

0.1 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 đồng 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.

0.2 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.

0.3 Định hướng giải pháp

Để giải quyết các vấn đề đã xác định ở phần 0.2, 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 đồ 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.

0.4 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.