­TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

──────── \* ───────

ĐỒ ÁN

**TỐT NGHIỆP ĐẠI HỌC**

NGÀNH CÔNG NGHỆ THÔNG TIN

**XÂY DỰNG HỆ THỐNG ĐIỂM DANH TRỰC TUYẾN**

|  |  |
| --- | --- |
| Giảng viên hướng dẫn: | PGS.TS Tạ Hải Tùng |
| Sinh viên thực hiện: | **Nguyễn Hữu Thắng** |
|  | **Vũ Hồng Thắng** |

HÀ NỘI 12-2018

# Lời cam kết

Họ và tên sinh viên : Nguyễn Hữu Thắng

Điện thoại liên lạc : 0356308865 Email : thangnh1997soict@gmail.com

Lớp : CN-CNTT03 K60 Hệ đào tạo : Chính quy

Họ và tên sinh viên : Vũ Hồng Thắng

Điện thoại liên lạc : 0356308865 Email : thangnh1997soict@gmail.com

Lớp : CN-CNTT03 K60 Hệ đào tạo : Chính quy

Chúng tôi – Nguyễn Hữu Thắng và Vũ Hồng Thắng – cam kết Đồ án Tốt nghiệp (ĐATN) là công trình nghiên cứu của bản thân chúng tôi dưới sự hướng dẫn của *học PGS- TS. Tạ Hải Tùng*. Các kết quả nêu trong ĐATN là trung thực, là thành quả của riêng chúng 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. Chúng 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  *Họ và tên sinh viên* |

# Lời cảm ơn

Để hoàn thành được đồ án tốt nghiệp này chúng tôi đã nhận được sự giúp đỡ rất nhiều từ thầy cô, gia đình, bạn bè, đồng nghiệp và người thân.

Trước hết, chúng tôi xin trân trọng gửi lời cảm ơn đến các thầy cô Viện Công nghệ thông tin – truyền thông nói riêng, và toàn thể các thầy cô trong trường đại học Bách Khoa Hà Nội nói chung đã tận tình giảng dạy chúng tôi trong suốt những năm tháng vừa qua.

Tiếp theo chúng tôi xin bày tỏ lòng kính trọng và biết ơn sâu sắc tới thầy Tạ Hải Tùng, người trực tiếp hướng dẫn và giúp đỡ chúng tôi thực hiện đồ án này.

Và chúng tôi cũng xin gửi lời cám ơn chân thành tới tập thể các anh, chị và bạn bè đang làm việc tại tổng công ty giải pháp doanh nghiệp Viettel đã nhiệt tình giúp đỡ chúng tôi trong suốt thời gian thực tập công ty. Đặc biệt là anh em trong phòng giao dịch điện tử.

Cuối cùng chúng tôi xin ngỏ lời cảm ơn chân thành tới gia đình, nơi nuôi dưỡng chúng tôi khôn lớn và những người bạn đã gắn bó với cả hai chúng tôi trong suốt thời gian học tập tại trường đại học Bách Khoa Hà Nội.

# Tóm tắt

Đồ án tốt nghiệp (ĐATN) được thực hiện với mục đích xây dựng hệ thống hỗ trợ việc điểm danh học sinh, sinh viên trong nhà trường. Hệ thống hướng tới tính ứng dụng cao, có khả năng thích ứng mềm dẻo với các yêu cầu từ phía người dùng vì vậy nên nội dung đồ án của tập trung vào các nhiệm vụ chính sau:

* Khảo sát, nghiên cứu để lấy được các quy trình nghiệp vụ và yêu cầu của hệ thống từ phía người dùng.
* Đưa ra được bản phân tích thiết kế chi tiết cho hệ thống dựa trên những khảo sát quy trình nghiệp vụ và yêu cầu người dùng của hệ thống điểm danh.
* Đưa ra và lựa chọn các công nghệ phù hợp để xây dựng hệ thống.
* Tiền hành triển khai và thử nghiệm đưa hệ thống vào sử dụng trong thực tế.

**Nội dung của đồ án bao gồm**:

* ***Mở đầu***

Giới thiệu chung về đồ án: tóm lược nhiệm vụ đề tài, môi trường thực hiện và bố cục tóm tắt của đồ án.

* ***Chương 1: Giới thiệu đề tài***

Giới thiệu cơ bản về điểm danh, các khảo sát về bài toán từ đó đưa ra mô tả tổng quan về bài toán cần giải quyết. Trong chương này cũng đưa ra các mục tiêu của đồ án. Tóm tắt cơ sở lý thuyết cùng với giới thiệu các công cụ sử dụng để thực hiện đồ án tốt nghiệp.

* ***Chương 2: Khảo sát và phân tích yêu cầu***

Nội dung của chương tập trung và việc phân tích các chức năng và dữ liệu của hệ thống dựa trên các khảo sát về các quy trình xử lý và yêu cầu người dùng. Tập trung sâu về việc phân tích các chức năng quan trọng và được cả hai thực hiện.

* ***Chương 3: Công nghệ sử dụng***
* ***Chương 4: Phát triển và triển khai hệ thống`***

Dựa vào những phân tích ở trên trong chương trình bày các thiết kế hệ thống về kiến trúc tổng quan, thiết kế chi tiết các lớp cơ bản và thiết kế chi tiết các bảng trong cơ sở dữ liệu.

Cùng với đó phần này sẽ trình bày việc xây dựng và thử nghiệm hệ thống, đánh giá tổng quan về ưu, nhược điểm cùng khả năng ứng dụng của hệ thống trong thực tế.

* ***Kết luận***

Phần cuối cùng của đồ án sẽ tập trung vào đánh giá các kết quả đạt được của đồ án và những hướng phát triển tương lai để hoàn thiện hệ thống.

# Abstract

The objectives of graduation project are to build a system supporting the rollcalling in the school system. The system is able to be highly applicable, flexibly adaptable to the users’ requirements; therefore, the content of project is focused on some main points:

* Making surveys to get procedures and system requirements.
* Exporting detailed design analysis report for the system based on above surveys and requirement of users.
* Recommending and then choosing the suitable technology in order to establish system.
* Setting up and testing system in the reality.

**The content of the thesis includes**:

* ***Introduction***

Provide objectives of the graduation thesis, the implementation environment and summary of thesis content.

* ***Chapter 1:* *Introduction and oriented solutions***

This part includes the about timekeeping, surveys about this problem and brief description about the problem to be solved. Provide objectives of the graduation thesis. Summary about the literature reviews together with introduction about the tools used to implement graduation project.

* ***Chapter 2:* *System's requirements and analysis***

The content of this chapter is focused on the analysis of data and functions of system based on surveys of processed and requirements of users. It also concentrates intensively on the analysis of important functions and is implemented by itself.

* ***Chapter 3:* *Technology used***
* ***Chapter 3:* *System's design and implement***

Based on the above analysis, this chapter is focused on presenting the system design of overall architecture, main classes and detailed design of the tables in the database.

Moreover, this section will show the development and testing of system as well as general evaluation about the advantages, disadvantages and the possible applications of the system in practice.

* ***Conclusion***

The main content of this chapter is to evaluate the outcomes of this graduation thesis and the direction of future development to improve the system.

# Mục lục

[Lời cam kết iii](#_Toc8244041)

[Lời cảm ơn iv](#_Toc8244042)

[Tóm tắt v](#_Toc8244043)

[Abstract vii](#_Toc8244044)

[Mục lục ix](#_Toc8244045)

[Danh mục hình vẽ xiii](#_Toc8244046)

[Danh mục bảng xiv](#_Toc8244047)

[Danh mục công thức xv](#_Toc8244048)

[Danh mục các từ viết tắt xvi](#_Toc8244049)

[Danh mục thuật ngữ xvii](#_Toc8244050)

[Chương 1 Giới thiệu đề tài 1](#_Toc8244051)

[1.1 Đặt vấn đề 1](#_Toc8244052)

[1.1.1. Ứng dụng công nghệ thông tin trong quản lý trường học 1](#_Toc8244053)

[1.1.2. Điểm danh 2](#_Toc8244054)

[1.2 Mục tiêu và phạm vi đề tài 3](#_Toc8244055)

[1.3 Định hướng giải pháp 4](#_Toc8244056)

[1.4 Bố cục đồ án 4](#_Toc8244057)

[Chương 2 Khảo sát và phân tích yêu cầu 6](#_Toc8244058)

[2.1 Khảo sát hiện trạng 6](#_Toc8244059)

[2.2 Tổng quan chức năng 7](#_Toc8244060)

[2.2.1 Biểu đồ use case tổng quan 7](#_Toc8244061)

[2.2.2 Biểu đồ use case phân rã XYZ 7](#_Toc8244062)

[2.2.3 Quy trình nghiệp vụ 7](#_Toc8244063)

[2.3 Đặc tả chức năng 8](#_Toc8244064)

[2.3.1 Đặc tả use case A 8](#_Toc8244065)

[2.3.2 Đặc tả use case B 8](#_Toc8244066)

[2.4 Yêu cầu phi chức năng 8](#_Toc8244067)

[Chương 3 Công nghệ sử dụng 9](#_Toc8244068)

[3.1 React-native 9](#_Toc8244069)

[3.2 Redux 9](#_Toc8244070)

[3.3 Redux-saga 10](#_Toc8244071)

[Chương 4 Phát triển và triển khai ứng dụng 13](#_Toc8244072)

[4.1 Thiết kế kiến trúc 13](#_Toc8244073)

[4.1.1 Lựa chọn kiến trúc phần mềm 13](#_Toc8244074)

[4.1.2 Thiết kế tổng quan 13](#_Toc8244075)

[4.1.3 Thiết kế chi tiết gói 14](#_Toc8244076)

[4.2 Thiết kế chi tiết 15](#_Toc8244077)

[4.2.1 Thiết kế giao diện 15](#_Toc8244078)

[4.2.2 Thiết kế lớp 15](#_Toc8244079)

[4.2.3 Thiết kế cơ sở dữ liệu 16](#_Toc8244080)

[4.3 Xây dựng ứng dụng 16](#_Toc8244081)

[4.3.1 Thư viện và công cụ sử dụng 16](#_Toc8244082)

[4.3.2 Kết quả đạt được 16](#_Toc8244083)

[4.3.3 Minh hoạ các chức năng chính 17](#_Toc8244084)

[4.4 Kiểm thử 17](#_Toc8244085)

[4.5 Triển khai 17](#_Toc8244086)

[Chương 5 Các giải pháp và đóng góp nổi bật 18](#_Toc8244087)

[5.1 Sử dụng ngôn ngữ react-native trong xây dựng mobile app. 18](#_Toc8244088)

[5.2 Xử lý những hạn chế trong việc điểm danh với chỉ GPS. 19](#_Toc8244089)

[5.3 Xuất báo cáo điểm danh. 20](#_Toc8244090)

[5.4 REST API 20](#_Toc8244091)

[Chương 6 Kết luận và hướng phát triển 21](#_Toc8244092)

[6.1 Kết luận 21](#_Toc8244093)

[6.1.1 Kết quả đạt được 21](#_Toc8244094)

[6.2 Hướng phát triển 22](#_Toc8244095)

[Tài liệu tham khảo 23](#_Toc8244096)

[Phụ lục A-1](#_Toc8244097)

[A Hướng dẫn viết đồ án tốt nghiệp A-1](#_Toc8244098)

[A.1 Quy định chung A-1](#_Toc8244099)

[A.2 Tạo đề mục A-2](#_Toc8244100)

[A.3 Bảng biểu A-2](#_Toc8244101)

[A.4 Hình vẽ A-3](#_Toc8244102)

[A.5 Tài liệu tham khảo A-4](#_Toc8244103)

[A.6 Công thức toán học A-5](#_Toc8244104)

[A.7 Tham chiếu chéo A-5](#_Toc8244105)

[A.8 Cập nhật mục lục và tham chiếu chéo A-6](#_Toc8244106)

[A.9 In quyển đồ án tốt nghiệp A-6](#_Toc8244107)

[B Đặc tả use case B-6](#_Toc8244108)

[B.1 Đặc tả use case “Thống kê tình hình mượn sách” B-6](#_Toc8244109)

[B.2 Đặc tả use case “Đăng ký làm thẻ mượn” B-6](#_Toc8244110)

[C Công nghệ sử dụng C-7](#_Toc8244111)

[C.1 Công nghệ bảo mật dữ liệu C-7](#_Toc8244112)

[C.2 Công nghệ blockchain C-7](#_Toc8244113)

[D Thiết kế gói D-7](#_Toc8244114)

[D.1 Thiết kế gói cho kiến trúc tổng quan D-7](#_Toc8244115)

[D.2 Thiết kế gói cho chức năng “Trả sách” D-7](#_Toc8244116)

[E Thiết kế lớp E-7](#_Toc8244117)

# Danh mục hình vẽ

[**Hình 1** Ví dụ biểu đồ phụ thuộc gói 9](#_Toc529131897)

[**Hình 2** Ví dụ thiết kế gói 10](#_Toc529131898)

[**Hình 3** Ví dụ hình vẽ A-4](#_Toc529131899)

# Danh mục bảng

[**Bảng 1** Danh sách thư viện và công cụ sử dụng 11](#_Toc529131900)

[**Bảng 2** Ví dụ sử dụng bảng A-3](#_Toc529131901)

# Danh mục công thức

[**Công thức 1** Khai triển Newton A-5](#_Toc529131902)

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.

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

### 1.1.1. Ứng dụng công nghệ thông tin trong quản lý trường học

Ứng dụng công nghệ thông tin trong quản lý trường học hiện đang được coi là xu thế tất yếu giúp các trường học phát giải quyết được vấn đề quản lý hiệu quả hơn hơn. Bởi thế các trường học đang dần hướng tới một mô hình trường học áp dụng một giải pháp tích hợp các ứng dụng công nghệ thông tin còn nhằm giúp cho việc quản lý các nguồn lực cán bộ, học sinh, sinh viên… của mình một cách hiệu quả, cung cấp thông tin một cách chính xác, kịp thời cho việc quản lý.

Với việc tận dụng được sức mạnh của công nghệ số trong việc quản lý các hoạt động của mình trường học có thể nâng cao rõ rệt hiệu quả trong quản lý

Trong chuỗi các ứng dụng công nghệ thông tin trong trường học đó thì điểm danh được coi là thành phần cơ bản không thể thiếu.

### 1.1.2. Điểm danh

Với hệ thống trường học tại Việt Nam, điểm danh là yêu cầu không thể thiếu bởi đó là cơ sở, căn cứ để đưa ra được tình hình, ý thức tham gia lớp học của mỗi giảng viên, giáo viên, học sinh, sinh viên… Về khái niệm điểm danh hiện có thể được hiểu như sau:

**Điểm danh** là hình thức ghi nhận thời gian tham gia lớp học của học sinh, sinh viên cũng như cán bộ giảng dạy qua đó đưa ra được những tiêu chí để đưa ra điểm chuyên cần cho mỗi học viên cũng như quản lý được thời gian trên giảng đường của các cán bộ giảng dạy có thể ảnh hưởng đến việc tính lương.

Có nhiều hình thức điểm danh:

* Điểm danh miệng truyền thống
* Điểm danh bằng nhận diện khuôn mặt
* Điểm danh bằng việc quét vân tay
* Điểm danh bằng việc định vị

....

Điểm danh thường trải qua việc tính toán cần sự chính xác để có được sự công bằng để có được sự công nhận xác đáng cho mỗi học viên, cán bộ giảng dạy ảnh hưởng trực tiếp đến quyền lợi của mỗi cá nhân.

Với mỗi hình thức điểm danh thì sẽ có những tiêu chí chung và cũng có những tiêu chí riêng mang đặc trưng của hình thức điểm danh đó. Nhưng xét thấy hình thức thứ nhất chưa hiệu quả còn mang tính đối phó, các hình thức thứ hai, ba rất hiệu quả nhưng tốn kém trong việc triển khai, rất ít cơ sở có thể đầu tư để thực hiện công việc này. ***Hình thức cuối cùng chính là hình thức sẽ được chúng tôi đề cập trong đồ án này cũng là hình thức giải quyết được các vấn đề mà các hình thức được đề cập ở trên chưa giải quyết được.***

**Điểm danh theo vị trí hiện tại**: dựa trên vị trí hiện tại của người dùng.

Các thông số cần thiết trong điểm danh theo vị trí:

* Vị trí hiện tại của người dùng
* Khoảng thời gian cho phép điểm danh đổi với học viên, cán bộ
* Thời gian bắt đầu và kết thúc mỗi tiết học
* Vì nếu chỉ lấy vị trí hiện tại xẩy ra nhiều trường hợp giả lập GPS nên bổ sung thêm địa chỉ MAC của wifi gần nhất, quét QR code sinh ra sau khi cán bộ giảng dạy điểm danh

## Mục tiêu và phạm vi đề tài

Dựa trên những vấn đề được nêu ra ở trên, mục tiêu của đề tài sẽ là:

* ***Về lý thuyết:***
* Nắm được các kiến thức về cơ sở dữ liệu, phân tích thiết kế hệ thống, quy trình xây dựng hệ thống, các kiến thức về lập trình trên môi trường Web.
* Tìm hiểu, nắm được các quy trình, nghiệp vụ trong việc điểm danh dựa trên vị trí hiện tại.
* Làm quen, sử dụng với công cụ lập trình Netbeans, Eclipse trong lập trình Java, làm quen và sử dụng hệ quản trị cơ sở dữ liệu MySQL để thao tác trên cơ sở dữ liệu.
* Làm quen với các công nghệ như: Spring, Ajax, JSP, React Native, JQUERY, BOOSTRAP… sử dụng các bộ công cụ hỗ trợ lập trình.
* ***Về sản phẩm:***
* Đưa ra bản phân tích thiết kế chi tiết cho hệ thống điểm danh trong nhà trường
* Tiến hành quá trình xây dựng hệ thống với các chức năng cơ bản giúp hệ thống có thể đưa vào sử dụng.

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 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.

## Định hướng giải pháp

Sử dụng các kiến thức đã được học và dựa trên các tài liệu thu thập được qua quá trình khảo sát để xây dựng bản phân tích tiết kế cho hệ thống hướng theo mô hình hướng đối tượng sử dụng UML

Xây dựng hệ thống theo mô hình Client – Server dựa trên 2 nền tảng Web, App:

• Đáp ứng được nhu cầu nhiều người sử dụng cùng lúc.

• Người dùng thuận tiện trong việc tiếp cận và thao tác trên hệ thống.

• Việc thay đổi và nâng cấp hệ thống dễ dàng do cơ sở dữ liệu và chương trình đều nằm trên máy chủ.

## 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 đồ án tốt nghiệp: Chương \*\*\* trình bày đóng góp chính của đồ án, đó là một nền tảng ABC cho phép khai phá và tích hợp nhiều nguồn dữ liệu, trong đó mỗi nguồn dữ liệu lại có định dạng đặc thù riêng. Nền tảng ABC được phát triển dựa trên khái niệm DEF, là các module ngữ nghĩa trợ giúp người dùng tìm kiếm, tích hợp và hiển thị trực quan dữ liệu theo mô hình cộng tác và mô hình phân tán.

**Chú ý:** Trong phần nội dung chính, mỗi chương của đồ án nên có phần *Tổng quan* và *Kết chương*. Hai phần này đều có định dạng văn bản “Normal”, sinh viên không cần tạo định dạng riêng, ví dụ như không in đậm/in nghiêng, không đóng khung, v.v.

Trong phần *Tổng quan* của chương N, sinh viên nên có sự liên kết với chương N-1 rồi trình bày sơ qua lý do có mặt của chương N và sự cần thiết của chương này trong đồ án. Sau đó giới thiệu những vấn đề sẽ trình bày trong chương này là gì, trong các đề mục lớn nào.

Ví dụ về phần *Tổng quan*: Chương 3 đã thảo luận về nguồn gốc ra đời, cơ sở lý thuyết và các nhiệm vụ chính của bài toán tích hợp dữ liệu. Chương 4 này sẽ trình bày chi tiết các công cụ tích hợp dữ liệu theo hướng tiếp cận “mashup”. Với mục đích và phạm vi của đề tài, sáu nhóm công cụ tích hợp dữ liệu chính được trình bày bao gồm: (i) nhóm công cụ ABC trong phần 4.1, (ii) nhóm công cụ DEF trong phần 4.2, nhóm công cụ GHK trong phần 4.3, v.v.

Trong phần *Kết chương*, sinh viên đưa ra một số kết luận quan trọng của chương. Những vấn đề mở ra trong *Tổng quan* cần được tóm tắt lại nội dung và cách giải quyết/thực hiện như thế nào. Sinh viên lưu ý không viết *Kết chương* giống hệt *Tổng quan*. Sau khi đọc phần *Kết chương*, người đọc sẽ nắm được sơ bộ nội dung và giải pháp cho các vấn đề đã trình bày trong chương. Trong *Kết chương*, Sinh viên nên có thêm câu liên kết tới chương tiếp theo.

Ví dụ về phần *Kết chương*: Chương này đã phân tích chi tiết sáu nhóm công cụ tích hợp dữ liệu. Nhóm công cụ ABC và DEF thích hợp với những bài toán tích hợp dữ liệu phạm vi nhỏ. Trong khi đó, nhóm công cụ GHK lại chứng tỏ thế mạnh của mình với những bài toán cần độ chính xác cao, v.v. Từ kết quả nghiên cứu và phân tích về sáu nhóm công cụ tích hợp dữ liệu này, tôi đã thực hiện phát triển phần mềm tự động bóc tách và tích hợp dữ liệu sử dụng nhóm công cụ GHK. Phần này được trình bày trong chương tiếp theo – Chương 5.

# ­­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

Hiện nay điểm danh đang là vấn đề nhức nhối đối với nhiều cơ sở giáo dục đặc biệt là giáo dục đại học. Cán bộ quản lý rất cần một công cụ để có thể đánh giá khách quan tình hình tham gia lớp học của học viên cũng như cán bộ giảng dạy qua đó có thể có được phản ánh chính xác nhất về tình hình học tập, giảng dạy của từng cá nhân.

Việc điểm danh miệng truyền thống không đảm bảo được tính chất khách quan cũng như có phần lạc hậu so với xã hội hiện tại. Những công nghệ điểm danh tiên tiến như quét vân tay, nhận dạng khuôn mặt… tốn quá nhiều kinh phí để lắp đặt và vận hành xem ra chưa phù hợp với các cơ sở giáo dục tại Việt Nam.

Tại Việt Nam, với tỉ lệ người sử dụng smart phone rất cao đặc biệt là sinh viên hầu như trên tay ai cũng có một chiếc smart phone thời thượng thì sẽ thật tuyệt vời nếu tận dụng được tài nguyên có sẵn này vào việc điểm danh. Thật vậy giải pháp điểm danh với chỉ một chiếc smart phone được nêu chi tiết dưới đây thật sự đã giải quyết được những thực trạng khó khăn được nêu ở trên. Đó là một bước cải tiến rất lớn trong việc điểm danh truyền thống cũng như tiết kiệm tối đa chi phí vận hành những hệ thống tiên tiến nhất nhưng hiệu quả cũng chẳng hề kém cạnh.

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.

## Tổng quan chức năng

### Biểu đồ use case tổng quan

A picture containing text, map

Description automatically generated

**Hình 1: Biểu đồ ues case tổng quan của hệ thống**

Hệ thống có 3 tác nhân tham gia là sinh viên, giảng viên và người quản trị. Trong đó, sinh viên là bên sử dụng hệ thống để xem thông tin thời khóa biểu và điểm danh; giảng viên sử dụng hệ thống để kích hoạt việc điểm danh và có thể thực hiện những điểm danh đặc biệt khi cần thiết; người quản trị sử dụng hệ thống để quản lí thông tin của các bên tham gia và các thành phần khác (như lớp học, phòng học, …). Giảng viên và người quản trị đều có thể yêu cầu hệ thống xuất báo cáo tổng hợp liên quan đến tình hình điểm danh của lớp.

### Biểu đồ use case phân rã “Rollcall”

A picture containing text, map

Description automatically generated

**Hình 2: Biểu đồ use case phân rã “Rollcall”**

Giảng viên thực hiện điểm danh để kích hoạt quy trình điểm danh, sau đó sinh viên sẽ điểm danh trong 1 khung thời gian nhất định. Trong một số trường hợp đặc biệt, giảng viên sẽ điểm danh cho sinh viên nếu chấp nhận lí do của sinh viên.

### Biểu đồ use case phân rã “Export report”

A picture containing text, map

Description automatically generated

**Hình 3: Biểu đồ use case phân rã “Export report”**

Giảng viên và người quản trị có thể yêu cầu hệ thống xuất ra các báo cáo tổng hợp về tình hình điểm danh của học sinh, giảng viên hoặc lớp học.

### Quy trình nghiệp vụ

*Quy trình nghiệp vụ điểm danh:*

A screenshot of a cell phone

Description automatically generated

**Hình 4: Quy trình nghiệp vụ điểm danh**

Nếu sản phẩm/hệ thống cần xây dựng có quy trình nghiệp vụ quan trọng/đáng chú ý, sinh viên cần mô tả và vẽ biểu đồ hoạt động minh họa quy trình nghiệp vụ đó. Sinh viên lưu ý đây không phải là luồng sự kiện của *từng use case*, mà là luồng hoạt động kết hợp nhiều use case để thực hiện một nghiệp vụ nào đó.

Ví dụ, một hệ thống quản lý thư viện có quy trình nghiệp vụ mượn trả với mô tả sơ bộ như sau: Sinh viên làm thẻ mượn, sau đó sinh viên đăng ký mượn sách, thủ thư cho mượn, và cuối cùng sinh viên trả lại sách cho thư viện. Một hệ thống có thể có một vài quy trình nghiệp vụ quan trọng như vậy.

## 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.

### 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.

# Công nghệ sử dụng

## React-native

React Native là công nghệ được tạo bởi Facebook, cho phép các dev sử dụng JavaScript để làm mobile apps trên cả Android và iOS với cảm nhận và giao diện native.

Chúng ta đã biết tới những framework như PhoneGap, hỗ trợ xây dựng những ứng dụng di động bằng bao nội dung web vào trong WebView, với phương châm là "Viết một lần, chạy mọi nơi" (Write once, run everywhere). Tuy nhiên những framework đó bộc lộ nhiều nhược điểm về hiệu năng cũng như trải nghiệm không hoàn toàn "native", vì vậy mà các lập trình viên vẫn thường ưa chuộng viết native app hơn.

React Native khác so với những framework trên, nó sử các Javascript component được hỗ trợ bởi các native component của IOS, Android vì vậy mà app bạn tạo nên là hoàn toàn native.

React Native không phải là một framework "Viết một lần, chạy mọi nơi". Bạn xây dựng UI bằng những component dành cho một nền tảng nhất định, vì vậy bạn không thể mang code đã viết cho iOS sang Android để chạy. Cái mà React Native làm là giúp bạn học được những kiến thức để phát triển ứng dụng trên đa nền tảng, còn được gọi là "Học một lần, viết mọi nơi" (Learn once, write everywhere).

## Redux

Quá trình hoạt động của ứng dụng web bất kỳ bị phụ thuộc khá nhiều yếu tố: dữ liệu từ máy chủ ban đầu, thao tác của người dùng (nhập dữ liệu, click menu, button…), dữ liệu cập nhật từ máy chủ, dữ liệu được tính toán trong ứng dụng, những yếu tố này còn gọi là nguồn dữ liệu. Những nguồn dữ liệu này đến từ những nơi khác nhau, bất kỳ khi nào, khiến cho ứng dụng rất khó kiểm soát, chúng tác động đến những thành phần đơn lẻ, hoặc nhiều thành phần trên ứng dụng, hoặc hiệu ứng dây chuyền. Chính vì sự phức tạp đó, là vấn đề mà Redux muốn giải quyết, tất cả các nguồn dữ liệu cần phải được quản lý và tạo thành một nguồn duy nhất, tin cậy.

Đối với ứng dụng React, việc truyền các thuộc tính từ component cha xuống component con và từ component con gọi lên component cha đôi khi là một ác mộng nếu như ứng dụng có quá nhiều tầng component. Với ứng dụng nhỏ thì vẫn có thể quản lý được nhưng nếu ứng dụng ngày càng lớn thì sẽ rất khó để quản lý. Với redux thì ta sẽ không phải lo về vấn đề đó nữa. Redux sẽ giải quyết việc này, ta có thể lấy về state của toàn bộ ứng dụng từ bất cứ component nào bằng cách sử dụng hàm *mapStateToProps* và chỉ định những state muốn lấy về.

Ví dụ sau khi học viên đăng nhập thông tin về phòng, lớp học của thời khóa biểu sẽ được tải về, nếu các component sau muốn sử dụng thì từ component xem thời khóa biểu phải truyền giá trị từ component thời khóa biểu sang, điều đấy sẽ tốn thời gian xử lí những việc không cần thiết*.* Redux đã giúp giải quyết việc đó bằng cách lưu hết trạng thái các câu trả lời vào một bể chứa được gọi là *store,* khi người dùng cập nhật câu trả lời thì những thay đổi đó sẽ được truyền vào bên trong *store, Các Component* khác bây giờ muốn lấy kết quả chỉ cần gọi trực tiếp bên trong *store* thông qua hàm *mapStateToProps.*

Đôi khi chúng ta muốn thay đổi state của các component khác mà không phải component cha hay các component anh em thì chúng ta sẽ phải gọi đến component ở mức độ cao nhất để thay đổi state. Điều đó khá là tốn công và không hợp lý cho lắm, nó sẽ dẫn đến việc khi ứng dụng lớn dần thì component level cao sẽ phải xử lý rất nhiều và rất phức tạp. Việc các component level cao phải xử lý rất nhiều logic cũng sẽ dẫn đến việc scale dự án trở nên khó khăn hơn. Với redux thì khác, xử lý logic về thay đổi state sẽ được thực hiện ở trong reducer, các hành động của người dùng sẽ được định nghĩa trong action, nó làm ứng dụng của chúng ta rất rõ ràng về mặt logic vì thế việc mở rộng hệ thống cũng sẽ dễ dàng hơn rất nhiều.

## Redux-saga

Redux Saga là một thư viện middleware trong React. (nằm giữa dispatch action và thay đổi state). Nhằm xử lý bất đồng bộ của việc lưu state trong react.



Trong ứng dụng này, chúng ta sử dụng Saga cho việc Fetch API để lấy dữ liệu về.

## Spring framework

Spring được phát minh bởi Rod Johnson. Nó được giới thiệu lần đầu tiên trong cuốn Expert One-on-One J2EE Design and Development, vào năm 2002. Trong đó, Rod giải thích các kinh nghiệm làm việc với J2EE và làm thế nào EJB mang đến thành công cho các project. Rod tin rằng một framework lightweight như JavaBean thì phù hợp cho nhu cầu của developer. Framework được mô tả đó được gọi Spring khi nó được đưa lên SourceForge vào tháng 2-2003. Vào lúc đó, Juergen Hoeller đã tham gia với Rod để cùng phát triển Spring, họ trở thành cánh tay phải của Spring. Sau đó họ cộng tác thêm một vài developer nữa. Gần đây, Rod và Juergen đã viết một cuốn sách có tên là:“Expert One-on-One J2EE Development without EJB” để mô tả Spring giải quyết nhiều vấn đề của J2EE như thế nào.

Cơ sở kiến trúc của Spring được phát triển bởi Rod vào năm 2000 (trước Struts). Cơ sở này được xây dựng từ kinh nghiệm xây dựng cơ sở hạ tầng trong một số project của Rod.

Spring là một J2EE application framework được cung cấp để quản lý các đối tượng nghiệp vụ và sự phụ thuộc giữa chúng. Ví dụ, với IoC (Inversion of Control), nó cho phép bạn xác định DAO (Data Access Object) phụ thuộc vào DataSource. Nó cũng cho phép developer viết các giao tiếp và đơn giản định nghĩa cài đặt trong XML file. Spring chứa nhiều lớp hỗ trợ các framework khác (như Hibernate và Struts) để tạo sự tích hợp dễ dàng hơn.

Áp dụng J2EE Design Pattern thì cồng kềnh và không cần thiết ở một số trường hợp. Spring thì giống design pattern nhưng mọi thứ thì đơn giản hơn. Ví dụ: thay vì viết ServerLocator để tìm Hibernate Session, bạn có thể cấu hình SessionFactory trong Spring. Điều này giúp bạn tập trung vào kỹ thuật (best practices) nhiều hơn là hình dung vào mẫu gần nhất.

Các module của spring trong dự án này bao gồm:

### Spring Boot

Spring Boot được xây dựng trên nền tảng Spring nhưng đã được giảm tải các cấu hinh(configuration) cho quá trình phát triển được nhanh chóng hơn bao giờ hết.

Spring Boot cung cấp mặc định các đoạn code và annotation configuration để phát triển dự án Spring trong thời gian ngắn.

Những lợi thế của Spring Boot:

* Nó dễ dàng cho việc phát triển ứng dụng dựa trên Spring với Java hoặc Groovy
* Nó giảm thiểu thời gian phát triển, tăng năng suất phát triển
* Tránh việc phải viết nhiều bản mẫu code, cấu hình Annotaion hoặc XML
* Dễ dàng trong việc tích hợp với hệ sinh thái của Spring như: Spring JDBC, Spring ORM, Spring Data, Spring Security...
* Nó theo cách tiếp cận "Opinionated Defaults Configuration" để giảm effort trong quá trình phát triển.
* Nó cung cấp các Embedded HTTP servers như Tomcat, Jetty... để phát triển và test một cách dễ dàng,
* Nó cung cấp công cụ CLI (command Line Interface) cho việc phát triển và test ứng dụng nhanh chóng và dễ dàng từ command line.
* Nó cung cấp rất nhiều các plugins để phát triển và test các ứng dụng Spring Boot nhanh chóng sử dụng các công cụ Build như Maven và Gradle
* Cung cấp nhiều plugins để làm việc với các embedded and in-memory databases

### Spring MVC

Spring có xây dựng một cơ chế có tên Spring MVC mà ở đó có các API cho phép việc xây dựng ứng dụng web được dễ dàng hơn và có quy tắc hơn. Quy tắc hơn thể hiện ở chỗ mọi thành phần được tạo ra, cài đặt và vận hành tuân theo một chuẩn thiết kế thống nhất, quy tắc đó là được gọi là MVC.

MVC lần lượt là ba chữ cái đầu tiên của ba từ Model, View và Controller. MVC là một mô hình ứng dụng mà ở đó các thành phần được phân tách ra thành các lớp riêng biệt với các nhiệm vụ đặc trưng. View sẽ là lớp cho các thành phần có chức năng hiển thị, giao tiếp trực tiếp với người dùng. Nhiệm vụ của các thành phần trong View là trình bày các dữ liệu từ Model đến người dùng cuối. Model là các thành phần có khả năng lưu trữ và vận chuyển thông tin. Quá trình ném dữ liệu vào Model sẽ được thực hiện bởi Controller. Controller là các thành phần giúp cho việc xử lý logic các thao tác nghiệp vụ. Nhiệm vụ của Controller là lấy dữ liệu từ Model, xử lý dữ liệu, và cập nhật lại dữ liệu vào Model.

Chương này có độ dài từ 10 đến 13 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).

# 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 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 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

## Sử dụng ngôn ngữ react-native trong xây dựng mobile app.

Đối với lập trình viên, đa phần chúng ta thường hay chọn lập trình web app thay vì mobile app. Với bản thân của chúng tôi cũng vậy, trước khi T.S Tạ Hải Tùng – người hướng dẫn trực tiếp đồ án này của chúng tôi đưa ra đề tài này, quả thật ban đầu chúng tôi rất ngại vì phải lập trình mobile app thật sự không quen thuộc đối với cả hai. Chúng tôi nhận thấy đa phần chúng ta thường ngại lập trình mobile app do lập trình mobile app rất khó và tốn thời gian là vì thực tế chúng ta cần tìm hiểu 2 hệ sinh thái hoàn toàn khác biệt. Nếu muốn lập trình app iOS, chúng ta phải học Swift hoặc Objective-C và Cocoa Pods. Nếu muốn lập trình app Android, chúng ta lại cần học Java và Android SDK. Việc học từng ngôn ngữ đó sẽ tốn của chúng ta rất nhiều thời gian. Điều tương tự cũng xảy ra với các frameworks: Cocoa Touch và Android SDK.

Nhưng các cụ ta vẫn thường nói “cái khó ló cái khôn”, quả đúng với chúng tôi trong hoàn cảnh này. Với một ít kiến thức js đã được học từ trước, khi bắt đầu xây dựng mobile app chúng tôi đã tìm tòi và lựa chọn lập trình trên React Native. Quả thật, phần lớn thời gian chúng tôi chỉ cần học 1 bộ công cụ. Có rất nhiều thứ để làm quen như: JavaScript, Node, React Native… nhưng chỉ có 1 công cụ duy nhất để học. Việc này thật sự tốn của chúng tôi ít thời gian hơn rất rất nhiều vì trước đấy chúng tôi chưa biết nhiều về Android Studio, Swift… Vậy nên chưa nói đến việc xây dựng mobile app đa nền tảng thì nếu chỉ phát triển trên duy nhất một nền tảng thì đã ngốn kha khá thời gian quý báu rồi.

Một điều nữa khiến chúng tôi khá tâm đắc nữa đó là khả năng sử dụng lại của React-native, tính năng “có thể chia sẻ code” này có rất nhiều lợi điểm như: tận dụng nguồn nhân lực tốt hơn, duy trì ít code hơn, ít bugs hơn, các tính năng trong cả 2 platforms cũng tương tự nhau…

Điều cuối cùng mà cũng là điều mà chúng tôi đã đề cập ở trên khiến không chỉ chúng tôi mà tất cả công đồng anh em lập trình viên khi lựa chọn công nghệ này đó là việc học 1 lần nhưng sử dụng được mọi platform. Từ khi bắt tay vào việc xây dựng mobile app bằng công nghệ này chúng tôi thấy rằng bởi vì tất cả code của Android và iOS sử dụng cùng bộ công cụ, nên ý tưởng có 1 team dev làm app cho cả 2 platform là thực hiện được – 1 điều ít khi xảy ra khi có rất ít dev lập trình cả 2 platform iOS và Android. Thậm chí, chúng tôi còn cho rằng team đang lập trình web app sử dụng React.js sẽ không phải cực khổ nữa khi học lập trình React Native và bắt đầu làm mobile app. Mặc dù phải công nhận đối với những ai chưa quen với lập trình js đặc biệt là reactjs thì khi bắt tay vào xây dựng không tránh khỏi việc vật lộn với ngôn ngữ mà chúng tôi vẫn hay nói vui với nhau là “God of bugs” này, nhưng chung quy lại chúng tôi vẫn cho rằng thật may mắn khi chọn công nghệ này làm công nghệ chính trong việc xây dựng mobile app điểm danh trực tuyến.

## Xử lý những hạn chế trong việc điểm danh với chỉ GPS.

Từ khi nhận được đề tài xây dựng hệ thống điểm danh trực tuyến bằng GPS cả hai chúng tôi đã ngồi lại teamwork với nhau rất nhiều về những case có thể khiến cho việc điểm danh trở nên quá lỏng lẻo. Nhận thấy việc nếu chỉ sử dụng GPS để điểm danh có thể khiến cho người dung có thể giả lập được GPS (ví dụ như việc rất nhiều người chơi pokemon go có thể giả lập được vị trí dẫn đến việc ngồi ở nhà vẫn có thể săn pokemon tận nước Mỹ xa xôi) và điểm danh một cách ngon lành gây mất công bằng và thật khó để chấp nhận.

Cả hai luôn đau đáu tìm cách giải quyết bài toán khó giải này và đây cũng là nguồn cơn của việc chúng tôi nhận ra cần thêm một thứ gì đấy đủ sức nặng để có thể siết chặt được sự lỏng lẻo mà GPS mang lại. Từ đó ý tưởng lấy địa chỉ MAC wifi (mã duy nhất được gán bởi nhà sản xuất cho từng phần cứng mạng), QR code (mã phản hồi nhanh) được hình thành. Ý tưởng lấy địa chỉ MAC wifi được xuất phát từ suy nghĩ việc wifi phủ song gần như toàn trường, thậm chí cán bộ giảng dạy có thể phát sóng wifi ngay trên chiếc smartphone của mình, còn QR code là ý tưởng làm cho sự ràng buộc, chặt chẽ của việc điểm danh được tăng cao hơn. Hai ý tưởng này quả thật đã khiến cho việc làm giả GPS gần như bất khả thi. Vì ngoài việc học viên gửi vị trí hiện tại lên server, giờ đây còn thêm việc gửi đúng chuỗi String trong Qr code hay việc validate đúng địa chỉ MAC wifi của lớp học.

Chưa hết, chúng tôi còn nhận thấy rằng hai ràng buộc trên vẫn là chưa đủ vì một học viên có thể sử dụng cùng một điện thoại để điểm danh cho những người bạn của mình khiến công sức thực hiện hai ràng buộc trên đổ xuống sông xuống biển. Vậy là ý tưởng lấy IMEI (Mã số nhận dạng di động quốc tế) được sử dụng. Đây là giá trị độc nhất mà mỗi điện thoại đều khác nhau nên nếu sinh viên điểm danh hộ bạn mình thì sẽ không được phép. Và phòng trường hợp thay đổi điện thoại, chúng tôi cho phép sinh viên được cập nhật giá trị này tối đa 3 lần trong mỗi kì học.

Quả thật với ba ràng buộc được nêu ra ở trên kết hợp với GPS đã khiến cho hệ thống vận hành trở nên công bằng và đáp ứng hiệu quả đối với yêu cầu bài toán.

## Xuất báo cáo điểm danh.

Quả thật đây là phần rất mới đối với cả hai chúng tôi và quả thật mất rất nhiều thời gian để xuất ra được báo cáo xlsx, pdf cuối cùng.

## REST API

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

### Kết quả đạt được

**Về mặt lý thuyết:**

* Nắm được các kiến thức về cơ sở dữ liệu, phân tích thiết kế hệ thống, quy trình xây dựng hệ thống, các kiến thức về lập trình trên môi trường Web, mobile app.
* Tìm hiểu, nắm được các quy trình, nghiệp vụ trong hệ thống điểm danh trực tuyến sử dụng GPS.
* Làm quen, sử dụng với công cụ lập trình Netbeans, Eclipse … trong lập trình java.
* Làm quen với các công nghệ như: React-native, Ajax, … sử dụng các bộ công cụ hỗ trợ lập trình.

**Về mặt sản phẩm:**

* Đưa ra được bản phân tích thiết kế chi tiết cho hệ thống điểm danh trực tuyến. Bản phân tích thiết kế hệ thống dựa trên yêu cầu đặt ra từ yêu cầu của người dùng.
* Tiến hành quá trình cài đặt và xây dựng hệ thống với các chức năng cơ bản giúp hệ thống có thể đưa vào vận hành. Các chức năng xây dựng được:
* Quản trị hệ thống
  + Quản trị lớp học.
  + Quản trị cán bộ giảng dạy.
  + Quản trị tình hình tham gia lớp học của cán bộ giảng dạy, học viên bằng cách xuất báo cáo
* Cán bộ giảng dạy
  + Xem lịch giảng dạy.
  + Điểm danh
  + Điểm danh cho sinh viên với những trường hợp đặc biệt
  + Xuất báo cáo
* Sinh viên
  + Xem thời khóa biểu.
  + Điểm danh
* Đảm bảo tính ổn định của hệ thống khi đưa vào hoạt động thực tế.

Dù rất cố gắng nhưng trong khoảng thời gian không dài, khối lượng công việc không phải là nhỏ nên hệ thống được xây dựng vẫn còn tồn tại những thiếu sót như màu sắc hiển thị còn đơn điệu, một số chức năng cần có sự hướng dẫn chi tiết cho người dùng thì việc sử dụng không dẫn tới sai sót cho hệ thống.

## Hướng phát triển

Hệ thống xây dựng với mục đích hỗ trợ các cơ sở giáo dục đặc biệt là giáo dục đại học trong việc điểm đanh để xác nhận thời gian tham gia lớp học của học viên, cán bộ giảng dạy. Với việc về cơ bản đã xây dựng được các chức năng để có thể đáp ứng được việc điểm danh cũng như xuất báo cáo đối với từng đối tượng người sử dụng.

Trong thời gian tới hệ thống sẽ được chỉnh sửa và hoàn thiện các chức năng, đưa các chức năng theo quy trình điện tử vào sử dụng để kiểm nghiệm, sửa đổi bổ sung cho giao diện ngày càng phù hợp và tiện lợi hơn cho người dùng. Cùng với đó là việc bổ sung các tiện ích theo yêu cầu của người dùng.

Được kỳ vọng sẽ là giải quyết được những vấn đề còn tồn tại trong việc điểm danh truyền thống cũng như điểm danh bằng các phương pháp tiên tiến. Hệ thống điểm danh bằng GPS hi vọng có thể phát triển các chức năng và có khả năng thương mại hóa.

# 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.

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, nhỏ hơn một chút so với font thông thường (13pt) của ĐATN. 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”.



**Hình 3** Ví dụ hình vẽ

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.

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.

Sinh viên chỉ cần in bìa giấy cứng mà không cần đóng quyển để tiết kiệm chi phí và giúp việc vận chuyển ĐATN dễ dàng hơn.

Quyển ĐATN nên được in một mặt trên các trang giấy A4.

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)