

ĐẠI HỌC BÁCH KHOA HÀ NỘI

# ĐỒ ÁN TỐT NGHIỆP

**Xây dựng hệ thống quản lý vận hành và bán hàng tốc độ cao cho nhà hàng cơm bình dân**

**LÊ ĐỨC LƯƠNG**

luong.ld232006p@sis.hust.edu.vn

**Chương trình đào tạo: Công nghệ thông tin**

**Giảng viên hướng dẫn:** Ths. Nguyễn Mạnh Tuấn

**Khoa:** Khoa học máy tính

**Trường:** Công nghệ thông tin và Truyền thông

**HÀ NỘI, 01/2026**

ĐẠI HỌC BÁCH KHOA HÀ NỘI

# ĐỒ ÁN TỐT NGHIỆP

Xây dựng hệ thống quản lý vận hành và bán hàng tốc độ cao cho nhà hàng cơm bình dân

LÊ ĐỨC LƯƠNG

luong.ld232006p@sis.hust.edu.vn

Chương trình đào tạo: Công nghệ thông tin

Giảng viên hướng dẫn: ThS. Nguyễn Mạnh Tuấn

Chữ kí GVHD

Khoa: Khoa học máy tính

Trường: Công nghệ Thông tin và Truyền thông

HÀ NỘI, 01/2026

# LỜI CẢM ƠN

Trước hết, em xin gửi lời cảm ơn chân thành đến Ban lãnh đạo và các thầy cô giáo Trường Công nghệ Thông tin và Truyền thông - Đại học Bách Khoa Hà Nội đã tận tình giảng dạy và trang bị cho em những kiến thức quý báu trong suốt thời gian ngồi trên ghế nhà trường.

Đặc biệt, em xin bày tỏ lòng biết ơn sâu sắc tới thầy ***Nguyễn Mạnh Tuấn***. Thầy đã trực tiếp hướng dẫn, định hướng và đưa ra những lời khuyên chuyên môn thiết thực giúp em hoàn thiện đồ án.

Cuối cùng, em xin cảm ơn gia đình, người yêu và bạn bè đã luôn là chỗ dựa vững chắc, động viên em vượt qua những khó khăn trong quá trình thực hiện đồ án này. Dù đã rất cố gắng, đồ án khó tránh khỏi những thiếu sót, em rất mong nhận được sự đóng góp ý kiến của quý Thầy, Cô.

Một lần nữa, em xin chân thành cảm ơn!

# TÓM TẮT NỘI DUNG ĐỒ ÁN

Mô hình kinh doanh quán cơm bình dân, một nét đặc trưng của ẩm thực Việt Nam phục vụ phần lớn sinh viên và người lao động, hiện đang đối mặt với những thách thức lớn trong khâu vận hành và quản lý. Các quy trình thủ công như ghi chép sổ sách hay trí nhớ của nhân viên thường dẫn đến sai sót trong tính toán, chậm trễ trong phục vụ giờ cao điểm và khó khăn trong việc kiểm soát nguyên liệu, gây thất thoát tài chính. Mặc dù thị trường đã có nhiều phần mềm quản lý nhà hàng (POS), nhưng đa số được thiết kế cho mô hình gọi món tiêu chuẩn với quy trình phức tạp, chi phí cao và không đáp ứng được tính linh hoạt đặc thù của quán cơm bình dân như thực đơn thay đổi theo ngày hay tốc độ phục vụ cực nhanh. Nhận thấy khoảng trống này, đồ án tập trung phát triển một hệ thống quản lý chuyên biệt, tối ưu hóa cho quy trình bán hàng nhanh gọn và quản lý linh hoạt.

Cách tiếp cận được lựa chọn là xây dựng một ứng dụng web dựa trên kiến trúc Client-Server, sử dụng ReactJS cho giao diện người dùng (Frontend) và Spring Boot cho phía máy chủ (Backend), kết hợp với cơ sở dữ liệu PostgreSQL. Lựa chọn này đảm bảo khả năng tương thích đa nền tảng, dễ dàng triển khai và mở rộng, đồng thời cung cấp trải nghiệm người dùng mượt mà nhờ công nghệ Single Page Application (SPA).

Giải pháp tổng thể bao gồm việc xây dựng các phân hệ chức năng cốt lõi: một giao diện bán hàng (POS) tối giản cho phép nhân viên tạo đơn và thanh toán chỉ trong vài thao tác chạm; một hệ thống quản lý thực đơn động giúp chủ quán dễ dàng cập nhật món ăn theo ngày; và cơ chế quản lý tồn kho thông minh dựa trên định lượng suất ăn thay vì nguyên liệu thô. Bên cạnh đó, hệ thống còn cung cấp các báo cáo thống kê trực quan về doanh thu và xu hướng tiêu dùng, hỗ trợ chủ quán ra quyết định nhập hàng chính xác.

Sinh viên thực hiện  
(Ký và ghi rõ họ tên)

## MỤC LỤC

<b>CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI.....</b>	<b>1</b>
1.1 Đặt vấn đề.....	1
1.2 Mục tiêu và phạm vi đề tài.....	2
1.3 Định hướng giải pháp.....	2
1.4 Bố cục đồ án .....	3
<b>CHƯƠNG 2. KHẢO SÁT VÀ PHÂN TÍCH YÊU CẦU.....</b>	<b>4</b>
2.1 Khảo sát hiện trạng .....	4
2.2 Tổng quan chức năng .....	4
2.2.1 Biểu đồ use case tổng quát .....	5
2.2.2 Biểu đồ use case phân rã XYZ .....	6
2.2.3 Quy trình nghiệp vụ .....	6
2.3 Đặc tả chức năng .....	6
2.3.1 Đặc tả use case A.....	6
2.3.2 Đặc tả use case B .....	6
2.4 Yêu cầu phi chức năng .....	6
<b>CHƯƠNG 3. CÔNG NGHỆ SỬ DỤNG.....</b>	<b>7</b>
<b>CHƯƠNG 4. THIẾT KẾ, TRIỂN KHAI VÀ ĐÁNH GIÁ HỆ THỐNG ....</b>	<b>8</b>
4.1 Thiết kế kiến trúc.....	8
4.1.1 Lựa chọn kiến trúc phần mềm .....	8
4.1.2 Thiết kế tổng quan.....	8
4.1.3 Thiết kế chi tiết gói .....	8
4.2 Thiết kế chi tiết.....	9
4.2.1 Thiết kế giao diện .....	9
4.2.2 Thiết kế lớp .....	9

4.2.3 Thiết kế cơ sở dữ liệu .....	10
4.3 Xây dựng ứng dụng.....	10
4.3.1 Thư viện và công cụ sử dụng .....	10
4.3.2 Kết quả đạt được .....	10
4.3.3 Minh họa các chức năng chính .....	10
4.4 Kiểm thử.....	10
4.5 Triển khai .....	11
<b>CHƯƠNG 5. CÁC GIẢI PHÁP VÀ ĐÓNG GÓP NỔI BẬT.....</b>	<b>12</b>
<b>CHƯƠNG 6. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN .....</b>	<b>13</b>
6.1 Kết luận .....	13
6.2 Hướng phát triển.....	13
<b>TÀI LIỆU THAM KHẢO.....</b>	<b>16</b>
<b>PHỤ LỤC.....</b>	<b>18</b>
<b>A. HƯỚNG DẪN VIẾT ĐỒ ÁN TỐT NGHIỆP .....</b>	<b>18</b>
A.1 Ngành học.....	19
A.2 Đánh dấu (bullet) và đánh số (numering) .....	19
A.3 Cách thêm bảng .....	20
A.4 Chèn hình ảnh .....	20
A.5 Tài liệu tham khảo .....	21
A.6 Cách viết phương trình và công thức toán học.....	21
A.7 Qui cách đóng quyển.....	21
<b>B. ĐẶC TẢ USE CASE.....</b>	<b>23</b>
B.1 Đặc tả use case “Thống kê tình hình mượn sách” .....	23
B.2 Đặc tả use case “Đăng ký làm thẻ mượn” .....	23

## DANH MỤC HÌNH VẼ

Hình 2.1	Biểu đồ use case tổng quát . . . . .	5
Hình 4.1	Ví dụ biểu đồ phụ thuộc gói . . . . .	8
Hình 4.2	Ví dụ thiết kế gói . . . . .	9
Hình A.1	Internet vạn vật . . . . .	20
Hình A.2	Quy cách đóng quyển đồ án . . . . .	22
Hình A.3	Quy cách đóng quyển đồ án . . . . .	22

## **DANH MỤC BẢNG BIỂU**

Bảng 4.1	Danh sách thư viện và công cụ sử dụng . . . . .	10
Bảng A.1	Table to test captions and labels. . . . .	20



## DANH MỤC THUẬT NGỮ VÀ TỪ VIẾT TẮT

Thuật ngữ	Ý nghĩa
API	Giao diện lập trình ứng dụng (Application Programming Interface)
EUD	Phát triển ứng dụng người dùng cuối(End-User Development)
GWT	Công cụ lập trình Javascript bằng Java của Google (Google Web Toolkit)
HTML	Ngôn ngữ đánh dấu siêu văn bản (HyperText Markup Language)
IaaS	Dịch vụ hạ tầng

# CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI

## 1.1 Đặt vấn đề

Trong bối cảnh chuyển đổi số đang diễn ra mạnh mẽ tại Việt Nam, ngành dịch vụ ăn uống (F&B) là một trong những lĩnh vực tiên phong trong việc áp dụng công nghệ để tối ưu hóa vận hành. Các hệ thống quản lý nhà hàng (POS), đặt món trực tuyến hay quản lý chuỗi cung ứng đã trở nên phổ biến, giúp các doanh nghiệp nâng cao hiệu quả kinh doanh và trải nghiệm khách hàng. Tuy nhiên, phần lớn các giải pháp hiện có trên thị trường thường tập trung vào mô hình nhà hàng gọi món, quán cà phê hoặc chuỗi thức ăn nhanh với thực đơn cố định và quy trình phục vụ tiêu chuẩn.

Ngược lại, mô hình kinh doanh "cơm bình dân" – một nét đặc trưng trong văn hóa ẩm thực và đời sống của người Việt, đặc biệt phục vụ đối tượng nhân viên văn phòng và sinh viên – lại có những đặc thù riêng biệt mà các phần mềm quản lý thông thường chưa đáp ứng tốt. Các đặc thù này bao gồm: áp lực phục vụ cực lớn trong khung giờ cao điểm ngắn (thường chỉ từ 11h30 đến 12h30); thực đơn thay đổi linh hoạt theo ngày dựa trên nguyên liệu chợ đầu mối; và quy trình bán hàng yêu cầu tốc độ xử lý giao dịch tính bằng giây để tránh ùn tắc.

Hiện nay, đa số các quán cơm bình dân vẫn vận hành theo phương thức thủ công hoặc sử dụng các phần mềm POS không chuyên dụng. Việc này dẫn đến nhiều bất cập: Thứ nhất, tốc độ bán hàng bị kìm hãm do nhân viên phải thực hiện nhiều thao tác chọn món phức tạp trên các giao diện không tối ưu, gây ùn tắc cục bộ vào giờ cao điểm. Thứ hai, việc quản lý tồn kho gặp khó khăn do không thể kiểm soát chính xác lượng "suất ăn" đã nấu so với nguyên liệu đầu vào, dẫn đến tình trạng thiếu món cục bộ hoặc dư thừa thức ăn gây lãng phí vào cuối ngày. Thứ ba, chủ kinh doanh thiếu các dữ liệu thống kê chính xác về thói quen ăn uống của khách hàng và xu hướng tiêu thụ từng món theo ngày để đưa ra quyết định nhập hàng tối ưu cho ngày hôm sau.

Việc thiếu vắng một công cụ quản lý chuyên biệt, có khả năng thích ứng với sự thay đổi thực đơn hàng ngày và đảm bảo tốc độ xử lý giao dịch cao, đang là rào cản lớn đối với sự phát triển và chuyên nghiệp hóa của mô hình kinh doanh này. Xuất phát từ nhu cầu thực tế đó, việc nghiên cứu và xây dựng một hệ thống quản lý đặc thù, tập trung vào tốc độ và sự linh hoạt cho mô hình cơm bình dân là một yêu cầu cấp thiết, mang lại giá trị thực tiễn cao cho cả người quản lý, nhân viên vận hành và khách hàng.

## 1.2 Mục tiêu và phạm vi đề tài

Trên thị trường hiện nay, các giải pháp quản lý nhà hàng phổ biến thường được thiết kế tối ưu cho mô hình gọi món tiêu chuẩn hoặc chuỗi thức ăn nhanh, dẫn đến tình trạng dư thừa tính năng nhưng lại thiếu đi sự linh hoạt cần thiết khi áp dụng vào đặc thù của quán cơm bình dân. Xuất phát từ thực tế đó, mục tiêu cốt lõi của đề tài là nghiên cứu và xây dựng một hệ thống quản lý chuyên biệt, tập trung giải quyết triệt để bài toán về tốc độ xử lý giao dịch trong khung giờ cao điểm và cơ chế quản lý thực đơn thay đổi theo ngày. Hệ thống hướng tới việc thay thế các phương thức ghi chép thủ công hoặc các phần mềm không chuyên, giúp chủ cửa hàng chuẩn hóa quy trình vận hành và kiểm soát hiệu quả nguồn lực kinh doanh.

Về phạm vi thực hiện, đồ án tập trung phát triển một ứng dụng web hoàn chỉnh bao gồm các phân hệ chức năng gắn liền với quy trình nghiệp vụ thực tế. Cụ thể, đối với nghiệp vụ bán hàng, hệ thống cung cấp giao diện POS tối giản nhằm giảm thiểu thao tác và rút ngắn thời gian thanh toán. Song song với đó, phân hệ quản trị được xây dựng với khả năng thiết lập thực đơn động linh hoạt và cơ chế quản lý tồn kho dựa trên số lượng suất ăn thành phẩm thay vì nguyên liệu thô. Ngoài ra, phạm vi đề tài cũng bao gồm việc xây dựng cổng thông tin cho phép khách hàng tra cứu thực đơn và đặt món trực tuyến, cùng với hệ thống báo cáo thống kê hỗ trợ chủ cửa hàng theo dõi doanh thu và xu hướng tiêu dùng theo thời gian thực.

## 1.3 Định hướng giải pháp

Để đảm bảo khả năng truy cập đa nền tảng và dễ dàng triển khai, hệ thống được xây dựng theo kiến trúc Client-Server. Cụ thể, phía máy khách (Client) sẽ được phát triển dưới dạng ứng dụng đơn trang (Single Page Application - SPA) nhằm tối ưu hóa trải nghiệm người dùng, giảm thiểu thời gian tải trang và tăng tốc độ tương tác – yếu tố then chốt cho môi trường phục vụ cường độ cao. Phía máy chủ (Server) sẽ được xây dựng theo mô hình RESTful API, đảm bảo tính tách biệt giữa giao diện và logic xử lý, thuận tiện cho việc mở rộng hoặc tích hợp với các hệ thống khác trong tương lai. Giải pháp cụ thể của đồ án tập trung vào việc xử lý hai bài toán cốt lõi là tốc độ và sự linh hoạt thông qua các kỹ thuật sau. Thứ nhất, đối với quy trình bán hàng, hệ thống sẽ tối ưu hóa giao diện người dùng (UI/UX) và luồng xử lý dữ liệu để giảm thiểu số bước thao tác, đồng thời áp dụng các cơ chế caching (lưu đệm) phía client để tăng tốc độ phản hồi ngay cả khi đường truyền mạng không ổn định. Thứ hai, đối với vấn đề thực đơn động và quản lý tồn kho, hệ thống sẽ thiết kế cơ sở dữ liệu linh hoạt cho phép định nghĩa món ăn theo ngày và xây dựng thuật toán tự động trừ kho dựa trên định lượng suất ăn đã được cấu hình trước, giúp cập nhật trạng thái "hết hàng" theo thời gian thực lên giao diện bán hàng và trang đặt

món của khách hàng. Đóng góp chính của đề án là cung cấp một giải pháp phần mềm quản lý toàn diện và chuyên biệt cho mô hình kinh doanh cơm bình dân, một phân khúc thị trường lớn nhưng chưa được phục vụ thỏa đáng bởi các giải pháp công nghệ hiện có. Kết quả đạt được là một hệ thống hoàn chỉnh có khả năng vận hành thực tế, giúp chủ cửa hàng chuẩn hóa quy trình, kiểm soát chặt chẽ nguồn lực và nâng cao chất lượng phục vụ, đồng thời tạo tiền đề dữ liệu cho các phân tích kinh doanh sâu hơn trong tương lai.

### 1.4 Bố cục đề án

Phần còn lại của báo cáo đề án tốt nghiệp này được tổ chức thành năm chương với nội dung cụ thể như sau.

Chương 2 tập trung vào việc khảo sát và phân tích yêu cầu của hệ thống. Trong chương này, em sẽ trình bày kết quả khảo sát thực tế về quy trình vận hành tại các quán cơm bình dân, từ đó xác định các tác nhân tham gia và xây dựng biểu đồ Use Case tổng quát cũng như các kịch bản nghiệp vụ chi tiết. Các yêu cầu phi chức năng về hiệu năng, bảo mật và tính khả dụng cũng được phân tích kỹ lưỡng tại đây làm cơ sở cho việc thiết kế hệ thống.

Chương 3 giới thiệu về các công nghệ và nền tảng được sử dụng để phát triển ứng dụng. Nội dung chương sẽ phân tích lý do lựa chọn kiến trúc Single Page Application (SPA) với ReactJS cho phía máy khách và Spring Boot cho phía máy chủ, cùng với hệ quản trị cơ sở dữ liệu PostgreSQL. Các công nghệ bổ trợ như WebSocket cho giao tiếp thời gian thực hay Redis cho cơ chế caching cũng được trình bày nhằm làm rõ tính phù hợp với bài toán đặt ra.

Chương 4 trình bày chi tiết về quá trình thiết kế, hiện thực hóa và đánh giá hệ thống. Nội dung bao gồm các thiết kế kiến trúc tổng thể, thiết kế cơ sở dữ liệu, và thiết kế giao diện người dùng. Phần này cũng mô tả quá trình xây dựng các chức năng cốt lõi như quản lý thực đơn động, xử lý đơn hàng tốc độ cao, và trình bày kết quả kiểm thử hệ thống để chứng minh tính đúng đắn và hiệu quả của giải pháp.

Chương 5 tổng hợp các đóng góp nổi bật và các giải pháp kỹ thuật đặc thù đã áp dụng để giải quyết bài toán nghiệp vụ. Tại đây, em sẽ đi sâu phân tích thuật toán quản lý tồn kho theo suất ăn và cơ chế tối ưu hóa thao tác người dùng trên giao diện bán hàng, những điểm làm nên sự khác biệt của hệ thống so với các sản phẩm đại trà.

Cuối cùng, Chương 6 đưa ra kết luận chung về những kết quả đã đạt được so với mục tiêu ban đầu, đồng thời chỉ ra những hạn chế còn tồn tại và đề xuất các hướng phát triển trong tương lai để hoàn thiện sản phẩm.

## CHƯƠNG 2. KHẢO SÁT VÀ PHÂN TÍCH YÊU CẦU

### 2.1 Khảo sát hiện trạng

Dựa trên thông tin khảo sát hiện trạng tại các quán cơm bình dân trên địa bàn Hà Nội, chúng tôi nhận thấy đa số các cơ sở vẫn vận hành theo quy trình truyền thống: khách hàng xếp hàng, trực tiếp chọn món tại quầy, nhân viên phục vụ mức đồ và báo giá ngay tại chỗ. Việc thanh toán thường diễn ra sau khi khách ăn xong dựa trên trí nhớ hoặc thẻ tạm. Quy trình này bộc lộ rõ những hạn chế vào giờ cao điểm, đặc biệt là tình trạng ùn tắc do tốc độ phục vụ chậm, rủi ro thất thoát doanh thu do sai sót trong tính toán, và khó khăn trong việc kiểm soát định lượng nguyên liệu tiêu hao so với số lượng suất ăn thực tế.

Qua phân tích các giải pháp phần mềm quản lý nhà hàng phổ biến như iPOS, CukCuk, và KiotViet, mặc dù các hệ thống này sở hữu tính năng mạnh mẽ và chuyên nghiệp, nhưng khi áp dụng vào mô hình cơm bình dân lại bộc lộ những bất cập nhất định. iPOS có chi phí đầu tư cao và quy trình thiết lập phức tạp, không phù hợp với quy mô nhỏ; CukCuk dù thân thiện hơn nhưng vẫn thiên về quản lý thực đơn cố định, gây khó khăn khi thay đổi món ăn hàng ngày; KiotViet mạnh về bán lẻ nhưng chưa tối ưu cho việc quản lý định lượng suất ăn chế biến với sự biến động nguyên liệu thường xuyên.

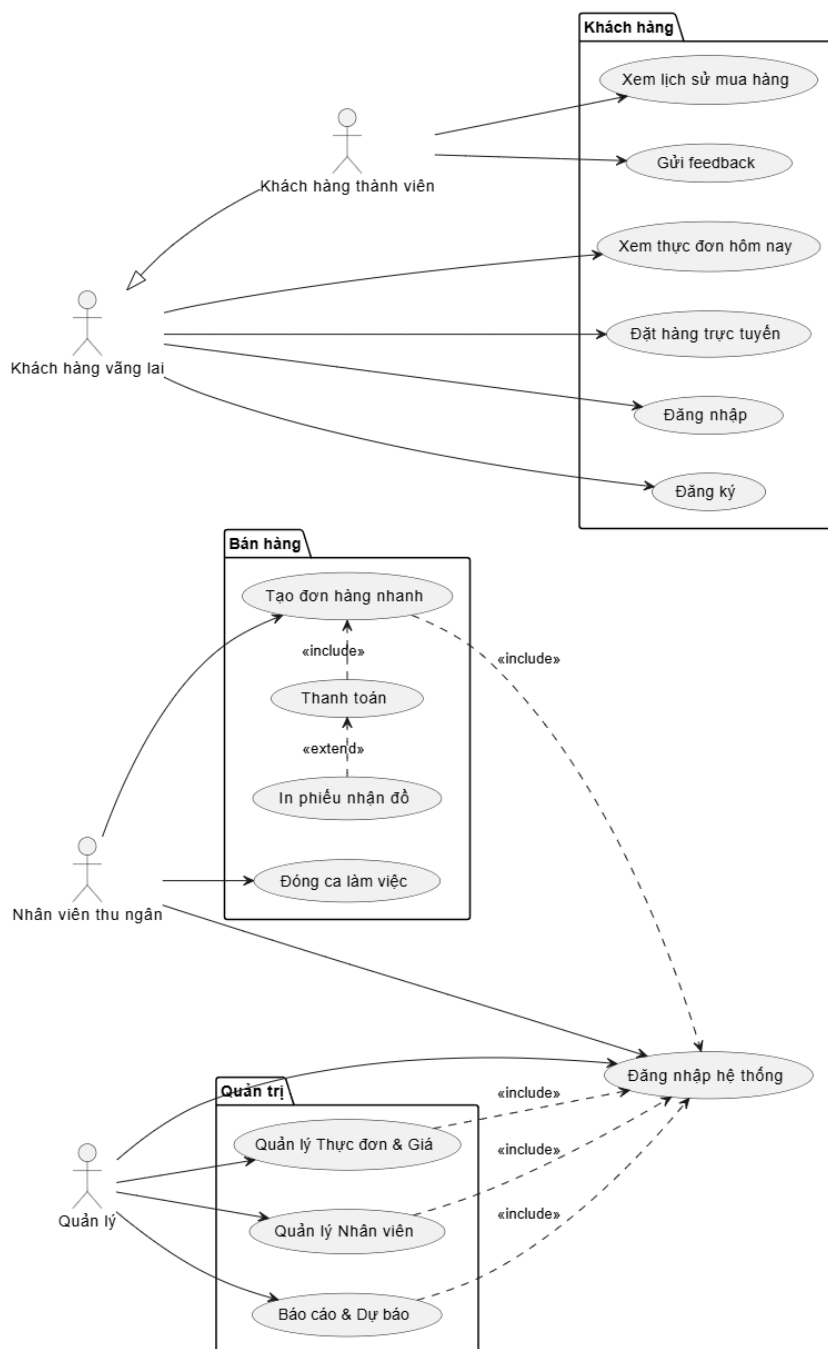
Từ những phân tích trên, có thể khẳng định nhu cầu cấp thiết về một hệ thống quản lý chuyên biệt cho mô hình cơm bình dân. Hệ thống này cần tập trung giải quyết triệt để bài toán về tốc độ xử lý giao dịch (dưới 30 giây/đơn), đơn giản hóa tối đa quy trình quản lý thực đơn động theo ngày, và cung cấp cơ chế kiểm soát tồn kho suất ăn trực quan, hiệu quả, giúp chủ quán tối ưu hóa vận hành và giảm thiểu thất thoát.

### 2.2 Tổng quan chức năng

Sau khi thu thập ý kiến của các chủ quán ăn bình dân, bạn bè (những người đã ăn những quán ăn ở xung quanh Hà Nội nhiều năm), cũng như những trải nghiệm thực tế của bản thân sau nhiều năm thưởng thức đồ ăn tại các quán ăn bình dân, em thiết kế hệ thống nhằm đáp ứng các yêu cầu nghiệp vụ cốt lõi của một quán cơm bình dân, tập trung vào việc tối ưu hóa quy trình bán hàng và quản lý vận hành. Về mặt tổng thể, hệ thống cung cấp các nhóm chức năng chính bao gồm quản lý bán hàng tại quầy, quản lý thực đơn và kho hàng, báo cáo thống kê, và cổng thông tin dành cho khách hàng. Đối với nhân viên thu ngân, hệ thống cung cấp công cụ tạo đơn hàng nhanh thông qua việc lựa chọn các gói giá định sẵn, hỗ trợ thanh toán đa phương thức và in hóa đơn tự động, giúp rút ngắn tối đa thời gian phục vụ trong

giờ cao điểm. Song song với đó, bộ phận quản lý được trang bị các công cụ để thiết lập thực đơn theo ngày, tùy chỉnh giá bán, và theo dõi biến động nguyên vật liệu dựa trên số lượng suất ăn tiêu thụ. Ngoài ra, hệ thống còn tích hợp khả năng dự báo nhu cầu nhập hàng dựa trên dữ liệu lịch sử và cung cấp các báo cáo trực quan về doanh thu, hiệu suất bán hàng, giúp chủ cửa hàng có cái nhìn toàn diện và đưa ra các quyết định kinh doanh chính xác.

### 2.2.1 Biểu đồ use case tổng quát



**Hình 2.1:** Biểu đồ use case tổng quát

### **2.2.2 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.

### **2.2.3 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.

## **2.3 Đặc tả chức năng**

Sinh viên lựa chọn từ 4 đến 7 use case quan trọng nhất của đề 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.

### **2.3.1 Đặc tả use case A**

### **2.3.2 Đặc tả use case B**

## **2.4 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.

### CHƯƠNG 3. CÔNG NGHỆ SỬ DỤNG

Chương này có độ dài không quá 10 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 đề 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 đề án. Sinh viên cũng có thể trình bày thêm nền tảng lý thuyết nào đó nếu cần dùng tới.

Với đề 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 trước đó.

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).



## CHƯƠNG 4. THIẾT KẾ, TRIỂN KHAI VÀ ĐÁNH GIÁ HỆ THỐNG

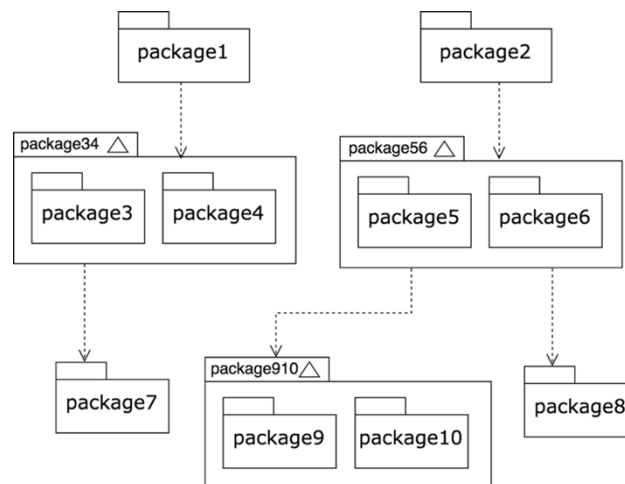
### 4.1 Thiết kế kiến trúc

#### 4.1.1 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.

#### 4.1.2 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 4.1



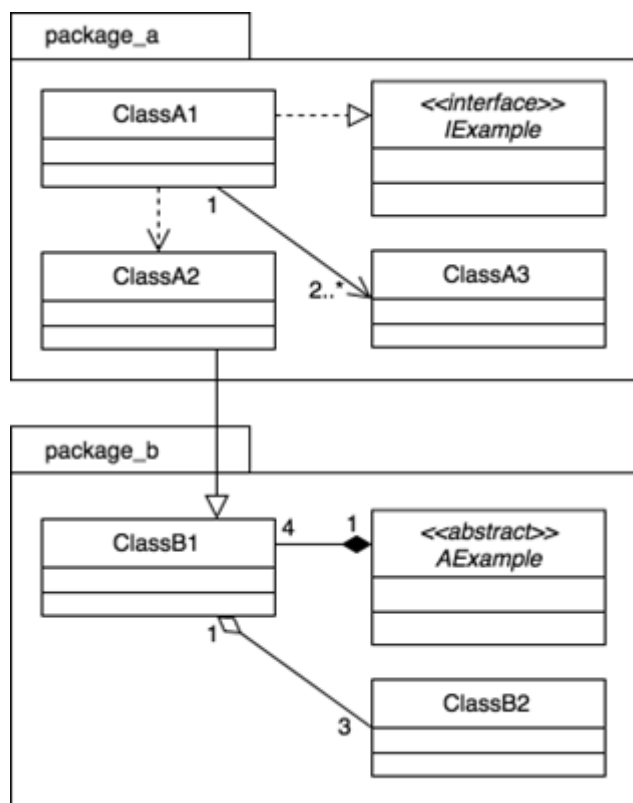
**Hình 4.1:** Ví dụ biểu đồ phụ thuộc gói

#### 4.1.3 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 4.2.

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 4.2.

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



Hình 4.2: Ví dụ thiết kế gói

## 4.2 Thiết kế chi tiết

### 4.2.1 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.

### 4.2.2 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).

### 4.2.3 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.)

## 4.3 Xây dựng ứng dụng

### 4.3.1 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 ???. Nếu có nhiều nội dung trình bày, sinh viên cần xoay ngang bảng.

Mục đích	Công cụ	Địa chỉ URL
IDE lập trình	Eclipse Oxygen a64 bit	<a href="http://www.eclipse.org/">http://www.eclipse.org/</a>
v.v.	v.v.	v.v.

**Bảng 4.1:** Danh sách thư viện và công cụ sử dụng

### 4.3.2 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.

### 4.3.3 Minh họa 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.

## 4.4 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.

### **4.5 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.)

## CHƯƠNG 5. CÁC GIẢI PHÁP VÀ ĐÓNG GÓP NỔI BẬT

Chương này có độ dài tối thiểu 5 trang, tối đa không giới hạn.<sup>1</sup> 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”.

---

<sup>1</sup>Trong trường hợp phần này dưới 5 trang thì sinh viên nên gộp vào phần kết luận, không tách ra một chương riêng rẽ nữa.

## **CHƯƠNG 6. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN**

### **6.1 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.

### **6.2 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.

## MỘT SỐ LƯU Ý VỀ 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] E. H. Hovy, "Automated discourse generation using discourse structure relations," *Artificial intelligence*, vol. 63, no. 1-2, pp. 341–385, 1993

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

[2] L. L. Peterson and B. S. Davie, *Computer networks: a systems approach*. Elsevier, 2007.

[3] N. T. 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>

[4] M. Poesio and B. Di Eugenio, "Discourse structure and anaphoric accessibility," in *ESSLLI workshop on information structure, discourse structure and discourse semantics*, Copenhagen, Denmark, 2001, pp. 129–143.

<**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>

[5] A. Knott, "A data-driven methodology for motivating a set of coherence relations," Ph.D. dissertation, The 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>

[6] T. Berners-Lee, *Hypertext transfer protocol (HTTP)*. [Online]. Available:

`ftp://info.cern.ch/pub/www/doc/http-spec.txt.Z` (visited on 09/30/2010).

[7] Princeton University, *Wordnet*. [Online]. Available: `http://www.cogsci.princeton.edu/~wn/index.shtml` (visited on 09/30/2010).



## TÀI LIỆU THAM KHẢO

- [1] E. H. Hovy, “Automated discourse generation using discourse structure relations,” *Artificial intelligence*, vol. 63, no. 1-2, pp. 341–385, 1993.
- [2] L. L. Peterson and B. S. Davie, *Computer networks: a systems approach*. Elsevier, 2007.
- [3] N. T. Hải, *Mạng máy tính và các hệ thống mở*. Nhà xuất bản giáo dục, 1999.
- [4] M. Poesio and B. Di Eugenio, “Discourse structure and anaphoric accessibility,” in *ESSLLI workshop on information structure, discourse structure and discourse semantics, Copenhagen, Denmark*, 2001, pp. 129–143.
- [5] A. Knott, “A data-driven methodology for motivating a set of coherence relations,” Ph.D. dissertation, The University of Edinburgh, UK, 1996.
- [6] T. Berners-Lee, *Hypertext transfer protocol (HTTP)*. [Online]. Available: <ftp://info.cern.ch/pub/www/doc/http-spec.txt>. Z (visited on 09/30/2010).
- [7] Princeton University, *Wordnet*. [Online]. Available: <http://www.cogsci.princeton.edu/~wn/index.shtml> (visited on 09/30/2010).

# PHỤ LỤC

## A. 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/choi 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.

### A.1 Ngành học

Sinh viên lưu ý viết đúng ngành/chuyên ngành trên bìa và trên gáy theo đúng quy định của Trường. Ngành học hay chuyên ngành học phụ thuộc vào ngành học mà sinh viên đăng ký. Sinh viên có thể đăng nhập trên trang quản lý học tập của mình để xem lại chính xác ngành học của mình.

Một số ví dụ sinh viên có thể tham khảo dưới đây, trong trường hợp có chuyên ngành thì sinh viên không cần ghi chuyên ngành:

- Đối với kỹ sư chính quy:
  - Từ K61 trở về trước: Ngành Kỹ thuật phần mềm
  - Từ K62 trở về sau: Ngành Khoa học máy tính
- Đối với cử nhân:
  - Ngành Công nghệ thông tin
- Đối với chương trình EliteTech:
  - Chương trình Việt Nhật/KSTN: Ngành Công nghệ thông tin
  - Chương trình ICT Global: Ngành Information Technology
  - Chương trình DS&AI: Ngành Khoa học dữ liệu

### A.2 Đánh dấu (bullet) và đánh số (numering)

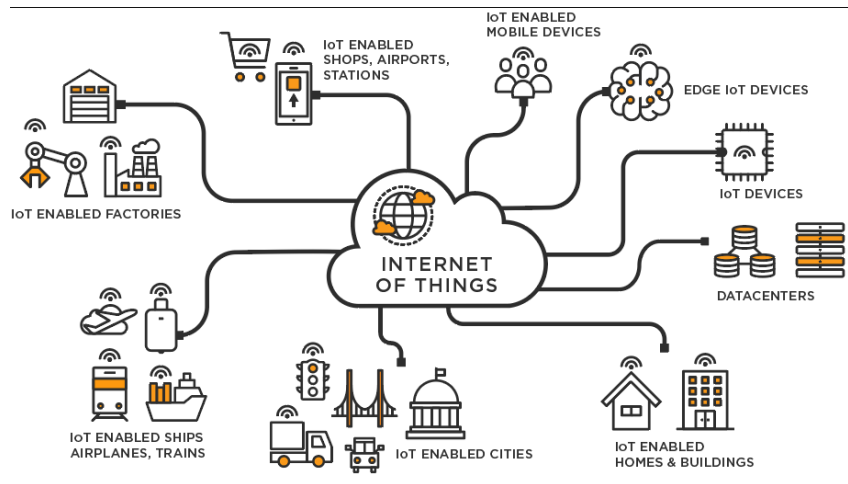
Việc sử dụng danh sách trong LaTeX khá đơn giản và không yêu cầu sinh viên phải thêm bất kỳ gói bổ sung nào. LaTeX cung cấp hai môi trường liệt kê đó là:

- Đánh dấu (bullet) là kiểu liệt kê không có thứ tự. Để sử dụng kiểu liệt kê đánh dấu, chúng ta khai báo như sau

```
\begin{itemize}
\item Nội dung thứ nhất được viết ở đây.
\item Nội dung thứ hai được viết ở đây.
\item ...
\end{itemize}
```

- Đánh số (numering) là kiểu liệt kê có thứ tự. Để sử dụng kiểu liệt kê đánh số, chúng ta khai báo như sau

```
\begin{enumerate}
\item Nội dung thứ nhất được viết ở đây.
\item Nội dung thứ hai được viết ở đây.
\item ...
\end{enumerate}
```



**Hình A.1:** Internet vạn vật

`\end{enumerate}`

Chú ý các nội dung trình bày trong cả hai môi trường liệt kê theo sau lệnh `\item`. Ngoài ra LaTeX còn cung cấp một số kiểu liệt kê khác, sinh viên có thể tham khảo tại <https://www.overleaf.com/learn/latex/Lists>

### A.3 Cách thêm bảng

Col1	Col2	Col2	Col3
1	6	87837	787
2	7	78	5415
3	545	778	7507
4	545	18744	7560
5	88	788	6344

**Bảng A.1:** Table to test captions and labels.

Bảng A.1 là ví dụ về cách tạo bảng. Tất cả các bảng biểu phải được đề cập đến trong phần nội dung và phải được phân tích và bình luận. Chú ý: Tạo bảng trong Latex khá phức tạp và mất thời gian, vì vậy sinh viên có thể sử dụng các công cụ hỗ trợ tạo bảng (Ví dụ: <https://www.tablesgenerator.com/>). Sinh viên có thể tìm hiểu sâu hơn về cách chèn ảnh trong Latex tại link <https://www.overleaf.com/learn/latex/Tables>.

### A.4 Chèn hình ảnh

Hình A.1 là ví dụ về cách chèn ảnh. Lưu ý chú thích của hình vẽ được đặt ngay dưới hình vẽ. Sinh viên có thể tìm hiểu sâu hơn về cách chèn ảnh trong Latex tại [https://www.overleaf.com/learn/latex/Inserting\\_Images](https://www.overleaf.com/learn/latex/Inserting_Images).

Chú ý, tất cả các hình vẽ phải được đề cập đến trong phần nội dung và phải được

phân tích và bình luận.

## A.5 Tài liệu tham khảo

### Cách liệt kê

Áp dụng cách liệt kê theo quy định của IEEE. Ví dụ của việc trích dẫn như sau **scott2013sdn**. Cụ thể, sinh viên sử dụng lệnh `\cite{}` như sau **ashton2009internet**. Chỉ những tài liệu được trích dẫn thì mới xuất hiện trong phần Tài liệu tham khảo. Tài liệu tham khảo cần có nguồn gốc rõ ràng và phải từ nguồn đáng tin cậy. Hạn chế trích dẫn tài liệu tham khảo từ các website, từ wikipedia.

### Các loại tài liệu tham khảo

Các nguồn tài liệu tham khảo chính là sách, bài báo trong các tạp chí, bài báo trong các hội nghị khoa học và các tài liệu tham khảo khác trên internet.

## A.6 Cách viết phương trình và công thức toán học

Các gói `amsmath`, `amssymb`, `amsfonts` hỗ trợ viết phương trình/công thức toán học đã được bổ sung sẵn ở phần đầu của file `main.tex`. Một ví dụ về tạo phương trình (A.1) như sau

$$F(x) = \int_b^a \frac{1}{3}x^3 \quad (\text{A.1})$$

Phương trình A.1 là ví dụ về phương trình tích phân. Một phương trình khác không được đánh số thứ tự (gán nhãn)

$$x[t_n] = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} X[f_k] e^{j2\pi nk/N}$$

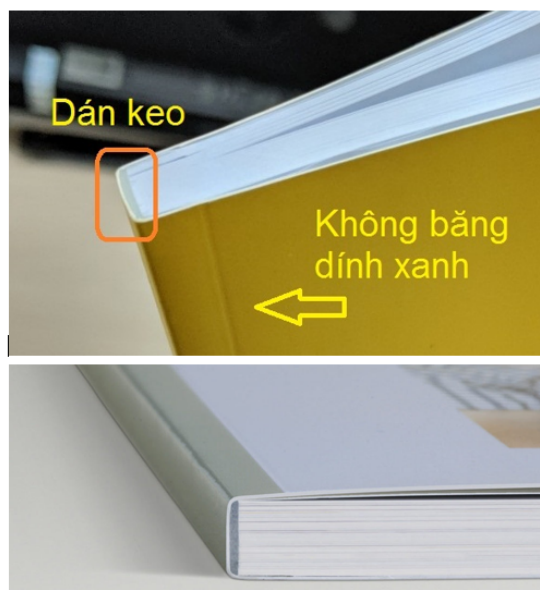
Phương trình này thể hiện phép biến đổi Fourier rời rạc ngược (IDFT).

## A.7 Qui cách đóng quyển

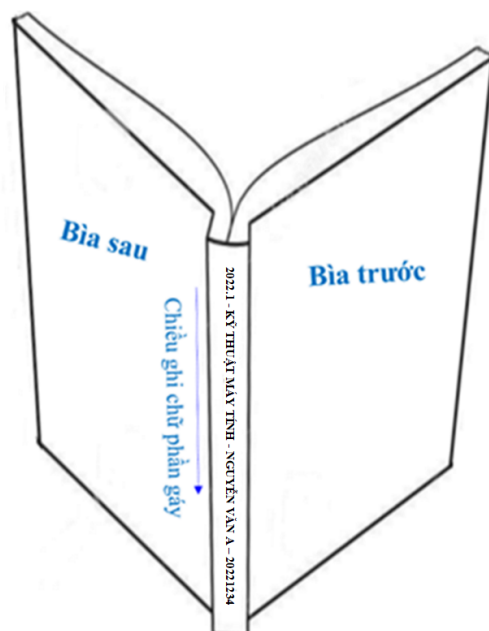
Phần bìa trước chế bản theo qui định; bìa trước và bìa sau là giấy liền khổ. Sử dụng keo nhiệt để dán gáy khi đóng quyển thay vì sử dụng băng dính và dập ghim như mô tả ở Hình A.3 Phần gáy ĐATN cần ghi các thông tin tóm tắt sau: Kỳ làm ĐATN - Ngành đào tạo - Họ và tên sinh viên - Mã số sinh viên. Ví dụ:

2022.1 - KỸ THUẬT MÁY TÍNH - NGUYỄN VĂN A - 20221234

Qui cách ghi chữ phần gáy như hình dưới đây:



**Hình A.2:** Quy cách đóng quyển đồ án



**Hình A.3:** Quy cách đóng quyển đồ án

## **B. 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.

### **B.1 Đặc tả use case “Thông kê tình hình mượn sách”**

...

### **B.2 Đặc tả use case “Đăng ký làm thẻ mượn”**

...