

TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

ĐỒ ÁN TỐT NGHIỆP

Hệ thống tự động triển khai mạng và ứng dụng phi tập trung dựa trên nền tảng Hyperledger Fabric

LÊ ĐỨC MẠNH

manh.ld176814@sis.hust.edu.vn

**Ngành Công nghệ thông tin và truyền thông
Chuyên ngành Hệ thống thông tin**

Giảng viên hướng dẫn: TS. Nguyễn Bình Minh

Chữ kí GVHD

Khoa: Công nghệ thông tin

Trường: Công nghệ thông tin và Truyền thông

HÀ NỘI, 07/2022

LỜI CẢM ƠN

Lời mở đầu, em xin cảm ơn trường Đại học Bách khoa Hà Nội đã tạo một môi trường học tập và nghiên cứu lý tưởng. Em tin rằng những kiến thức chuyên sâu, những trải nghiệm quý báu có được tại đây sẽ là hành trang vững chãi trên con đường đời phía sau mái trường Đại học.

Em muốn gửi lời cảm ơn đầu tới các thầy cô viện Công nghệ Thông tin và Truyền Thông đã chỉ dạy trong 5 năm học vừa qua. Đặc biệt em xin cảm ơn TS. Nguyễn Bình Minh và Th.S Đinh Hữu Hải Quân đã tận tình hướng dẫn, tạo điều kiện tốt nhất để em có thể hoàn thành quyển đồ án tốt nghiệp này.

Cảm ơn những người bạn chân tình đã cùng nhau chia sẻ biết bao vui buồn. Tôi sẽ luôn trân trọng những kỉ niệm về quãng đời sinh viên không thể quên này.

Lời cuối cùng, xin cảm ơn gia đình đã luôn ủng hộ, hỗ trợ con. Cảm ơn bố và mẹ đã nuôi con trưởng thành và quan tâm chăm sóc con. Hai người luôn là điểm tựa vững chãi cuộc đời con.

TÓM TẮT NỘI DUNG ĐỒ ÁN

Cùng với cách mạng công nghiệp lần thứ 4, công nghệ chuỗi khối được cho là một trong

MỤC LỤC

CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI.....	1
1.1 Đặt vấn đề.....	1
1.2 Mục tiêu và phạm vi đề tài.....	2
1.2.1 Giải pháp liên quan	2
1.2.2 Mục tiêu và phạm vi.....	2
1.3 Định hướng giải pháp.....	3
1.3.1 Triển khai mạng.....	3
1.3.2 Triển khai ứng dụng phi tập trung.....	3
1.4 Bố cục đồ án	3
CHƯƠNG 2. HYPERLEDGER FABRIC.....	4
2.1 Tổng quan	4
2.2 Kiến trúc mạng Hyperledger Fabric.....	4
2.3 Khởi tạo một mạng Hyperledger Fabric	5
2.3.1 Bước 1: Khởi tạo Orderer Node.....	5
2.3.2 Bước 2: Thêm Tổ chức quản trị	5
2.3.3 Bước 3: Định nghĩa Consortium	6
2.3.4 Bước 4: Tạo kênh tương ứng với Consortium	6
2.3.5 Bước 5: Thêm Peer node.....	7
2.3.6 Bước 6: Cài đặt Hợp đồng thông minh	7
2.3.7 Bước 7: Khởi tạo mạng hoàn tất	8
CHƯƠNG 3. KHẢO SÁT VÀ PHÂN TÍCH YÊU CẦU.....	9
3.1 Tổng quan chức năng	9
3.1.1 Biểu đồ use case tổng quan	9

3.2 Đặc tả chức năng	9
3.2.1 Đặc tả use case Tạo mạng	10
3.2.2 Đặc tả use case Thêm node ngoài	11
3.2.3 Đặc tả use case Thêm tổ chức.....	12
3.2.4 Đặc tả use case Xóa mạng.....	13
3.2.5 Đặc tả use case Tạo Ứng dụng phi tập trung.....	14
3.2.6 Đặc tả use case Tải SDK.....	15
3.2.7 Đặc tả use case Cập nhật Ứng dụng phi tập trung	16
3.2.8 Đặc tả use case Xóa Ứng dụng phi tập trung	17
CHƯƠNG 4. CÔNG NGHỆ SỬ DỤNG.....	18
CHƯƠNG 5. THỰC NGHIỆM VÀ ĐÁNG GIÁ	19
5.1 Thiết kế kiến trúc.....	19
5.1.1 Lựa chọn kiến trúc phần mềm	19
5.1.2 Thiết kế tổng quan.....	19
5.1.3 Thiết kế chi tiết gói	19
5.2 Thiết kế chi tiết.....	20
5.2.1 Thiết kế giao diện	20
5.2.2 Thiết kế lớp	20
5.2.3 Thiết kế cơ sở dữ liệu	21
5.3 Xây dựng ứng dụng.....	21
5.3.1 Thư viện và công cụ sử dụng	21
5.3.2 Kết quả đạt được	21
5.3.3 Minh họa các chức năng chính	21
5.4 Kiểm thử.....	21
5.5 Triển khai	22

CHƯƠNG 6. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	23
6.1 Kết luận.....	23
6.2 Hướng phát triển.....	23
TÀI LIỆU THAM KHẢO.....	26
PHỤ LỤC.....	28
A. HƯỚNG DẪN VIẾT ĐỒ ÁN TỐT NGHIỆP	28
A.1 Ngành học.....	29
A.2 Đánh dấu (bullet) và đánh số (numering)	29
A.3 Cách thêm bảng	29
A.4 Chèn hình ảnh	30
A.5 Tài liệu tham khảo	30
A.6 Cách viết phương trình và công thức toán học.....	31
B. ĐẶC TẢ USE CASE.....	32
B.1 Đặc tả use case “Thống kê tình hình mượn sách”	32
B.2 Đặc tả use case “Đăng ký làm thẻ mượn”	32

DANH MỤC HÌNH VẼ

Hình 2.1	Kiến trúc một mạng Hyperledger Fabric cơ bản	4
Hình 2.2	Bước 1: Khởi tạo Orderer Node	5
Hình 2.3	Bước 2: Thêm Tổ chức quản trị	6
Hình 2.4	Bước 3: Định nghĩa Consortium	6
Hình 2.5	Bước 4: Tạo kênh tương ứng với Consortium	7
Hình 2.6	Bước 5: Thêm Peer node	7
Hình 2.7	Bước 6: Cài đặt Hợp đồng thông minh	8
Hình 2.8	Bước 7: Khởi tạo mạng hoàn tất	8
Hình 3.1	Biểu đồ usecase Tổng quan	9
Hình 5.1	Ví dụ biểu đồ phụ thuộc gói	19
Hình 5.2	Ví dụ thiết kế gói	20
Hình A.1	Internet vạn vật	30

DANH MỤC BẢNG BIỂU

Bảng 3.1	Đặc tả ca sử dụng “Tạo mạng”	10
Bảng 3.2	Đặc tả ca sử dụng “Thêm Node ngoài”	11
Bảng 3.3	Đặc tả ca sử dụng “Thêm tổ chức”	12
Bảng 3.4	Đặc tả ca sử dụng “Xóa mạng”	13
Bảng 3.5	Đặc tả ca sử dụng “Tạo Ứng dụng phi tập trung”	14
Bảng 3.6	Đặc tả ca sử dụng “Tải SDK”	15
Bảng 3.7	Đặc tả ca sử dụng “Cập nhật Ứng dụng phi tập trung”	16
Bảng 3.8	Đặc tả ca sử dụng “Xóa Ứng dụng phi tập trung”	17
Bảng 5.1	Danh sách thư viện và công cụ sử dụng	21
Bảng A.1	Table to test captions and labels.	30

DANH MỤC THUẬT NGỮ VÀ TỪ VIẾT TẮT

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

CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI

1.1 Đặt vấn đề

Công nghệ chuỗi khối (blockchain) được cho là một trong những công nghệ cốt lõi trong cuộc cách mạng công nghiệp lần thứ 4. Tính đột phá của công nghệ chuỗi khối nằm ở việc cho phép những giao dịch có giá trị lớn được thực hiện một cách minh bạch, chính xác, công bằng mà không cần đến sự chứng thực và phân xử của một bên thứ ba. Ví dụ như mạng chuỗi khối đầu tiên, Bitcoin, cho phép hai bên ở mọi nơi chuyển tiền (đồng Bitcoin) mà không cần thông qua một bên trung gian (thường là ngân hàng). Mạng Ethereum tiến thêm một bước nữa khi không chỉ đối với giao dịch tiền tệ, Ethereum còn đảm bảo các nghiệp vụ phức tạp với nhiều điều khoản và điều kiện sẽ được các bên liên quan thực thi một cách chuẩn chỉ thông qua ứng dụng phi tập trung.

Để có thể loại bỏ người trung gian khỏi các giao dịch và tác vụ cốt lõi nằm ở một cuốn sổ cái kỹ thuật số. Cuốn sổ cái này là một chuỗi các khối dữ liệu (nên có tên là chuỗi khối) và có các đặc tính sau:

- Tính phân tán: Cuốn sổ cái được lưu trữ trên nhiều máy tính khác nhau, đảm bảo khả năng chịu lỗi.
- Tính bảo mật: Dữ liệu lưu trữ trên cuốn sổ cái là không thể giả mạo.
- Tính minh bạch: Dữ liệu một khi đã được cập nhật trên sổ cái thì không ai có thể thay đổi hay xóa bỏ.

Ứng dụng đặc tính này trong các nghiệp vụ, những giao dịch liên quan đến nghiệp vụ đó một khi đã được lưu lại trên blockchain thì sẽ tồn tại mãi mãi trên sổ cái. Thông qua nội dung của cuốn sổ cái, việc truy xuất nguồn gốc và xác nhận tính minh bạch của nghiệp vụ có thể được thực hiện một cách nhanh chóng và tự động.

Mạng Bitcoin và Ethereum nêu ở trên là mạng chuỗi khối công khai (public blockchain). Đối với loại mạng này, mọi người đều có thể tham gia, thực hiện tương tác hay truy vấn dữ liệu. Tuy nhiên, đối với một số nghiệp vụ, việc thông tin giao dịch và số liệu hoạt động của các doanh nghiệp bị công khai có thể là một vấn đề lớn. Do vậy việc sử dụng loại mạng công khai vào trong nghiệp vụ giữa các tổ chức, liên doanh đôi khi là bất khả thi. Để giải quyết vấn đề này mạng chuỗi khối riêng tư (private blockchain) ra đời. Để có thể được ứng dụng vào các hoạt động giữa các doanh nghiệp, ngoài tính phân tán, bảo mật và minh bạch dữ liệu như mạng công khai, mạng riêng tư thường thêm có các đặc điểm sau:

- Danh tính của những bên tham gia vào mạng cần phải xác thực được.
- Không phải ai cũng có thể tham gia, truy vấn dữ liệu hay tương tác với mạng. Chỉ các cá nhân, tổ chức có đủ quyền mới có thể thực hiện các hành động này.
- Tốc độ xử lý giao dịch cao hơn so với mạng chuỗi khối công khai.

Với những ưu điểm mà mạng chuỗi khối riêng tư mang lại, việc các doanh nghiệp ứng dụng mạng riêng tư vào nghiệp vụ chắc chắn có thể giúp nâng cao hiệu suất công việc. Tuy vậy, việc triển khai, quản lý hệ thống mạng này yêu cầu cơ sở hạ tầng và kiến thức chuyên môn đặc thù. Thêm vào đó, việc viết và triển khai một ứng dụng phi tập trung phục vụ cho nghiệp vụ cũng là một thử thách lớn, đặc biệt là với những ai chưa từng tiếp xúc với mạng chuỗi khối, một công nghệ mới và có phần phức tạp.

Nhận thấy vấn đề trên, tôi đã quyết định phát triển một hệ thống tự động triển khai mạng cùng với đó là ứng dụng phi tập trung dựa trên nền tảng Hyperledger Fabric - một nền tảng mạng chuỗi khối riêng tư nổi tiếng nhất hiện nay. Thông qua giao diện trực quan của hệ thống, một lập trình viên dù cho không có kiến thức về mạng chuỗi khối cũng có thể dễ dàng triển khai hạ tầng mạng chuỗi khối riêng tư và các ứng dụng phi tập trung phục vụ cho nhiều yêu cầu nghiệp vụ khác nhau.

1.2 Mục tiêu và phạm vi đề tài

1.2.1 Giải pháp liên quan

Ông lớn Amazon cung cấp một giải pháp để hỗ trợ quá trình triển khai mạng Hyperledger Fabric - Amazon Managed Blockchain. Với thế mạnh về điện toán đám mây của mình, thông qua dịch vụ Managed Blockchain, người dùng có thể triển khai một mạng Hyperledger Fabric cho riêng mình trên cơ sở hạ tầng của Amazon chỉ với vài cái nhấp chuột. Quá trình quản lý và theo dõi hoạt động của mạng cũng được đơn giản hóa, tối ưu trải nghiệm người dùng.

Tuy quy trình khởi tạo và quản trị mạng có thể được thực hiện thông qua giao diện trực quan, việc phát triển ứng dụng phi tập trung phục vụ cho các nghiệp vụ trên mạng này lại không được như vậy. Người dùng phải tự mình lập trình vào phải tự chạy các câu lệnh trên terminal để triển khai ứng dụng đó lên mạng. Do vậy quá trình phát triển ứng dụng phi tập trung vẫn sẽ yêu cầu kiến thức chuyên sâu về Hyperledger Fabric. Lập trình viên sẽ cần phải học để viết và triển khai hợp đồng thông minh, một quá trình tiêu tốn nhiều thời gian lẫn công sức.

1.2.2 Mục tiêu và phạm vi

Với mục tiêu đơn giản hóa nhất có thể quá trình ứng dụng mạng chuỗi khối riêng tư vào các nghiệp vụ doanh nghiệp, hệ thống trong đề án này hướng đến việc cho

phép những lập trình viên dù cho không có kiến thức về mạng chuỗi khối cũng có thể dễ dàng triển khai mạng cùng với đó là ứng dụng phi tập trung dựa trên nền tảng Hyperledger Fabric. Để đạt được điều này, 2 mục tiêu sau được đề ra:

- Cho phép cấu hình và triển khai một mạng Hyperledger Fabric thông qua giao diện trực quan.
- Cho phép thiết kế và triển khai ứng dụng phi tập trung thông qua giao diện trực quan. Cung cấp phương thức để có thể tương tác với hợp đồng thông minh đó mà không yêu cầu kiến thức đặc thù.

1.3 Định hướng giải pháp

1.3.1 Triển khai mạng

Cơ sở hạ tầng để triển khai mạng cần được suy tính kỹ lưỡng để đảm bảo mạng hoạt động ổn định và hiệu quả nhất. Để đơn giản hóa quá trình triển khai và quản lý mạng Hyperledger Fabric, người dùng sẽ được phép tùy chỉnh cấu hình thông qua một giao diện. Sau đó, hệ thống sẽ triển khai một mạng với cấu hình tương ứng lên điện toán đám mây. Người dùng sẽ không cần chú ý quá nhiều đến chi tiết phần cứng mà chỉ cần quan tâm đến cấu hình của cơ sở hạ tầng.

1.3.2 Triển khai ứng dụng phi tập trung

Điều quan trọng nhất trong việc triển khai ứng dụng là thiết kế kiến trúc sao cho phù hợp với nghiệp vụ. Để bao quát được nhiều nghiệp vụ nhất có thể, hệ thống sẽ cho phép người dùng triển khai ứng dụng dựa trên mô hình cơ sở dữ liệu quan hệ. Nhiều thực thể sở hữu các thuộc tính khác nhau có thể được định nghĩa. Các thực thể này có thể có nhiều liên kết với nhau (quan hệ một-một, một-nhiều, nhiều-nhiều). Dựa vào kiến trúc tổng quan của các thực thể này, một ứng dụng phi tập trung tương ứng sẽ được tự động sinh ra. Sau khi được triển khai, người dùng có thể tải bộ SDK về để tương tác với mạng chuỗi khối thông qua các hàm đọc ghi sửa xóa các thực thể trên. Quá trình hình thành và thay đổi của các thực thể này sẽ được lưu lại vĩnh viễn, đảm bảo việc xác thực và truy vấn dữ liệu nghiệp vụ thông qua mạng chuỗi khối.

1.4 Bố cục đồ án

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

- Chương 2: Giới thiệu kiến trúc mạng chuỗi khối Hyperledger Fabric.
- Chương 3: Khảo sát phân tích yêu cầu.
- Chương 4: Trình bày về các công nghệ sử dụng.
- Chương 5: Kết luận và định hướng phát triển trong tương lai.

CHƯƠNG 2. HYPERLEDGER FABRIC

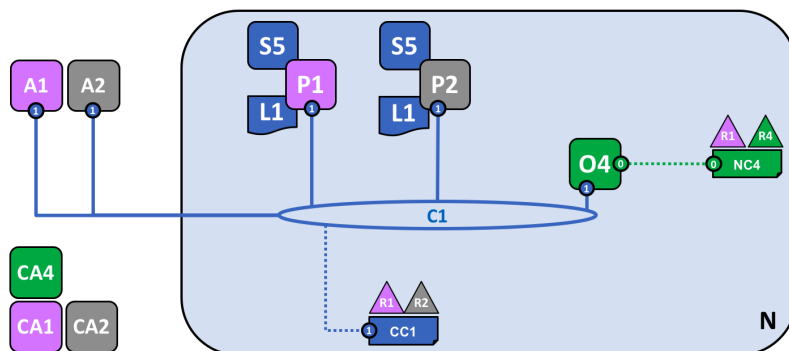
Để có thể triển khai một mạng Hyperledger Fabric, kiến thức về kiến trúc của mạng là yêu cầu tiên quyết. Chương này sẽ trình bày về tổng quan kiến trúc và các thành phần trong một mạng Hyperledger Fabric.

2.1 Tổng quan

Như đã giới thiệu, Hyperledger Fabric là một trong những nền tảng mạng chuỗi khối riêng tư nổi tiếng nhất. Mạng Hyperledger Fabric hỗ trợ các tính năng sau:

- **Quản lý định danh:** Hyperledger Fabric sử dụng hạ tầng khóa công khai và chữ ký số để xác thực, cấp quyền cho các tác nhân trong mạng.
- **Xử lý giao dịch tách biệt:** Để tăng khả năng xử lý song song, việc cập nhật giao dịch mới vào sổ cái và xác thực tính hợp lệ của các giao dịch được tách riêng biệt.
- **Chaincode:** Là ứng dụng phi tập trung (hợp đồng thông minh) mà thông qua đó có thể tương tác với sổ cái kỹ thuật số. Chaincode này có thể được viết bởi các ngôn ngữ lập trình phổ thông thay vì phải viết trên các ngôn ngữ lập trình chuyên biệt.
- **Tính mô-đun:** Các thành phần trong mạng có thể được tùy chỉnh, thay đổi.

2.2 Kiến trúc mạng Hyperledger Fabric



Hình 2.1: Kiến trúc một mạng Hyperledger Fabric cơ bản. Nguồn [1]

Các thành phần trong mạng ở Hình 2.1 bao gồm:

- **N:** Mạng (Network).
- **NC:** Cấu hình mạng (Network Configuration).
- **C:** Kênh (Channel): Tập trung các Tổ chức có chung mục đích kinh doanh.
- **CC:** Cấu hình kênh (Channel Configuration).

- **R:** Tổ chức (Organization).
- **O:** Orderer Node: Node duy nhất trong mạng xử lý quá trình đồng thuận.
- **P:** Peer Node: Nơi lưu trữ sổ cái kỹ thuật số. Các tương tác với hợp đồng thông minh (chaincode) đều phải thông qua Peer node.
- **S:** Hợp đồng thông minh (chaincode).
- **L:** Sổ cái kỹ thuật số (Ledger).
- **CA:** Nhà cung cấp chứng thực số (Certificate authority): Cung cấp danh tính cho tất cả các thành phần trong mạng.
- **A:** Ứng dụng hay giao diện (Application): Hỗ trợ tương tác hệ thống.

2.3 Khởi tạo một mạng Hyperledger Fabric

2.3.1 Bước 1: Khởi tạo Orderer Node

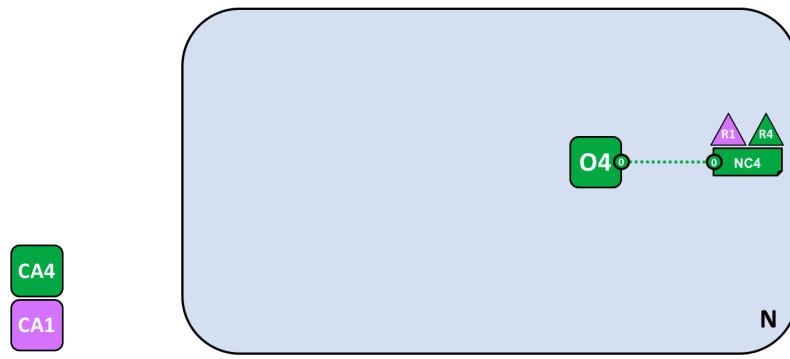


Hình 2.2: Bước 1: Khởi tạo Orderer Node. Nguồn [1]

Bước đầu tiên là khởi tạo một Orderer Node. Như ở Hình 2.2, để triển khai mạng N, trước nhất cần có Orderer Node O4 được khởi chạy và tùy chỉnh với cấu hình mạng NC4. Cấu hình mạng NC4 giao quyền quản trị mạng cho tổ chức R4. Là một mạng chuỗi khối riêng tư, việc định danh các tác nhân là bắt buộc. Nhà cung cấp chứng thực từ số CA4 sẽ cung cấp định danh cho các node và cá nhân thuộc tổ chức R4. O4 cũng trực thuộc tổ chức R4 và sẽ được R4 cung cấp cho một định danh. Tất cả các hành động sau này như thêm tổ chức vào vào mạng, thêm kênh, cài đặt chaincode cho kênh, khởi tạo chaincode, yêu cầu thực thi chaincode, ... đều phải đi qua Orderer O4 này. Và trong Hyperledger Fabric, tất cả các hành động trên đều là giao dịch (transaction).

2.3.2 Bước 2: Thêm Tổ chức quản trị

Cấu hình mạng NC4 ban đầu chỉ cho phép Tổ chức R4 có quyền quản trị trên mạng. R4 có thể thêm R1 làm một tổ chức quản trị khác nữa trong mạng. Sau thời điểm này hai tổ chức R1 và R4 sẽ đều có quyền quản trị mạng tương đương nhau.



Hình 2.3: Bước 2: Thêm Tổ chức quản trị. Nguồn [1]

Tương tự như CA4, nhà cung cấp chứng từ số CA1 sẽ cung cấp chứng từ số cho các thực thể thuộc R1.

2.3.3 Bước 3: Định nghĩa Consortium

Hiện tại dù mạng có thể được quản trị bởi R1 và R4, vẫn chưa có nhiều nghiệp vụ có thể được thực hiện. Để có thể được ứng dụng vào các hoạt động kinh doanh, điều đầu tiên cần thực hiện là định nghĩa một Consortium. Consortium có thể hiểu là một liên doanh - kết nối nhiều tổ chức mà có hoạt động liên quan đến nhau.



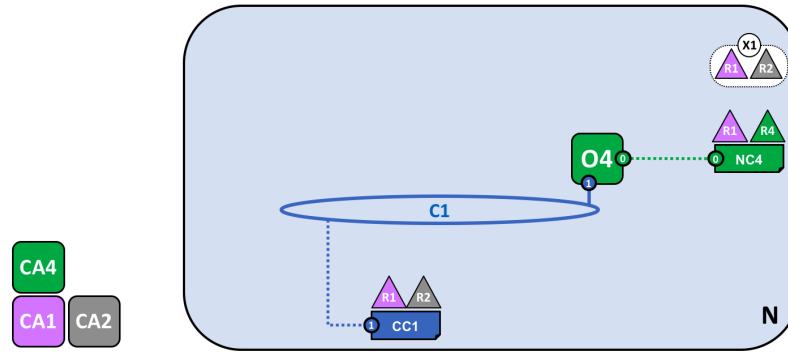
Hình 2.4: Bước 3: Định nghĩa Consortium. Nguồn [1]

Một quản trị viên mạng (R1 hoặc R4) định nghĩa một Consortium X1 với hai tổ chức, R1 và R2. Định nghĩa của consortium này được lưu trữ trong cấu hình mạng NC4 và sẽ được sử dụng ở các bước phát triển mạng tiếp theo. CA2 cũng được sử dụng để cung cấp định danh cho R2. Lưu ý là số lượng tổ chức trong một Consortium là tùy ý.

2.3.4 Bước 4: Tạo kênh tương ứng với Consortium

Kênh có thể được hiểu là một phương tiện truyền thông tin mà các tổ chức cùng chung một Consortium có thể sử dụng để giao tiếp với nhau.

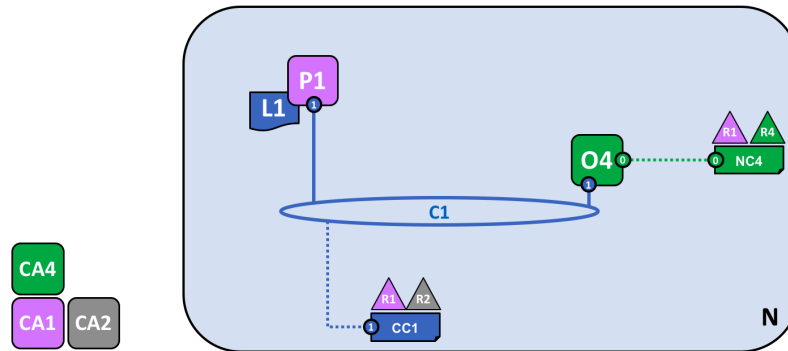
Kênh C1 được tạo sử dụng Consortium X1. Cấu hình kênh CC1 của C1 được tách biệt hoàn toàn với cấu hình mạng NC4. CC1 được quản lý bởi R1 và R2, 2 tổ



Hình 2.5: Bước 4: Tạo kênh tương ứng với Consortium. Nguồn [1]

chức này có quyền ngang nhau đối với C1. R4 dù là một quản trị mạng nhưng sẽ không có quyền gì đối với CC1.

2.3.5 Bước 5: Thêm Peer node



Hình 2.6: Bước 5: Thêm Peer node. Nguồn [1]

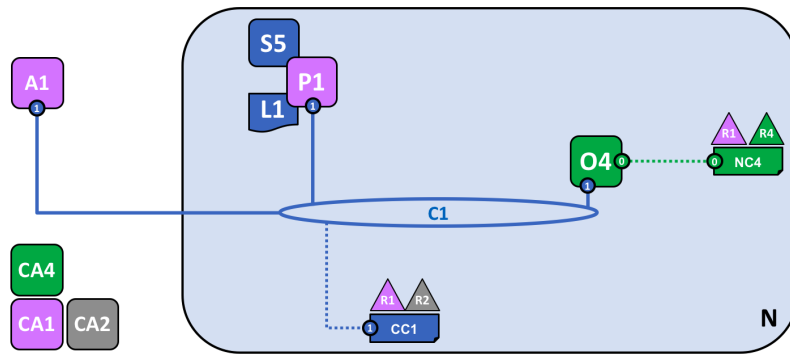
Peer P1 được thêm vào kênh C1. Trên kênh C1 sẽ tồn tại một sổ cái duy nhất và P1 lưu trữ một bản sao L1 của cuốn sổ cái này. Có thể nói, L1 được lưu trữ vật lý trên P1 và được lưu trữ một cách lô-gíc trên kênh C1. P1 thuộc tổ chức R1, do đó sẽ được cung cấp một định danh từ CA1. Khi được khởi chạy, P1 sẽ gửi yêu cầu tham gia kênh C1 tới O4. Dựa trên định danh của P1 và cấu hình kênh CC1, quyền của P1 trên kênh sẽ được xác định.

2.3.6 Bước 6: Cài đặt Hợp đồng thông minh

Sau khi đã tồn tại sổ cái, hợp đồng thông minh có thể được triển khai để ứng dụng vào nghiệp vụ.

Hợp đồng thông minh S5 được cài đặt trên Peer P1. Ứng dụng ngoài A1 có thể sử dụng S5 để truy cập vào sổ cái L1 thông qua peer P1. A1 để có thể được sử dụng cũng sẽ cần định danh, ví dụ A1 có thể sở hữu một định danh được CA1 cung cấp.

Cần lưu ý rằng A1 không thể tương tác trực tiếp sổ cái L1 trực tiếp thông qua P1, mà tất cả quyền truy cập tới sổ cái sẽ được quản lý thông qua S5. Có thể hiểu



Hình 2.7: Bước 6: Cài đặt Hợp đồng thông minh. Nguồn [1]

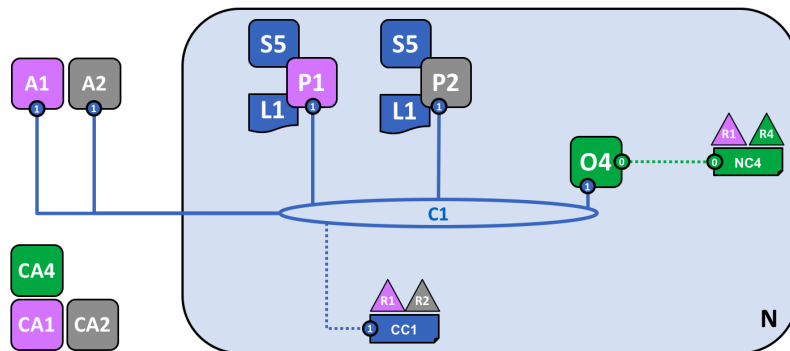
là S5 định nghĩa và cung cấp một tập hợp các phương thức mà theo đó sổ cái L1 có thể được truy vấn hay cập nhật.

S5 sẽ không chỉ được cài đặt trên P1, mà còn trên cả kênh C1. Điều này đảm bảo các thành phần kết nối với C1 sẽ biết đến S5. Tuy nhiên những thành phần khác ngoài P1 sẽ không thể nhìn thấy cấu trúc chi tiết của S5.

Một kênh có thể có nhiều Hợp đồng thông minh.

2.3.7 Bước 7: Khởi tạo mạng hoàn tất

Để hoàn tất khởi tạo mạng, Tổ chức R2 sẽ được thêm vào.



Hình 2.8: Bước 7: Khởi tạo mạng hoàn tất. Nguồn [1]

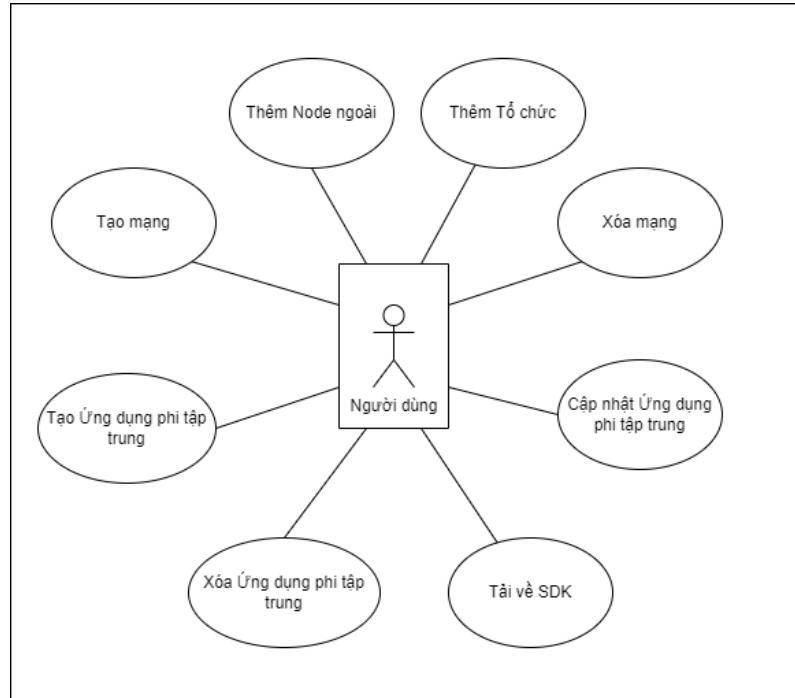
R2 sẽ thêm Peer node P2, cài đặt S5 trên đó. P2 cũng sẽ có một bản sao của sổ cái L1. A2 sử dụng định danh được cung cấp bởi CA2 có thể tương tác với L1 thông qua S5. Lúc này, hai tổ chức R1 và R2 đã có thể giao dịch với nhau thông qua kênh C1.

Chương 2 đã trình bày về tổng quan kiến trúc của một mạng Hyperledger Fabric. Đây sẽ là tiền đề để thiết kế một hệ thống có thể tự động triển khai mạng và ứng dụng phi tập trung. Chương tiếp theo sẽ phân tích các yêu cầu dựa trên quá trình khởi tạo một mạng như đã nói ở chương này.

CHƯƠNG 3. KHẢO SÁT VÀ PHÂN TÍCH YÊU CẦU

3.1 Tổng quan chức năng

3.1.1 Biểu đồ use case tổng quan



Hình 3.1: Biểu đồ usecase Tổng quan

Hình 3.1 là biểu đồ usecase tổng quát của hệ thống. Hệ thống chỉ bao gồm duy nhất một tác nhân, đó là người dùng. Người dùng có thể tạo, xóa mạng Hyperledger Fabric. Đối với các mạng đã được tạo, người dùng còn thể thêm một node ngoài vào trong hệ thống, cùng với đó là thêm một tổ chức mới. Trên các mạng đó, người dùng còn có thể tạo một ứng dụng phi tập trung mới, cập nhật và xóa các ứng dụng phi tập trung đã tồn tại. Cuối cùng, người dùng có thể tải về các SDK để tương tác với các ứng dụng phi tập trung tương ứng.

3.2 Đặc tả chức năng

3.2.1 Đặc tả use case Tạo mạng

Bảng 3.1: Đặc tả ca sử dụng “Tạo mạng”

Mã use case:	UC01		
Tên use case:	Tạo mạng		
Tác nhân:	Người dùng		
Mô tả:	Người dùng tạo một mạng Hyperledger Fabric mới		
Điều kiện tiên quyết:	Người dùng đã đăng nhập vào hệ thống		
Hậu điều kiện:	Hiện thị kết quả tạo mạng		
Luồng sự kiện chính:	STT	Thực hiện bởi	Hành động
	1	Người dùng	Nhấn nút tạo mạng trên giao diện
	2	Người dùng	Nhập thông tin cấu hình mạng: (i) Tên mạng, (ii) Số máy ảo, (iii) Cấu hình mỗi máy ảo, (iv) Số Tổ chức, (v) Số peer node cho mỗi tổ chức
	3	Hệ thống	Tự động triển khai mạng theo thông tin người dùng đã nhập
Luồng sự kiện con:	Không có.		
Ngoại lệ:	Thông tin nhập vào không hợp lệ. Hệ thống sẽ thông báo lỗi.		
Bao gồm:	Không có.		

3.2.2 Đặc tả use case Thêm node ngoài**Bảng 3.2:** Đặc tả ca sử dụng “Thêm Node ngoài”

Mã use case:	UC02		
Tên use case:	Thêm Node ngoài		
Tác nhân:	Người dùng		
Mô tả:	Tác nhân thêm một node ngoài vào một mạng đang hoạt động của mình		
Điều kiện tiên quyết:	Người dùng đã tạo một mạng		
Hậu điều kiện:	Hiện thị kết quả thêm node ngoài		
Luồng sự kiện chính:	STT	Thực hiện bởi	Hành động
	1	Người dùng	Nhấn nút thêm node ngoài trên giao diện
	2	Người dùng	Nhập thông tin: (i) Tên node, (ii) Tổ chức Node sẽ thuộc về, (iii) Địa chỉ IP và cổng mạng node sẽ hoạt động
	3	Hệ thống	Thực hiện sinh các tệp tin để có cho phép một Node ngoài với thông tin người dùng đã cung cấp có thể tham gia vào mạng
	4	Người dùng	Tải tệp tin hệ thống sinh ra về
	5	Người dùng	Khởi chạy các tệp tin được hệ thống sinh trên máy có đại chỉ Ip và cổng mạng lúc nhập để máy đó tham gia vào mạng
Luồng sự kiện con:	Không có.		
Ngoại lệ:	Thông tin nhập vào không hợp lệ, mạng chưa hoạt động. Hệ thống sẽ thông báo lỗi.		
Bao gồm:	Không có.		

3.2.3 Đặc tả use case Thêm tổ chức

Bảng 3.3: Đặc tả ca sử dụng “Thêm tổ chức”

Mã use case:	UC03		
Tên use case:	Thêm tổ chức		
Tác nhân:	Người dùng		
Mô tả:	Người dùng thêm một tổ chức mới vào một mạng đang hoạt động của mình		
Điều kiện tiên quyết:	Người dùng đã tạo một mạng		
Hậu điều kiện:	Hiện thị kết quả thêm tổ chức		
Luồng sự kiện chính:	STT	Thực hiện bởi	Hành động
	1	Người dùng	Nhấn nút thêm tổ chức trên giao diện
	2	Người dùng	Nhập thông tin tổ chức mới: (i) Tên tổ chức, (ii) Số peer thuộc tổ chức mới
	3	Hệ thống	Tự động thêm một tổ chức mới vào mạng
Luồng sự kiện con:	Không có.		
Ngoại lệ:	Thông tin nhập vào không hợp lệ, mạng chưa hoạt động. Hệ thống sẽ thông báo lỗi.		
Bao gồm:	Không có.		

3.2.4 Đặc tả use case Xóa mạng**Bảng 3.4:** Đặc tả ca sử dụng “Xóa mạng”

Mã use case:	UC04		
Tên use case:	Xóa mạng		
Tác nhân:	Người dùng		
Mô tả:	Người dùng xóa một mạng của mình		
Điều kiện tiên quyết:	Người dùng đã tạo một mạng		
Hậu điều kiện:	Hiện thị kết quả xóa mạng		
Luồng sự kiện chính:	STT	Thực hiện bởi	Hành động
	1	Người dùng	Nhấn nút xóa mạng trên giao diện
	2	Người dùng	Xác nhận sẽ xóa mạng
	3	Hệ thống	Gỡ bỏ toàn bộ các thành phần trong mạng
Luồng sự kiện con:	Không có.		
Ngoại lệ:	Mạng chưa hoạt động. Hệ thống sẽ báo lỗi		
Bao gồm:	Không có.		

3.2.5 Đặc tả use case Tạo Ứng dụng phi tập trung

Bảng 3.5: Đặc tả ca sử dụng “Tạo Ứng dụng phi tập trung”

Mã use case:	UC05		
Tên use case:	Tạo Ứng dụng phi tập trung		
Tác nhân:	Người dùng		
Mô tả:	Người dùng tạo Ứng dụng phi tập trung trên một mạng đang hoạt động của mình		
Điều kiện tiên quyết:	Người dùng đã tạo một mạng		
Hậu điều kiện:	Hiện thị kết quả tạo ứng dụng phi tập trung		
Luồng sự kiện chính:	STT	Thực hiện bởi	Hành động
	1	Người dùng	Nhấn nút tạo ứng dụng trên giao diện
	2	Người dùng	Nhập thông tin: (i) Tên ứng dụng, (ii) Mạng mà ứng dụng sẽ được triển khai lên, (iii) Kiểu mã hóa
	3	Người dùng	Thao tác trên giao diện để khai báo kiến trúc của ứng dụng: (i) Tạo thực thể, (ii) Thêm thuộc tính cho các thực thể, (iii) Thêm quan hệ cho các thực thể (1-1, 1-n, n-n)
	4	Người dùng	Xác nhận tạo ứng dụng
	5	Hệ thống	Tự động triển khai ứng dụng tương ứng lên mạng đã chọn
	6	Hệ thống	Sinh ra SDK để có thể sử dụng ứng dụng.
Luồng sự kiện con:	Không có.		
Ngoại lệ:	Thông tin nhập vào không hợp lệ, mạng chưa hoạt động. Hệ thống sẽ thông báo lỗi.		
Bao gồm:	Không có.		

3.2.6 Đặc tả use case Tải SDK**Bảng 3.6:** Đặc tả ca sử dụng “Tải SDK”

Mã use case:	UC06		
Tên use case:	Tải SDK		
Tác nhân:	Người dùng		
Mô tả:	Người dùng tải SDK tương ứng với một ứng dụng của mình		
Điều kiện tiên quyết:	Người dùng đã tạo một ứng dụng phi tập trung		
Hậu điều kiện:	Hiện thị kết quả tải về SDK		
Luồng sự kiện chính:	STT	Thực hiện bởi	Hành động
	1	Người dùng	Nhấn nút tải SDK trên giao diện
	2	Hệ thống	Tải về tệp tin chứa SDK
	3	Người dùng	Sử dụng SDK
Luồng sự kiện con:	Không có.		
Ngoại lệ:	Ứng dụng chưa hoạt động. Hệ thống sẽ thông báo lỗi.		
Bao gồm:	Không có.		

3.2.7 Đặc tả use case Cập nhật Ứng dụng phi tập trung

Bảng 3.7: Đặc tả ca sử dụng “Cập nhật Ứng dụng phi tập trung”

Mã use case:	UC07		
Tên use case:	Cập nhật Ứng dụng phi tập trung		
Tác nhân:	Người dùng		
Mô tả:	Người dùng cập nhật một ứng dụng phi tập trung của mình		
Điều kiện tiên quyết:	Người dùng đã tạo một ứng dụng phi tập trung		
Hậu điều kiện:	Hiện thị kết quả cập nhật ứng dụng		
Luồng sự kiện chính:	STT	Thực hiện bởi	Hành động
	1	Người dùng	Nhấn nút cập nhật ứng dụng trên giao diện
	2	Người dùng	Nhập thông tin mới: Tên mới cho ứng dụng
	3	Người dùng	Thao tác trên giao diện để chỉnh sửa kiến trúc của ứng dụng: (i) Thêm thực thể, (ii) Thêm thuộc tính cho thực thể mới và cũ, (iii) Thêm quan hệ cho các thực thể mới và cũ (1-1, 1-n, n-n)
	4	Người dùng	Xác nhận cập nhật ứng dụng
	5	Hệ thống	Tự động triển khai cập nhật ứng dụng
Luồng sự kiện con:	6	Hệ thống	Cập nhật SDK mới
	Không có.		
Ngoại lệ:	Thông tin nhập vào không hợp lệ, mạng chưa hoạt động, ứng dụng chưa hoạt động. Hệ thống sẽ thông báo lỗi.		
Bao gồm:	Không có.		

3.2.8 Đặc tả use case Xóa Ứng dụng phi tập trung

Bảng 3.8: Đặc tả ca sử dụng “Xóa Ứng dụng phi tập trung”

Mã use case:	UC08		
Tên use case:	Xóa Ứng dụng phi tập trung		
Tác nhân:	Người dùng		
Mô tả:	Người dùng xóa một ứng dụng phi tập trung của mình		
Điều kiện tiên quyết:	Người dùng đã tạo một ứng dụng phi tập trung		
Hậu điều kiện:	Hiện thị kết quả xóa ứng dụng		
Luồng sự kiện chính:	STT	Thực hiện bởi	Hành động
	1	Người dùng	Nhấn nút xóa ứng dụng trên giao diện
	2	Người dùng	Xác nhận xóa ứng dụng
	3	Hệ thống	Xóa ứng dụng khỏi mạng
Luồng sự kiện con:	Không có.		
Ngoại lệ:	Mạng chưa hoạt động, ứng dụng chưa hoạt động. Hệ thống sẽ thông báo lỗi.		
Bao gồm:	Không có.		

CHƯƠNG 4. 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 5. THỰC NGHIỆM VÀ ĐÁNG GIÁ

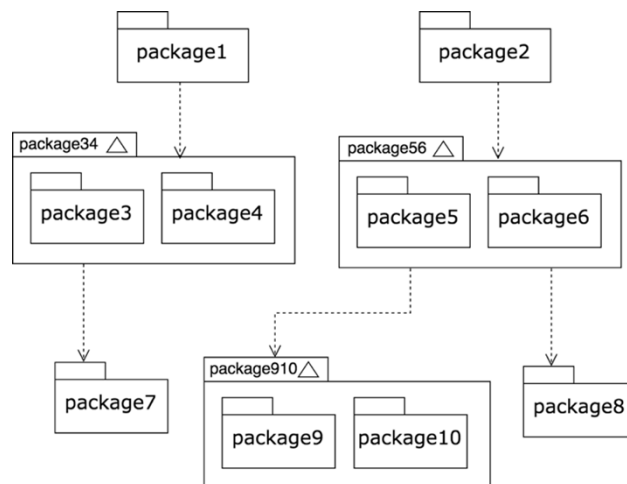
5.1 Thiết kế kiến trúc

5.1.1 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.

5.1.2 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 5.1



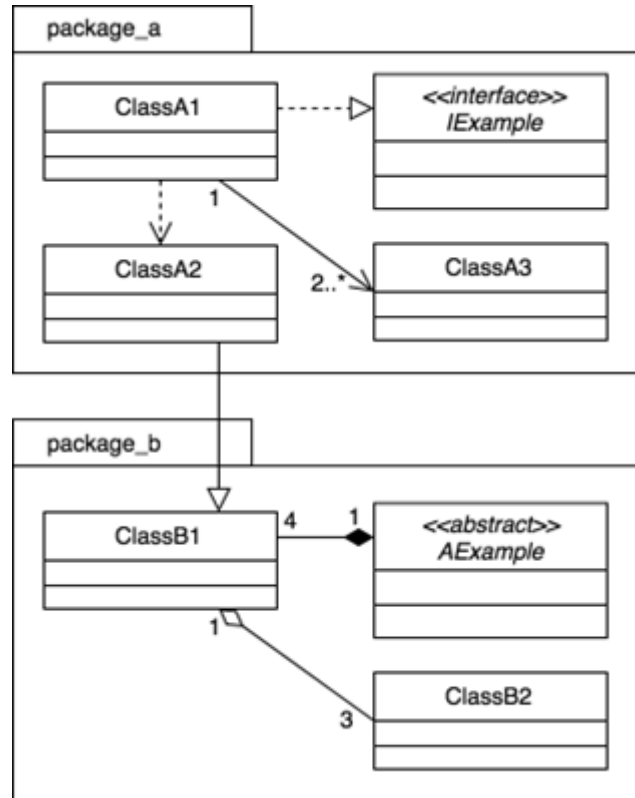
Hình 5.1: Ví dụ biểu đồ phụ thuộc gói

5.1.3 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 5.2.

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 5.2.

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



Hình 5.2: Ví dụ thiết kế gói

5.2 Thiết kế chi tiết

5.2.1 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.

5.2.2 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).

5.2.3 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.)

5.3 Xây dựng ứng dụng

5.3.1 Thư viện và công cụ sử dụng

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

Mục đích	Công cụ	Địa chỉ URL
IDE lập trình	Eclipse Oxygen a64 bit	http://www.eclipse.org/
v.v.	v.v.	v.v.

Bảng 5.1: Danh sách thư viện và công cụ sử dụng

5.3.2 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.

5.3.3 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.

5.4 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.

5.5 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 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 >

[2] 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>

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

[4] 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>

[5] 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>

[6] 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>

[7] 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).

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

TÀI LIỆU THAM KHẢO

- [1] *Fabric guide*, <https://hyperledger-fabric.readthedocs.io/en/release-2.2/network/network.html>, Last visited 14-May-2019.
- [2] E. H. Hovy, “Automated discourse generation using discourse structure relations,” *Artificial intelligence*, vol. 63, no. 1-2, pp. 341–385, 1993.
- [3] L. L. Peterson and B. S. Davie, *Computer networks: a systems approach*. Elsevier, 2007.
- [4] N. T. Hải, *Mạng máy tính và các hệ thống mở*. Nhà xuất bản giáo dục, 1999.
- [5] 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.
- [6] A. Knott, “A data-driven methodology for motivating a set of coherence relations,” Ph.D. dissertation, The University of Edinburgh, UK, 1996.
- [7] 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).
- [8] Princeton University, *Wordnet*. [Online]. Available: <http://www.cogsci.princeton.edu/~wn/index.shtml> (visited on 09/30/2010).

PHỤ LỤC

A. HƯỚNG DẪN VIẾT ĐỒ ÁN TỐT NGHIỆP

Quy định chung

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

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

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

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

Sinh viên tuyệt đối không trình bày ĐATN theo kiểu viết ý hoặc gạch đầu dòng. ĐATN không phải là một slide thuyết trình; khi người đọc không hiểu sẽ không có ai giải thích hộ. Sinh viên cần viết thành các đoạn văn và phân tích, diễn giải đầy đủ, rõ ràng. Câu văn cần đúng ngữ pháp, đầy đủ chủ ngữ, vị ngữ và các thành phần câu. Khi thực sự cần liệt kê, sinh viên nên liệt kê theo phong cách khoa học với các ký tự La Mã. Ví dụ, nhiều sinh viên luôn cảm thấy hối hận vì (i) chưa cố gắng hết mình, (ii) chưa sắp xếp thời gian học/choi một cách hợp lý, (iii) chưa tìm được người yêu để chia sẻ quãng đời sinh viên vất vả, và (iv) viết ĐATN một cách cẩu thả.

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

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

trong từng trang báo cáo ĐATN.

A.1 Ngành học

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

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

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

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

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

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

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

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

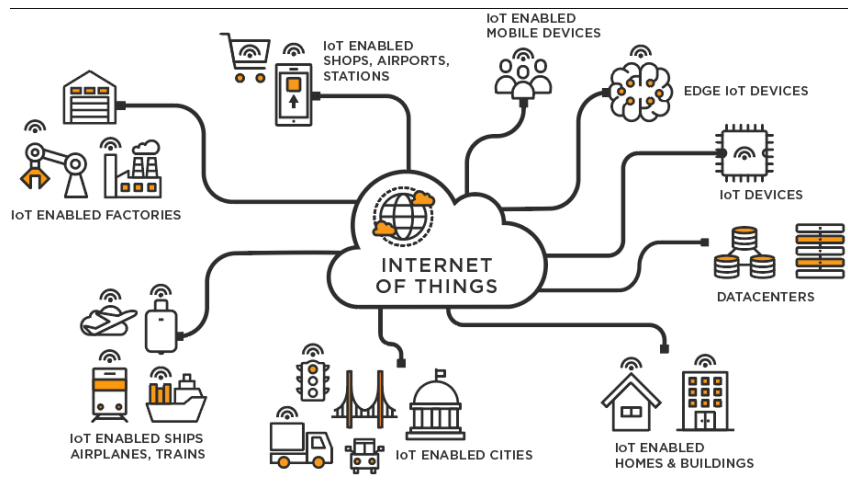
Chú ý các nội dung trình bày trong cả hai môi trường liệt kê theo sau lệnh `\item`. Ngoài ra LaTeX còn cung cấp một số kiểu liệt kê khác, sinh viên có thể tham khảo tại <https://www.overleaf.com/learn/latex/Lists>

A.3 Cách thêm bảng

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>.

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.



Hình A.1: Internet vạn vật

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.

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).

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”

...