**TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI**

**ĐỒ ÁN TỐT NGHIỆP**

**Xây dựng ứng dụng Bkbank Mobile Banking**

**ĐÀO SỸ MẠNH**

Manhds.173253@sis.hust.edu.vn

**Ngành Công nghệ Thông tin và Truyền thông**

**Chuyên ngành Công nghệ phần mềm**

|  |  |
| --- | --- |
| **Giảng viên hướng dẫn:** | ThS. Nguyễn Tiến Thành |
| **Bộ môn:** | Công nghệ phần mềm |
| **Viện:** | Công nghệ thông tin – Truyền thông |
| **HÀ NỘI, 07/2022** | |

# Lời cam kết

Họ và tên sinh viên: Đào Sỹ Mạnh

Điện thoại liên lạc: 0967780420 Email: manh.ds173253@sis.hust.edu.vn

Lớp: KHMT.01 Hệ đào tạo: Đại học đại trà

Tôi – *Đào Sỹ Mạnh* – cam kết Đồ án Tốt nghiệp (ĐATN) là công trình nghiên cứu của bản thân tôi dưới sự hướng dẫn của *ThS. Nguyễn Tiến Thành*. Các kết quả nêu trong ĐATN là trung thực, là thành quả của riêng tôi, không sao chép theo bất kỳ công trình nào khác. Tất cả những tham khảo trong ĐATN – bao gồm hình ảnh, bảng biểu, số liệu, và các câu từ trích dẫn – đều được ghi rõ ràng và đầy đủ nguồn gốc trong danh mục tài liệu tham khảo. Tôi xin hoàn toàn chịu trách nhiệm với dù chỉ một sao chép vi phạm quy chế của nhà trường.

|  |  |
| --- | --- |
|  | *Hà Nội, ngày tháng năm*  Tác giả ĐATN  Mạnh  *Đào Sỹ Mạnh* |

# Lời cảm ơn

Lời đầu tiên em xin gửi lời cảm ơn chân thành đến các thầy cô trong trường Đại học Bách Khoa Hà Nội cũng như các thầy cô trong viện Công nghệ thông tin và Truyền thông đã mang đến cho em nhiều kiến thức và kinh nghiệm trong 5 năm qua, cung cấp hành trang quý báu và hữu ích cho tương lai sắp tới.

Em cũng xin gửi lời cảm ơn sâu sắc đến ThS. Nguyễn Tiến Thành. Thầy là người đã tận tình hướng dẫn, giúp đỡ để em có thể hoàn thành ĐATN này một cách tốt nhất.

Cuối cùng, em xin cảm ơn gia đình, bạn bè, đồng nghiệp - những người luôn ở bên cạnh, cổ vũ và động viên tinh thần, giúp em vượt qua những khó khăn trong suốt quá trình học tập tại đại học Bách Khoa Hà Nội.

Do thời gian làm đồ án có hạn, cũng như kiến thức chuyên môn còn non kém, ĐATN của em còn nhiều thiết sót và hạn chế. Vậy nên em rất mong nhận được những ý kiến góp ý từ thầy cô và các bạn để có thể thấy được ưu nhược điểm bản thân, từ đó hoàn thiện báo cáo cũng như xây dựng kinh nghiệm vững vàng hơn.

Em xin chân thành cảm ơn!

# Tóm tắt

Hiện nay cùng với sự phát triển của xã hội, kinh tế đi kèm với đời sống mỗi người nâng cao lên, nhu cầu giao dịch, trao đổi giá trị, cụ thể là tiền mặt giữa các cá nhân và tổ chức tăng lên một cách đáng kể. Việc sử dụng trực tiếp bằng tiền mặt dần phát sinh nhiều hạn chế như việc phải trao đổi trực tiếp, khó quản lý, lưu giữ. Từ đó, sự xuất hiện của các ngân hàng giúp việc quản lý tiền bạc, giao dịch của mọi người trở nên thuận tiện, khoa học hơn. Tuy nhiên trở ngại là việc phải trực tiếp sử dụng dịch vụ tại các điểm, quầy giao dịch. Một hướng giải pháp hiệu quả cho vấn đề ở trên là ứng dụng internet, công nghệ thông tin vào trong việc quản lý tiền bạ, giao dịch của người dùng. Thay vì phải giao dịch trực tiếp hoặc sử dụng tiền mặt thì mọi người có thể thực hiện điều đó một cách trực truyến qua internet.Với những lý do trên thì một ứng dụng điện thoại có khả năng thực hiện các giao dịch trực tuyến là cần thiết. ĐATN này hướng tới xây dựng một ứng dụng di động có thể thực hiện các trao đổi, giao dịch, dịch vụ trực tuyến nhanh chóng, hiệu quả tiết kiệm thời gian, công sức của người dùng.

# Abstract

Mục này khuyến khích sinh viên viết lại mục “Tóm tắt” đồ án tốt nghiệp ở trang trước bằng tiếng Anh. Phần này phải có đầy đủ các nội dung như trong phần tóm tắt bằng tiếng Việt.

Sinh viên không nhất thiết phải trình bày mục này. Nhưng nếu lựa chọn trình bày, sinh viên cần đảm bảo câu từ và ngữ pháp chuẩn tắc, nếu không sẽ có tác dụng ngược, gây phản cảm.

# Mục lục

[Lời cam kết ii](#_Toc44922331)

[Lời cảm ơn iii](#_Toc44922332)

[Tóm tắt iv](#_Toc44922333)

[Abstract v](#_Toc44922334)

[Mục lục vi](#_Toc44922335)

[Danh mục hình vẽ ix](#_Toc44922336)

[Danh mục bảng x](#_Toc44922337)

[Danh mục công thức xi](#_Toc44922338)

[Danh mục các từ viết tắt xii](#_Toc44922339)

[Danh mục thuật ngữ xiii](#_Toc44922340)

[Chương 1 Giới thiệu đề tài 1](#_Toc44922341)

[1.1 Đặt vấn đề 1](#_Toc44922342)

[1.2 Mục tiêu và phạm vi đề tài 2](#_Toc44922343)

[1.3 Định hướng giải pháp 2](#_Toc44922344)

[1.4 Bố cục đồ án 2](#_Toc44922345)

[Chương 2 Khảo sát và phân tích yêu cầu 4](#_Toc44922346)

[2.1 Khảo sát hiện trạng 4](#_Toc44922347)

[2.2 Tổng quan chức năng 4](#_Toc44922348)

[2.2.1 Biểu đồ use case tổng quan 4](#_Toc44922349)

[2.2.2 Biểu đồ use case phân rã XYZ 5](#_Toc44922350)

[2.2.3 Quy trình nghiệp vụ 5](#_Toc44922351)

[2.3 Đặc tả chức năng 5](#_Toc44922352)

[2.3.1 Đặc tả use case A 5](#_Toc44922353)

[2.3.2 Đặc tả use case B 5](#_Toc44922354)

[2.4 Yêu cầu phi chức năng 5](#_Toc44922355)

[Chương 3 Công nghệ sử dụng 6](#_Toc44922356)

[Chương 4 Phát triển và triển khai ứng dụng 7](#_Toc44922357)

[4.1 Thiết kế kiến trúc 7](#_Toc44922358)

[4.1.1 Lựa chọn kiến trúc phần mềm 7](#_Toc44922359)

[4.1.2 Thiết kế tổng quan 7](#_Toc44922360)

[4.1.3 Thiết kế chi tiết gói 8](#_Toc44922361)

[4.2 Thiết kế chi tiết 8](#_Toc44922362)

[4.2.1 Thiết kế giao diện 8](#_Toc44922363)

[4.2.2 Thiết kế lớp 9](#_Toc44922364)

[4.2.3 Thiết kế cơ sở dữ liệu 9](#_Toc44922365)

[4.3 Xây dựng ứng dụng 9](#_Toc44922366)

[4.3.1 Thư viện và công cụ sử dụng 9](#_Toc44922367)

[4.3.2 Kết quả đạt được 9](#_Toc44922368)

[4.3.3 Minh hoạ các chức năng chính 10](#_Toc44922369)

[4.4 Kiểm thử 10](#_Toc44922370)

[4.5 Triển khai 10](#_Toc44922371)

[Chương 5 Các giải pháp và đóng góp nổi bật 11](#_Toc44922372)

[Chương 6 Kết luận và hướng phát triển 12](#_Toc44922373)

[6.1 Kết luận 12](#_Toc44922374)

[6.2 Hướng phát triển 12](#_Toc44922375)

[Tài liệu tham khảo 13](#_Toc44922376)

[Phụ lục A-1](#_Toc44922377)

[A Hướng dẫn viết đồ án tốt nghiệp A-1](#_Toc44922378)

[A.1 Quy định chung A-1](#_Toc44922379)

[A.2 Ngành học A-2](#_Toc44922380)

[A.3 Tạo đề mục A-2](#_Toc44922381)

[A.4 Bảng biểu A-3](#_Toc44922382)

[A.5 Hình vẽ A-4](#_Toc44922383)

[A.6 Tài liệu tham khảo A-4](#_Toc44922384)

[A.7 Công thức toán học A-5](#_Toc44922385)

[A.8 Tham chiếu chéo A-5](#_Toc44922386)

[A.9 Cập nhật mục lục và tham chiếu chéo A-6](#_Toc44922387)

[A.10 In quyển đồ án tốt nghiệp A-6](#_Toc44922388)

[B Đặc tả use case B-8](#_Toc44922389)

[B.1 Đặc tả use case “Thống kê tình hình mượn sách” B-8](#_Toc44922390)

[B.2 Đặc tả use case “Đăng ký làm thẻ mượn” B-8](#_Toc44922391)

[C Công nghệ sử dụng C-8](#_Toc44922392)

[C.1 Công nghệ bảo mật dữ liệu C-8](#_Toc44922393)

[C.2 Công nghệ blockchain C-8](#_Toc44922394)

[D Thiết kế gói D-8](#_Toc44922395)

[D.1 Thiết kế gói cho kiến trúc tổng quan D-8](#_Toc44922396)

[D.2 Thiết kế gói cho chức năng “Trả sách” D-9](#_Toc44922397)

[E Thiết kế lớp E-9](#_Toc44922398)

# Danh mục hình vẽ

[**Hình 1** Ví dụ biểu đồ phụ thuộc gói 7](#_Toc27562454)

[**Hình 2** Ví dụ thiết kế gói 8](#_Toc27562455)

[**Hình 3** Ví dụ hình vẽ A-4](#_Toc27562456)

[**Hình 4** Quy cách đóng quyển A-6](#_Toc27562457)

[**Hình 5** Quy cách ghi chữ phần gáy A-6](#_Toc27562458)

[**Hình 6** Hướng dẫn thiết lập in hai mặt A-7](#_Toc27562459)

# Danh mục bảng

[**Bảng 1** Danh sách thư viện và công cụ sử dụng 9](#_Toc27562460)

[**Bảng 2** Ví dụ sử dụng bảng A-2](#_Toc27562461)

# Danh mục công thức

[**Công thức 1** Khai triển Newton A-5](#_Toc27562462)

Lưu ý: Nếu ĐATN có ít hơn ba công thức toán học, sinh viên có thể xóa bỏ mục này.

# Danh mục các từ viết tắt

|  |  |
| --- | --- |
| API | Application Programming Interface  Giao diện lập trình ứng dụng |
| **EUD** | End-User Development  Phát triển ứng dụng người dùng cuối |
| **GWT** | Google Web Toolkit  Công cụ lập trình Javascript bằng Java của Google |
| **HTML** | HyperText Markup Language  Ngôn ngữ đánh dấu siêu văn bản |
| **CNTT** | Công nghệ thông tin |
| **ĐATN** | Đồ án tốt nghiệp |
| **SV** | Sinh viên |

# Danh mục thuật ngữ

|  |  |
| --- | --- |
| Browser | Trình duyệt |
| **Cache memory** | Bộ nhớ đệm |
| **E-commerce** | Thương mại điện tử |
| **Bloatware** | Ứng dụng nhà sản xuất tích hợp vào thiết bị |
| **Interpreter** | Trình thông dịch |
| **Compiler** | Trình biên dịch |

# Giới thiệu đề tài

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 word này, chỉ chỉnh sửa nội dung, và không viết trên file word 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 bổ 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

## Đặt vấn đề

Ngày nay với sự phát triển không ngừng của kinh tế thì việc giao dịch, trao đổi tăng lên đáng kể. Và song song với đó con người cần quản lý tiền bạc của mình một cách chặt chẽ và khoa học hơn. Việc sử dụng trực tiếp bằng tiền mặt dần phát sinh nhiều hạn chế như việc phải trao đổi trực tiếp, khó quản lý, lưu giữ, nhiều thủ tục rườm rà không cần thiết gây mất thời gian của người thực hiện giao dịch

Hiện tại các ngân hàng đã giúp được mọi người trong việc quản lý, lưu giữ tiền bạc. Bên cạnh đó, các giao dịch được quản lý khoa học hơn, rõ ràng hơn, minh bạch hơn, tuy nhiên vẫn còn những rào cản về các thủ tục rườm rà cũng như việc phải có mặt trực tiếp để sử dụng các dịch vụ.

Nếu xây dựng một ứng dụng được quản lý bởi các ngân hàng thì mọi việc trở nên tiện lợi và nhanh chóng hơn. Một ứng dụng có thể thực hiện giao dịch, quản lý tiền bạc một các trực tuyến, mọi người chỉ cần thực hiện vài thao tác đơn giản để đăng ký sử dụng ứng dụng, sau đó mọi giao dịch cũng như quản lý đều được thực hiện một cách trực tuyến, tiết kiệm được rất nhiều thời gian công sức. Ngoài ra ứng dụng có thể đáp ứng được các nhu cầu cá nhân hằng ngày như nạp mua mã thẻ điện thoại, thanh toán các hoá đơn online.

Tại thời điểm hiện tại đã có rất nhiều ngân hàng triển khai ứng dụng như vậy, tuy nhiên bước xác minh thông tin của nhiều ứng dụng này chưa được thực sự tốt và công việc xử lý nằm toàn bộ ở phía server. Vì vậy ngoài module xử lý ở server thì em có phát triển thêm module cho phép nhận diện một số lỗi trong quy trình xác thực ở ngay trên client là app moblie. Điều này cải thiện trải nghiệm người dùng rất nhiều bằng việc có thể nhanh chóng thông báo một số lỗi phổ biến của việc cung cấp ảnh xác thực thay vì phải dùng đường truyền mạng gửi tới server và đợi kết quả. Ngoài ra việc xử lý trên client còn giúp server giảm tải được công việc

## Mục tiêu và phạm vi đề tài

Với những vấn đề đã trình bày trong mục 1.1, trong ĐATN này tôi sẽ phát triển một hệ thống banking online có các chức năng quản lý thông tin và thực hiện các giao dịch trực tuyến. Các nhóm chức năng chính dự định sẽ được triển khai là

* Thứ nhất, hệ thống có chức năng xác nhận thông tin người dùng. Khi đăng ký thành công, mặc định người dùng sẽ ở trạng thái chưa xác nhận, khi đó hệ thống chuyển hướng về màn hình xác nhận thông tin. Tại màn hình này, người dùng cung cấp tên, số thẻ căn cước, ảnh chụp mặt trước thẻ căn cước. Hệ thống ứng dụng mobile sẽ kiểm tra những thông tin này trước khi chuyển đến server.
* Thứ hai, người dùng có thể cài đặt thông tin cá nhân như cài đặt mật mã, cài đặt thẻ vật lý, quản lý danh bạ.
* Thứ ba, hệ thống có các chức năng về giao dịch. Người dùng có thể chuyển khoản nội bộ, chuyển khoản online để chuyển tiền qua lại. Sử dụng chức năng mua mã thẻ hoặc nạp thẻ điện thoại trực tuyến. Ngoài ra có thể gửi tiền tiết kiệm trực tuyến và tất toán trực tuyến khi cần. Cuối cùng là có thể tra cứu lại lịch sử các giao dịch trong một khoảng thời gian nhất định.
* Thức tư, người quản trị có thể xem thống kê, quản lý các thành phần của hệ thống như quản lý người dùng, quản lý tin tức, quản lý lãi xuất, quản lý hỏi đáp.

Để có thể phát triển một hệ thống đáp ứng các chức năng trên, hệ thống cần được thiết kế với một kiến trúc đáp ứng được nhiều người sử dụng cùng lúc, hiệu năng ổn định và dễ dàng mở rộng. Hơn nữa, giao diện sử dụng dành cho người dùng cũng phải được thiết kế tinh gọn, tối giản mà lại dễ sử dụng, không gây nhiều khó khăn với những người mới tiếp cận, tăng trải nghiệm cho người dùng. Ngoài ra, hệ thống cũng cần đáp ứng được các yêu cầu về kỹ thuật, chẳng hạn như tìm kiếm và sửa lỗi, bảo trì và thêm mới chức năng.

## Định hướng giải pháp

Để xây dựng hệ thống banking online theo mục tiêu đã trình bày ở trên thì em định hướng xây dựng nền tảng ứng dụng mobile cho người dùng và nền tảng website cho người quản trị

Mã nguồn của ứng dụng sẽ chia làm bốn phần:

* Backend: xử lý các nghiệp vụ logic, kết nối với dữ liệu để thực hiện các tính năng
* Service xác thực thông tin: để xác thực thông tin của người dùng dựa trên thẻ căn cước và chân dung
* Giao diện mobile: Hiển thị dữ liệu, tương tác với người dung
* Giao diện quản lý cho người quản trị: Giúp người quản trị có thể kiểm soát, quản lý các thầnh phần của ứng dụng

Phần backend sẽ được xây dựng trên platform NodeJs, sử dụng ngôn ngữ lập trình JavaScript. Với ưu điểm nhanh hơn và nhẹ hơn của JavaScipt so với các ngôn ngữ lập trình khác, ứng dụng xây dựng sẽ đảm bảo một hiệu năng tốt nhất cho người dùng. Ngoài ra, đồ án sử dụng Socket.IO để đưa thông báo đến người dùng một cách tức thì.

Service xác thực thông tin được xây dựng dựa trên mô hình Yolov5LiteS để phát hiện vị trí gương mặt trong các ảnh, sau đó những gương mặt này được so khớp với nhau bằng mô hình MobileFaceNet.

Phần giao diện mobile sử dụng mobile UI framework Flutter với ngôn ngữ lập trình Dart. Với Flutter thì ứng dụng có thể sử dụng trên cả iOS cũng như Android

Giao diện quản lý cho người quản trị được xây dựng bằng ReactJs, một thư viện javascript để hiển thị giao diện cho người quản trị tương tác. Với việc được thiết kế theo mô hình single page application, giao diện quản trị sẽ đem lại cho người dùng cảm giác mượt mà.

Đối với ĐATN này, dữ liệu được lưu trữ bằng cơ sở dữ liệu MySQL. PlatForm NodeJs hỗ trợ thư viện Sequelize giúp tối ưu hóa câu truy vấn, giảm thiểu thời gian khi truy cập dữ liệu và dễ dàng mở rộng tổng quan chức năng.

Mô hình em sử dụng cho đồ án là mô hình Client-Server, một mô hình đang được sử dụng rộng rãi trong các dự án. Với mô hình này thì phía máy khách sẽ yêu cầu các thông tin cũng như giao tiếp với máy chủ bằng việc sử dụng các API.

.

## Bố cục đồ án

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

Với những yêu cầu đã đặt ra ở chương 1, chương 2 sẽ trình bày tổng quan về các chức năng trong hệ thống đồng thời đặc tả một số usecase chính.

Trong chương 3, tôi sẽ trình bày các công nghệ được sử dụng trong quá trình xây dựng và hoàn thành đồ án. Với mỗi công nghệ sẽ có cơ sở lý thuyết, công dụng cũng như sự phù hợp để lựa chọn trong ĐATN này.

Khi đã xác định được các công nghệ sử dụng ở chương 3 thì chương 4 sẽ tập trung trình bày chi tiết về phân tích thiết kế hệ thống và trình bày sâu hơn về luồng hoạt động của một số nghiệp vụ chính trong hệ thống, thiết kế giao diện, thiết kế lớp, thiết kế cơ sở dữ liệu, sau đó là triển khai và cuối cùng là kiểm thử.

Chương 5 trình bày về các giải pháp và đóng góp nổi bật của cá nhân với ĐATN này, đồng thời nêu ra các khó khăn và cách giải quyết những khó khẳn đó trong quá trình hoàn thành ĐATN.

Cuối cùng, chương 6 là phần kết luận, chương này sẽ tổng kết lại quá trình xây dựng ứng dụng, những ưu, nhược điểm đã đạt được cũng như định hướng phát triển mở rộng trong tương lai.

# ­­Khảo sát và phân tích yêu cầu

Chương này có độ dài từ 9 đến 11 trang. Với đồ án nghiên cứu, sinh viên đổi tên chương thành “Tình hình nghiên cứu hiện nay” (Related works – State of the art) và phân tích rõ ngữ cảnh bài toán cũng như các kết quả nghiên cứu tương tự. Nếu là đồ án nghiên cứu, các đề mục cần thay đổi cho phù hợp. Sinh viên cần trao đổi kỹ với GV hướng dẫn để đưa ra được đề mục phù hợp nhất.

Với phương pháp phân tích thiết kế hướng đối tượng, sinh viên sử dụng biểu đồ use case theo hướng dẫn của template này. Với các phương pháp khác, sinh viên trao đổi với giáo viên hướng dẫn để đổi tên và sắp xếp lại đề mục cho phù hợp. Ví dụ, thay vì sử dụng biểu đồ use case, sinh viên đi theo hướng tiếp cận Agile có thể dùng User Story.

## Khảo sát hiện trạng

***Thông thường, khảo sát chi tiết về hiện trạng và yêu cầu của phần mềm sẽ được lấy từ ba nguồn chính, đó là (i) người dùng/khách hàng, (ii) các hệ thống đã có, (iii) và các ứng dụng tương tự.***

***Sinh viên cần tiến hành phân tích, so sánh, đánh giá chi tiết ưu nhược điểm của các sản phẩm/nghiên cứu hiện có. Sinh viên có thể lập bảng so sánh nếu cần thiết. Kết hợp với khảo sát người dùng/khách hàng (nếu có), sinh viên nêu và mô tả sơ lược các tính năng phần mềm quan trọng cần phát triển.***

Trên thị trường hiện nay xuất hiện rất nhiều ứng dụng cho mục đích giao dịch, quản lý tài sản trực tuyến, nổi bật và phổ biến nhất là ứng dụng mobile banking của các ngân hàng hoặc các ví điện tử. Trong các ứng dụng mobile banking có Agribank E-Mobile banking, một ứng dụng tương đối phổ biến và cũng là của ngân hàng thương mại được nhà nước nắm giữ 100% vốn điều lệ. Về thị trường bên ví điện tử thì phổ biến và được nhiều người sử dụng đang là Momo

|  |  |  |  |
| --- | --- | --- | --- |
| Tiêu chí | Agribank | Viettinbank | Momo |
| Ngôn ngữ |  |  |  |
| Phí duy trì | 10.000/ tháng |  | Miễn phí |
| Phí nhận sms biến động số dư | 8.000/tháng |  | Miễn phí |
| Sử dụng xác thực eKYC | có | không | có |
| Phát hiện lỗi thông tin xác thực trước khi gửi server | Không | Không | Không |
| Nạp tiền điện thoại | Có | Có | Có |
| Mua mã thẻ | Có | Không | Có |
| Phí chuyển khoản nội bộ | Miễn phí | Miễn phí | Miễn phí |
| Phí chuyển khoản liên ngân hàng | Miễn phí | Miễn phí | Có phí |
| Gửi tiết kiệm | Có | Có | Không |
| Thanh toán hoá đơn | Có | Có | Có |
| Quản lý thẻ vật lý | Có | Có | Không |

Từ những so sánh trên, tôi sẽ phát triển hệ thống banking online có các chức năng cơ bản như quản lý thông tin cá nhân, chuyển khoản nội bộ, liên ngân hàng, gửi tiết kiệm, nạp tiền hoặc mua mã thẻ điện thoại và có thể xác thực thông tin trực tuyến với giao diện dễ dàng và phản hồi nhanh chóng. Phía người quản trị có thể quản lý các thành phẩn của hệ thống như quản lý người dùng, tin tức, lãi xuất, thông tin ứng dụng, hỏi đáp, địa chỉ giao dịch

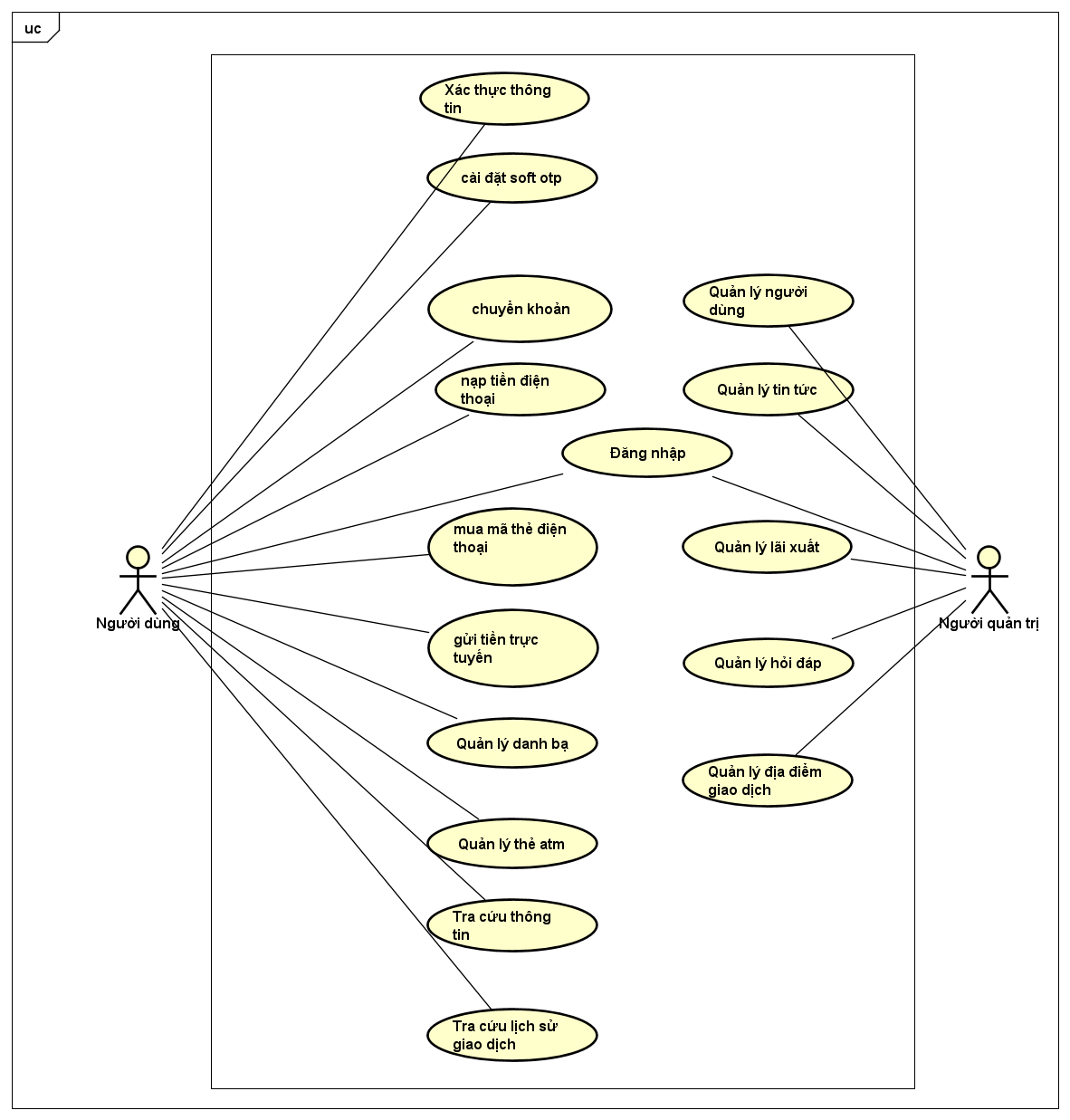
## Tổng quan chức năng

Phần 2.2 này có nhiệm vụ tóm tắt các chức năng của phần mềm. Trong phần này, sinh viên lưu ý chỉ mô tả chức năng mức cao (tổng quan) mà không đặc tả chi tiết cho từng chức năng. Đặc tả chi tiết được trình bày trong phần 2.3.

### Biểu đồ use case tổng quan

***Sinh viên vẽ biểu đồ use case tổng quan và giải thích các tác nhân tham gia là gì, nêu vai trò của từng tác nhân, và mô tả ngắn gọn các use case chính.***

Hệ thống mobile banking gồm nhiều các thành phần chức năng khác nhau. Hình 1 mô tả toàn bộ chức năng tổng quan của hệ thống

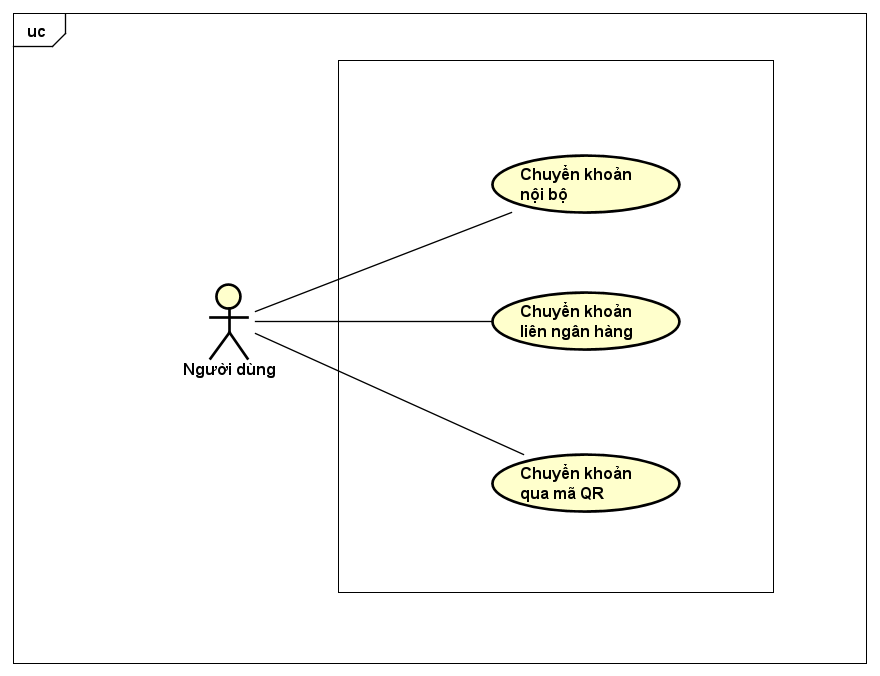


Hình . Biểu đồ usecase tổng quan

Hệ thống gồm có 2 tác nhân là người dùng và người quản trị. Người dùng sau khi đăng ký và xác thực thông tin thì có thể cài đặt các thông tin như soft otp, hạn mức giao dịch, thẻ vật lý. Ngoài ra có thể thực hiện các giao dịch trực tuyến: chuyển khoản nội bộ và liên ngân hàng, nạp hoặc mua mã thẻ điện thoại, gửi tiền trực tuyến. Bên cạnh đó một số chức năng hỗ trợ cung cấp thông tin đến người dùng như tra cứu thông tinm tra cứu lịch sử giao dịch. Người quản trị cũng là một tác nhân chính, có thể quản lý các thành phần của hệ thống như quản lý người dùng, quản lý tin tức, quản lý lãi xuất, quản lý hỏi đáp, quản lý địa điểm giao dịch.

### Biểu đồ use case phân rã “Chuyển khoản”

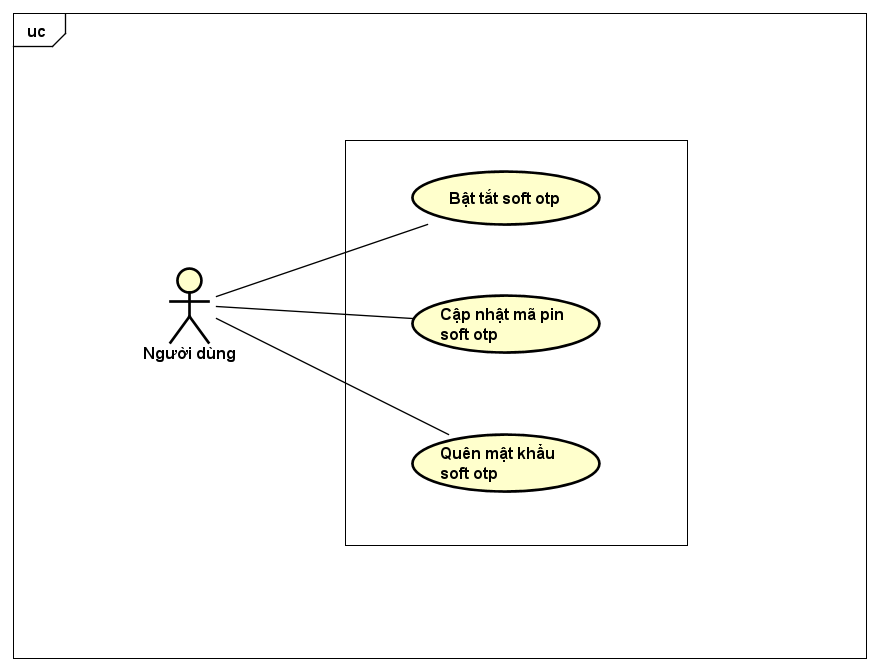
Hệ thống có ba hình thức chuyển khoản chính là chuyển khoản nội bộ để chuyển tiền giữa những người dùng trong cùng hệ thống, liên ngân hàng để chuyển tiền với những người sử dụng ngân hàng khác hoặc sử dụng mã qr để thực hiện thao tác nhanh chóng, Hình 2 dưới đây sẽ biểu diễn



Hình . Biểu đồ usecase phân rã “Chuyển khoản”

### Biểu đồ use case phân rã “Cài đặt soft otp”

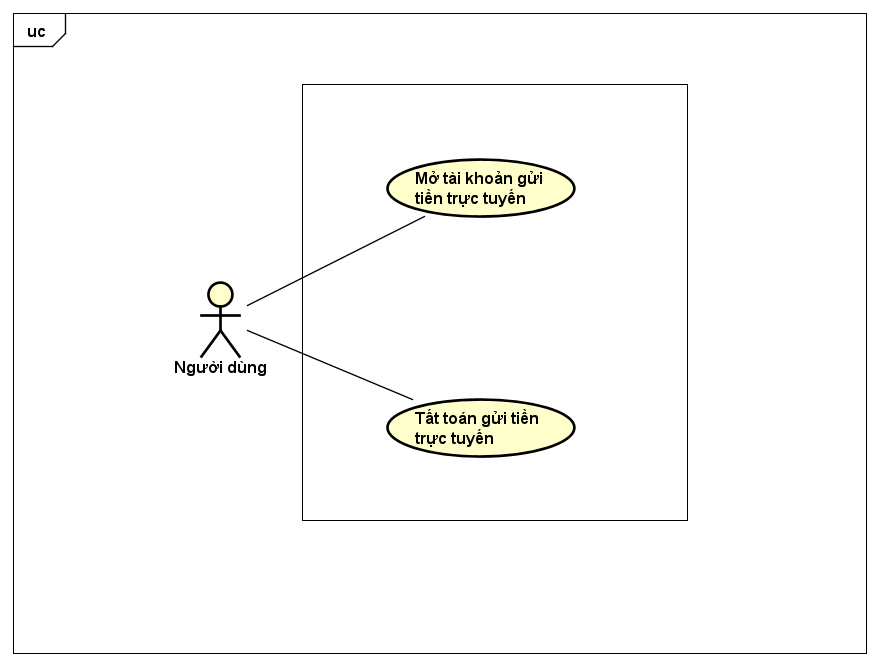
Một trong phương thức xác thực trước khi thực hiện giao dịch là soft otp, là một mã pin 6 ký tự, người dùng có thể bật, tắt, cập nhật, quên mã pin



Hình . Biểu đồ use case phân rã “cài đặt soft otp”

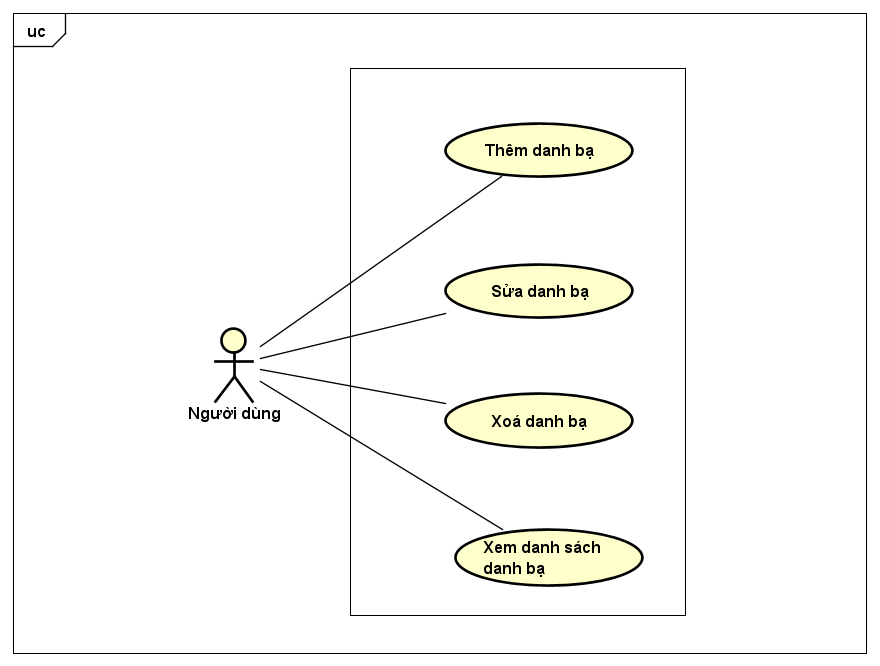
### Biểu đồ use case phân rã “Gửi tiền trực tuyến”

Hệ thống mobile banking hỗ trợ gửi tiền trực tuyến. Khi muốn bắt đầu gửi tiết kiệm, người dùng sử dụng tính năng mở tài khoản gửi tiền trực tuyến với các thông tin như kỳ hạn, số tiền, hình thức gia hạn. Khi muốn nhận lại số tiền gốc và lãi thì người dùng cần tất toán tài khoản trực tuyến. Hình 4 sẽ mô tả các chức năng trên.



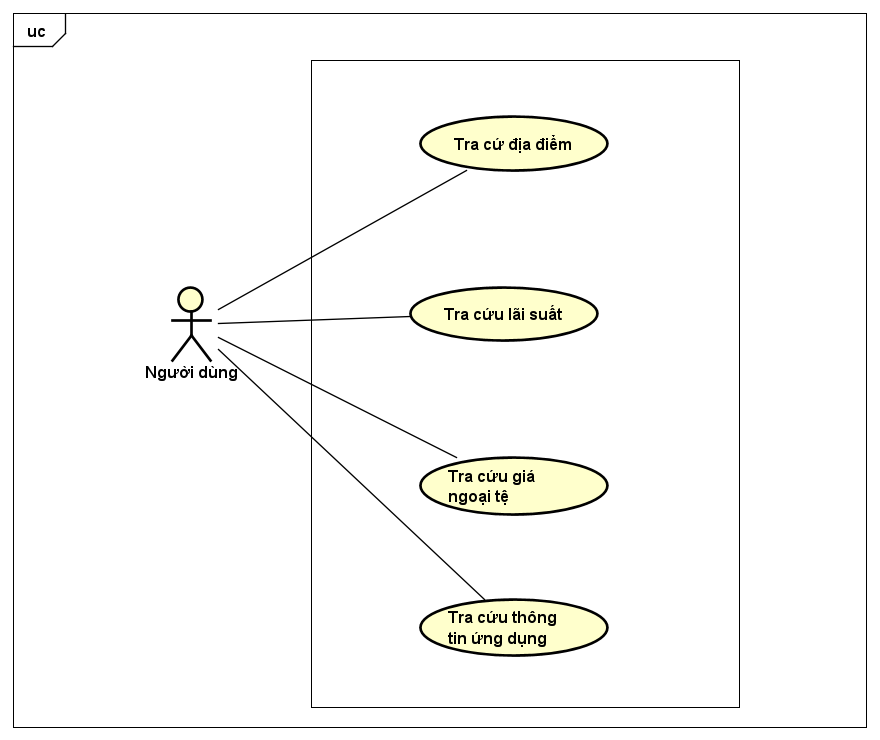
Hình . Phân rã usecase gửi tiền trực tuyển

### Biểu đồ use case phân rã “Quản lý danh bạ”



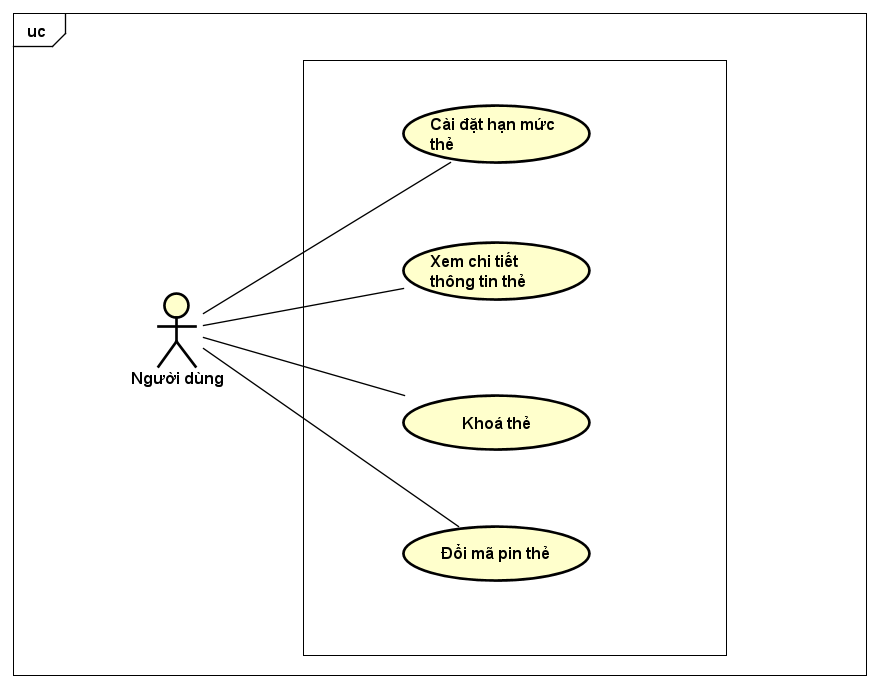
Hình . Biểu đồ usecase phân rã "Quản lý danh bạ"

### Biểu đồ use case phân rã “Tra cứu thông tin”



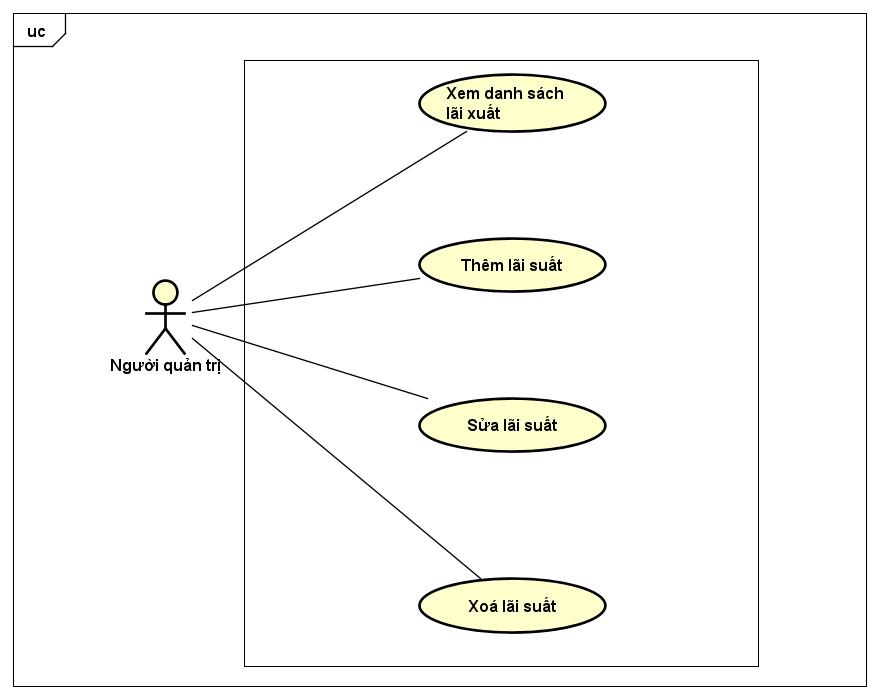
Hình . Biểu đồ use case phân rã “Tra cứu thông tin”

### Biểu đồ use case phân rã “Quản lý thẻ atm”



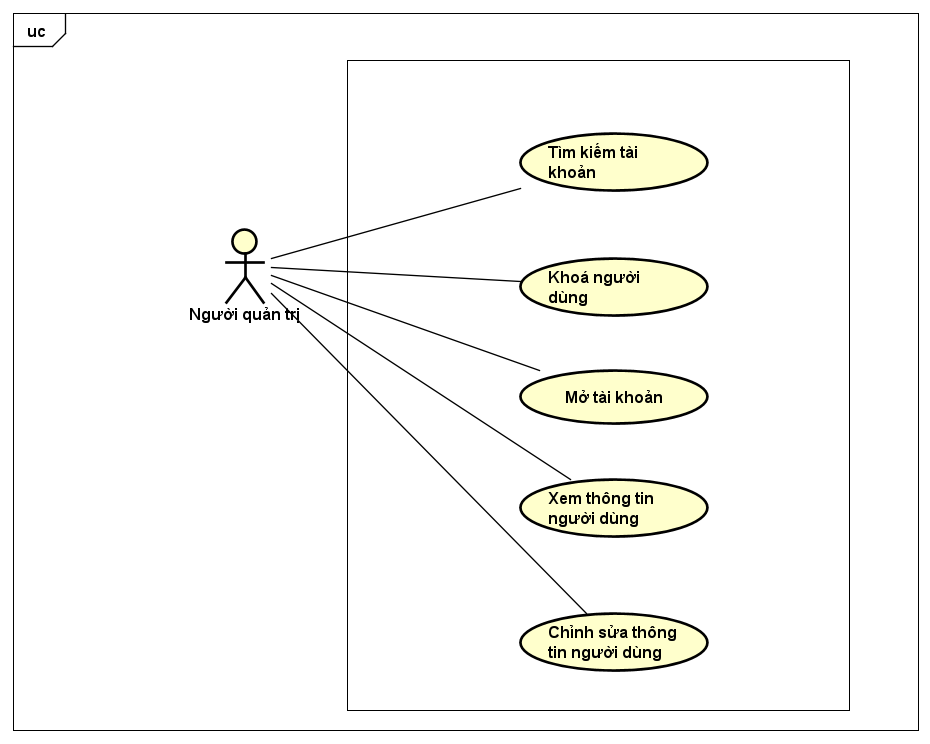
Hình . Biểu đồ use case phân rã “Quản lý thẻ atm”

### Biểu đồ use case phân rã “Quản lý lãi suất”



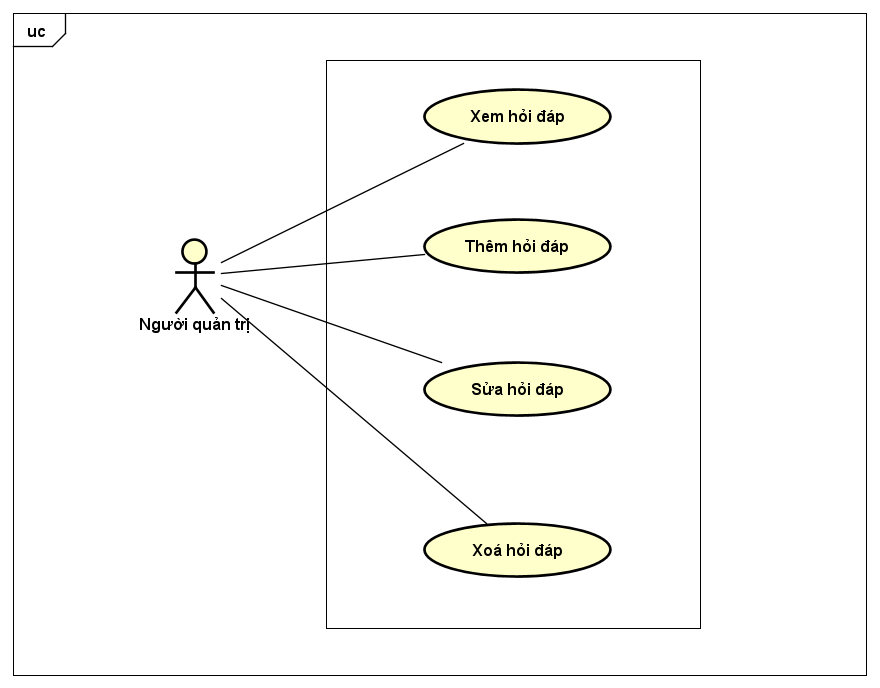
Hình . Biểu đồ use case phân rã “Quản lý lãi suất”

### Biểu đồ use case phân rã “Quản lý người dùng”



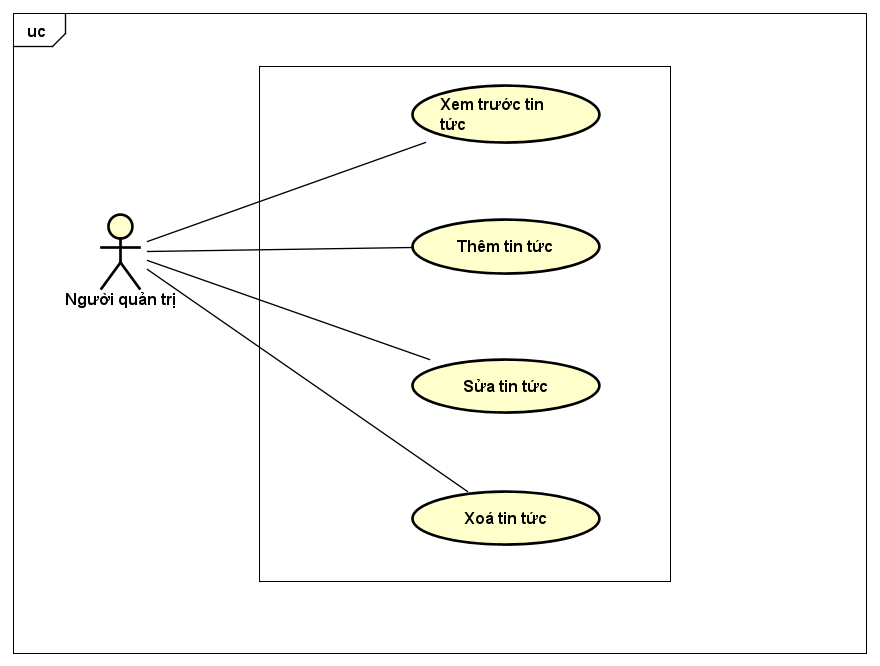
Hình . Biểu đồ use case phân rã “Quản lý người dùng”

### Biểu đồ use case phân rã “Quản lý hỏi đáp”



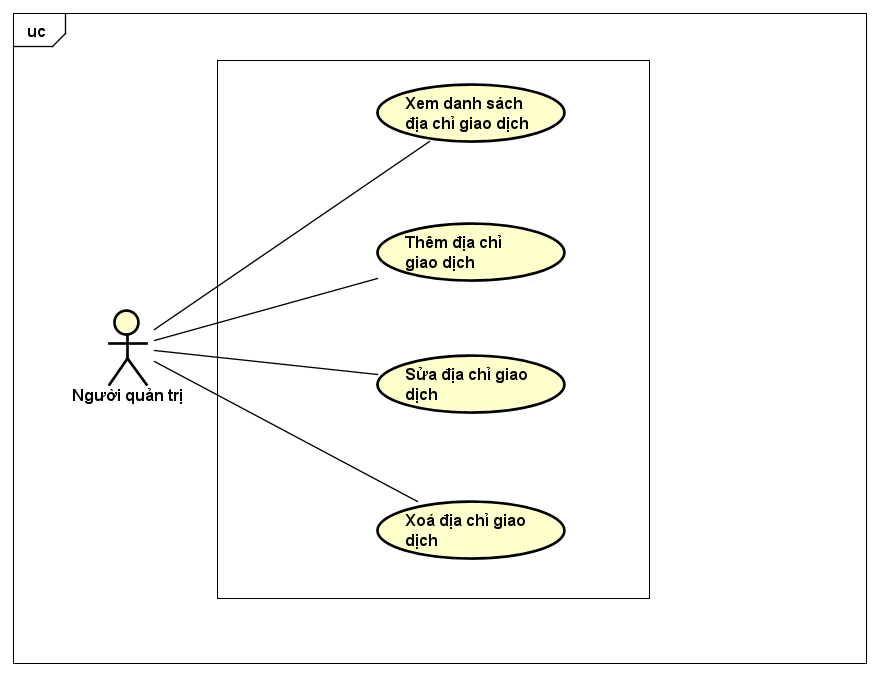
Hình . Biểu đồ use case phân rã “Quản lý hỏi đáp”

### Biểu đồ use case phân rã “Quản lý tin tức”



Hình . Biểu đồ use case phân rã “Quản lý tin tức”

### Biểu đồ use case phân rã “Quản lý địa điểm giao dịch”



Hình . Biểu đồ use case phân rã “Quản lý địa điểm giao dịch”

### Quy trình nghiệp vụ

Quy trình xác thực thông tin

Quy trình chuyển khoản

Quy trình gửi tiền tiết kiệm online

## Đặc tả chức năng

Sinh viên lựa chọn từ 4 đến 7 use case quan trọng nhất của đồ án để đặc tả chi tiết. Mỗi đặc tả bao gồm ít nhất các thông tin sau: (i) Tên use case, (ii) Luồng sự kiện (chính và phát sinh), (iii) Tiền điều kiện, và (iv) Hậu điều kiện. Sinh viên chỉ vẽ bổ sung biểu đồ hoạt động khi đặc tả use case phức tạp.

* Xác minh thông tin
* Chuyển khoản
* Đổi mã pin soft otp
* Mua mã thẻ điện thoại
* Mở tài khoản trực tuyến
* Tra cứu lịch sử giao dịch
* Chỉnh sửa tin tức

### Đặc tả use case A

### Đặc tả use case B

## 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.

### Độ tin cậy

* Phân quyền: đối với mỗi tác nhân sẽ có những quyền hạn truy cập và sử dụng các chức năng nhất định.
* Các thông tin như mật khẩu, mã pin, mã thẻ trước khi lưu trữ cần được mã hoá để tránh tấn công (sử dụng mã hoá theo chuẩn bcrypt).
* Có khả năng sao lưu và phục hồi dữ liệu một cách an toàn.
* Các giao dịch đều là minh bạch, được lưu trữ lại và có thể tra cứu lại khi cần thiết.
* Sử dụng mã pin, mật khẩu, sms để xác thực các giao dịch tránh việc tài khoản dễ dàng được sử dụng cho mục đích xấu bởi người khác

### Tính dễ bảo trì

Hệ thống vẫn còn đang tiếp tục trong quá trình phát triển nên cần được thiết kế để dễ dàng mở rộng, thêm nới các chức năng. Những chức năng được them mới này sẽ không ảnh hưởng đến sự hoạt động ổn định của hệ thống trước đó

### Yêu cầu về giao diện người dùng

* Giao diện cần đẹp, đơn giản, dễ học, dễ dùng, phù hợp với đa phần các lứa tuổi.
* Phân chia các tính năng rõ rang, rành mạch.
* Tương thích, thống nhất trên các kích thước màn hình khác nhau

Trong chương 2 này tôi đã trình bày các đánh giá của mình về một số sản phẩm phổ biến, đang được sử dụng bởi nhiều người từ đó đưa ra các chức năng cần có trong hệ thống và tiến hành phân tích một số usecase quan trọng. Ngoài ra các yêu cầu phi chức năng cũng được đề cập giúp nâng cao tính bảo mật, trải nghiệp người dùng. Chương 3 tôi sẽ trình bày các công cụ và công nghệ sử dụng xuyên suốt quán trình phát triển hệ thống.

# 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 1.3.

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).

* Javascript
* Nodejs express làm backend
* Mysql sequelize database
* Flutter làm app
* Reactjs antd design Tailwind làm cms
* Python Flash làm server nhận diện ai
* 2 Module AL
* Docker CI CD triển khai lên server thuận tiện hơn

Trong chương 2, tôi đã trình bày một cách tổng quan về các chức năng chính của hệ thống mobile banking. Trong chương này, tôi sẽ trình bày các công nghệ sử dụng trong quá trình xây dựng và phát triển hệ thống.

## Javascript

Javascrip có tên viết tắt là JS. Đây là một ngôn ngữ lập trình theo kịch bản, được hỗ trợ bởi hầu hết các trình duyệt. Quy tắc, cách viết của ngôn ngữ này đơn giản, ngắn gọn giúp người dùng dễ hiểu, dễ học

Thường được sử dụng để tạo các website. Khi đó, HTML như bộ khung, CSS trang chí các thành phần thì JS chịu trách nhiệm chính trong việc tương tác với người dùng. Hiện nay có rất nhiều các framework, thư viện xây dựng bằng javascript, điển hình như ReactJs, Vuejs, Angular.

Ngoài ra những năm gần đây, Javascript còn được sử dụng để viết mã nguồn phía máy chủ nhờ vào môi trường runtime của Nodejs. Một số framework phía máy chủ phổ biến như Express, Koa …

Việc lựa chọn JS là ngôn ngữ chính để xây dựng backend cũng như hệ thống cms quản lý cho người quản trị đảm bảo hiệu quả cao cũng như tính dễ sử dụng.

## Platform NodeJs và Framework Express

Nodejs là một nền rảng được xây dựng dựa trên V8 Javascript Engine. Theo khảo sát của StackOverflow thì năm 2021 thì nodejs chiếm 34% số lượng người dùng trong tổng các nền tảng và ngôn ngữ đang được sử dụng. Sự xuất hiện của nodejs khiển cho javascrip giờ đây có thể thực thi ngay cả phía máy chủ. Cộng đồng sử dụng Nodejs rất lớn mạnh, có rất nhiều trang web lớn đang sử dụng NodeJs để viết chương trình, có thể kể đến: Netflix, Trello, Paypal, Ebay…

Express là một framework miễn phí cho Nodejs. Sử dụng Express giúp tạo dựng bộ khung ứng dụng một cách nhanh chóng và mang lại nhiều lợi ích:

* Phát triển máy chủ nhanh hơn
* Dễ dàng xây dựng thành phần định tuyến
* Dễ gỡ lỗi khi viết mã nguồn
* Hệ thống thư viện phong phú, đa dạng, giúp giải quyết nhanh chóng các vấn đề cần giải quyết

Việc chọn Nodejs để phát triển các API phía backend không chỉ tiết kiệm thời gian xây dựng hệ thống, nó còn giúp hệ thống dễ dàng chỉnh sửa và mở rộng. Mang đến trải nghiệm tối ưu cho người dùng.

## Mysql và Sequelize

Mysql là hệ thống quản trị cơ sở dữ liệu mở phổ biến và được ưa chuộn trong quá trình phát triên ứng dụng. Mysql sử dụng mô hình dữ liệu quan hệ, được biết đến với tốc độ khá cao, ổn định, hoạt động được trên nhiều hệ điều hành.

Bên cạnh mô hình dữ liệu quan hệ còn có mô hình dữ liệu phi quan hệ NoSql. Trong mô hình này, mọi dữ liệu sẽ không bị rang buộc về quan hệ, thời gian truy xuất các dữ liệu độc sẽ nhanh hơn so với cơ sở dữ liệu quan hệ. Tuy nhiên trong các trường hợp cần truy xuất liên quan các dữ liệu khác thì cơ sở dữ liệu quan hệ lại phát huy tính hiệu quả của nó.

Hệ thống mobile banking là một hệ thống có các nghiệp vụ về kinh tế, ngân hang nên các dữ liệu có mối quan hệ chặt chẽ với nhau, vì vậy việc chọn hệ quản trị dữ liệu MySQL sẽ phù hợp hơn. Ngoài ra còn giúp hệ thống dễ dàng mở rộng, phát triển sau này.

Trong hệ thống này, để thao tác với MySql dễ dàng, thuận tiện thì một javascript ORM là Sequelize đã được sử dụng. Sequelize cung cấp các chức năng chính như tạo model để dễ dàng quản lý thay vì tạo trực tiếp bảng, truy vấn nhanh gọn với các method có sẵn thay vì phải viết từng câu truy vấn thuần, ngoài ra còn hỗ trợ relations, transaction…

## ReactJs và Antd design

ReactJs là một thư viện Javascript mã nguồn mở, được thiết kế bởi Facebook hỗ trợ xây dựng các thành phần giao diện nhanh gọn và tiện lợi. Thay vì phải nhúng javasctipt vào code HTML thì ReactJs đã xây dựng cú pháp JSX giúp thực hiện việc này dễ dàng hơn. Thành phần cơ bản của ReactJs là components, chúng giúp người dùng tái sử dụng các đoạn mã nguồn một cách hiệu quả để đảm bảo tính thống nhất cũng việc giảm thời gian việt mã nguồn.

Ứng dụng sử dụng ReactJs để xây dựng thuộc loại ứng dụng Single Page, các mã nguồn JS sẽ được trình duyệt thực thi phần lớn trong lần tải trang đầu tiên. Điều này đem lại trải nghiệm mượt mà đồng thời giảm bớt gành nặng phía backend.

Antd Design là một tập hợp các components của React được xây dựng theo các tiêu chuẩn về giao diện và trải nghiệm người dùng. Nó cung cấp hết các component trong một ứng dụng web hiện đại như Layout, Button, Icon, DatePicker, Input …

Với những yêu điểm trên thì việc lửa chọn sử dụng ReactJs kết hợp với Antd Design giúp thành phần cms quản lý cho người quản trị của ĐATN trở nên thân thiện, trải nghiệm người dùng tốt và trên hết là giảm ngắn thời gian triển khai.

## Flutter

Flutter là mã nguồn mở miễn phí mobile UI framework sử dụng ngôn ngữ dart của Google. Framework này giúp tạo ra các giao diện chất lượng cao trên iOS và Android trong khoản thời gian ngắn. Do việc sử dụng ngôn ngữ Dart nên Flutter rất dễ học, ngoài ra có rất nhiều lý do để lựa chọn Flutter cho hệ thống này, một số ưu điểm nổi bật như:

* Phát triển ứng dụng nhanh.
* Chỉ cần viết một lần duy nhất cho tất cả các ứng dụng.
* Giao diện người dùng đẹp và linh hoạt.
* Hỗ trợ nhiều widget khác nhau.
* Ứng dụng có hiệu năng cao so với các framework tương tự.

## Các mô hình Yolov5LiteS, MobileFaceNet va Flash

YOLOv5 là một mô hình Object Detection thuộc họ mô hình YOLO. So với YOLOv4 thì YOLOv5 có độ chính xác tương đương nhưng nhưng lại có tốc độ nhanh hơn khi thực thi dự đoán.

MobileFaceNet dựa trên siam network, biểu diễn các bức ảnh trong một không gian eucledean n chiều sao cho khoảng cách giữa các embedding càng nhỏ, mức độ tương đồng càng lớn.

Flash là một web framework, nó là một Python module cho phép bạn triển khải ứng dụng một các dễ dàng với các chức năng như định tuyến url, template engine.

ĐATN này đã sử dụng mô hình YOLOv5 để nhận diện các khuôn mặt trong ảnh thẻ căn cước cũng như ảnh chân dung. Sau khi đã xác định được ví trí khuôn mặt từ các ảnh thì sử dụng mô hình MobileFaceNet để tính toán, đưa ra kết quả về độ tương đồng giữa 2 khuôn mặt. Tuy nhiên phần hệ thống xác thực thông tin đang sử dụng ngôn ngữ Python, khác với ngôn ngữ Javascript của backend nên việc sử dụng Flash để hai dịch vụ giao tiếp qua API là phù hợp.

## Docker

Docker là một nền tảng mở giúp building, deploying và running ứng dụng dễ dàng hơn bằng cách sử dụng các containers trên nền tảng ảo hoá. Các containers cho phép lập trình viên đóng gói một ứng dụng với tất cả các phần cần thiết, chẳng hạn như thư viện và các phụ thuộc khác, và gói tất cả dưới dạng một package. Docker đang được sử dụng rất nhiều vì tính dễ dùng, tốc độ và khả năng mở rộng.

Vì nền tảng xây dựng ĐATN, cụ thể là cơ sở dũ liệu MySql là windows còn nền tảng triển khai ở server là linux nên việc sử dụng docker là cần thiết và tiện lợi.

## CI CD

CI/CD là một bộ đôi công việc bao gồm CI (Continuous Integration) và CD (Continuous Delivery), nói cách khác là quá trình tích hợp thường xuyên hơn cũng như cập nhật lên phiên bản mới nhanh hơn. Nó là quá trình làm việc liên tục và tự động hoá phần mềm.

Thay vì phải thao tác trên server mỗi lần thay đổi mã nguồn ở máy tính cá nhân và muốn nó đồng bộ với server thì nhờ CI/CD, các thao tác test, build, run trên server diễn ra một các tự động khi có thay đổi ở repo.

Vì hệ thống mobile banking có nhiều dịch vụ như xác thực, các đầu api, hay giao diện cms quản lý cho người quản trị nên khi ứng dụng CI/CD vào giúp việc triển khai mã nguồn lên server tiết kiệm thời gian đáng kể so với phương pháp thủ công.

Chương 3 này tôi đã trình bày các công nghệ mà em sử dụng để xây dựng hệ thống. Tiếp theo trong chương 4 tôi sẽ nói chi tiết về thiết kế kiến trúc, thiết kế giao diện, cơ sở dữ liệu

# Phát triển và triển khai ứng dụng

## Thiết kế kiến trúc

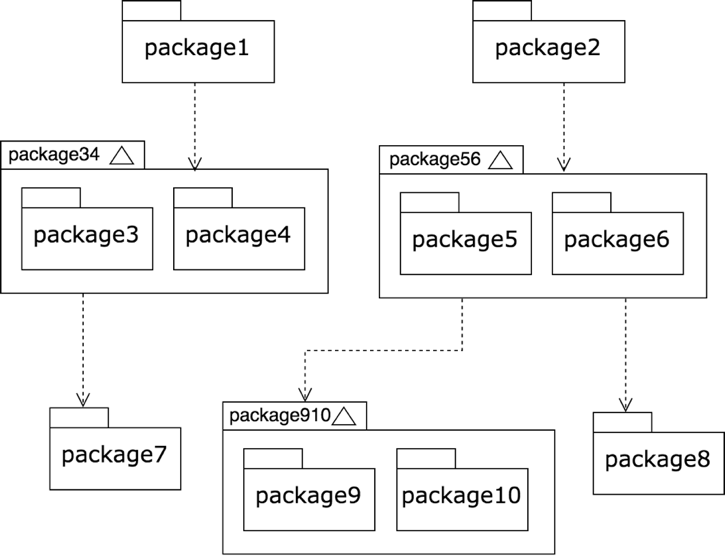
### Lựa chọn kiến trúc phần mềm

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.

### 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 1.



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

### 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 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 Hình 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

## Thiết kế chi tiết

### 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.

### 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).

### 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.)

## Xây dựng ứng dụng

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

**Bảng 1** Danh sách thư viện và công cụ sử dụng

|  |  |  |
| --- | --- | --- |
| **Mục đích** | **Công cụ** | **Địa chỉ URL** |
| IDE lập trình | Eclipse Oxygen 64 bit | http://www.eclipse.org/ |
| v.v. | v.v. | v.v. |

### 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.

### Minh hoạ 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.

## 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.

## 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.)

# Các giải pháp và đóng góp nổi bật

Chương này có độ dài tối thiểu 5 trang, tối đa không giới hạn. 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”.

# Kết luận và hướng phát triển

## 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.

## 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.

# Tài liệu tham khảo

Lưu ý: Sinh viên không được đưa bài giảng/slide, các trang Wikipedia, hoặc các trang web thông thường làm tài liệu tham khảo.

Một trang web được phép dùng làm tài liệu tham khảo **chỉ khi** nó là công bố chính thống của cá nhân hoặc tổ chức nào đó. Ví dụ, trang web đặc tả ngôn ngữ XML của tổ chức W3C https://www.w3.org/TR/2008/REC-xml-20081126/ là TLTK hợp lệ.

Có năm loại tài liệu tham khảo mà sinh viên phải tuân thủ đúng quy định về cách thức liệt kê thông tin như sau. Lưu ý: các phần văn bản trong cặp dấu < > dưới đây chỉ là hướng dẫn khai báo cho từng loại tài liệu tham khảo; sinh viên cần xóa các phần văn bản này trong ĐATN của mình.

<**Bài báo đăng trên tạp chí khoa học**: Tên tác giả, tên bài báo, tên tạp chí, volume, từ trang đến trang (nếu có), nhà xuất bản, năm xuất bản>

1. Hovy E. H., Automated Discourse Generation Using Discourse Structure Relations, Artificial Intelligence, Elsevier Science Publishers, 63: 341-385, 1993.

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

1. Peterson L. L. and Davie B. S., Computer Networks: A Systems Approach, 2nd ed., Mogran-Kaufmann, 1999.
2. Nguyễn Thúc 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>

1. Poesio M. and Di Eugenio B., Discourse Structure and Anaphoric Accessibility, In Proc. of the ESSLLI Workshop on Information Structure, Discourse Structure and Discourse Semantics, Helsinki, 2001.

<**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>

1. Knott D., A Data-Driven Methodology for Motivating a Set of Coherence Relations, Ph.D. Thesis, 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>

1. Berners-Lee T., Hypertext Transfer Protocol (HTTP), CERN, ftp:/info.cern.ch/pub/www/doc/http-spec.txt.Z, last visited May 2010.
2. Princeton University, WordNet, http://www.cogsci.princeton.edu/~wn/index.shtml, last visited May 2010.

# Phụ lục

Phần phụ lục là không bắt buộc. Nếu sinh viên không có nhu cầu trình bày thêm, có thể xóa bỏ phần này. Lưu ý là phần phụ lục chỉ được đánh chỉ mục đến cấp 2, sinh viên không được phép chia nhỏ hơn nữa.

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.

Ngành học

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

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

* Đối với kỹ sư chính quy:
  + Từ K61 trở về trước: Ngành Kỹ thuật phần mềm
  + Từ K62 trở về sau: Ngành Khoa học máy tính
* Đối với cử nhân:
  + Ngành Công nghệ thông tin
* Đối với chương trình EliteTech:
  + Chương trình Việt Nhật/KSTN: Ngành Công nghệ thông tin
  + Chương trình ICT Global: Ngành Information Technology
  + Chương trình DS&AI: Ngành Khoa học dữ liệu
* Đối với kỹ sư chương trình đào tạo quốc tế SIE
  + Chương trình LTU: Ngành Công nghệ thông tin/Information Technology
  + Chương trình VUW: Ngành Kỹ thuật Phần mềm
  + Chương trình G-INP: Ngành Hệ thống thông tin

Tạo đề mục

Đề mục giúp tạo bố cục cho tài liệu. Để các tính năng tự động – ví dụ tính năng cập nhật mục lục, hoặc tính năng tham chiếu chéo – của Word hoạt động được, sinh viên cần tuân thủ theo các style đã tạo trong tài liệu này. Để hiển thị các style này, sinh viên vào tab Home trong thanh Ribbon của Word.

Để tạo đề mục cấp 1, 2, 3, 4, 5, sinh viên gõ tiêu đề cho đề mục của mình rồi chọn các style là Heading 1, 2, 3, 4, 5 tương ứng. Sinh viên hạn chế dùng tới đề mục cấp 4, và phải trong trường hợp thực sự cần thiết mới dùng đến đề mục cấp 5.

Phần phụ lục chỉ cho phép có hai cấp tiêu đề. Hai style tương ứng với hai cấp này là “Heading 7, Phụ lục cấp 1” và “Heading 8, Phụ lục cấp 2”.

Bảng biểu

Sinh viên lưu ý không để bảng tràn ra lề (margin) trên, dưới, trái hoặc phải của trang. Do không gian nhỏ hẹp, bảng nên có font là 12pt hoặc nhỏ hơn. Độ dãn dòng của bảng nên là 1 line. Căn lề bảng là căn giữa, nhưng nội dung văn bản trong bảng nên được căn lề trái.

Sinh viên có thể viết tắt các từ trong bảng để tiết kiệm không gian nhưng phải giải thích các từ viết tắt này ở phần Chú thích bảng. Ví dụ áp dụng được minh họa trong Bảng 2.

**Bảng 2** Ví dụ sử dụng bảng

Chú thích: Y: Year; RS: Risk Set; G: Graduated; AB: Academically Excluded;C: Censored; HRG: Hazard Ratio – Graduated

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Y** | **RS** | **G** | **AE** | **C** | **HRG**  **(%)** |
| **1** | 11.959 | 0 | 725 | 1619 | 0,0 |
| **2** | 10.457 | 0 | 474 | 1513 | 0,0 |
| **3** | 7.365 | 1213 | 335 | 966 | 16,7 |
| **4** | 900 | 599 | 145 | 405 | 55,3 |

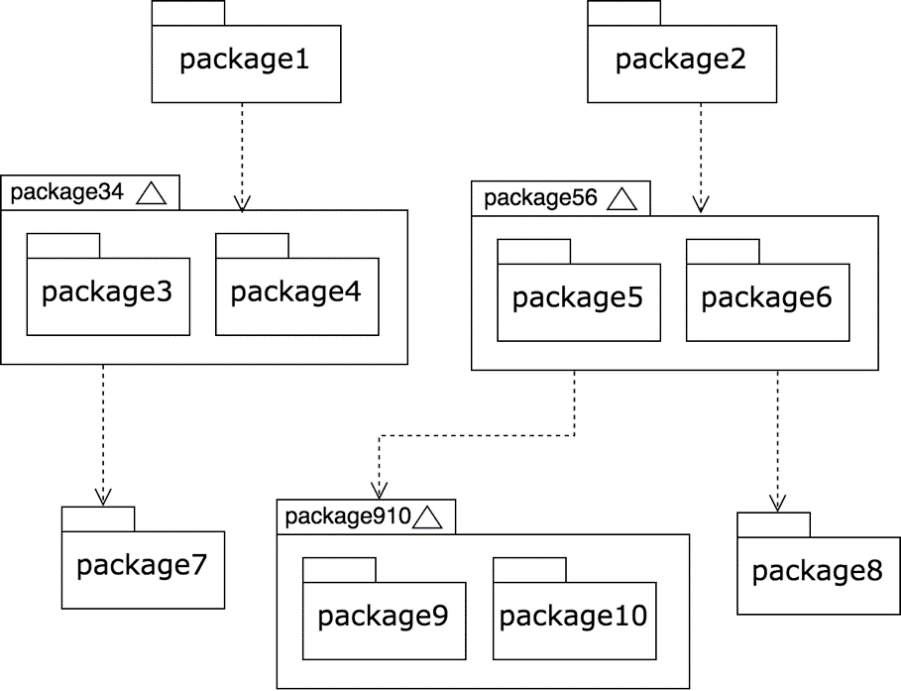
Sinh viên được tùy ý lựa chọn định dạng (template) cho các bảng trong ĐATN của mình, nhưng phải đảm bảo tính nhất quán trên toàn ĐATN. Template bảng phải đảm bảo phần heading cột trong bảng có font in đậm và nổi bật (highlighted) hơn so với các nội dung khác trong bảng.

Vì bảng có thể kéo dài nhiều trang, tiêu đề của bảng nên để ở phần đầu của bảng. Sinh viên không thêm tiêu đề bảng bằng tay. Để thêm tiêu đề bảng tự động, sinh viên nhấn chuột phải vào bảng, chọn “Insert Caption”, chọn “Label” là “Bảng”, rồi nhấn nút “OK”. Sau đó, sinh viên nhập vào nội dung tiêu đề và căn chỉnh “Center” cho tiêu đề này. Lưu ý, sinh viên cần bôi đậm bằng tay cụm từ **Bảng n**. Kết quả thu được có dạng như sau “**Bảng 2** Ví dụ sử dụng bảng”. Sinh viên nên xoay ngang trang giấy trong trường hợp bảng có nhiều cột với nhiều nội dung văn bản.

Hình vẽ

Tương tự như bảng, sinh viên không được để hình vẽ tràn lề trang. Căn lề cho hình vẽ là căn giữa (Center). Cách thêm tiêu đề hình vẽ tương tự như cách thêm tiêu đề bảng, nhưng sinh viên chọn “Label” là “Hình” thay vì “Bảng”.

Tiêu đề hình vẽ phải đặt ở dưới hình vẽ. Nếu hình vẽ được copy từ trên mạng, sinh viên bắt buộc phải ghi rõ nguồn. Sinh viên nên thống nhất công cụ sử dụng và style cho hình vẽ trong toàn ĐATN. Các chi tiết trong hình vẽ phải được bố trí gọn gàng; chữ trong hình phải đảm bảo nhìn được rõ nét khi in báo cáo trên giấy A4. Khi resize ảnh, cần giữ nguyên tỷ lệ dài rộng, tránh làm méo hoặc vỡ hình. Ví dụ hình vẽ được minh họa trong Hình 3.



**Hình 0.1** Ví dụ hình vẽ

Tài liệu tham khảo

Sinh viên cần hạn chế tối đa dùng trang Web làm tài liệu tham khảo. Chỉ chấp nhận trang Web làm TLTK khi trang đó là nơi công bố chính thức của tổ chức hoặc cá nhân 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.

Khi giới thiệu về công cụ, API, thư viện, hoặc nền tảng nào đó, sinh viên có thể đưa ra địa chỉ URL của các tiện ích này. Sinh viên lưu ý địa chỉ URL đó không phải là tài liệu tham khảo. Trong các trường hợp tương tự như vậy, sinh viên nên tạo “Footnote”. Sinh viên tạo “Footnote” bằng cách vào mục “References”, chọn “Insert Footnote”. Ví dụ tạo Footnote như sau: TensorFlow[[1]](#footnote-1) là nền tảng học máy mã nguồn mở đang được sử dụng rộng rãi hiện nay. Lưu ý: số Footnote phải đặt sát với từ được mô tả. Như trong ví dụ trên, số 1 được đặt ngay cạnh chữ TensorFlow (không có dấu cách).

Công thức toán học

Giống như bảng, hình vẽ, và tài liệu tham khảo, công thức toán học cần được đánh số, giải thích, và tham chiếu đầy đủ.

Để thêm tiêu đề (caption) cho công thức, sinh viên đặt con trỏ văn bản vào dòng văn bản dưới công thức. Sau đó vào menu “Insert” (lưu ý là menu Insert, không phải là tab Insert trong thanh Ribbon) và chọn mục “Caption”. Từ Popup Menu hiện ra, sinh viên chọn “Label” là “Công thức”, rồi nhấn nút “OK”. Sinh viên bôi đậm chữ **Công thức n**, và căn tiêu đề ra giữa. Công thức 1 là ví dụ mẫu cho sinh viên tham khảo.

**Công thức 1** Khai triển Newton

Tham chiếu chéo

Tham chiếu chéo (Cross-reference) là tiện ích hữu hiệu cho người viết báo cáo. Nó giúp tạo các liên kết tham chiếu (hyperlink) tới các hình ảnh, bảng biểu, tài liệu tham khảo, và các đề mục một cách tự động. Ví dụ, ngay trong câu này, một tham chiếu đã được tạo ra tới mục 2.1. Người đọc dễ dàng nhấp chuột vào liên kết 2.1 để ngay lập tức chuyển đến mục đó.

Để tạo tham chiếu chéo tới các đề mục (heading), sinh viên vào tab “References”, rồi tìm và nhấn chọn “Cross-refrence”. SV chọn “Refrence type” là “Heading” và chọn “Insert Reference to” là “Heading number (no context)”. Sau đó, SV chọn phần đề mục muốn tham chiếu rồi bấm “Insert”.

Để tạo tham chiếu chéo tới các hình vẽ, bảng biểu và công thức, sinh viên cũng vào tab “References” và chọn “Cross-refrence”. SV chọn “Refrence type” là “Hình”, “Bảng”, hoặc “Công thức’. SV chọn “Insert Reference to” là “Only label and number”, trỏ đến phần muốn tham chiếu rồi bấm “Insert”. Nếu font chữ trong liên kết tham chiếu tạo ra được in đậm (bold), SV chuyển về dạng thường cho chuẩn tắc.

Thực hiện tương tự các bước mô tả ở trên, sinh viên có thể tạo tham chiếu chéo tới các tài liệu tham khảo. Sinh viên chọn “Refrence type” là “Numbered item”, chọn “Insert Reference to” là “Paragraph number”, trỏ đến phần tài liệu tham khảo muốn tham chiếu rồi bấm “Insert”. Ví dụ, tham chiếu chéo tới tài liệu tham khảo [2], [3], [4] đã được tạo.

Cập nhật mục lục và tham chiếu chéo

Trong suốt quá trình viết ĐATN, sinh viên sẽ tạo ra nhiều xáo trộn như thay đổi vị trí hình và bảng, thay đổi thứ tự tài liệu tham khảo, thay đổi tên đề mục, v.v. Vì vậy, để hoàn thành ĐATN, sinh viên cần cập nhật lại các thành phần mục lục, danh sách hình ảnh/bảng/công thức và tham chiếu chéo được tạo tự động trong ĐATN của mình.

Để cập nhật các thành phần này, sinh viên bấm Ctrl+A để chọn toàn báo cáo, nhấn chuột phải và chọn “Update Field”, rồi liên tục chọn mục “Entire table” khi được hỏi trong hộp thoại Popup Menu. Sau đó sinh viên search chữ “Error” trên toàn báo cáo để kiểm tra xem có lỗi đánh chỉ mục hoặc lỗi tham chiếu nào không. Đồng thời, sinh viên nên tự soát lại bằng tay toàn bộ nội dung quyển ĐATN để tránh mọi sai sót.

In quyển đồ án tốt nghiệp

Do hiện nay có nhiều phiên bản Word cho nhiều nền tảng máy tính, sinh viên nhất thiết phải xuất ĐATN ra định dạng PDF rồi mang tới cửa hàng in ấn để tránh sai sót. Quyển ĐATN nên được in một mặt trên các trang giấy A4.

Khi đóng quyển, 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** (Xem Hình 4).



**Hình 0.2** Quy cách đóng quyển

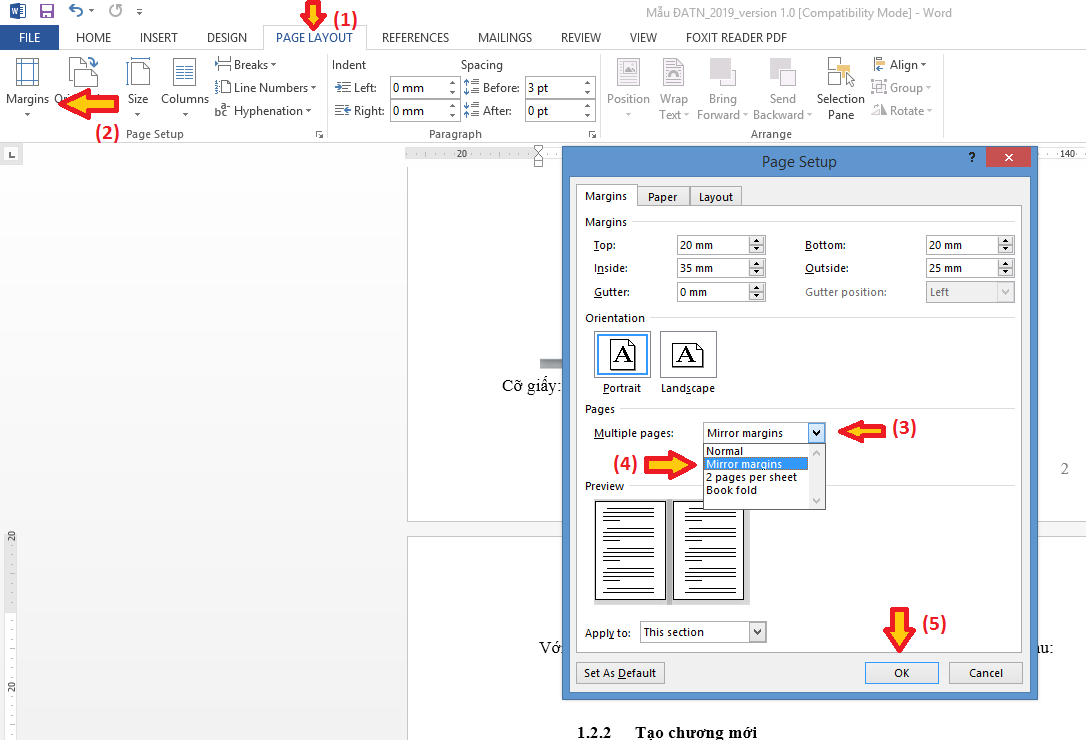
Phần gáy ĐATN được đóng theo quy cách như Hình 5, với các thông tin cần ghi bao gồm *Kỳ làm ĐATN - Ngành đào tạo - Họ và tên sinh viên - Mã số sinh viên*



**Hình 0.3** Quy cách ghi chữ phần gáy

Sinh viên có thể in một mặt hoặc hai mặt. Template này mặc định ở chế độ in một mặt. Để chuyển sang chế độ in hai mặt, sinh viên phải thiết lập lại page layout như sau. Sinh viên vào tab Page Layout, chọn Margins, chọn Custom Margin, trong popup hiện ra, ở mục Multiple pages, chọn Mirror margins, và nhấn OK (Xem **Hình 6**). Do trang đầu tiên là trang bìa của đồ án, nên khi in chế độ hai mặt có thể cần chèn (insert) 1 trang trắng sau trang bìa để đảm bảo trang “Lời cam kết” sẽ là trang lẻ. Để tránh sai sót gây lãng phí, sinh viên cần kiểm tra kỹ trang chẵn lẽ trước khi in.

Sinh viên lưu ý, để đóng quyển theo đúng quy cách với keo nhiệt và chữ in trên gáy, độ dày tối thiểu cho tổng các trang là 0.5cm. Sinh viên tự cân nhắc in một mặt hoặc hai mặt.



**Hình 0.4** Hướng dẫn thiết lập in hai mặt

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.

Đặc tả use case “Thống kê tình hình mượn sách”

Cách trình bày tương tự như phần hướng dẫn ở mục 2.3.

Đặc tả use case “Đăng ký làm thẻ mượn”

Cách trình bày tương tự như phần hướng dẫn ở mục 2.3.

Công nghệ sử dụng

Công nghệ bảo mật dữ liệu

Công nghệ blockchain

Thiết kế gói

Thiết kế gói cho kiến trúc tổng quan

Thiết kế gói cho chức năng “Trả sách”

Thiết kế lớp

1. https://www.tensorflow.org/, lần truy cập cuối: 28/06/2018 [↑](#footnote-ref-1)