

## TIÊU CHÍ ĐÁNH GIÁ CHUNG CHO CÁC ĐỀ TÀI ĐỒ ÁN

Để đảm bảo đồ án đạt yêu cầu của môn học Nhập môn Công nghệ Phần mềm, các đề tài cần đáp ứng các tiêu chí sau:

### 1. Quy trình phát triển phần mềm

- Áp dụng quy trình phát triển phần mềm: Agile-Scrum (Ưu tiên), Waterfall, Spiral).
- Có lập kế hoạch phát triển theo Sprint nếu áp dụng Agile-Scrum
- Báo cáo rõ ràng các giai đoạn trong quy trình:
  - Thu thập yêu cầu
  - Phân tích và thiết kế
  - Cài đặt và kiểm thử
  - Tổng hợp và hướng phát triển

### 2. Phân tích và mô hình hóa yêu cầu

- Xác định rõ yêu cầu hệ thống và phạm vi đồ án.
- Sử dụng các công cụ mô hình hóa:
  - Use Case Diagram (Xác định chức năng chính)
  - ERD (Entity Relationship Diagram) (Thiết kế cơ sở dữ liệu)
  - DFD (Data Flow Diagram) (Mô tả luồng xử lý dữ liệu)
  - UML (Class Diagram, Sequence Diagram, Activity Diagram, State Diagram, ...)

### 3. Thiết kế kiến trúc phần mềm

#### 3.1 Chọn mô hình kiến trúc phù hợp với hệ thống:

- Monolithic Architecture (Kiến trúc đơn khối): Phù hợp với hệ thống nhỏ hoặc MVP.
- Client-Server Architecture: Hệ thống có phân chia giữa máy khách (Client) và máy chủ (Server).
- Layered Architecture (Kiến trúc phân lớp): Tách biệt các thành phần giao diện, xử lý nghiệp vụ, và truy xuất dữ liệu.
- Model-View-Controller (MVC): Phù hợp với ứng dụng web và desktop.
- Microservices Architecture (Kiến trúc dịch vụ nhỏ): Dành cho hệ thống lớn, dễ mở rộng.
- Event-Driven Architecture: Phù hợp với hệ thống cần xử lý sự kiện theo thời gian thực.

### 3.2 Nguyên tắc thiết kế kiến trúc:

- Tính mô-đun(Module): Hệ thống được chia thành các mô-đun độc lập, dễ bảo trì.
- Tính mở rộng (Scalability): Hệ thống có thể mở rộng theo nhu cầu sử dụng.
- Tính bảo mật (Security): Kiểm soát truy cập, mã hóa dữ liệu, bảo vệ thông tin người dùng.
- Tính tái sử dụng (Reusability): Sử dụng lại các thành phần phần mềm khi cần thiết.
- Tính hiệu suất (Performance): Đảm bảo thời gian phản hồi nhanh và tối ưu tài nguyên.

### 3.3 Tài liệu kiến trúc phần mềm phải bao gồm:

- Sơ đồ kiến trúc tổng quan: Biểu diễn cấu trúc phần mềm, các thành phần và cách chúng tương tác.
- Mô tả chi tiết các thành phần: Giải thích từng phần trong hệ thống (Frontend, Backend, Database, API, ...).
- Quy trình xử lý dữ liệu: Cách dữ liệu di chuyển trong hệ thống (luồng dữ liệu).
- Mô hình triển khai: Cách hệ thống được triển khai trên môi trường thực tế (Cloud, On-premise, Hybrid).
- Tài liệu hướng dẫn phát triển: Quy tắc lập trình, tiêu chuẩn mã nguồn.

## 4. Thiết kế cơ sở dữ liệu và quản lý dữ liệu

- Xây dựng cơ sở dữ liệu đầy đủ và tối ưu hóa dữ liệu.
- Đảm bảo tính đúng đắn, toàn vẹn dữ liệu và khả năng mở rộng.
- Cung cấp tài liệu chi tiết về:
  - Danh sách các bảng dữ liệu
  - Mối quan hệ giữa các thực thể
  - Mô tả dữ liệu và ràng buộc toàn vẹn

## 5. Kiểm thử phần mềm và báo cáo lỗi

- Xây dựng test case và kiểm thử phần mềm theo nhiều cấp độ:
  - Unit Testing (Kiểm thử đơn vị)
  - Integration Testing (Kiểm thử tích hợp)
  - System Testing (Kiểm thử hệ thống)
  - User Acceptance Testing (UAT) (Kiểm thử chấp nhận từ người dùng)
- Báo cáo chi tiết về các lỗi đã phát hiện và cách khắc phục.
- Sử dụng công cụ kiểm thử tự động nếu có (Selenium, JUnit, Postman).

## **6. Triển khai thực tế và tài liệu hướng dẫn**

- Cung cấp bản cài đặt phần mềm và tài liệu hướng dẫn sử dụng.
- Phần mềm có thể chạy trên nhiều nền tảng (Windows, Web, Mobile).
- Hướng dẫn cách cấu hình và cài đặt phần mềm.

## **7. Báo cáo đầy đủ, có nhận xét và hướng phát triển**

- Tài liệu báo cáo cần có các phần chính:
  - Giới thiệu đề tài
  - Phân tích yêu cầu hệ thống
  - Thiết kế hệ thống và database
  - Kiểm thử và kết quả chạy thử
  - Nhận xét & đánh giá
  - Hướng phát triển phần mềm trong tương lai
- Nhóm cần ghi rõ các hạn chế của phần mềm và đề xuất cải tiến.

--- The End ---