo tham chiếu chéo tới các tài liệu tham khảo. Sinh viên chọn “Refrence type” là “Numbered item”, chọn “Insert Reference to” là “Paragraph number”, trỏ đến phần tài liệu tham khảo muốn tham chiếu rồi bấm “Insert”. Ví dụ, tham chiếu chéo tới tài liệu tham khảo [2], [3], [4] đã được tạo.

Cập nhật mục lục và tham chiếu chéo

Trong suốt quá trình viết ĐATN, sinh viên sẽ tạo ra nhiều xáo trộn như thay đổi vị trí hình và bảng, thay đổi thứ tự tài liệu tham khảo, thay đổi tên đề mục, v.v. Vì vậy, để hoàn thành ĐATN, sinh viên cần cập nhật lại các thành phần mục lục, danh sách hình ảnh/bảng/công thức và tham chiếu chéo được tạo tự động trong ĐATN của mình.

Để cập nhật các thành phần này, sinh viên bấm Ctrl+A để chọn toàn báo cáo, nhấn chuột phải và chọn “Update Field”, rồi liên tục chọn mục “Entire table” khi được hỏi trong hộp thoại Popup Menu. Sau đó sinh viên search chữ “Error” trên toàn báo cáo để kiểm tra xem có lỗi đánh chỉ mục hoặc lỗi tham chiếu nào không. Đồng thời, sinh viên nên tự soát lại bằng tay toàn bộ nội dung quyển ĐATN để tránh mọi sai sót.

In quyển đồ án tốt nghiệp

Do hiện nay có nhiều phiên bản Word cho nhiều nền tảng máy tính, sinh viên nhất thiết phải xuất ĐATN ra định dạng PDF rồi mang tới cửa hàng in ấn để tránh sai sót.

Sinh viên chỉ cần in bìa giấy cứng mà không cần đóng quyển để tiết kiệm chi phí và giúp việc vận chuyển ĐATN dễ dàng hơn.

Quyển ĐATN nên được in một mặt trên các trang giấy A4.

Đặc tả use case

Nếu trong nội dung chính không đủ không gian cho các use case khác (ngoài các use case nghiệp vụ chính) thì đặc tả thêm cho các use case đó ở đây.

Đặc tả use case “Thống kê tình hình mượn sách”

Cách trình bày tương tự như phần hướng dẫn ở mục 2.3.

Đặc tả use case “Đăng ký làm thẻ mượn”

Cách trình bày tương tự như phần hướng dẫn ở mục 2.3.

Công nghệ sử dụng

Công nghệ bảo mật dữ liệu

Công nghệ blockchain

Thiết kế gói

Thiết kế gói cho kiến trúc tổng quan

Thiết kế gói cho chức năng “Trả sách”

Thiết kế lớp

1. https://www.tensorflow.org/, lần truy cập cuối: 28/06/2018 [↑](#footnote-ref-1)