

ĐẠI HỌC BÁCH KHOA HÀ NỘI

## ĐỒ ÁN TỐT NGHIỆP

Thiết kế xây dựng công nghệ thực tế ảo và ứng dụng

NGUYỄN VĂN A

nguyenvanabc@sis.hust.edu.vn

Chương trình đào tạo: Công nghệ thông tin Việt-Nhật

Giảng viên hướng dẫn: PGS. TS. Phạm Văn ABC

Chữ kí GVHD

Khoa:

Kỹ thuật máy tính

Trường:

Công nghệ Thông tin và Truyền thông

HÀ NỘI, 06/2022

## 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, nên viết phần cảm ơn ngắn gọn, tránh dùng các từ sáo rỗng, giới hạn trong khoảng 100-150 từ.

## TÓM TẮT NỘI DUNG ĐỒ ÁN

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.

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

## **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

<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.....	3
1.4 Bố cục đồ án .....	5
<b>CHƯƠNG 2. KHẢO SÁT VÀ PHÂN TÍCH YÊU CẦU.....</b>	<b>10</b>
2.1 Khảo sát hiện trạng .....	10
2.1.1 Bối cảnh và xu hướng thị trường.....	10
2.1.2 Phân tích các ứng dụng giao đồ ăn hiện có .....	11
2.1.3 Nhu cầu và thách thức của người dùng .....	13
2.2 Tổng quan chức năng .....	15
2.2.1 Biểu đồ use case tổng quát .....	15
2.2.2 Biểu đồ use case phân rã XYZ .....	18
2.2.3 Quy trình nghiệp vụ .....	19
2.3 Đặc tả chức năng .....	19
2.3.1 Đặc tả use case A.....	19
2.3.2 Đặc tả use case B .....	19
2.4 Yêu cầu phi chức năng .....	19
<b>CHƯƠNG 3. CÔNG NGHỆ SỬ DỤNG.....</b>	<b>20</b>
<b>CHƯƠNG 4. THIẾT KẾ, TRIỂN KHAI VÀ ĐÁNH GIÁ HỆ THỐNG ....</b>	<b>21</b>
4.1 Thiết kế kiến trúc.....	21
4.1.1 Lựa chọn kiến trúc phần mềm .....	21
4.1.2 Thiết kế tổng quan.....	21
4.1.3 Thiết kế chi tiết gói .....	21

4.2 Thiết kế chi tiết.....	22
4.2.1 Thiết kế giao diện .....	22
4.2.2 Thiết kế lớp .....	22
4.2.3 Thiết kế cơ sở dữ liệu .....	23
4.3 Xây dựng ứng dụng.....	23
4.3.1 Thư viện và công cụ sử dụng .....	23
4.3.2 Kết quả đạt được .....	23
4.3.3 Minh họa các chức năng chính .....	23
4.4 Kiểm thử.....	23
4.5 Triển khai .....	24
<b>CHƯƠNG 5. CÁC GIẢI PHÁP VÀ ĐÓNG GÓP NỔI BẬT.....</b>	<b>25</b>
<b>CHƯƠNG 6. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN .....</b>	<b>26</b>
6.1 Kết luận .....	26
6.2 Hướng phát triển.....	26
<b>TÀI LIỆU THAM KHẢO.....</b>	<b>29</b>
<b>PHỤ LỤC.....</b>	<b>31</b>
<b>A. HƯỚNG DẪN VIẾT ĐỒ ÁN TỐT NGHIỆP .....</b>	<b>31</b>
A.1 Ngành học.....	32
A.2 Đánh dấu (bullet) và đánh số (numering) .....	32
A.3 Cách thêm bảng .....	33
A.4 Chèn hình ảnh .....	33
A.5 Tài liệu tham khảo .....	34
A.6 Cách viết phương trình và công thức toán học.....	34
A.7 Qui cách đóng quyển.....	34
<b>B. ĐẶC TẢ USE CASE.....</b>	<b>36</b>
B.1 Đặc tả use case “Thống kê tình hình mượn sách” .....	36

B.2 Đặc tả use case “Đăng ký làm thẻ mượn” .....	36
--------------------------------------------------	----





## DANH MỤC HÌNH VẼ

Hình 2.1	Biểu đồ use case tổng quát của hệ thống YummyZoom . . .	16
Hình 4.1	Ví dụ biểu đồ phụ thuộc gói . . . . .	21
Hình 4.2	Ví dụ thiết kế gói . . . . .	22
Hình A.1	Internet vạn vật . . . . .	33
Hình A.2	Quy cách đóng quyển đồ án . . . . .	35
Hình A.3	Quy cách đóng quyển đồ án . . . . .	35

## **DANH MỤC BẢNG BIỂU**

Bảng 4.1	Danh sách thư viện và công cụ sử dụng . . . . .	23
Bảng A.1	Table to test captions and labels. . . . .	33

## DANH MỤC THUẬT NGỮ VÀ TỪ VIẾT TẮT

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 cuộc cách mạng công nghiệp 4.0, lĩnh vực công nghệ thực phẩm (FoodTech), đặc biệt là các nền tảng giao đồ ăn trực tuyến, đã trải qua sự phát triển mạnh mẽ và trở thành một phần không thể thiếu trong đời sống hiện đại. Theo báo cáo của Momentum Works, thị trường giao đồ ăn tại Việt Nam đã đạt mức tăng trưởng 26% trong năm 2024, cao nhất khu vực Đông Nam Á, với tổng giá trị giao dịch (GMV) tăng từ 1.4 tỷ USD vào năm 2023 lên 1.8 tỷ USD vào năm 2024. Sự tăng trưởng này được thúc đẩy bởi sự phổ biến của điện thoại thông minh, lối sống bận rộn của người dân đô thị và nhu cầu ngày càng cao về sự tiện lợi trong ăn uống. Kết quả khảo sát cho thấy có đến 30% người dùng lựa chọn hình thức đặt đồ ăn qua ứng dụng cho bữa trưa, chứng tỏ thói quen này đã trở nên phổ biến.

Hiện tại, thị trường giao đồ ăn trực tuyến tại Việt Nam đang bị thống trị bởi hai nền tảng lớn là GrabFood và ShopeeFood, tạo nên một thế song cực với thị phần lần lượt là 48% và 47%. Hai nền tảng này cạnh tranh khốc liệt thông qua các chiến dịch khuyến mãi lớn, xây dựng hệ sinh thái dịch vụ đa dạng và phát triển các chương trình khách hàng thân thiết. GrabFood tận dụng lợi thế từ hệ sinh thái "siêu ứng dụng" Grab với các dịch vụ đi chuyển, giao hàng và thanh toán điện tử, cùng với chương trình GrabRewards và gói thành viên GrabUnlimited. Trong khi đó, ShopeeFood khai thác hiệu quả mô hình tích hợp với nền tảng thương mại điện tử Shopee và văn hóa "săn sale" của người tiêu dùng Việt Nam thông qua các chương trình voucher và khuyến mãi liên tục.

Tuy nhiên, bên cạnh những thành công đạt được, các nền tảng hiện tại vẫn chưa giải quyết triệt để một vấn đề quan trọng đối với một nhóm người dùng cụ thể. Đó là nhóm sinh viên đại học và nhân viên văn phòng, những người thường xuyên có nhu cầu đặt đồ ăn theo nhóm trong giờ nghỉ trưa ngắn ngủi. Mặc dù cả GrabFood và ShopeeFood đều cung cấp tính năng đặt hàng nhóm (Group Order), quy trình thanh toán của họ vẫn tồn tại một "nỗi đau" đáng kể. Trong mô hình hiện tại, một người (chủ nhóm) phải đứng ra thanh toán toàn bộ đơn hàng, sau đó phải mất công thu lại tiền từ từng thành viên trong nhóm. Quy trình này không chỉ gây bất tiện và mất thời gian mà còn tạo ra tâm lý ngại ngùng, đặc biệt trong môi trường công sở. Hơn nữa, tình huống quên thu tiền hoặc thành viên quên trả tiền diễn ra thường xuyên, gây khó chịu và ảnh hưởng đến mối quan hệ trong nhóm.

Vấn đề này có tầm quan trọng đáng kể vì nó ảnh hưởng trực tiếp đến trải nghiệm ăn uống tập thể, một nét văn hóa phổ biến tại Việt Nam. Việc giải quyết hiệu quả

bài toán này không chỉ cải thiện trải nghiệm người dùng mà còn mở ra cơ hội cho một nền tảng mới tiếp cận thị trường ngách với giá trị khác biệt rõ ràng. Với quy mô đông đảo của nhóm đối tượng sinh viên và nhân viên văn phòng tại các thành phố lớn, tiềm năng thị trường cho giải pháp này là đáng kể. Nếu được triển khai thành công, mô hình này còn có thể được áp dụng cho các tình huống đặt hàng nhóm khác như tiệc tại văn phòng, sự kiện nhóm hoặc các buổi họp mặt gia đình.

## 1.2 Mục tiêu và phạm vi đề tài

Để xác định hướng đi phù hợp cho dự án YummyZoom, việc phân tích và đánh giá các nền tảng giao đồ ăn hàng đầu trên thị trường là điều kiện tiên quyết. GrabFood với thế mạnh là hệ sinh thái dịch vụ toàn diện, bao gồm di chuyển, giao hàng và thanh toán điện tử, đã xây dựng được lòng trung thành cao từ nhóm người dùng từ 35 tuổi trở lên thông qua các chương trình như GrabRewards và GrabUnlimited. ShopeeFood, với chiến lược tận dụng nền tảng thương mại điện tử Shopee, tập trung vào nhóm người dùng trẻ từ 16-24 tuổi bằng các chương trình khuyến mãi và voucher liên tục. Cả hai nền tảng đều cung cấp đầy đủ các tính năng cơ bản như tìm kiếm nhà hàng, đặt hàng, thanh toán đa dạng, theo dõi đơn hàng và đánh giá. Đặc biệt, cả hai đều có tính năng đặt hàng nhóm, cho phép nhiều người cùng tham gia vào một đơn hàng.

Tuy nhiên, qua quá trình phân tích, một hạn chế quan trọng đã được xác định. Tính năng đặt hàng nhóm hiện tại trên cả GrabFood và ShopeeFood đều yêu cầu chủ nhóm phải thanh toán toàn bộ đơn hàng, sau đó tự thu tiền từ các thành viên khác. Quy trình này tạo ra nhiều bất tiện: người chủ nhóm phải tạm ứng một khoản tiền lớn, mất thời gian để thu lại từng phần, và có nguy cơ không thu đủ tiền do các thành viên quên hoặc trì hoãn. Đối với nhóm sinh viên và nhân viên văn phòng, những người thường xuyên đặt đồ ăn chung trong giờ nghỉ trưa ngắn, vấn đề này càng trở nên bất tiện và gây mất thời gian quý báu. Hơn nữa, tình huống này còn có thể tạo ra những khoảnh khắc ngượng ngùng khi phải nhắc nhở đồng nghiệp về việc trả tiền, ảnh hưởng đến mối quan hệ trong nhóm.

Dựa trên phân tích trên, dự án YummyZoom được xác định với mục tiêu chính là xây dựng một ứng dụng giao đồ ăn tập trung vào việc giải quyết vấn đề đặt hàng nhóm thông qua cơ chế thanh toán phân tán. Để đạt được mục tiêu này, hệ thống sẽ được phát triển với các chức năng cốt lõi sau đây. Thứ nhất, các chức năng cơ bản đáp ứng tiêu chuẩn ngành bao gồm hệ thống quản lý tài khoản cho ba vai trò (khách hàng, nhà hàng, quản trị viên), tìm kiếm và khám phá nhà hàng với bộ lọc đa dạng, đặt hàng cá nhân với khả năng tùy chỉnh món ăn, quản lý giỏ hàng và áp dụng mã khuyến mãi, thanh toán mô phỏng thông qua chế độ thử nghiệm của Stripe, theo dõi trạng thái đơn hàng, và hệ thống đánh giá nhà hàng. Thứ hai, tính năng đột phá

tạo nên lợi thế cạnh tranh chính là TeamCart (Giỏ hàng nhóm), cho phép mỗi thành viên trong nhóm tự thêm món ăn mình muốn và tự thanh toán cho phần của riêng mình, kèm theo khả năng cập nhật thời gian thực để các thành viên có thể theo dõi tiến trình của nhóm. Thứ ba, các chức năng quản lý dành cho nhà hàng bao gồm đăng ký và quản lý hồ sơ, quản lý thực đơn và tính năng đánh dấu món hết hàng, tạo và quản lý các chương trình khuyến mãi, xử lý đơn hàng với thông báo thời gian thực, và xem đánh giá từ khách hàng. Cuối cùng, các chức năng quản trị bao gồm bảng điều khiển tổng quan về các chỉ số hoạt động của hệ thống, quản lý và phê duyệt đăng ký nhà hàng mới, và giám sát nội dung như đánh giá và chương trình khuyến mãi.

Để đảm bảo tính khả thi trong khuôn khổ đồ án tốt nghiệp, phạm vi dự án được xác định rõ ràng với các giới hạn hợp lý. Dự án sẽ tập trung phát triển ba vai trò chính là khách hàng, nhà hàng và quản trị viên, với đầy đủ luồng đặt hàng cá nhân và nhóm. Quá trình giao hàng sẽ được mô phỏng thông qua việc cập nhật các trạng thái đơn hàng thay vì triển khai hệ thống theo dõi tài xế thực tế. Các chức năng nằm ngoài phạm vi dự án bao gồm module ứng dụng dành cho tài xế với các nghiệp vụ phức tạp như đăng ký, định tuyến và quản lý thu nhập, hệ thống theo dõi vị trí GPS thời gian thực, các thuật toán gợi ý sử dụng trí tuệ nhân tạo, chương trình khách hàng thân thiết với tích điểm và đổi thưởng, và tích hợp các cổng thanh toán thực tế như VNPAY hay MoMo. Các quyết định giới hạn này được đưa ra nhằm tập trung nguồn lực vào việc hoàn thiện tính năng cốt lõi là TeamCart và đảm bảo chất lượng của các chức năng nền tảng, thay vì dàn trải cho quá nhiều tính năng phức tạp vượt quá khả năng thực hiện trong thời gian và nguồn lực giới hạn của một đồ án tốt nghiệp.

### 1.3 Định hướng giải pháp

Để giải quyết các vấn đề đã xác định ở phần 1.2, dự án YummyZoom được phát triển theo định hướng ứng dụng kiến trúc phần mềm hiện đại và các công nghệ phù hợp. Về mặt kiến trúc, dự án áp dụng Clean Architecture (Kiến trúc sạch) kết hợp với Domain-Driven Design (DDD - Thiết kế hướng miền) làm nền tảng. Clean Architecture được lựa chọn vì khả năng tách biệt logic nghiệp vụ khỏi các chi tiết kỹ thuật, giúp hệ thống dễ bảo trì, kiểm thử và mở rộng trong tương lai. Domain-Driven Design được áp dụng để mô hình hóa các quy tắc nghiệp vụ phức tạp của lĩnh vực giao đồ ăn thông qua các khái niệm như tập hợp (Aggregate), thực thể (Entity), đối tượng giá trị (Value Object) và sự kiện (Domain Event), đảm bảo tính nhất quán dữ liệu và phản ánh chính xác các quy trình thực tế.

Về công nghệ triển khai, dự án sử dụng .NET 9 và C# cho phát triển phần backend. .NET 9 được lựa chọn vì khả năng xử lý hiệu năng cao, hỗ trợ lập trình

bất đồng bộ (async/await) cần thiết cho các tính năng thời gian thực, và một hệ sinh thái phong phú với nhiều thư viện hỗ trợ. Entity Framework Core được sử dụng làm công cụ ánh xạ quan hệ đối tượng (Object-Relational Mapping - ORM), cho phép áp dụng phương pháp ưu tiên mã nguồn (code-first) phù hợp với DDD và tự động hóa việc quản lý lược đồ cơ sở dữ liệu thông qua migrations. Đặc biệt, để triển khai tính năng TeamCart với khả năng cập nhật thời gian thực, dự án tích hợp SignalR, một thư viện hỗ trợ giao tiếp hai chiều qua WebSocket, cho phép các thành viên trong nhóm nhìn thấy các thay đổi ngay lập tức khi có người thêm hoặc xóa món ăn.

Giải pháp tổng thể của YummyZoom được thiết kế theo kiến trúc phân lớp với các thành phần chính sau. Lớp miền (Domain layer) chứa các tập hợp (aggregate) cốt lõi như User, Restaurant, Order, Menu và TeamCart, đóng gói toàn bộ quy tắc nghiệp vụ và đảm bảo tính nhất quán của dữ liệu. Lớp ứng dụng (Application layer) triển khai các ca sử dụng thông qua Commands và Queries theo mô hình CQRS (Command Query), xử lý các yêu cầu từ người dùng và điều phối các đối tượng miền. Lớp hạ tầng (Infrastructure layer) cung cấp các triển khai cụ thể cho tầng lưu trữ bền vững sử dụng Entity Framework Core với PostgreSQL, quản lý định danh thông qua ASP.NET Identity, và giao tiếp thời gian thực sử dụng SignalR. Lớp giao diện (Web layer) triển khai các điểm cuối API theo chuẩn RESTful sử dụng ASP.NET Core, xử lý xác thực và phân quyền thông qua JWT tokens. Luồng hoạt động chính của hệ thống bắt đầu từ khách hàng duyệt nhà hàng và thực đơn, sau đó có thể chọn đặt hàng cá nhân hoặc tạo một TeamCart. Đối với TeamCart, chủ nhóm tạo giỏ hàng và chia sẻ liên kết, các thành viên tham gia và tự thêm món ăn, mỗi người tự thanh toán cho phần của mình, và khi tất cả đã hoàn tất, giỏ hàng nhóm được chuyển thành một đơn hàng duy nhất gửi đến nhà hàng. Nhà hàng nhận đơn hàng, xử lý và cập nhật trạng thái theo thời gian thực, trong khi quản trị viên giám sát toàn bộ hoạt động của hệ thống.

Đóng góp chính của đề án thể hiện ở ba khía cạnh quan trọng. Thứ nhất, dự án thành công trong việc áp dụng Clean Architecture kết hợp Domain-Driven Design vào một miền nghiệp vụ phức tạp như giao đồ ăn, với nhiều tác nhân tương tác và quy trình nghiệp vụ đa dạng. Thứ hai, giải pháp TeamCart với cơ chế thanh toán phân tán là một đột phá so với các ứng dụng hiện tại, giải quyết trực tiếp "nỗi đau" thực tế của người dùng khi đặt hàng nhóm. Cơ chế này không chỉ loại bỏ gánh nặng tài chính cho chủ nhóm mà còn tạo ra sự công bằng và minh bạch trong thanh toán. Thứ ba, việc triển khai tính năng cộng tác thời gian thực cho phép nhiều người cùng tương tác với một giỏ hàng chung một cách mượt mà, với các thay đổi được đồng bộ ngay lập tức giữa tất cả các thành viên. Về kết quả đạt được, dự án đã xây

dựng thành công một ứng dụng sản phẩm khả dụng tối thiểu (MVP) hoàn chỉnh với đầy đủ các tính năng cốt lõi, tính năng TeamCart hoạt động ổn định với khả năng cập nhật thời gian thực, kiến trúc hệ thống rõ ràng và dễ bảo trì, và hiệu năng đáp ứng yêu cầu với thời gian phản hồi dưới 2 giây cho hầu hết các thao tác.

### 1.4 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 khảo sát và phân tích yêu cầu hệ thống. Trong chương này, tôi thực hiện phân tích chi tiết các ứng dụng giao đồ ăn hàng đầu tại thị trường Việt Nam, đặc biệt là GrabFood và ShopeeFood, để xác định các tiêu chuẩn ngành và tìm ra cơ hội khác biệt hóa. Tiếp đến, tôi xây dựng biểu đồ ca sử dụng (use case) tổng quát và các biểu đồ ca sử dụng phân rã cho các chức năng chính, sau đó tiến hành đặc tả chi tiết cho các ca sử dụng quan trọng nhất của hệ thống. Đặc biệt, chương này tập trung vào việc mô tả chi tiết quy trình nghiệp vụ của tính năng TeamCart, làm rõ sự khác biệt so với tính năng đặt hàng nhóm truyền thống. Cuối chương, tôi trình bày các yêu cầu phi chức năng về hiệu năng (performance), khả năng sử dụng (usability), bảo mật (security) và khả năng mở rộng (scalability) mà hệ thống cần đáp ứng.

Chương 3, giới thiệu về các công nghệ và phương pháp được sử dụng trong dự án. Chương này bắt đầu với việc trình bày chi tiết về Clean Architecture và các nguyên lý thiết kế của nó, sau đó giải thích cách áp dụng Domain-Driven Design để mô hình hóa miền giao đồ ăn với các khái niệm như tập hợp (Aggregate), thực thể (Entity) và đối tượng giá trị (Value Object). Tiếp theo, tôi phân tích các công nghệ cụ thể bao gồm .NET 9 với ASP.NET Core cho phát triển phần backend, Entity Framework Core cho truy cập dữ liệu với phương pháp ưu tiên mã nguồn (code-first), SignalR cho giao tiếp thời gian thực, và ASP.NET Identity kết hợp JWT cho xác thực và phân quyền. Mỗi công nghệ được phân tích về lý do lựa chọn, các lựa chọn thay thế có thể xem xét, và cách thức áp dụng cụ thể vào YummyZoom.

Chương 4 trình bày chi tiết về thiết kế, triển khai và đánh giá hệ thống. Đầu tiên, tôi mô tả kiến trúc tổng thể của hệ thống theo Clean Architecture với bốn lớp chính (Domain, Application, Infrastructure và Web) và các mối quan hệ phụ thuộc giữa chúng. Tiếp theo, tôi trình bày thiết kế chi tiết cho các tập hợp miền (domain aggregates) quan trọng nhất như User, Restaurant, Order, Menu và đặc biệt là TeamCart với các thực thể (entity) và đối tượng giá trị (value object) bên trong. Phần thiết kế cơ sở dữ liệu trình bày lược đồ cơ sở dữ liệu với các bảng, quan hệ và ràng buộc được sinh ra từ mô hình miền thông qua Entity Framework Core migrations. Tôi cũng trình bày thiết kế các điểm cuối API theo chuẩn RESTful,



cách tổ chức bộ điều khiển (controllers) và các tầng trung gian (middleware) được sử dụng. Phần cuối chương trình bày về quá trình xây dựng ứng dụng với các công cụ và thư viện được sử dụng, cũng như kết quả kiểm thử thông qua kiểm thử đơn vị (unit tests) cho lớp miền, kiểm thử tích hợp (integration tests) cho lớp hạ tầng và kiểm thử chức năng (functional tests) cho lớp ứng dụng.

Chương 5 tập trung vào các giải pháp kỹ thuật và đóng góp nổi bật của đồ án. Tôi phân tích chi tiết cách áp dụng Clean Architecture vào miền phức tạp của giao đồ ăn, làm rõ cách tách biệt logic nghiệp vụ khỏi các vấn đề hạ tầng và lợi ích của việc này trong kiểm thử và bảo trì. Tiếp theo, tôi trình bày giải pháp xử lý thông báo thời gian thực sử dụng SignalR, bao gồm cách thiết kế các trung tâm (hubs), quản lý kết nối và phát thông điệp đến các máy khách. Phần quan trọng nhất là phân tích chi tiết thiết kế và triển khai của tính năng TeamCart, từ mô hình miền với tập hợp TeamCart, các quy tắc nghiệp vụ đảm bảo tính nhất quán, cơ chế khóa khi chuyển đổi trạng thái, đến việc xử lý thanh toán phân tán và đồng bộ hóa thời gian thực giữa các thành viên. Mỗi giải pháp được so sánh với cách tiếp cận của các ứng dụng hiện có để làm rõ tính đột phá và hiệu quả của phương pháp được áp dụng.

Chương 6 tổng kết lại toàn bộ công việc đã thực hiện và các kết quả đạt được. Tôi đánh giá những thành công của dự án trong việc xây dựng một ứng dụng sản phẩm khả dụng tối thiểu hoàn chỉnh với tính năng TeamCart hoạt động ổn định và hiệu quả. Đồng thời, tôi cũng thẳng thắn chỉ ra những hạn chế còn tồn tại như việc chưa triển khai ứng dụng di động gốc (native mobile app), chưa có tích hợp học máy (machine learning) và AI cho gợi ý cá nhân hóa, và việc mô phỏng quá trình giao hàng thay vì tích hợp với hệ thống tài xế thực tế. Cuối cùng, tôi đề xuất các hướng phát triển trong tương lai bao gồm phát triển ứng dụng di động cho iOS và Android, tích hợp các cổng thanh toán thực tế phổ biến tại Việt Nam như MoMo và VNPAY, xây dựng chương trình khách hàng thân thiết với tích điểm và đổi thưởng, áp dụng học máy để gợi ý món ăn và nhà hàng phù hợp, và triển khai kiến trúc microservices để tăng khả năng mở rộng của hệ thống khi số lượng người dùng tăng lên. [Sửa sau]

## PHẦN HƯỚNG DẪN TỪ MẪU BÁO CÁO

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 này, chỉ chỉnh sửa nội dung, và không viết trên file mới.

**Khi đóng quyển ĐATN**, sinh viên cần lưu ý tuân thủ hướng dẫn ở phụ lục A.9

**SV cần đặc biệt lưu ý cách hành văn.** Mỗi đoạn văn không được quá dài và cần có ý tứ rõ ràng, bao gồm duy nhất một ý chính và các ý phân tích hỗ trợ để làm rõ hơn ý chính. Các câu văn trong đoạn phải đầy đủ chủ ngữ vị ngữ, cùng hướng đến chủ đề chung. Câu sau phải liên kết với câu trước, đoạn sau liên kết với đoạn trước. Trong văn phong khoa học, sinh viên không được dùng từ trong văn nói, không dùng các từ phóng đại, thái quá, các từ thiếu khách quan, thiên về cảm xúc, về quan điểm cá nhân như “tuyệt vời”, “cực hay”, “cực kỳ hữu ích”, v.v. Các câu văn cần được tối ưu hóa, đảm bảo rất khó để thể thêm hoặc bớt đi được dù chỉ một từ. Cách diễn đạt cần ngắn gọn, súc tích, không dài dòng.

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

### Hướng dẫn viết phần 1.1 - 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. Sinh viên cần lưu ý phần này chỉ trình bày vấn đề, tuyệt đối không trình bày giải pháp.

### **Hướng dẫn viết phần 1.2 - 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 **có các chức năng chính 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.

### **Hướng dẫn viết phần 1.3 - 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 đề 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.

### **Hướng dẫn viết phần 1.4 - Bố cục đề án**

Phần còn lại của báo cáo đề 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 đề án tốt nghiệp: Chương \*\*\* trình bày đóng góp chính của đề 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.

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

Trong chương này, nghiên cứu tiến hành khảo sát và phân tích toàn diện về hiện trạng của ngành giao đồ ăn trực tuyến, từ đó xác định được các yêu cầu cần thiết cho việc phát triển hệ thống YummyZoom. Chương bao gồm bốn phần chính: khảo sát hiện trạng thị trường và các ứng dụng tương tự hiện có, tổng quan về các chức năng cần thiết của hệ thống thông qua biểu đồ ca sử dụng (use case) và quy trình nghiệp vụ, đặc tả chi tiết các ca sử dụng quan trọng nhất, và cuối cùng là xác định các yêu cầu phi chức năng của hệ thống. Thông qua việc phân tích các ứng dụng như GrabFood, ShopeeFood cùng với khảo sát nhu cầu thực tế của người dùng và nhà hàng, chương này sẽ làm rõ những khoảng trống còn tồn tại trong thị trường hiện tại và định hướng các tính năng độc đáo mà YummyZoom cần phát triển để tạo ra lợi thế cạnh tranh.

### 2.1 Khảo sát hiện trạng

#### 2.1.1 Bối cảnh và xu hướng thị trường

Trong bối cảnh chuyển đổi số mạnh mẽ và sự thay đổi thói quen tiêu dùng của người dân Việt Nam, ngành giao đồ ăn trực tuyến đã trải qua một giai đoạn tăng trưởng ấn tượng trong những năm gần đây. Theo báo cáo của Hiệp hội Thương mại điện tử Việt Nam (VECOM), thị trường giao đồ ăn trực tuyến tại Việt Nam đạt giá trị khoảng 1,2 tỷ USD vào năm 2023 và dự kiến sẽ tăng trưởng với tốc độ trung bình 15-20% mỗi năm trong giai đoạn 2024-2028. Sự bùng nổ này được thúc đẩy bởi nhiều yếu tố quan trọng, bao gồm sự phát triển của công nghệ di động, tăng cường khả năng tiếp cận internet băng rộng, và đặc biệt là sự thay đổi mạnh mẽ trong hành vi tiêu dùng sau đại dịch COVID-19.

Đại dịch COVID-19 đã tạo ra một bước ngoặt quan trọng trong ngành dịch vụ ăn uống, khi các biện pháp giãn cách xã hội và phong tỏa đã buộc người tiêu dùng phải chuyển sang sử dụng các dịch vụ giao đồ ăn trực tuyến như một giải pháp thay thế cho việc dùng bữa tại nhà hàng. Điều đáng chú ý là sau khi các biện pháp hạn chế được nới lỏng, thói quen sử dụng dịch vụ giao đồ ăn trực tuyến vẫn được duy trì và tiếp tục phát triển, cho thấy sự chuyển đổi căn bản trong hành vi tiêu dùng. Theo khảo sát của công ty nghiên cứu thị trường Nielsen Vietnam, khoảng 78% người tiêu dùng tại các thành phố lớn như Hà Nội và TP.HCM đã sử dụng dịch vụ giao đồ ăn trực tuyến ít nhất một lần trong năm 2023, trong đó có 45% sử dụng thường xuyên với tần suất từ 2-3 lần mỗi tuần.

Xu hướng phát triển của thị trường giao đồ ăn trực tuyến không chỉ thể hiện qua con số tăng trưởng về doanh thu mà còn qua sự đa dạng hóa trong nhu cầu của

người tiêu dùng. Khách hàng ngày càng đòi hỏi cao hơn về chất lượng dịch vụ, tính tiện lợi, tốc độ giao hàng, và đặc biệt là trải nghiệm người dùng trên ứng dụng di động. Đồng thời, nhu cầu về tính cá nhân hóa và các tính năng xã hội hóa như đặt hàng nhóm, chia sẻ trải nghiệm ẩm thực cũng đang trở thành các yếu tố quan trọng trong việc lựa chọn nền tảng giao đồ ăn. Điều này tạo ra cơ hội lớn cho các ứng dụng mới có khả năng đáp ứng những nhu cầu chưa được thỏa mãn một cách đầy đủ bởi các nền tảng hiện có.

Từ góc độ doanh nghiệp, ngành nhà hàng và dịch vụ ăn uống cũng đã chứng kiến sự thay đổi mạnh mẽ trong mô hình kinh doanh. Nhiều nhà hàng truyền thống đã phải chuyển đổi sang mô hình hybrid, kết hợp giữa phục vụ tại chỗ và giao hàng trực tuyến để duy trì hoạt động kinh doanh. Đặc biệt, các nhà hàng quy mô vừa và nhỏ đang tìm kiếm những nền tảng giao đồ ăn có chính sách hoa hồng hợp lý và cung cấp các công cụ quản lý hiệu quả để tối ưu hóa hoạt động kinh doanh của mình. Điều này không chỉ tạo ra nhu cầu về các giải pháp công nghệ mới mà còn mở ra cơ hội cho các ứng dụng có thể hỗ trợ tốt hơn cho cộng đồng nhà hàng địa phương.

### **2.1.2 Phân tích các ứng dụng giao đồ ăn hiện có**

Để hiểu rõ bối cảnh cạnh tranh và xác định được các cơ hội phát triển cho YummyZoom, nghiên cứu tiến hành phân tích chi tiết hai ứng dụng giao đồ ăn đang thống lĩnh thị trường Việt Nam hiện nay: GrabFood và ShopeeFood. Theo báo cáo của Momentum Works năm 2024, hai nền tảng này chiếm giữ gần như toàn bộ thị phần với GrabFood dẫn đầu ở mức 48% và ShopeeFood bám sát với 47%, tạo nên một thế song cực trong ngành giao đồ ăn trực tuyến tại Việt Nam.

#### **a, GrabFood - Sức mạnh từ hệ sinh thái siêu ứng dụng**

GrabFood không đơn thuần là một ứng dụng giao đồ ăn độc lập mà là một thành phần quan trọng trong hệ sinh thái siêu ứng dụng Grab. Lợi thế cạnh tranh cốt lõi của GrabFood xuất phát từ khả năng tận dụng tối đa các dịch vụ khác trong hệ sinh thái này, bao gồm dịch vụ di chuyển (GrabBike, GrabCar), giao hàng (GrabExpress), và hệ thống thanh toán tích hợp (ví điện tử Moca). Sự liên kết chặt chẽ này cho phép Grab có được một lượng khách hàng khổng lồ có sẵn, đội ngũ tài xế đông đảo, và đặc biệt là một kho dữ liệu người dùng phong phú để tối ưu hóa trải nghiệm dịch vụ.

Từ góc độ tính năng, GrabFood excel trong việc cung cấp trải nghiệm theo dõi đơn hàng theo thời gian thực với khả năng hiển thị vị trí tài xế trên bản đồ một cách chính xác. Hệ thống gợi ý món ăn của GrabFood được đánh giá cao nhờ việc áp dụng các thuật toán học máy để phân tích hành vi người dùng và đưa ra các đề xuất

cá nhân hóa. Đồng thời, GrabFood cũng có một hệ thống đánh giá và phản hồi khá hoàn chỉnh, cho phép người dùng đánh giá cả nhà hàng và tài xế, từ đó tạo ra một cơ chế kiểm soát chất lượng dịch vụ hiệu quả.

Tuy nhiên, GrabFood cũng tồn tại những điểm yếu đáng chú ý. Chi phí sử dụng dịch vụ của GrabFood thường cao hơn so với các đối thủ, với phí giao hàng và phí dịch vụ có thể lên đến 15-20% giá trị đơn hàng. Giao diện người dùng của ứng dụng Grab, do phải tích hợp nhiều dịch vụ, đôi khi trở nên phức tạp và khó điều hướng, đặc biệt đối với người dùng lớn tuổi hoặc ít am hiểu công nghệ. Về tính năng đặt hàng nhóm, mặc dù GrabFood có hỗ trợ chức năng này, nhưng quy trình vẫn yêu cầu một người làm chủ nhóm phải đứng ra thanh toán toàn bộ đơn hàng, sau đó tự thu lại tiền từ các thành viên khác.

### **b, ShopeeFood - Thống lĩnh bằng văn hóa sẵn khuyến mãi**

ShopeeFood, với tiền thân là Now.vn - một trong những nền tảng giao đồ ăn tiên phong tại Việt Nam, đã được tích hợp sâu vào hệ sinh thái thương mại điện tử Shopee sau khi SEA Group mua lại. Điểm mạnh nổi bật nhất của ShopeeFood nằm ở chiến lược marketing aggressive với vô số chương trình khuyến mãi, voucher giảm giá và ưu đãi miễn phí giao hàng. Nền tảng này rất thành công trong việc khai thác tâm lý "săn sale" đặc trưng của người tiêu dùng Việt Nam, đặc biệt là thông qua việc tận dụng các sự kiện mua sắm lớn của Shopee như 11/11, 12/12 để thu hút người dùng.

Về mặt tính năng, ShopeeFood có giao diện tương đối đơn giản và trực quan, phù hợp với đối tượng người dùng trẻ tuổi. Hệ thống tìm kiếm và lọc món ăn được thiết kế khá hiệu quả, cho phép người dùng dễ dàng tìm thấy các món ăn theo sở thích và ngân sách. ShopeeFood cũng tích hợp tốt với ví điện tử ShopeePay, tạo ra một trải nghiệm thanh toán mượt mà và thường đi kèm với các ưu đãi hoàn tiền. Đặc biệt, nền tảng này có điểm mạnh trong việc hỗ trợ các cửa hàng nhỏ lẻ, quán ăn vặt với chính sách hoa hồng cạnh tranh.

Tuy nhiên, ShopeeFood cũng đối mặt với một số thách thức đáng kể. Chất lượng dịch vụ giao hàng không ổn định là một vấn đề thường xuyên được người dùng phàn nàn, với tình trạng giao hàng chậm trễ và thái độ phục vụ của tài xế chưa đồng đều. Số lượng nhà hàng đối tác và phạm vi phủ sóng của ShopeeFood vẫn còn hạn chế hơn so với GrabFood, đặc biệt tại các khu vực ngoại thành. Về tính năng đặt hàng nhóm, ShopeeFood cũng gặp phải vấn đề tương tự như GrabFood khi yêu cầu một người phải đứng ra thanh toán cho toàn bộ nhóm.

### **c, Phân tích so sánh và nhận định tổng quan**

Qua việc phân tích chi tiết hai nền tảng dẫn đầu thị trường, có thể thấy rằng cả GrabFood và ShopeeFood đều đã xây dựng được những lợi thế cạnh tranh riêng biệt nhưng cũng tồn tại những khoảng trống có thể khai thác. GrabFood thành công nhờ hệ sinh thái tích hợp và chất lượng dịch vụ ổn định, trong khi ShopeeFood chiến thắng bằng chiến lược giá cả cạnh tranh và chương trình khuyến mãi hấp dẫn. Tuy nhiên, cả hai nền tảng đều chưa giải quyết được hoàn toàn vấn đề trong quy trình đặt hàng nhóm, nơi mà một người phải gánh vác trách nhiệm tài chính cho cả nhóm.

Đây chính là cơ hội mà YummyZoom có thể tận dụng thông qua tính năng TeamCart độc đáo, cho phép mỗi thành viên trong nhóm tự thanh toán phần của mình. Bên cạnh đó, cả hai nền tảng hiện tại đều có giao diện khá phức tạp với nhiều tính năng không cần thiết, tạo ra cơ hội cho một ứng dụng tập trung vào sự đơn giản và trải nghiệm người dùng tối ưu. Đối với nhóm khách hàng mục tiêu là sinh viên và nhân viên văn phòng - những người có nhu cầu đặt hàng nhóm cao và thời gian nghỉ trưa hạn chế, việc có một giải pháp đơn giản, nhanh chóng và giải quyết được vấn đề thanh toán nhóm sẽ mang lại giá trị thực tiễn đáng kể.

#### **2.1.3 Nhu cầu và thách thức của người dùng**

Dựa trên việc phân tích thị trường và các ứng dụng hiện có, có thể nhận diện được những nhu cầu cốt lõi và thách thức chưa được giải quyết triệt để trong lĩnh vực giao đồ ăn trực tuyến. Việc hiểu rõ các nhu cầu này không chỉ giúp định hướng phát triển sản phẩm mà còn tạo cơ sở để YummyZoom xây dựng được lợi thế cạnh tranh bền vững.

##### **a, Nhu cầu cốt lõi của người dùng cuối**

Người dùng cuối, đặc biệt là nhóm sinh viên và nhân viên văn phòng, có những nhu cầu rất cụ thể khi sử dụng dịch vụ giao đồ ăn trực tuyến. Nhu cầu quan trọng nhất là tính tiện lợi và tốc độ trong việc đặt hàng, đặc biệt trong những khoảng thời gian nghỉ trưa ngắn ngủi. Họ mong muốn có thể tìm kiếm, lựa chọn và đặt hàng một cách nhanh chóng mà không phải trải qua quá nhiều bước phức tạp. Đồng thời, sự đa dạng trong lựa chọn món ăn và nhà hàng cũng là yếu tố quan trọng, giúp người dùng có thể thỏa mãn các sở thích ẩm thực khác nhau và tránh cảm giác nhàm chán.

Tính minh bạch về giá cả, phí dịch vụ và thời gian giao hàng là một nhu cầu không thể thiếu khác. Người dùng muốn biết chính xác họ sẽ phải trả bao nhiêu tiền và món ăn sẽ đến khi nào, từ đó có thể lập kế hoạch thời gian và ngân sách một cách hợp lý. Đặc biệt quan trọng là nhu cầu về trải nghiệm đặt hàng nhóm không rườm rà, khi một nhóm bạn bè hoặc đồng nghiệp muốn cùng đặt món từ một nhà



hàng. Hiện tại, việc phải có một người đứng ra làm chủ nhóm và thanh toán toàn bộ, sau đó thu lại tiền từ các thành viên khác, tạo ra không ít phiền phức và ngại ngùng trong các mối quan hệ xã hội.

Cuối cùng, nhu cầu về thanh toán linh hoạt và an toàn cũng rất được quan tâm. Người dùng muốn có nhiều lựa chọn về phương thức thanh toán và đảm bảo rằng thông tin tài chính của họ được bảo vệ một cách tối ưu. Đặc biệt với thế hệ trẻ, việc thanh toán điện tử nhanh chóng và thuận tiện thường được ưu tiên hơn so với thanh toán tiền mặt.

### **b, Nhu cầu của nhà hàng đối tác**

Từ góc độ nhà hàng, đặc biệt là các cơ sở kinh doanh quy mô vừa và nhỏ, họ cần có công cụ quản lý đơn hàng hiệu quả và dễ sử dụng. Nhà hàng muốn có thể nhận thông báo đơn hàng mới một cách kịp thời, xem chi tiết đơn hàng rõ ràng, và cập nhật trạng thái chuẩn bị món ăn một cách thuận tiện. Việc quản lý thực đơn cũng cần phải linh hoạt, cho phép nhà hàng dễ dàng thêm món mới, cập nhật giá cả, và đặt trạng thái "hết hàng" khi cần thiết.

Hệ thống khuyến mãi linh hoạt là một nhu cầu quan trọng khác của nhà hàng. Họ muốn có thể tự tạo và quản lý các chương trình giảm giá, coupon theo ý muốn để thu hút khách hàng mà không phụ thuộc hoàn toàn vào các chương trình khuyến mãi của nền tảng. Điều này giúp nhà hàng chủ động hơn trong việc điều chỉnh chiến lược kinh doanh và cạnh tranh trên thị trường.

Chi phí hoa hồng hợp lý cũng là mối quan tâm lớn của các nhà hàng, đặc biệt là những cơ sở nhỏ lẻ với biên lợi nhuận không cao. Họ mong muốn có thể tiếp cận được khách hàng mới thông qua nền tảng trực tuyến mà không phải gánh chịu chi phí quá lớn. Cuối cùng, nhà hàng cũng cần được hỗ trợ kỹ thuật kịp thời khi gặp vấn đề trong quá trình sử dụng hệ thống.

### **c, Những thách thức chưa được giải quyết tốt**

Mặc dù thị trường đã có những ứng dụng thành công, vẫn tồn tại một số thách thức chưa được giải quyết một cách tối ưu. Thách thức lớn nhất và rõ ràng nhất là quy trình đặt hàng nhóm phức tạp với vấn đề thanh toán tập trung. Việc một người phải đứng ra thanh toán cho cả nhóm không chỉ tạo ra gánh nặng tài chính tạm thời mà còn gây ra những tình huống khó xử trong các mối quan hệ xã hội, đặc biệt khi có thành viên quên hoặc chậm trả tiền.

Giao diện ứng dụng quá phức tạp với nhiều tính năng không cần thiết là một thách thức khác. Các ứng dụng hiện tại thường cố gắng tích hợp quá nhiều chức năng, dẫn đến việc người dùng, đặc biệt là những người ít am hiểu công nghệ, cảm

thấy khó khăn trong việc điều hướng và sử dụng. Điều này đặc biệt trở nên rõ ràng khi người dùng có thời gian hạn chế, như trong giờ nghỉ trưa, và muốn đặt hàng một cách nhanh chóng.

Thiếu giải pháp tối ưu cho nhóm khách hàng trẻ, đặc biệt là sinh viên và nhân viên văn phòng, cũng là một khoảng trống đáng chú ý. Nhóm người dùng này có thói quen và nhu cầu rất cụ thể: họ thường đặt hàng theo nhóm, có ngân sách hạn chế, ưa thích sự đơn giản và tốc độ, nhưng lại chưa có ứng dụng nào thực sự được thiết kế riêng để phục vụ những đặc thù này.

### **d, Định hướng phát triển cho YummyZoom**

Dựa trên việc phân tích các nhu cầu và thách thức trên, YummyZoom được định hướng phát triển theo những nguyên tắc rõ ràng. Trước hết, ứng dụng sẽ tập trung vào việc tạo ra một trải nghiệm đơn giản và trực quan, loại bỏ những tính năng phức tạp không cần thiết để người dùng có thể đặt hàng một cách nhanh chóng và hiệu quả nhất. Giao diện được thiết kế với triết lý "ít hơn là nhiều hơn", ưu tiên tính dễ sử dụng và tốc độ thao tác.

Giải quyết triệt để vấn đề đặt hàng nhóm thông qua tính năng TeamCart là định hướng cốt lõi của YummyZoom. Thay vì yêu cầu một người thanh toán cho cả nhóm, mỗi thành viên sẽ tự chịu trách nhiệm thanh toán phần món ăn của mình, từ đó loại bỏ hoàn toàn những phiền phức và ngại ngùng trong các mối quan hệ xã hội. Tính năng này không chỉ là một cải tiến kỹ thuật mà còn mang lại giá trị thực tiễn cao cho người dùng.

Cuối cùng, YummyZoom được định hướng để phục vụ đặc thù của sinh viên và nhân viên văn phòng - những người có nhu cầu đặt hàng nhóm cao, thời gian hạn chế, và ưa thích sự tiện lợi. Từ việc tối ưu hóa quy trình đặt hàng cho đến việc cung cấp các tùy chọn thanh toán phù hợp với thói quen của thế hệ trẻ, tất cả đều được thiết kế với sự hiểu biết sâu sắc về nhu cầu và hành vi của nhóm khách hàng mục tiêu này.

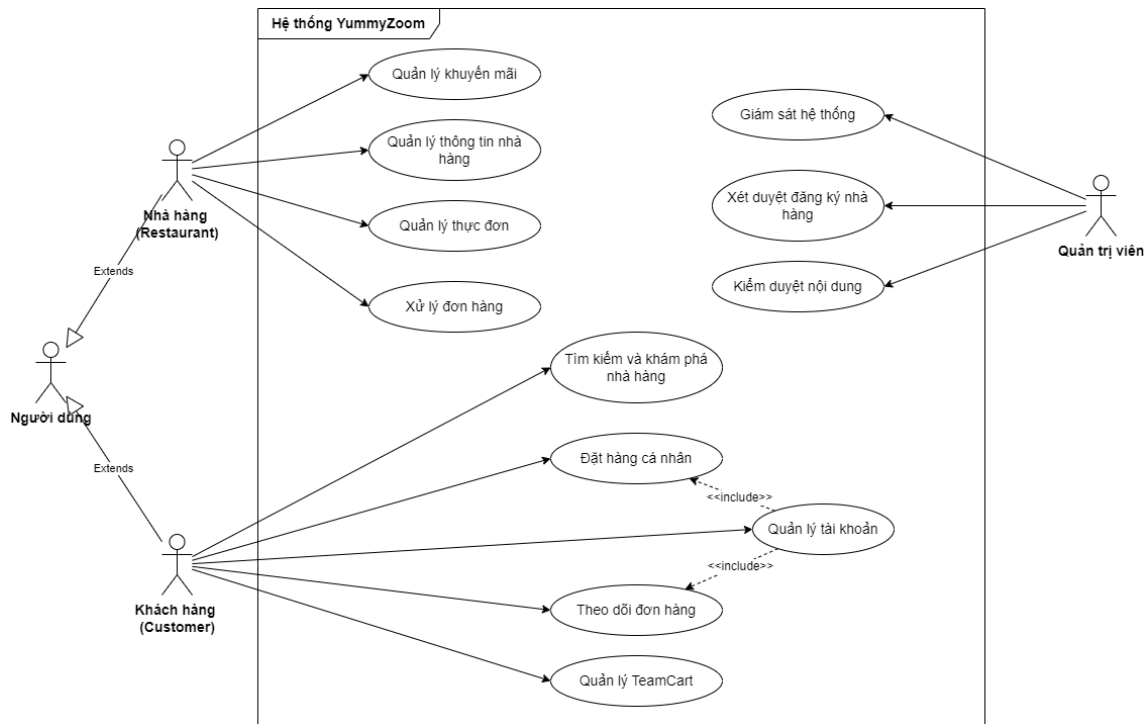
## **2.2 Tổng quan chức năng**

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.

### **2.2.1 Biểu đồ use case tổng quát**

Hình 2.1 minh họa biểu đồ use case tổng quát của hệ thống YummyZoom, thể hiện tất cả các tác nhân (actors) chính và các chức năng cốt lõi mà họ có thể thực hiện trên nền tảng. Biểu đồ được thiết kế theo nguyên tắc phân nhóm rõ ràng các

use case theo từng đối tượng người dùng, giúp dễ dàng nhận diện phạm vi và ranh giới trách nhiệm của từng thành phần trong hệ thống.



Hình 2.1: Biểu đồ use case tổng quát của hệ thống YummyZoom

### a, Các tác nhân chính trong hệ thống

Hệ thống YummyZoom được thiết kế để phục vụ ba nhóm tác nhân chính, mỗi nhóm có vai trò và quyền hạn khác nhau trong quy trình vận hành nền tảng giao đồ ăn trực tuyến.

**Khách hàng (Customer)** Đây là tác nhân chính và quan trọng nhất của hệ thống, bao gồm tất cả người dùng cuối có nhu cầu đặt món ăn thông qua nền tảng. Nhóm tác nhân này bao gồm cả khách hàng cá nhân thực hiện đặt hàng riêng lẻ và những người tham gia vào các đơn hàng nhóm thông qua tính năng TeamCart. Khách hàng có quyền truy cập vào toàn bộ các tính năng liên quan đến việc tìm kiếm, đặt hàng, thanh toán và theo dõi đơn hàng, đồng thời có thể đóng góp phản hồi thông qua hệ thống đánh giá.

**Nhà hàng (Restaurant)** Tác nhân này đại diện cho các đối tác kinh doanh của nền tảng, bao gồm chủ nhà hàng và nhân viên được ủy quyền quản lý hoạt động trên hệ thống. Nhà hàng có trách nhiệm cung cấp thông tin chính xác về cơ sở kinh doanh, quản lý thực đơn một cách hiệu quả, xử lý các đơn hàng đến từ khách hàng, và thực hiện các chiến lược marketing thông qua hệ thống khuyến mãi. Đây là tác nhân then chốt quyết định chất lượng dịch vụ và trải nghiệm khách hàng trên nền

tăng.

**Quản trị viên (Admin)** Tác nhân có quyền hạn cao nhất trong hệ thống, chịu trách nhiệm giám sát và điều hành toàn bộ hoạt động của nền tảng. Quản trị viên có khả năng can thiệp vào mọi khía cạnh của hệ thống, từ việc phê duyệt các nhà hàng mới tham gia nền tảng, kiểm soát chất lượng nội dung, cho đến việc xử lý các tình huống đặc biệt và đảm bảo hệ thống vận hành một cách ổn định. Vai trò này đặc biệt quan trọng trong việc duy trì uy tín và chất lượng dịch vụ của toàn bộ nền tảng.

### **b, Các use case chính và mô tả chức năng**

Dựa trên phân tích nhu cầu và đặc thù của nền tảng giao đồ ăn trực tuyến, hệ thống YummyZoom được thiết kế với 14 use case chính, được phân nhóm theo từng tác nhân để đảm bảo tính logic và dễ quản lý.

**Nhóm use case dành cho Khách hàng** **Quản lý tài khoản:** Use case cơ bản cho phép khách hàng thực hiện các thao tác liên quan đến việc tạo mới, đăng nhập, và cập nhật thông tin cá nhân. Đây là use case có mối quan hệ include với các use case đặt hàng, do việc xác thực danh tính là bắt buộc trước khi thực hiện bất kỳ giao dịch nào.

**Tìm kiếm nhà hàng:** Cho phép khách hàng duyệt và tìm kiếm nhà hàng dựa trên nhiều tiêu chí khác nhau như vị trí địa lý, loại món ăn, mức giá, và đánh giá. Use case này được thiết kế để mang lại trải nghiệm khám phá phong phú và cá nhân hóa.

**Đặt hàng cá nhân:** Quy trình đặt hàng truyền thống cho một khách hàng, bao gồm việc chọn món, tùy chỉnh, và thanh toán. Use case này yêu cầu khách hàng phải đăng nhập (mối quan hệ include với "Quản lý tài khoản").

**Quản lý TeamCart:** Đây là tính năng độc đáo và nổi bật nhất của YummyZoom, cho phép nhiều khách hàng cùng tham gia vào một đơn hàng nhóm mà mỗi người có thể tự thanh toán phần của mình. Tính năng này giải quyết triệt để vấn đề thanh toán tập trung trong các đơn hàng nhóm truyền thống.

**Theo dõi đơn hàng:** Cung cấp khả năng theo dõi trạng thái đơn hàng theo thời gian thực, giúp khách hàng nắm được tình hình chuẩn bị và giao hàng của nhà hàng.

**Đánh giá nhà hàng:** Cho phép khách hàng chia sẻ trải nghiệm thông qua hệ thống đánh giá sao và nhận xét văn bản, đóng góp vào việc xây dựng uy tín cho nhà hàng và hỗ trợ khách hàng khác trong việc lựa chọn.

**Nhóm use case dành cho Nhà hàng Quản lý thông tin nhà hàng:** Cho phép nhà hàng cập nhật các thông tin cơ bản như tên, địa chỉ, giờ hoạt động, thông tin liên lạc, và trạng thái nhận đơn hàng.

**Quản lý thực đơn:** Use case phức tạp bao gồm việc tạo và tổ chức menu, thêm/sửa/xóa món ăn, cập nhật giá cả, quản lý tình trạng còn hàng, và thiết lập các tùy chọn customization cho từng món ăn.

**Xử lý đơn hàng:** Quy trình nhận và xử lý đơn hàng từ khách hàng, bao gồm chấp nhận/từ chối đơn hàng, cập nhật trạng thái chuẩn bị, và đánh dấu sẵn sàng giao hàng.

**Quản lý khuyến mãi:** Cho phép nhà hàng tự tạo và quản lý các chương trình khuyến mãi, mã giảm giá với các điều kiện áp dụng linh hoạt để thu hút và giữ chân khách hàng.

**Nhóm use case dành cho Quản trị viên Quản lý hệ thống:** Cung cấp dashboard tổng quan với các metrics quan trọng về hoạt động của nền tảng, giúp admin nắm bắt tình hình kinh doanh và đưa ra các quyết định vận hành phù hợp.

**Duyệt đăng ký nhà hàng:** Quy trình onboarding cho các nhà hàng mới muốn tham gia nền tảng, bao gồm việc xác minh thông tin, kiểm tra giấy tờ pháp lý, và phê duyệt hoặc từ chối đơn đăng ký.

**Kiểm duyệt nội dung:** Giám sát và kiểm soát chất lượng nội dung do người dùng tạo ra, đặc biệt là các đánh giá và nhận xét, để đảm bảo tính phù hợp và chính xác.

**Giám sát vận hành:** Theo dõi toàn bộ hoạt động giao dịch trên nền tảng, can thiệp khi cần thiết để xử lý các tình huống đặc biệt hoặc khiếu nại từ người dùng.

Tổng thể, 14 use case này tạo nên một hệ sinh thái hoàn chỉnh cho nền tảng giao đồ ăn trực tuyến, trong đó tính năng TeamCart là điểm nhấn độc đáo giúp YummyZoom khác biệt so với các đối thủ cạnh tranh hiện tại trên thị trường.

### 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 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 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

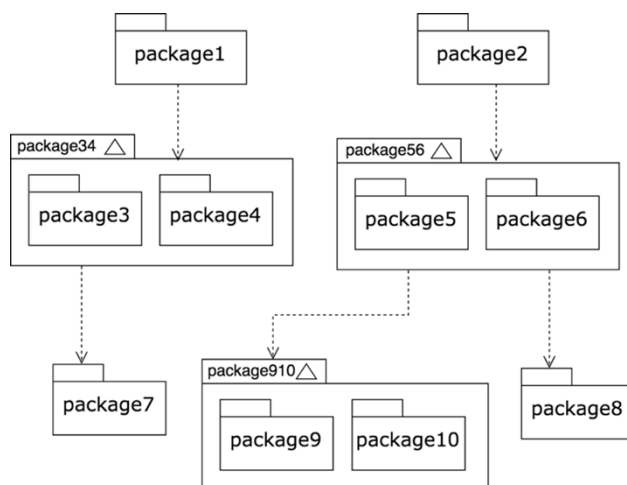
### 4.1 Thiết kế kiến trúc

#### 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

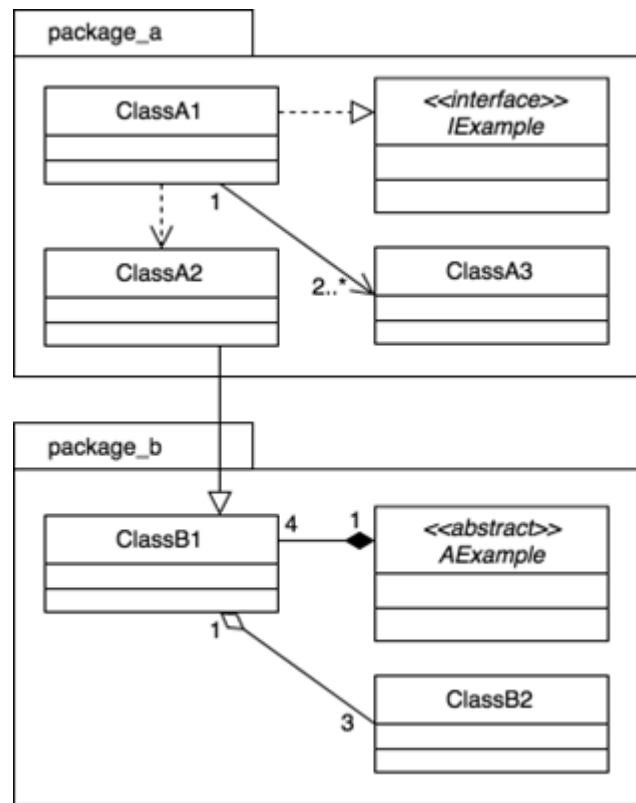
#### 4.1.3 Thiết kế chi tiết gói

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 4.1. 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)*. Accessed: Sep. 30, 2010. [Online]. Available: <ftp://info.cern.ch/pub/www/doc/http-spec.txt.Z>
- [7] Princeton University, *Wordnet*. Accessed: Sep. 30, 2010. [Online]. Available: <http://www.cogsci.princeton.edu/~wn/index.shtml>



## **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 DATN 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 DATN.

Tất cả các hình vẽ, bảng biểu, công thức, và tài liệu tham khảo trong DATN 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 DATN theo kiểu viết ý hoặc gạch đầu dòng. DATN 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 DATN 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)

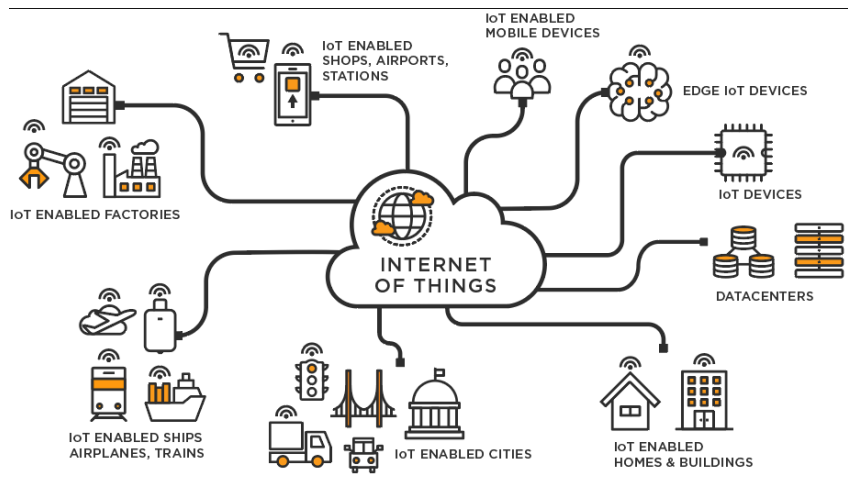
Việc sử dụng danh sách trong LaTeX khá đơn giản và không yêu cầu sinh viên phải thêm bất kỳ gói bổ sung nào. LaTeX cung cấp hai môi trường liệt kê đó là:

- 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 ...
```



**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)

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

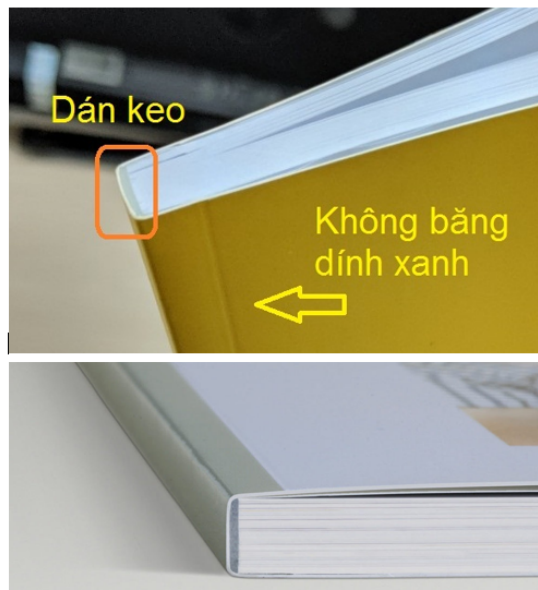
Phương trình này thể hiện phép biến đổi Fourier rời rạc ngược (IDFT).

## A.7 Qui cách đóng quyển

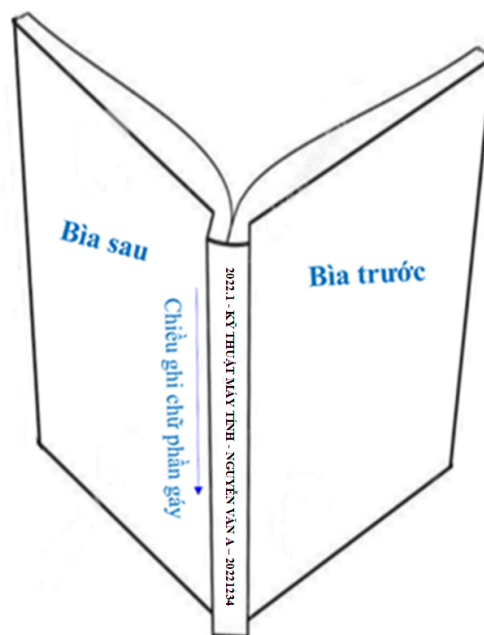
Phần bìa trước chế bản theo qui định; bìa trước và bìa sau là giấy liền khổ. Sử dụng keo nhiệt để dán gáy khi đóng quyển thay vì sử dụng băng dính và dập ghim như mô tả ở Hình A.2 Phần gáy ĐATN cần ghi các thông tin tóm tắt sau: Kỳ làm ĐATN - Ngành đào tạo - Họ và tên sinh viên - Mã số sinh viên. Ví 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 “Thống kê tình hình mượn sách”**

...

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

...