

HƯỚNG DẪN TRÌNH BÀY BÁO CÁO

BÀI TẬP TIỂU LUẬN, ĐỒ ÁN, DỰ ÁN, KHÓA LUẬN

(Áp dụng cho các nhóm sinh viên thực hiện đề tài do thầy **Mai Văn Mạnh** hướng dẫn)

Cập nhật lần cuối: 25/08/2024

Tùy theo đặc thù của từng đề tài và yêu cầu từ giảng viên hướng dẫn, cách viết báo cáo có thể khác nhau. Tuy nhiên, mục tiêu chung của mọi báo cáo là cung cấp một cái nhìn toàn diện về quá trình nghiên cứu và triển khai đề tài của nhóm. Nội dung báo cáo cần bao gồm: phần giới thiệu khái quát về đề tài; lý do lựa chọn đề tài và ý nghĩa thực tiễn; kiến thức nền tảng và tài liệu tham khảo liên quan đến việc thực hiện; các khái niệm, lý thuyết mà nhóm đã nghiên cứu; cùng với đó là thiết kế hệ thống, sơ đồ, bảng biểu, hình ảnh minh họa cho quá trình thực hiện. Phần báo cáo cũng cần trình bày các đề xuất của nhóm, các đóng góp chính, hoặc những tính năng nổi bật của website/phần mềm, kết quả đạt được, và có thể so sánh với những kết quả từ các nghiên cứu/sản phẩm trước đó. Cuối cùng, báo cáo nên kết thúc với phần kết luận, tài liệu tham khảo và phụ lục bổ sung (nếu có).

Mục lục gợi ý dành cho các đề tài dự án liên quan đến phát triển phần mềm

- Các trang bìa và phụ lục theo quy định
- Trang tóm tắt (Abstract)
- Chương 1 – Giới thiệu (Introduction)
- Chương 2 – Cơ sở lý thuyết (Literature Review)
- Chương 3 – Phân tích yêu cầu (Requirements Analysis)
- Chương 4 – Thiết kế hệ thống (System Design)
- Chương 5 – Phát triển và Kiểm thử (Development and Testing)
- Chương 6 – Triển khai và Đánh giá (Deployment and Evaluation)
- Chương 7 – Kết luận (Conclusion)
- Tài liệu tham khảo, và các trang phụ lục theo yêu cầu

Nếu ở quy mô nhỏ (tiểu luận) thì một số chương có thể được gộp chung với nhau, ví dụ chương 3 và chương 4 (thậm chí cả chương 5) có thể được gộp chung với nhau thành một chương, tùy vào độ dài nội dung của báo cáo. Trong trường hợp cần thiết, chương 5 có thể được tách riêng ra thành các chương nhỏ hơn để nội dung có thể được triển khai một cách cụ thể, rõ ràng hơn.

Dưới đây là giải thích chi tiết hơn về mục lục của báo cáo, và các nội dung cần được đề cập trong báo cáo:

1. Các trang thông tin cần thiết theo quy định ở biểu mẫu của Khoa/Trường.
2. Các trang mục lục nội dung, mục lục ảnh/bảng/công thức (nếu có).

3. Trang **Tóm tắt (Abstract)**: Mô tả ngắn gọn về sản phẩm, lý do phát triển, và vấn đề mà sản phẩm giải quyết. Ví dụ nếu sản phẩm là một ứng dụng di động, tóm tắt sẽ mô tả chức năng chính của ứng dụng (ví dụ ứng dụng cho phép người dùng đăng tải và chia sẻ hình ảnh), cùng với công nghệ nổi bật (ví dụ: sử dụng AI để gợi ý tìm kiếm hoặc phân loại hình ảnh). Tóm lược kết quả và đóng góp của đề tài. Trang tóm tắt này không nên viết quá một trang A4.

CHƯƠNG 1: GIỚI THIỆU (INTRODUCTION)

1.1 Bối cảnh và lý do thực hiện đề tài: Giới thiệu về vấn đề mà nhóm muốn giải quyết. Ví dụ: “Hiện nay, nhu cầu học ngoại ngữ trực tuyến ngày càng tăng, nhưng các ứng dụng hiện tại chưa tối ưu về tính cá nhân hóa nội dung học tập.”

1.2 Mục tiêu của đề tài: Trình bày rõ ràng mục tiêu cụ thể mà sản phẩm hướng tới. Ví dụ: “Xây dựng một ứng dụng di động giúp người học luyện nghe và nói tiếng Anh với phản hồi từ hệ thống trí tuệ nhân tạo (AI).”

1.3 Phạm vi của đề tài: Giới hạn phạm vi của sản phẩm. Ví dụ: “Ứng dụng này chỉ hỗ trợ người dùng Android, chưa hỗ trợ iOS.”

1.4 Phương pháp nghiên cứu: trình bày các phương pháp nhóm đã sử dụng để nghiên cứu và phát triển sản phẩm. Điều này có thể bao gồm cả phương pháp nghiên cứu tài liệu, khảo sát người dùng, và các phương pháp kỹ thuật như phát triển Agile hoặc Waterfall.

1.5 Ý nghĩa thực tiễn của đề tài: Mô tả những lợi ích và giá trị mà sản phẩm của nhóm có thể mang lại cho xã hội hoặc một nhóm đối tượng cụ thể. Điều này có thể bao gồm việc giải quyết các vấn đề hiện hữu hoặc cải thiện một quy trình nhất định. Ví dụ *"Ứng dụng này có ý nghĩa thực tiễn lớn trong việc hỗ trợ người học tiếng Anh nâng cao kỹ năng nghe và nói thông qua công nghệ nhận diện giọng nói AI. Bên cạnh đó, ứng dụng còn giúp giáo viên theo dõi tiến độ học tập của học viên một cách dễ dàng và có thể điều chỉnh phương pháp giảng dạy dựa trên dữ liệu thu thập được."*

1.6 Cấu trúc của đề tài: Nêu tổng quát nội dung các chương. Ví dụ: “Chương 2 phân tích yêu cầu hệ thống, chương 3 trình bày thiết kế hệ thống, chương 4 mô tả quá trình triển khai,...”

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT (LITERATURE REVIEW)

2.1 Các sản phẩm/công trình tương tự (related works)

Mục tiêu của phần này là để đánh giá các sản phẩm hiện có trên thị trường, xem xét các tính năng, ưu nhược điểm của những sản phẩm đó so với sản phẩm nhóm đang phát triển. Mỗi sản phẩm tương tự nên được trình bày rõ ràng với các mục chính như tên sản phẩm, nhà phát triển,

tính năng chính, công nghệ sử dụng, ưu điểm và nhược điểm. Đưa ra nhận xét về những điểm hạn chế mà sản phẩm của nhóm có thể khắc phục hoặc cải thiện.

2.2 Các công nghệ liên quan: Mô tả chi tiết các công nghệ được sử dụng trong sản phẩm của nhóm và tại sao nhóm chọn những công nghệ này so với các giải pháp khác.

- Giới thiệu tổng quan về công nghệ: Mô tả từng công nghệ một cách ngắn gọn, giải thích công nghệ đó là gì và chức năng của nó trong hệ thống của bạn. Một số công nghệ là cốt lõi trong đề tài thì nên mô tả kỹ hơn, chi tiết hơn về kiến trúc và nguyên lý hoạt động của nó. Nên có những diagram hoặc ảnh minh họa để làm rõ vấn đề và giúp người đọc dễ hình dung hơn (ví dụ ảnh minh họa so sánh nguyên lý hoạt động giữa docker và máy ảo truyền thống sử dụng thông qua hypervisor).
- Giải thích lý do nhóm chọn công nghệ này cho sản phẩm của mình, so sánh với các công nghệ khác để làm rõ sự ưu việt của nó.
- Cung cấp các ví dụ cụ thể về việc sử dụng công nghệ đó trong sản phẩm của nhóm.

2.3 Các nguyên lý và mô hình lý thuyết: trình bày các nguyên lý lý thuyết và mô hình phát triển phần mềm nhóm đã áp dụng trong quá trình thiết kế và triển khai sản phẩm. Phần này giúp làm rõ sự hiểu biết của nhóm về các khái niệm nền tảng và cách nhóm đã áp dụng chúng vào sản phẩm thực tế.

- Giải thích nguyên lý hoặc mô hình: Trình bày lý thuyết về các nguyên lý hoặc mô hình phát triển phần mềm mà nhóm sử dụng: ví dụ microservice, async communication, message broker, decoupling architecture, single page application, progressive web app...
- Ứng dụng vào sản phẩm: Cụ thể hóa việc sử dụng mô hình này trong sản phẩm của nhóm, đưa ra ví dụ thực tế.
- Với các mô hình, nguyên lý như vậy thì các nhóm nên có ảnh minh họa rõ ràng, dễ hiểu, cân nhắc tự vẽ lại thông qua các công cụ vẽ.

CHƯƠNG 3: PHÂN TÍCH YÊU CẦU (REQUIREMENTS ANALYSIS) (có thể gộp chung với chương 4 nếu nội dung hai chương không quá dài).

Chương này là bước quan trọng đầu tiên trong quá trình phát triển phần mềm, nơi các nhóm sẽ xác định những gì hệ thống cần thực hiện dựa trên nhu cầu của người dùng hoặc tổ chức. Chương này không chỉ giúp người đọc hiểu rõ về mục tiêu của hệ thống mà còn làm cơ sở cho việc thiết kế và triển khai sau này.

Nội dung của chương này cần làm rõ được những câu hỏi cơ bản như:

- Hệ thống sẽ phục vụ đối tượng nào?

- Hệ thống cần thực hiện những tác vụ gì?

3.1 Yêu cầu từ người dùng (User Requirements): Phân tích các yêu cầu của người dùng (end-user). Đây có thể là những người sử dụng hệ thống trực tiếp hoặc gián tiếp, như quản trị viên, khách hàng, nhân viên... Liệt kê các nhóm người dùng chính của hệ thống và vai trò của họ. Đôi khi có thể sử dụng sơ đồ stakeholder để minh họa các nhóm liên quan. Xác định các yêu cầu mà từng nhóm người dùng cần từ hệ thống. Ví dụ: Khách hàng cần đăng ký tài khoản, duyệt sản phẩm, thêm sản phẩm vào giỏ hàng và thanh toán (cần mô tả chi tiết hơn chứ không phải chỉ ghi ra vài dòng như vậy).

3.2 Yêu cầu chức năng: Chi tiết các chức năng mà sản phẩm sẽ thực hiện. Ví dụ: Đăng ký và đăng nhập người dùng, luyện nghe và phát âm với phản hồi tức thì, theo dõi tiến độ học tập.

3.2 Yêu cầu phi chức năng: Các yếu tố như hiệu năng, bảo mật, khả năng mở rộng. Ví dụ: “Ứng dụng phải đáp ứng ít nhất 1000 người dùng đồng thời và có thời gian phản hồi dưới 1 giây.”

3.3 Sơ đồ use case và các UML diagrams khác: Cần lưu ý với mỗi diagram khi đưa vào báo cáo, cần có sự đầu tư chỉnh chu, để đảm bảo ảnh/diagram có chữ rõ nét, không quá nhỏ, khi in ra có thể dễ đọc. Không vẽ/screenshot các nội dung dư thừa. Mỗi nội dung cần phải có thêm văn bản mô tả chi tiết, tránh trường hợp chỉ cung cấp ảnh/diagram và không giải thích gì thêm.

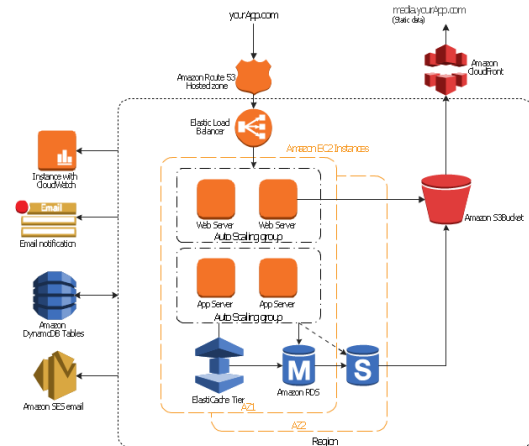
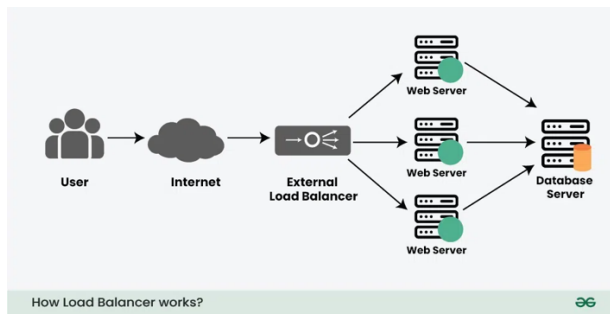
3.4 Một số nội dung khác: Các nhóm hoàn toàn có thể bổ sung thêm các nội dung khác liên quan đến quá trình phân tích yêu cầu đề tài của nhóm mình.

CHƯƠNG 4: THIẾT KẾ HỆ THỐNG (SYSTEM DESIGN)

Chương “Thiết kế hệ thống” đóng vai trò quan trọng trong việc xác định các thành phần và kiến trúc của hệ thống sau khi các yêu cầu đã được phân tích rõ ràng. Nó chuyển đổi các yêu cầu thành các mô hình kỹ thuật, mô tả cách hệ thống sẽ hoạt động, cách các thành phần giao tiếp với nhau và dữ liệu sẽ được quản lý như thế nào.

4.1 Kiến trúc hệ thống: Mô tả kiến trúc tổng thể của hệ thống, cách các thành phần chính của hệ thống được tổ chức và giao tiếp với nhau.

- Kiến trúc tổng quan: Trình bày các thành phần chính của hệ thống như Frontend, Backend, Cơ sở dữ liệu, API, và bất kỳ dịch vụ bên thứ ba nào được tích hợp.
- Kiến trúc Client-Server hoặc Microservices: Lựa chọn kiểu kiến trúc phù hợp và giải thích tại sao.
- Sơ đồ kiến trúc: Minh họa sơ đồ kiến trúc hệ thống để giúp người đọc hiểu rõ sự tương tác giữa các thành phần.



Trong chương này bắt buộc phải có một diagram mô tả kiến trúc hệ thống theo cách giống hai ví dụ bên trên, trong đó liệt kê các thành phần của hệ thống, cách mà chúng tương tác với nhau để giúp người đọc dễ dàng hình dung được tổng quan về hệ thống. Tất nhiên bên cạnh cung cấp diagram, báo cáo cũng phải trình bày để giải thích bằng lời về các thành phần có trong diagram và nguyên lý hoạt động của chúng.

4.2 Thiết kế cơ sở dữ liệu

Mô tả cách tổ chức dữ liệu trong hệ thống và các bảng, quan hệ giữa chúng. Giải thích lựa chọn loại cơ sở dữ liệu (quan hệ như MySQL, PostgreSQL hoặc NoSQL như MongoDB) và lý do chọn lựa. Mô tả các thực thể (entities) chính của hệ thống, các thuộc tính của chúng và mối quan hệ giữa các thực thể. Chi tiết về các bảng, các thuộc tính của bảng, các khóa chính, khóa ngoại và các ràng buộc. Sử dụng ERD (hoặc diagram phù hợp) để minh họa các thực thể và mối quan hệ. Cần lưu ý nếu diagram quá lớn phải chỉnh sao cho phù hợp, đảm bảo chữ không quá nhỏ và vẫn có tính thẩm mỹ. Ngược lại, nếu diagram quá lớn thì cần thiết kế sao cho khi đưa vào báo cáo chữ vẫn đủ lớn để có thể đọc được, tránh trường hợp scale toàn bộ diagram lại thành một hình rất nhỏ, không thể hoặc rất khó để đọc nội dung.

4.3 Thiết kế giao diện người dùng (User Interface Design) Trình bày cách hệ thống tương tác với người dùng thông qua giao diện, bao gồm cấu trúc màn hình, chức năng và sự tiện dụng (usability).

- **Bố cục giao diện:** Trình bày các màn hình chính của hệ thống (như trang chủ, trang quản lý tài khoản, trang sản phẩm).
- **Luồng người dùng (User Flow):** Mô tả cách người dùng sẽ tương tác với các chức năng của hệ thống.
- **Wireframes hoặc Mockups:** Cung cấp bản phác thảo giao diện để minh họa trực quan.

4.4 Thiết kế các API và dịch vụ (API and Service Design) Mô tả các API và dịch vụ mà hệ thống sẽ cung cấp để phục vụ các thành phần khác hoặc tương tác với dịch vụ bên ngoài.

- Thiết kế API: Mô tả các API mà hệ thống cung cấp, bao gồm các phương thức HTTP (GET, POST, PUT, DELETE) và các URL.
- Mô hình truyền dữ liệu: Trình bày cách dữ liệu được truyền tải qua API (JSON, XML).
- Tài liệu API: Cung cấp chi tiết về cách sử dụng API, các tham số đầu vào và đầu ra.

4.5 Thiết kế bảo mật (Security Design) Trình bày các biện pháp bảo mật được thiết kế để bảo vệ hệ thống khỏi các mối đe dọa như tấn công mạng và vi phạm dữ liệu.

Xác thực và phân quyền (Authentication and Authorization): Mô tả cách người dùng đăng nhập vào hệ thống và cách phân quyền truy cập.

- Mã hóa dữ liệu: Giải thích cách mã hóa dữ liệu nhạy cảm (ví dụ: thông tin thanh toán).
- Biện pháp phòng ngừa tấn công: Mô tả các biện pháp phòng chống tấn công như SQL Injection, Cross-Site Scripting (XSS), Cross-Site Request Forgery (CSRF).

4.6 Thiết kế hiệu năng và khả năng mở rộng (Performance and Scalability Design) Đảm bảo hệ thống có khả năng hoạt động hiệu quả khi số lượng người dùng tăng cao và có thể mở rộng khi cần thiết..

- Tối ưu hiệu năng: Mô tả các giải pháp như caching, load balancing để tăng tốc độ phản hồi.
- Khả năng mở rộng (Scalability): Giải thích cách hệ thống có thể mở rộng theo chiều ngang (thêm máy chủ) hoặc chiều dọc (nâng cấp máy chủ).
- Thiết kế chịu tải cao (High Availability Design): Đưa ra các biện pháp dự phòng để hệ thống không bị gián đoạn khi gặp sự cố.

CHƯƠNG 5 – PHÁT TRIỂN VÀ KIỂM THỬ HỆ THỐNG (DEPLOYMENT & TESTING)

Chương này tập trung vào cách thực hiện và cấu hình hệ thống dựa trên thiết kế đã đề xuất và mô tả về quá trình phát triển phần mềm của nhóm. Đây là phần rất quan trọng vì nó cho thấy cách chuyển đổi từ lý thuyết thiết kế thành thực tiễn triển khai. Có thể đề cập tới các nội dung như sau:

- Môi trường phát triển và công cụ: Trình bày các công cụ và môi trường phát triển đã sử dụng (IDE, hệ thống quản lý phiên bản Git).

- Ngôn ngữ lập trình và framework: Mô tả chi tiết về ngôn ngữ lập trình và các framework bạn chọn. Ví dụ: “Ứng dụng di động được phát triển bằng React Native, backend sử dụng Node.js và Express.”
- Cấu trúc mã nguồn: Giải thích sơ lược cấu trúc mã nguồn của dự án. Ví dụ: “Các thành phần frontend được chia thành các module riêng biệt để dễ quản lý.”
- Các thách thức trong quá trình phát triển: Trình bày những khó khăn gặp phải trong quá trình lập trình và cách khắc phục. Ví dụ: “Vấn đề về tối ưu hóa mô hình nhận diện giọng nói dẫn đến việc phải sử dụng TensorFlow Lite thay cho TensorFlow tiêu chuẩn.”
- Triển khai cơ sở dữ liệu (Database Deployment) Mô tả quy trình thiết lập và triển khai cơ sở dữ liệu, đảm bảo hệ thống có thể lưu trữ và truy xuất dữ liệu hiệu quả.
- Mô tả triển khai ứng dụng phần mềm trên các máy chủ hoặc dịch vụ đám mây (nếu dùng).
 - Triển khai Backend: Mô tả quá trình cài đặt và cấu hình backend, bao gồm việc thiết lập server (nginx, Apache) và cài đặt các dịch vụ liên quan.
 - Triển khai Frontend: Trình bày cách triển khai giao diện web (SPA) trên máy chủ hoặc dịch vụ lưu trữ như Netlify, Vercel.
 - Tích hợp với cơ sở dữ liệu và API: Mô tả quá trình kết nối ứng dụng với cơ sở dữ liệu và API. Ví dụ: Triển khai ứng dụng ReactJS lên Netlify, và backend Node.js kết nối với cơ sở dữ liệu trên AWS RDS.

CHƯƠNG 6: TRIỂN KHAI VÀ ĐÁNH GIÁ (DEPLOYMENT AND EVALUATION)

Chương này tập trung vào các hoạt động kiểm thử nhằm đảm bảo hệ thống hoạt động chính xác và ổn định. Kiểm thử không chỉ nhằm phát hiện lỗi mà còn để xác nhận rằng hệ thống đáp ứng các yêu cầu đề ra. Có thể đề cập các nội dung như sau:

- Chiến lược kiểm thử: Giải thích phương pháp kiểm thử (kiểm thử đơn vị, kiểm thử tích hợp, kiểm thử hệ thống).
- Kiểm thử tự động: Mô tả cách bạn thiết lập kiểm thử tự động nếu có. Ví dụ: “Sử dụng Jenkins để thiết lập CI/CD và tự động hóa các bài kiểm thử đơn vị.”
- Kiểm thử hiệu năng: Đánh giá hiệu suất của hệ thống dưới tải nặng. Ví dụ: “Hệ thống có thể xử lý 200 yêu cầu kiểm tra giọng nói trong một giây với thời gian phản hồi trung bình là 300ms.”

- Kiểm thử khả dụng: Mô tả kết quả kiểm thử với người dùng cuối và những cải tiến dựa trên phản hồi.
- Theo dõi lỗi và khắc phục: Trình bày cách quản lý lỗi (sử dụng JIRA, GitHub Issues).
- Kiểm thử hiệu năng và khả năng mở rộng (Performance and Scalability Testing): Đảm bảo hệ thống có thể hoạt động hiệu quả dưới tải trọng cao và có khả năng mở rộng khi cần thiết. Đo lường thời gian phản hồi, tốc độ xử lý và khả năng đáp ứng của hệ thống dưới các điều kiện khác nhau. Các loại kiểm thử hiệu năng:
 - o Load Testing: Kiểm tra hệ thống dưới tải trọng bình thường.
 - o Stress Testing: Xem hệ thống hoạt động như thế nào dưới tải trọng cực kỳ cao.
 - o Endurance Testing: Kiểm tra hiệu năng của hệ thống khi chạy liên tục trong một khoảng thời gian dài.

CHƯƠNG 7: KẾT LUẬN

Đây là nơi các nhóm phân tích và diễn giải kết quả, đồng thời đưa ra những đánh giá và nhận xét về các phát hiện của nhóm. Chương này chủ yếu tóm lược lại những gì đã báo cáo ở các chương trước, chương tổng kết có thể được chia làm 3 mục con riêng biệt:

6.1 Kết quả đạt được: Trình bày một cách rõ ràng và chi tiết các kết quả thu được từ sản phẩm của nhóm. Phân tích và giải thích các kết quả thu được, làm rõ ý nghĩa của chúng và liên hệ với các mục tiêu nghiên cứu

6.2 Những ưu điểm và khuyết điểm.

6.3 Hướng phát triển trong tương lai.

NHỮNG ĐIỀU TUYỆT ĐỐI CẦN TRÁNH

- Viết sai tên trường, tên khoa, tên đề tài, tên, và mã số của thành viên nhóm mình.
- Viết sai tên, sai học vị của giảng viên hướng dẫn.
- Sao chép nội dung trên mạng và dán trực tiếp vào báo cáo.
- Sử dụng công cụ AI để tạo nội dung báo cáo.
- Lạm dụng cách trình bày theo hướng gạch đầu dòng (bullet).

NHỮNG ĐIỀU TUYỆT ĐỐI CẦN TRÁNH

- Viết sai tên đề tài.
- Viết sai tên, mã số của mình và thành viên nhóm mình.
- Viết sai tên giảng viên hướng dẫn.
- Sao chép nội dung trên mạng và dán trực tiếp vào báo cáo.
- Sử dụng công cụ AI để tạo nội dung báo cáo.
- Lạm dụng cách trình bày theo hướng gạch đầu dòng (bullet)

NHỮNG LỖI THƯỜNG GẶP TRONG BÁO CÁO

- Viết sai chính tả.
- Nội dung tìm hiểu sơ sài, viết báo cáo cầu thả.
- Không xóa các comment, các dữ liệu mẫu trong mẫu báo cáo mà Khoa cung cấp.
- Trình bày xấu, không có sự đồng nhất về font chữ, kích thước chữ, sự đồng nhất về bullet & numbering...
- Các lỗi cơ bản do cầu thả như: viết in hoa tên mình nhưng lại viết thường tên giảng viên hướng dẫn, để nguyên tên người hướng dẫn là “Nguyễn Văn A” như trong mẫu cung cấp, ghi sai học vị của giảng viên hướng dẫn...

Đây là ví dụ của việc lạm dụng trình bày theo hướng sử dụng gạch đầu dòng (bullet) quá nhiều. Trong báo cáo tiểu luận/đồ án/khóa luận, sinh viên nên trình bày nội dung theo từng đoạn văn, mỗi đoạn văn gồm nhiều câu viết liên tục.

NHỮNG LỖI THƯỜNG GẶP KHÁC TRONG BÁO CÁO

- Viết sai chính tả: sinh viên cần kiểm tra kỹ chính tả trước khi gửi báo cáo cho giảng viên chấm, hoặc kể cả khi gửi báo cáo cho giảng viên hướng dẫn review/góp ý.
- Nội dung tìm hiểu sơ sài, viết báo cáo cầu thả, thiếu sự đầu tư.
- Không xóa các comment, các dữ liệu mẫu trong mẫu báo cáo mà Khoa cung cấp.
- Trình bày xấu, không có sự đồng nhất về font chữ, kích thước chữ, sự đồng nhất về bullet & numbering, không đồng nhất và mức độ dẫn dòng.
- Các lỗi cơ bản do cầu thả như: viết in hoa tên mình nhưng lại viết thường tên giảng viên hướng dẫn, để nguyên tên người hướng dẫn là “Nguyễn Văn A” như trong mẫu cung cấp, ghi sai học vị của giảng viên hướng dẫn...

NHỮNG ĐIỀU NÊN LÀM

- Kiểm tra lỗi chính tả thật kỹ trước khi nộp bài, trước khi gửi cho giảng viên hướng dẫn.
- Trình bày trang bìa thật cẩn thận, nên viết IN HOA tên đề tài, kiểm tra logo trường nếu độ phân giải thấp phải thay đổi ảnh khác và scale lại cho phù hợp tỷ lệ với trang giấy.
- Sử dụng font chữ Times New Roman, kích thước 13pt, chế độ giãn dòng là 1.5 và **căn đều** cho văn bản. Các tiêu đề có font lớn hơn một chút, ví dụ 15pt.
- Phải có mục lục nội dung được sinh tự động và nên có **tối đa là 3 cấp độ**, hạn chế sử dụng tới 4 cấp độ hoặc hơn nếu không cần thiết (ví dụ mục 4.1.2.4).
- Nếu có hình ảnh, bảng, ký hiệu thì phải có mục lục riêng cho các phần này, cách làm tương tự như mục lục nội dung. Nếu không có bảng nào trong báo cáo thì **ĐỪNG** làm một trang trống với tiêu đề “DANH MỤC BẢNG”.
- Nội dung của báo cáo nên viết theo dạng đoạn văn, hạn chế viết theo dạng liệt kê từng dòng. Ở những nơi sử dụng liệt kê thì nên có văn bản giới thiệu, giải thích về các nội dung liệt kê đó. Ví dụ ở mục “chức năng của ứng dụng”, các nhóm nên tránh gạch đầu dòng và ghi từng chức năng ra rồi kết thúc. Cách làm tốt hơn là ghi tên từng chức năng ở mỗi gạch đầu dòng và viết đoạn văn mô tả, giới thiệu thêm về chức năng đó.
- Không được sao chép nội dung trên các blog và đưa vào báo cáo, cần phải đọc hiểu và tự trình bày lại theo cách hiểu của mình, tránh trường hợp copy-paste mà không hiểu đoạn văn đó nói về cái gì.
- Các hình ảnh, bảng, sơ đồ trong báo cáo cần có caption, sử dụng font nhỏ hơn khoảng 1-2pt so với font của văn bản, in nghiêng, căn giữa và phải đánh số. Ví dụ hình 1.2, bảng 3.1 (tên chương kèm theo tên hình) hoặc hình 1, hình 2, hình 3 (đánh số tăng dần từ 1).
- Đối với các bảng, nên sử dụng font chữ nhỏ hơn khoảng 2pt so với văn bản, nội dung trong bảng nên chọn căn giữa (vertical & horizontal) ở hầu hết các trường hợp. Tiêu đề bảng nên chỉnh in đậm và IN HOA, nếu cần có thể chỉnh background các ô tiêu đề hơi xám một chút.
- Khi kết thúc một chương, **cần sang trang mới** để bắt đầu chương tiếp theo, không được bắt đầu chương mới trong cùng 1 trang với chương cũ vì xấu và làm cho người đọc bị sót chương.
- Hạn chế đưa mã nguồn vào báo cáo. Một số đề tài đặc thù có thể đưa vào nhưng phải trình bày font code riêng và dễ đọc, cần diễn giải về đoạn source code đó. Không được chụp ảnh màn hình source code từ phần mềm IDE và dán thẳng vào báo cáo mà nên

copy source vào sau đó format lại font chữ, kích thước và màu nền cho hợp lệ. Không nên sử dụng background màu đen cho code block.

- Không đề cập quá chi tiết về cách cài phần mềm, cài tool trong báo. Nếu đề cập về việc cài công cụ chỉ cần trình bày đơn giản như: muốn chạy Java thì cần JDK (giới thiệu sơ về JDK), JDK có thể được tải về từ website ... sau khi cài đặt cần thiết lập biến môi trường... như vậy là đủ. Không nói những vấn đề như tại giao diện A chúng ta cần nhấn Next, giao diện B chọn chữ “xyz”...
- Khi đưa hình ảnh, bảng, sơ đồ vào báo cáo thì cần có đoạn văn giới thiệu, diễn giải về các hình ảnh, bảng, sơ đồ này chứ không chỉ đưa vào cho vui. Ví dụ khi làm app mobile và giới thiệu về phần giao diện, một số nhóm chỉ chụp screenshot các màn hình và đưa vào báo cáo rồi ghi ngắn gọn “màn hình đăng nhập”, “màn hình đăng xuất”... cách làm này là chưa đạt, cần phải mô tả thêm về các màn hình này như nó có các chức năng nào, user làm được những gì ở màn hình này hoặc là thứ tự các bước diễn ra theo trình tự nào...
- In báo cáo 1 mặt và không cần in màu nếu không cần thiết (trừ luận văn tốt nghiệp, hầu hết các báo cáo khác không cần in màu, trừ khi các nhóm chủ định in màu để nhấn mạnh nội dung trong ảnh).
- Với các đề tài mang tính tìm hiểu chẳng hạn như “tìm hiểu các cách kết nối cơ sở dữ liệu trong Java” thì ngoài việc tìm hiểu ra, cũng nên có thêm các phần so sánh đối chiếu và phân tích cách nào tốt hơn cách nào hoặc khi nào thì dùng cách này, khi nào thì dùng cách kia.
- Với tất cả các đề tài, làm đầy đủ và chi tiết là điều không thừa. Ví dụ với đề tài bài tập lớn là “Tìm hiểu cách kết nối cơ sở dữ liệu trong Java và viết website minh họa”, nếu nhóm làm một website sơ sài chẳng hạn như tạo database có 1 bảng sinh viên với thông tin của 1 sinh viên sau đó website đọc thông tin sinh viên này rồi hiện lên dưới dạng plan text, thì demo này cũng được gọi là “kết nối cơ sở dữ liệu và hiển thị trên website” tuy nhiên nó quá sơ sài. Cách làm tốt hơn là nên có một bảng với nhiều records, ví dụ một bảng nhiều sinh viên và ở phía website nên cung cấp các chức năng như: hiển thị danh sách sinh viên, thêm một sinh viên mới, xóa một sinh viên, cập nhật một sinh viên... nếu kết hợp được với bootstrap và jquery để làm cho trang sinh động hơn thì đó là một điều tốt.
- Với các đề tài cuối kỳ như “xây dựng website bán điện thoại di động” thì sản phẩm các em đem nộp, đem chấm (ở điều kiện lý tưởng) phải là một trang web hoạt động ổn như các yêu cầu đã đề ra, có nhiều danh mục (samsung, apple, nokia..) mỗi danh mục có

nhiều thiết bị điện thoại với nhiều mức giá khác nhau, thậm chí mỗi điện thoại cũng nên có slide show các hình ảnh liên quan... Tránh trường hợp một số nhóm làm đề tài bán điện thoại mà website mở lên chỉ có lèo tèo vài cái điện thoại và nhóm khi báo cáo thì trả lời với giảng viên chấm bài rằng “cái này thêm nhanh mà, chẳng qua em chưa thêm”. Các em cần lưu ý rằng sản phẩm khi các em báo cáo chính là sản phẩm dùng để tính điểm, nếu làm sơ sài thì các em sẽ nhận được điểm thấp tương ứng (mặc dù các em đã có thể làm tốt hơn). Giống người chơi game show các chương trình gọi vốn cũng vậy, nếu sản phẩm của họ rất tốt nhưng màn thuyết trình của họ quá tệ không đủ để thuyết phục các shark thì họ cũng phải chấp nhận việc thất bại, không thuyết phục được đầu tư từ các shark. Vì vậy cần phải hoàn thiện tốt nhất có thể khi nộp bài trên hệ thống elearning cũng như khi báo cáo trực tiếp với giảng viên.

**** Các nhóm lưu ý, kể từ khi hướng dẫn này được cung cấp đến các nhóm thì các nhóm phải có trách nhiệm đọc hiểu và áp dụng vào báo cáo của mình. Ngoài ra các nhóm cũng tự chủ động tìm hiểu thêm để trình bày báo cáo một cách tốt nhất có thể.*