

ĐẠI HỌC BÁCH KHOA HÀ NỘI

ĐỒ ÁN TỐT NGHIỆP

Xây dựng ứng dụng đặt người lái xe hộ

Triệu Tuyên Nhâm

nham.tt204771@sis.hust.edu.vn

Trường Công nghệ thông tin và truyền thông

Giảng viên hướng dẫn: TS. Đỗ Tiến Dũng

Chữ kí GVHD

Khoa:

Kỹ thuật máy tính

Trường:

Công nghệ Thông tin và Truyền thông

HÀ NỘI, 12/2024

LỜI CẢM ƠN

Trong quá trình thực hiện và hoàn thành Đồ án tốt nghiệp, em xin gửi lời cảm ơn sâu sắc đến tất cả những người đã đồng hành, động viên và hỗ trợ em trong suốt thời gian qua.

Trước tiên, em xin gửi lời cảm ơn chân thành tới TS. Đỗ Tiến Dũng, người thầy đã tận tâm chỉ dẫn và đưa ra những góp ý giúp đỡ em từng bước trong quá trình thực hiện đồ án.

Em cũng xin chân thành cảm ơn Đại học Bách Khoa Hà Nội, cùng quý thầy cô trong trường đặc biệt là các thầy cô của Trường Công nghệ Thông tin và Truyền thông, những người đã truyền đạt cho em những kiến thức quý giá và tạo điều kiện thuận lợi cho quá trình học tập và nghiên cứu của em.

Bên cạnh đó, em không thể không nhắc đến gia đình và những người bạn thân yêu. Chính sự quan tâm, động viên và chia sẻ của mọi người đã tiếp thêm sức mạnh để em vượt qua khó khăn và hoàn thành đồ án này.

Cuối cùng, em rất mong nhận được những góp ý từ thầy cô và các bạn để có thể cải thiện và hoàn thiện đồ án của mình hơn nữa.

Em xin chân thành cảm ơn!

TÓM TẮT NỘI DUNG ĐỒ ÁN

Từ lâu, vấn đề tai nạn giao thông do sử dụng rượu bia luôn là một mối lo ngại lớn đối với nhà nước và người dân. Mặc dù nhà nước đã ban hành nhiều nghị định xử phạt những cá nhân vi phạm như: phạt hành chính, tước giấy phép lái xe, ... Tuy nhiên việc loại bỏ văn hóa uống rượu bia là điều không thể và không nên. Vì vậy, để giải quyết vấn đề này, các giải pháp cho phép người dân vừa có thể sử dụng đồ uống có cồn như rượu bia mà vẫn an toàn khi tham gia giao thông đã xuất hiện như đặt xe ôm, đặt taxi hay như đặt người lái xe hộ đã xuất hiện. Tuy nhiên, ở Việt Nam thì việc đặt xe ôm hay taxi đã khá phổ biến nhưng bên cạnh đó dịch vụ đặt người lái hộ vẫn còn khá mới mẻ và chưa được phổ biến rộng rãi.

Do đó, em quyết định xây dựng một ứng dụng cho phép mọi người có thể đặt người lái hộ khi cần thiết. Mục tiêu của ứng dụng này là đem lại cho người dùng trải nghiệm tốt và dễ dàng để sử dụng.

Để có thể thực hiện được điều này, đồ án là sự kết hợp của framework Flutter, NestJS và Firebase. Bên cạnh đó còn có sự hỗ trợ đến từ API của Google Map Platform hỗ trợ việc tìm kiếm, định tuyến các địa điểm cũng như tính toán khoảng cách và độ ưu tiên trong việc phân phối tài xế 1 cách hiệu quả.

Qua đó, ứng dụng đã phần nào giải quyết được các vấn đề ban đầu, mang lại trải nghiệm mượt mà, tiện lợi cho người dùng. Ứng dụng không chỉ giải quyết được vấn đề tai nạn giao thông do sử dụng rượu bia mà còn góp phần nâng cao ý thức chấp hành luật an toàn giao thông của mọi người, từ đó tạo nên một môi trường giao thông văn minh, an toàn hơn.

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

MỤC LỤC

| | |
|---|-----------|
| CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI..... | 1 |
| 1.1 Đặt vấn đề..... | 1 |
| 1.2 Mục tiêu và phạm vi đề tài..... | 1 |
| 1.3 Định hướng giải pháp..... | 1 |
| 1.4 Bố cục đồ án | 1 |
| CHƯƠNG 2. KHẢO SÁT VÀ PHÂN TÍCH YÊU CẦU..... | 4 |
| 2.1 Khảo sát hiện trạng | 4 |
| 2.2 Tổng quan chức năng | 5 |
| 2.2.1 Biểu đồ use case tổng quát | 5 |
| 2.2.2 Biểu đồ use case phân rã Xác thực | 6 |
| 2.2.3 Biểu đồ use case phân rã Xác thực | 7 |
| 2.2.4 Biểu đồ use case phân rã Đặt tài xế | 8 |
| 2.2.5 Biểu đồ use case phân rã Nhận chuyển..... | 9 |
| 2.2.6 Quy trình nghiệp vụ | 10 |
| 2.2.7 Quy trình nghiệp vụ Đặt tài xế | 10 |
| 2.3 Đặc tả chức năng | 11 |
| 2.3.1 Đặc tả Đặt tài xế | 11 |
| 2.3.2 Đặc tả use case Nhận chuyển..... | 12 |
| 2.3.3 Đặc tả use case Tìm kiếm tài xế..... | 13 |
| 2.3.4 Đặc tả use case Chỉnh sửa thông tin | 14 |
| 2.4 Yêu cầu phi chức năng | 14 |
| CHƯƠNG 3. CÔNG NGHỆ SỬ DỤNG..... | 15 |
| 3.1 Flutter..... | 15 |
| 3.1.1 Giới thiệu | 15 |

| | |
|---|-----------|
| 3.1.2 Mục đích sử dụng..... | 15 |
| 3.1.3 Ưu - Nhược điểm | 15 |
| 3.2 NestJS | 16 |
| 3.2.1 Giới thiệu | 16 |
| 3.2.2 Mục đích sử dụng..... | 16 |
| 3.2.3 Ưu - Nhược điểm | 16 |
| 3.3 Firebase | 17 |
| 3.3.1 Giới thiệu | 17 |
| 3.3.2 Mục đích sử dụng..... | 17 |
| 3.3.3 Ưu - Nhược điểm | 17 |
| CHƯƠNG 4. THIẾT KẾ, TRIỂN KHAI VÀ ĐÁNH GIÁ HỆ THỐNG | 18 |
| 4.1 Thiết kế kiến trúc..... | 18 |
| 4.1.1 Lựa chọn kiến trúc phần mềm | 18 |
| 4.1.2 Thiết kế tổng quan..... | 18 |
| 4.1.3 Thiết kế chi tiết gói | 18 |
| 4.2 Thiết kế chi tiết..... | 19 |
| 4.2.1 Thiết kế giao diện | 19 |
| 4.2.2 Thiết kế lớp | 19 |
| 4.2.3 Thiết kế cơ sở dữ liệu | 20 |
| 4.3 Xây dựng ứng dụng..... | 20 |
| 4.3.1 Thư viện và công cụ sử dụng | 20 |
| 4.3.2 Kết quả đạt được | 20 |
| 4.3.3 Minh họa các chức năng chính | 20 |
| 4.4 Kiểm thử..... | 20 |
| 4.5 Triển khai | 21 |
| CHƯƠNG 5. CÁC GIẢI PHÁP VÀ ĐÓNG GÓP NỔI BẬT..... | 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..... | 27 |
| A. HƯỚNG DẪN VIẾT ĐỒ ÁN TỐT NGHIỆP | 27 |
| A.1 Ngành học..... | 28 |
| A.2 Đánh dấu (bullet) và đánh số (numering) | 28 |
| A.3 Cách thêm bảng | 28 |
| A.4 Chèn hình ảnh | 29 |
| A.5 Tài liệu tham khảo | 29 |
| A.6 Cách viết phương trình và công thức toán học..... | 30 |
| A.7 Qui cách đóng quyển..... | 30 |
| 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 | Biểu đồ use case tổng quát | 5 |
| Hình 2.2 | Biểu đồ usecase Xác thực | 6 |
| Hình 2.3 | Biểu đồ usecase Quản lý tài khoản | 7 |
| Hình 2.4 | Biểu đồ usecase Đặt tài xế | 8 |
| Hình 2.5 | Biểu đồ usecase Nhận chuyển | 9 |
| Hình 2.6 | Quy trình nghiệp vụ Đặt tài xế | 10 |
| Hình 4.1 | Ví dụ biểu đồ phụ thuộc gói | 18 |
| Hình 4.2 | Ví dụ thiết kế gói | 19 |
| Hình A.1 | Internet vạn vật | 29 |
| Hình A.2 | Qui cách đóng quyển đồ án | 30 |
| Hình A.3 | Qui cách đóng quyển đồ án | 31 |

DANH MỤC BẢNG BIỂU

| | | |
|----------|--|----|
| Bảng 2.1 | Đặc tả usecase Đặt tài xế. | 11 |
| Bảng 2.2 | Đặc tả usecase Nhận chuyển. | 12 |
| Bảng 2.3 | Đặc tả usecase Tìm kiếm tài xế. | 13 |
| Bảng 2.4 | Đặc tả usecase Thay đổi thông tin cá nhân. | 14 |
| Bảng 4.1 | Danh sách thư viện và công cụ sử dụng | 20 |
| Bảng A.1 | Table to test captions and labels. | 29 |

CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI

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.

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.

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 đồ án là gì, kết quả đạt được là gì.

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

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 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 đề 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 đề 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

2.1 Khảo sát hiện trạng

Hiện nay trên thị trường đã có một số ứng dụng hỗ trợ việc tìm và đặt người lái hộ như: FastGo, ViSafe, GOCheap,... Dưới đây là đánh giá về các ứng dụng này để tìm hiểu về các tính năng cũng như hạn chế của chúng:

FastGo: FastGo là một trong những ứng dụng thuê lái xe hộ khi say được yêu thích hiện nay. Được phát triển bởi công ty cổ phần FastGo Việt Nam, app này cung cấp dịch vụ lái xe chuyên nghiệp và an toàn cho người sử dụng.

- Ưu điểm:
 - Giao diện đơn giản, dễ sử dụng
 - Bảo mật thông tin cá nhân
 - Tính năng đặt xe và theo dõi hành trình dễ dàng
 - Hỗ trợ khách hàng 24/7
- Hạn chế:
 - Cần phải kết nối mạng ổn định
 - Không có tính năng đón khách

ViSafe: ViSafe là một ứng dụng thuê lái xe hộ khi say được phát triển bởi Công ty Cổ Phần An Toàn Giao Thông Việt Nam. App này nhắm đến việc cung cấp dịch vụ an toàn và chất lượng cho người sử dụng.

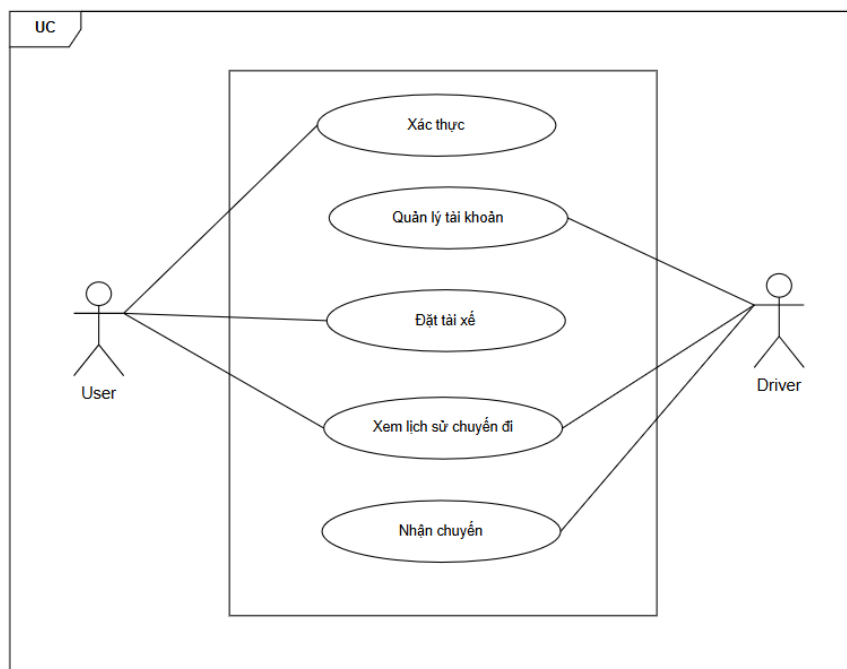
- Ưu điểm:
 - Giao diện đơn giản, dễ sử dụng
 - Bảo mật thông tin cá nhân
 - Hỗ trợ khách hàng 24/7
 - Đa dạng tính năng
- Hạn chế:
 - Không có tính năng theo dõi hành trình

GOCheap: GOCheap là một trong những app đặt lái xe hộ khi say tuyệt vời nhất ở thời điểm hiện tại. Ứng dụng này được phát triển và quản lý bởi Công ty TNHH GOCheap. Ứng dụng này nhắm đến mục tiêu cung cấp dịch vụ thuê lái xe hộ chất lượng và uy tín nhất cho người dùng.

- Ưu điểm:
 - Tiện lợi, nhanh chóng
 - Dịch vụ chất lượng
 - Bảo mật tốt
 - Hỗ trợ 24/7
- Hạn chế:
 - Giao diện chưa không thân thiện

2.2 Tổng quan chức năng

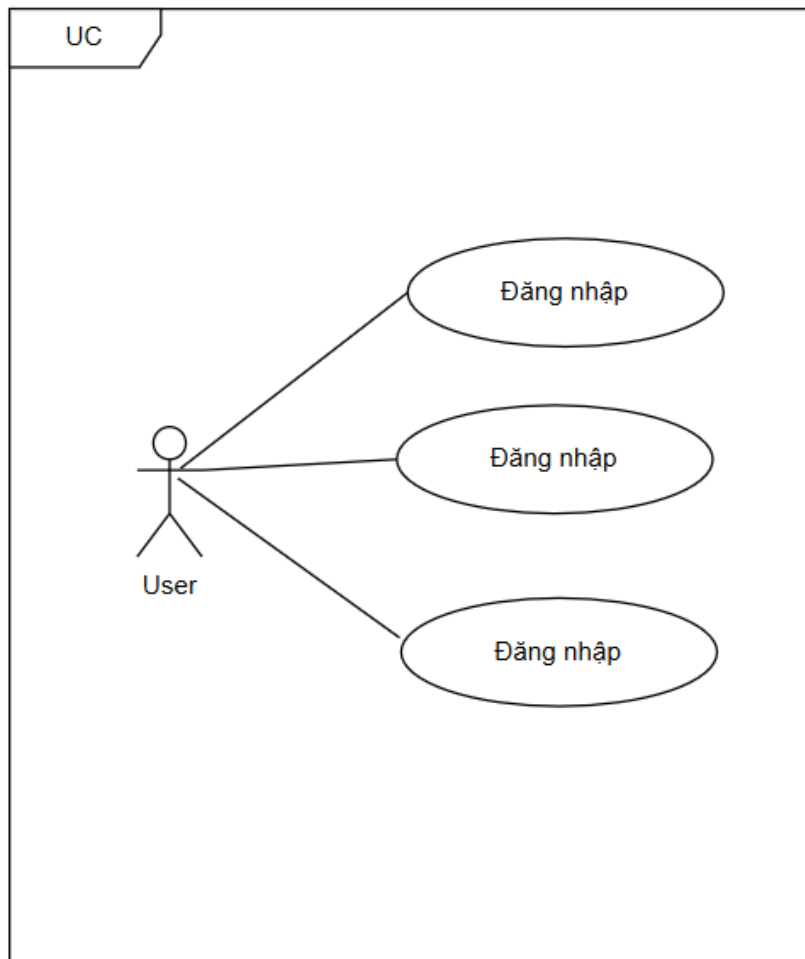
2.2.1 Biểu đồ use case tổng quát



Hình 2.1: Biểu đồ use case tổng quát

Hệ thống gồm 2 tác nhân là User và Driver. Để sử dụng các chức năng trong ứng dụng thì cả user và driver đều cần đăng nhập bằng số điện thoại và mật khẩu. Hệ thống bao gồm các usecase chính sau (i) usecase xác thực: xử lý việc đăng nhập, đăng ký của người dùng; (ii) usecase quản lý tài khoản: quản lý thông tin cá nhân, phương tiện của người dùng và tài xế; (iii) usecase đặt tài xế: tìm kiếm địa điểm, xem giá cước, chọn phương tiện, đặt tài xế; (iv) usecase xem lịch sử: người dùng xem lại lịch sử các chuyến đi của mình; (v) usecase nhận chuyến: tài xế nhận được thông tin về chuyến đi, xác nhận hoặc hủy chuyến

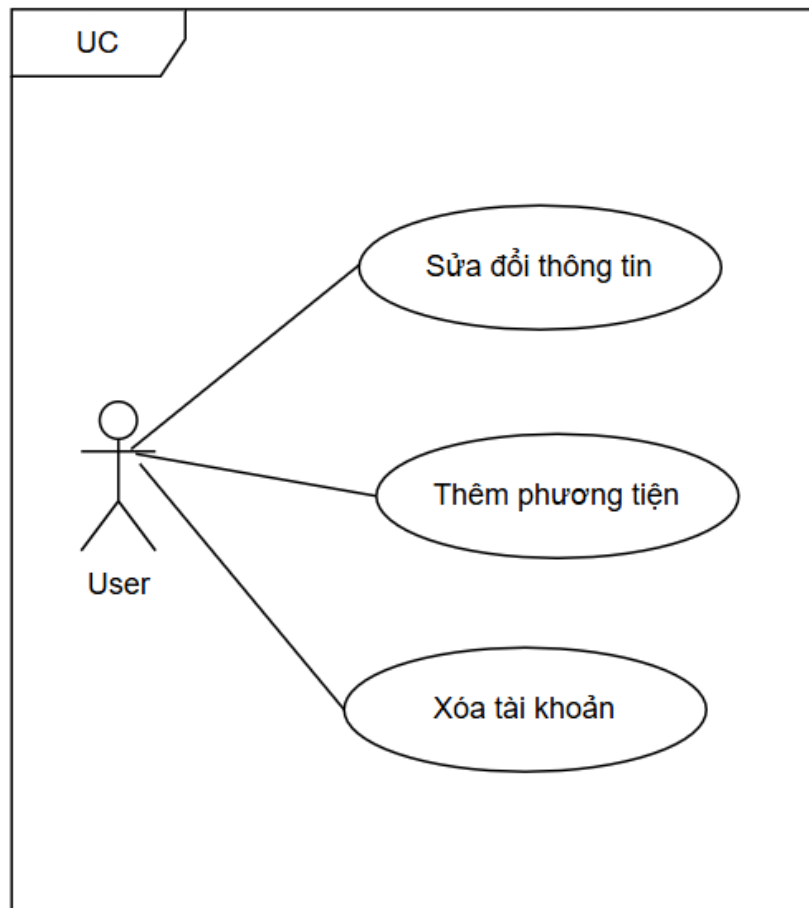
2.2.2 Biểu đồ use case phân rã Xác thực



Hình 2.2: Biểu đồ usecase Xác thực

Usecase Xác thực bao gồm những chức năng chính sau (i) đăng nhập: người dùng nhập số điện thoại và mật khẩu để đăng nhập vào hệ thống, (ii) đăng ký: người dùng nhập các thông tin bắt buộc để đăng ký, (iii) đăng xuất: Đăng xuất khỏi tài khoản đang được đăng nhập.

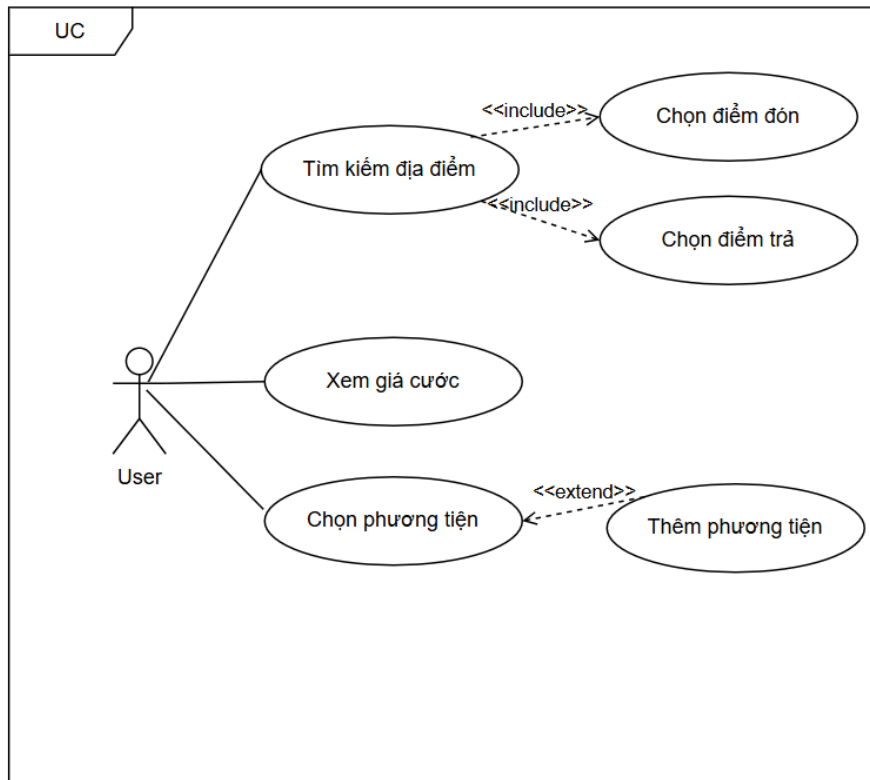
2.2.3 Biểu đồ use case phân rã Xác thực



Hình 2.3: Biểu đồ usecase Quản lý tài khoản

Usecase Quản lý tài khoản bao gồm những chức năng chính sau (i) sửa đổi thông tin cá nhân: người dùng sửa đổi các thông tin cá nhân của mình, (ii) thêm phương tiện: người dùng thêm phương tiện muốn lái hộ, (iii) xóa tài khoản: người dùng xóa tài khoản của mình.

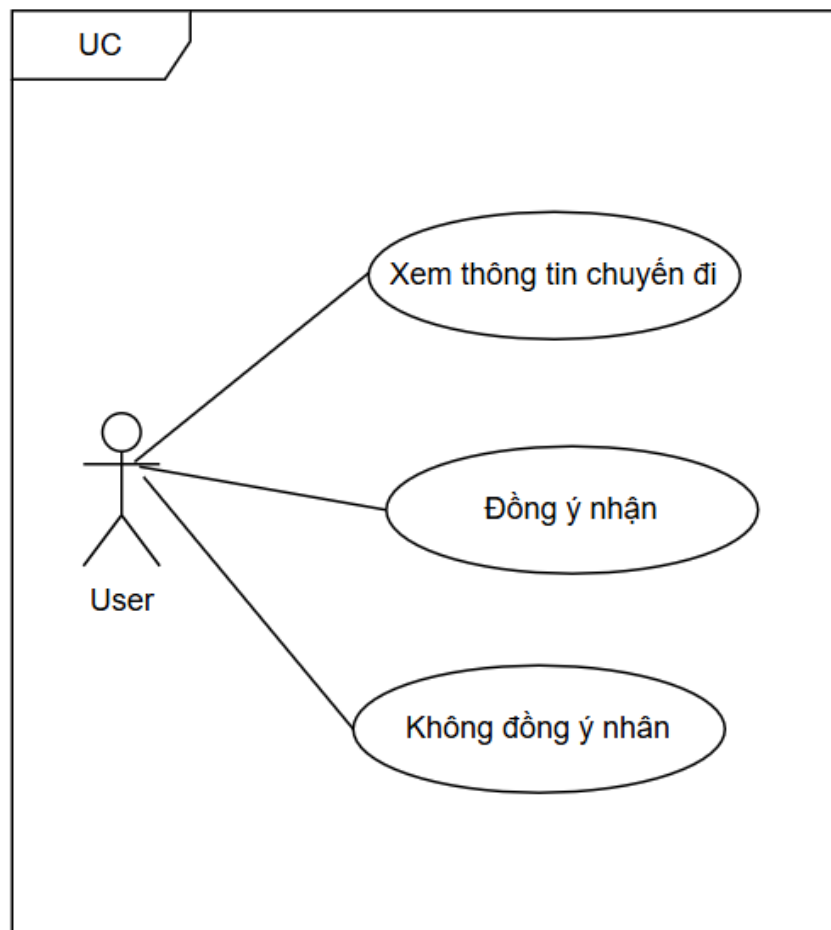
2.2.4 Biểu đồ use case phân rã Đặt tài xế



Hình 2.4: Biểu đồ usecase Đặt tài xế

Usecase Đặt tài xế bao gồm những chức năng chính sau (i) tìm kiếm địa điểm: người dùng nhập điểm đến, điểm đón, (ii) xem giá cước: người dùng xem giá cước của các chuyến đi, (iii) chọn phương tiện: người dùng chọn phương tiện để đặt chuyến.

2.2.5 Biểu đồ use case phân rã Nhận chuyến

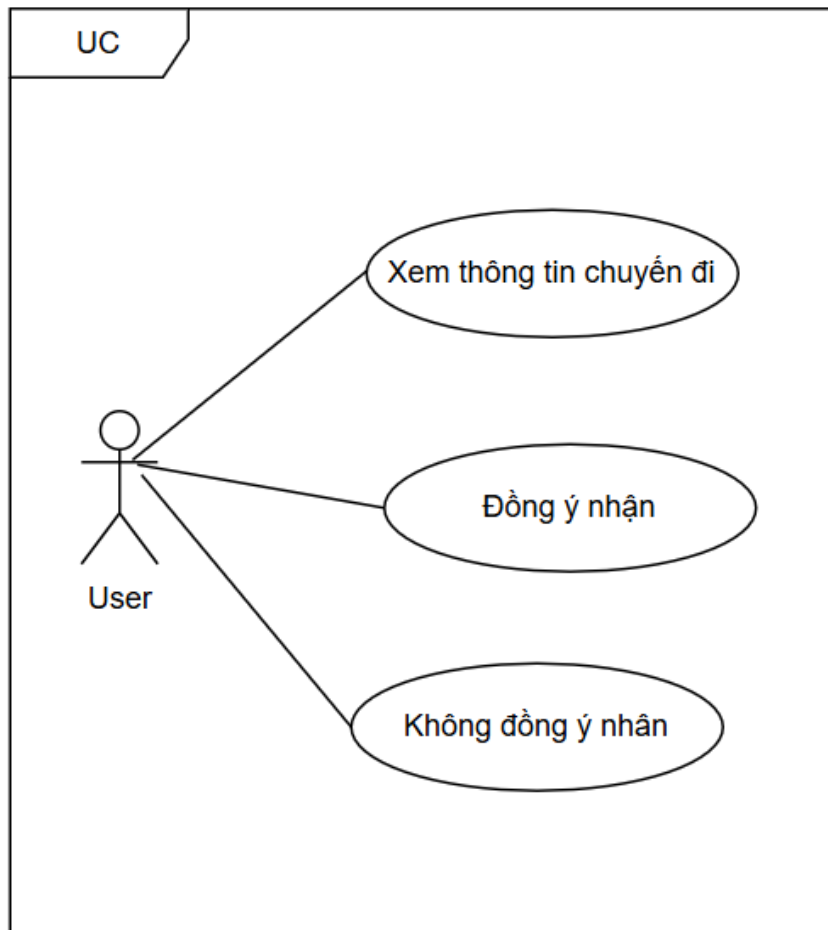


Hình 2.5: Biểu đồ usecase Nhận chuyến

Usecase Nhận chuyến bao gồm những chức năng chính sau (i) xem thông tin về chuyến đi: tài xế xem thông tin về chuyến đi như giá cước, địa điểm, (ii) đồng ý nhận: tài xế chấp nhận chuyến xe này, (iii) không đồng ý nhận: tài xế không chấp nhận chuyến đi và bỏ qua chuyến đi.

2.2.6 Quy trình nghiệp vụ

2.2.7 Quy trình nghiệp vụ Đặt tài xế



Hình 2.6: Quy trình nghiệp vụ Đặt tài xế

Hình 2.6

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

2.3.1 Đặc tả Đặt tài xế

| | |
|-------------------------------|--|
| Mã usecase | UC-1 |
| Tên usecase | Đặt tài xế |
| Mô tả | Usecase này mô tả quá trình đặt tài xế của người dùng |
| Tác nhân | Người dùng |
| Tiền điều kiện | Người dùng đăng nhập vào hệ thống |
| Luồng sự kiện chính | <ol style="list-style-type: none"> 1. Người dùng từ màn hình chính nhấn vào nút "Ô tô" của phần đặt 2. Người dùng tìm kiếm địa điểm đón và địa điểm trả 3. Người dùng nhấn nút "Tiếp tục" 4. Server tính toán quãng đường và giá tiền 5. Người dùng xem quãng đường đi chuyển và giá tiền 6. Người chọn phương tiện cần lái hộ 7. Người dùng nhấn nút "Đặt tài xế" 8. Server tìm kiếm tài xế phù hợp |
| Luồng sự kiện thay thế | 8a. Hệ thống thông báo cho người dùng biết không tìm được tài xế |
| Hậu điều kiện | Hệ thống chuyển sang màn hình theo dõi tài xế di chuyển |
| Luồng ngoại lệ | Không |

Bảng 2.1: Đặc tả usecase Đặt tài xế.

2.3.2 Đặc tả use case Nhận chuyển

| | |
|-------------------------------|--|
| Mã usecase | UC-2 |
| Tên usecase | Nhận chuyển |
| Mô tả | Usecase này mô tả quá trình nhận chuyển của tài xế |
| Tác nhân | Tài xế |
| Tiền điều kiện | - Người dùng đăng nhập vào hệ thống - Người dùng ở chế độ online |
| Luồng sự kiện chính | 1. Hệ thống gửi thông báo tới tài xế đang có người muốn đặt chuyển 2. Tài xế nhận thông báo về thông tin chuyển đi 3. Người dùng nhấn nút "Tiếp tục" |
| Luồng sự kiện thay thế | 3a. Người dùng ấn nút 'X' để không chấp nhận chuyển xe 3b. Hệ thống chờ 10 giây nếu người dùng không thực hiện thao tác thì tự động bỏ qua chuyển xe. |
| Hậu điều kiện | Hệ thống chuyển sang màn hình chuyển đi |
| Luồng ngoại lệ | Không |

Bảng 2.2: Đặc tả usecase Nhận chuyển.

2.3.3 Đặc tả use case Tìm kiếm tài xế

| | |
|-------------------------------|---|
| Mã usecase | UC-3 |
| Tên usecase | Tìm kiếm tài xế |
| Mô tả | Usecase này mô tả quá trình tìm kiếm tài xế phù hợp |
| Tác nhân | Người dùng, Hệ thống, Tài xế |
| Tiền điều kiện | - Người dùng đăng nhập vào hệ thống - Tài xế ở chế độ online |
| Luồng sự kiện chính | 1. Người dùng gửi thông tin về chuyến đi đến server 2. Server lọc ra những người phù hợp 3. Server sắp xếp danh sách những người phù hợp 4. Server gửi thông báo lần lượt tới danh sách tài xế |
| Luồng sự kiện thay thế | Không |
| Hậu điều kiện | Server gửi thông báo cho người dùng đã tìm được tài xế |
| Luồng ngoại lệ | Hệ thống gửi thông báo không tìm được tài xế |

Bảng 2.3: Đặc tả usecase Tìm kiếm tài xế.

2.3.4 Đặc tả use case **Chỉnh sửa thông tin**

| | |
|-------------------------------|---|
| Mã usecase | UC-4 |
| Tên usecase | Thay đổi thông tin cá nhân |
| Mô tả | Usecase này mô tả quá trình thay đổi thông tin cá nhân |
| Tác nhân | Người dùng |
| Tiền điều kiện | - Người dùng đăng nhập vào hệ thống |
| Luồng sự kiện chính | 1. Người dùng nhấn nút "Tài khoản" 2. Hệ thống hiển thị thông tin của người dùng 3. Người dùng nhập thông tin cần thay đổi 4. Người dùng nhấn nút "Cập nhật" |
| Luồng sự kiện thay thế | 4a. Hệ thống thông báo cập nhật thành |
| Hậu điều kiện | Hệ thống hiển thị thông tin mới được chỉnh sửa |
| Luồng ngoại lệ | Hệ thống gửi thông báo chỉnh sửa thông tin thất bại |

Bảng 2.4: Đặc tả usecase Thay đổi thông tin cá nhân.

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

3.1 Flutter

3.1.1 Giới thiệu

Flutter là một framework phát triển giao diện người dùng (UI) mã nguồn mở do Google phát triển. Nó cho phép xây dựng các ứng dụng đa nền tảng (cross-platform) từ một cơ sở mã duy nhất. Flutter sử dụng ngôn ngữ lập trình Dart và cung cấp một bộ widget phong phú để tạo giao diện người dùng đẹp mắt và linh hoạt.

Flutter được sử dụng để phát triển ứng dụng di động (Android, iOS), ứng dụng web, và thậm chí cả ứng dụng desktop (Windows, MacOS, Linux).

3.1.2 Mục đích sử dụng

Trong dự án này Flutter được sử dụng để xây dựng UI cho các màn hình của ứng dụng.

3.1.3 Ưu - Nhược điểm

1. Ưu điểm

- Phát triển đa nền tảng: Với mã nguồn duy nhất, Flutter cho phép phát triển ứng dụng cho nhiều nền tảng, giảm thời gian và công sức cho việc phát triển và bảo trì.
- Hiệu suất cao: Flutter biên dịch mã trực tiếp sang mã máy (native) thông qua Dart's Ahead-of-Time (AOT) compilation, đảm bảo hiệu suất gần như ứng dụng gốc (native).
- Giao diện đẹp và linh hoạt: Flutter cung cấp một bộ widget phong phú, được tùy chỉnh cao, giúp tạo ra giao diện người dùng đẹp mắt và đồng nhất trên các nền tảng.
- Cộng đồng phát triển mạnh mẽ: Flutter có một cộng đồng rộng lớn và tài liệu phong phú, giúp hỗ trợ giải quyết vấn đề nhanh chóng.

2. Nhược điểm

- Dung lượng ứng dụng lớn: Các ứng dụng Flutter thường có dung lượng lớn hơn so với ứng dụng native, đặc biệt khi so với ứng dụng iOS.
- Thiếu một số thư viện native: Một số tính năng hoặc thư viện native cần phải được phát triển thêm bằng cách viết mã native (Android: Kotlin/Java, iOS: Swift/Objective-C).

- Hỗ trợ native không đầy đủ: Mặc dù Flutter cung cấp rất nhiều tính năng, nhưng với các ứng dụng cần tích hợp sâu vào hệ điều hành (như sử dụng Bluetooth, cảm biến), sẽ phải viết mã native thủ công.

3.2 NestJS

3.2.1 Giới thiệu

NestJS là một framework mạnh mẽ để xây dựng ứng dụng phía server (backend) trong Node.js. Nó được xây dựng trên nền tảng của TypeScript và sử dụng các concept phổ biến từ Angular như Dependency Injection, Decorators, và Modularity. NestJS kết hợp các đặc điểm tốt nhất của các framework hiện có (như Express hoặc Fastify) và cung cấp một cấu trúc ứng dụng rõ ràng, dễ mở rộng.

3.2.2 Mục đích sử dụng

- NestJS được sử dụng để viết các API cho ứng dụng
- Kết hợp với Firebase, API của Google Map Platform để xây dựng các dịch vụ backend

3.2.3 Ưu - Nhược điểm

1. Ưu điểm

- Kiến trúc module rõ ràng, dễ mở rộng: NestJS cung cấp cấu trúc module có tổ chức, giúp phát triển và bảo trì code dễ dàng hơn.
- Hỗ trợ TypeScript: NestJS được xây dựng trên TypeScript, giúp phát triển ứng dụng một cách an toàn và hiệu quả hơn.
- Dependency Injection: NestJS sử dụng Dependency Injection để quản lý các dependencies giữa các module và services, giúp code trở nên linh hoạt và dễ dàng trong việc kiểm thử phần mềm.
- Cộng đồng và tài liệu phong phú: NestJS có một cộng đồng đông đảo và tài liệu chính thức chi tiết giúp lập trình viên học tập và triển khai 1 cách dễ dàng.

2. Nhược điểm

- Khó tiếp cận với người mới bắt đầu: Cấu trúc module, dependency injection và các khái niệm khác của NestJS có thể gây khó khăn cho người mới bắt đầu.
- Kích thước bundle: Do sử dụng TypeScript và nhiều tính năng, kích thước bundle của ứng dụng NestJS có thể lớn hơn so với các ứng dụng sử dụng các framework nhẹ hơn.

3.3 Firebase

3.3.1 Giới thiệu

Firebase là một nền tảng phát triển ứng dụng di động và web do Google cung cấp. Nó cung cấp nhiều dịch vụ đám mây (Backend-as-a-Service) giúp các nhà phát triển tập trung vào việc xây dựng ứng dụng mà không cần quản lý cơ sở hạ tầng phía sau. Firebase cung cấp nhiều tính năng như xác thực người dùng, cơ sở dữ liệu thời gian thực, lưu trữ đám mây, phân tích và nhiều dịch vụ khác.

3.3.2 Mục đích sử dụng

Trong dự án này, Firebase được sử dụng để:

- Realtime Database: Lưu trữ và quản lý dữ liệu thời gian thực
- Cloud Storage: Lưu trữ và quản lý files và hình ảnh
- Cloud Messaging: Gửi thông báo đẩy (push notifications) đến người dùng

3.3.3 Ưu - Nhược điểm

1. Ưu điểm

- Phát triển nhanh chóng: Firebase cung cấp nhiều dịch vụ được xây dựng sẵn, giúp giảm thời gian phát triển ứng dụng.
- Khả năng mở rộng: Firebase được xây dựng trên cơ sở hạ tầng của Google, cho phép ứng dụng dễ dàng mở rộng quy mô.
- Bảo mật cao: Được hỗ trợ bởi Google với các tính năng bảo mật mạnh mẽ
- Hỗ trợ đa nền tảng: Firebase hỗ trợ phát triển ứng dụng cho nhiều nền tảng như iOS, Android và web.

2. Nhược điểm

- Giá thành cao: Firebase có giá thành cao, đặc biệt là với các dịch vụ nâng cao.
- Phụ thuộc vào Google: Hoàn toàn phụ thuộc vào hạ tầng của Google
- Tính linh hoạt hạn chế: Không thể tùy chỉnh sâu như các giải pháp backend tự xây dựng

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

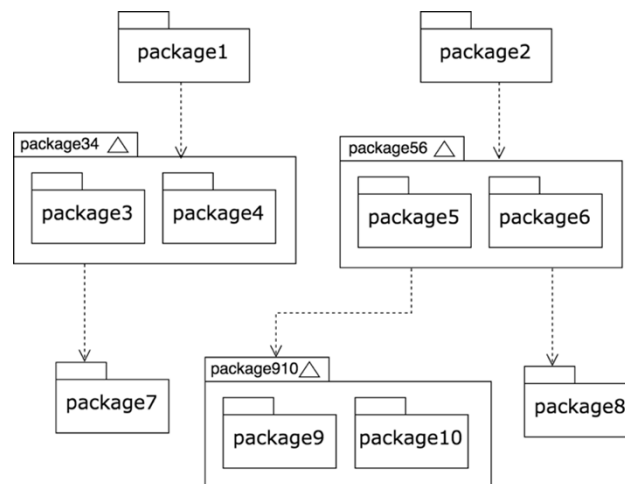
4.1 Thiết kế kiến trúc

4.1.1 Lựa chọn kiến trúc phần mềm

Mục này có độ dài từ một đến ba trang. Sinh viên cần lựa chọn kiến trúc phần mềm cho ứng dụng của mình như: kiến trúc ba lớp MVC, MVP, SOA, Microservice, v.v. rồi giải thích sơ bộ về kiến trúc đó (không giải thích chi tiết/dài dòng). Sử dụng kiến trúc phần mềm đã chọn ở trên, sinh viên mô tả kiến trúc cụ thể cho ứng dụng của mình. Gợi ý: sinh viên áp dụng lý thuyết chung vào hệ thống/sản phẩm của mình như thế nào, có thay đổi, bổ sung hoặc cải tiến gì không. Ví dụ, thành phần M trong kiến trúc lý thuyết MVC sẽ là những thành phần cụ thể nào (ví dụ: là interface I + class C1 + class C2, v.v.) trong kiến trúc phần mềm của sinh viên.

4.1.2 Thiết kế tổng quan

Sinh viên vẽ biểu đồ gói UML (UML package diagram), nêu rõ sự phụ thuộc giữa các gói (package). SV cần vẽ các gói sao cho chúng được phân theo các tầng rõ ràng, không được sắp đặt package lộn xộn trong hình vẽ. Sinh viên chú ý các quy tắc thiết kế (Các gói không phụ thuộc lẫn nhau, gói tầng dưới không phụ thuộc gói tầng trên, không phụ thuộc bỏ qua tầng, v.v.) và cần giải thích sơ lược về mục đích/nhiệm vụ của từng package. SV tham khảo ví dụ minh họa trong Hình 4.1



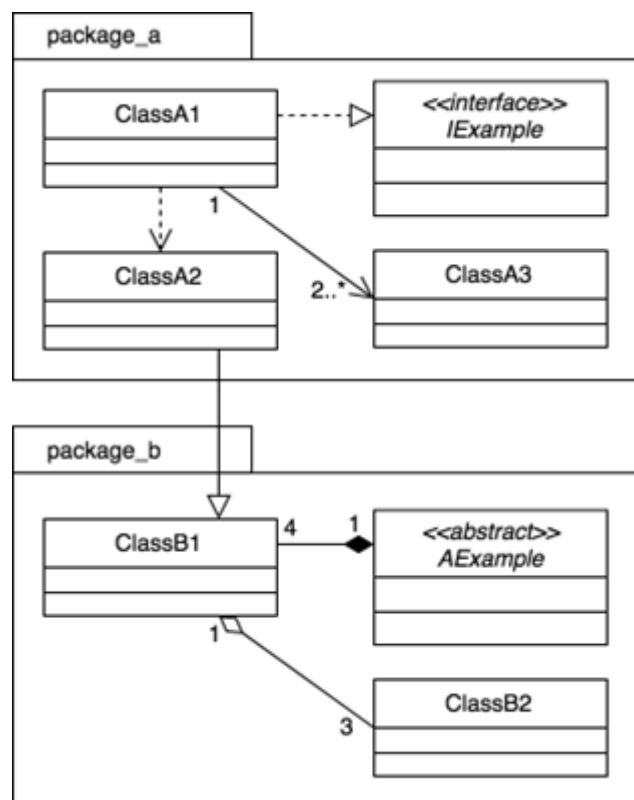
Hình 4.1: Ví dụ biểu đồ phụ thuộc gói

4.1.3 Thiết kế chi tiết gói

Sinh viên thiết kế và lần lượt vẽ biểu đồ thiết kế cho từng package, hoặc một nhóm các package liên quan để giải quyết một vấn đề gì đó. Khi vẽ thiết kế gói, sinh viên chỉ cần đưa tên lớp, không cần chỉ ra các thành viên phương thức và thuộc tính. SV tham khảo ví dụ minh họa trong Hình 4.2.

Sinh viên cần vẽ rõ ràng quan hệ giữa các lớp trong biểu đồ. Các quan hệ bao gồm: phụ thuộc (dependency), kết hợp (association), kết tập (aggregation), hợp thành (composition), kế thừa (inheritance), và thực thi (implementation). Các quan hệ này đều đã được minh họa trong 4.2.

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



Hình 4.2: Ví dụ thiết kế gói

4.2 Thiết kế chi tiết

4.2.1 Thiết kế giao diện

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

4.2.2 Thiết kế lớp

Phần này có độ dài từ ba đến bốn trang. Sinh viên trình bày thiết kế chi tiết các thuộc tính và phương thức cho một số lớp chủ đạo/quan trọng nhất của ứng dụng

(từ 2-4 lớp). Thiết kế chi tiết cho các lớp khác, nếu muốn trình bày, sinh viên đưa vào phần phụ lục.

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

4.2.3 Thiết kế cơ sở dữ liệu

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

4.3 Xây dựng ứng dụng

4.3.1 Thư viện và công cụ sử dụng

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

| Mục đích | Công cụ | Địa chỉ URL |
|---------------|------------------------|---|
| IDE lập trình | Eclipse Oxygen a64 bit | http://www.eclipse.org/ |
| 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.¹ 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”.

¹Trong 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 >

hovy1993automated 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>

peterson2007computer L. L. Peterson and B. S. Davie, *Computer networks: a systems approach*. Elsevier, 2007.

NguyenThucHai 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>

poesio2001discourse 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>

knott1996data 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>

BernersTim 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).

LectureA 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/chơi một cách hợp lý, (iii) chưa tìm được người yêu để chia sẻ quãng đời sinh viên vất vả, và (iv) viết ĐATN một cách cẩu thả.

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

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

trong từng trang báo cáo ĐATN.

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}
```

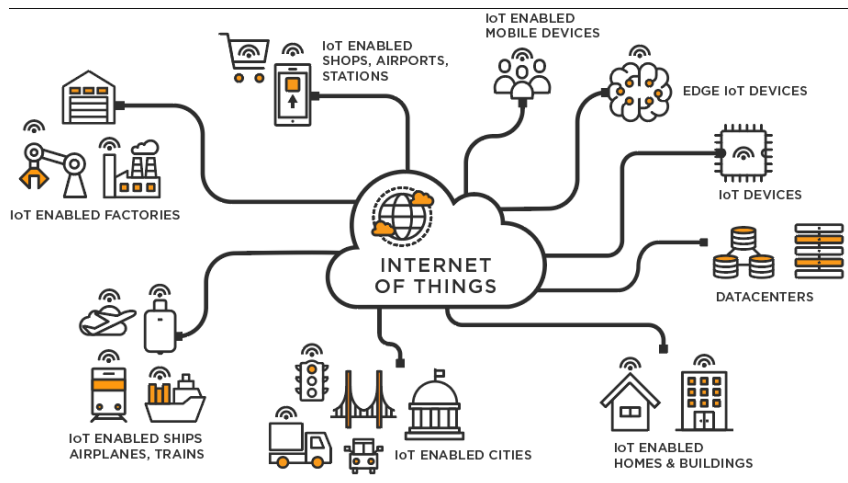
Chú ý các nội dung trình bày trong cả hai môi trường liệt kê theo sau lệnh `\item`. Ngoài ra LaTeX còn cung cấp một số kiểu liệt kê khác, sinh viên có thể tham khảo tại <https://www.overleaf.com/learn/latex/Lists>

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.



Hình A.2: Quy cách đóng quyển đồ án

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)

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

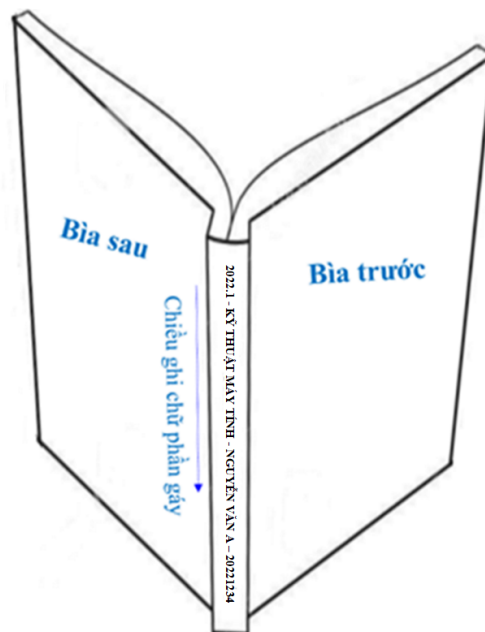
Phương trình này thể hiện phép biến đổi Fourier rời rạc ngược (IDFT).

A.7 Quy cách đóng quyển

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

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

Quy cách ghi chữ phần gáy như hình dưới đây:



Hình A.3: Qui 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”

...