

ĐỒ ÁN MÔN HỌC

Môn học: Nguyên lý xây dựng phần mềm

Thời gian thực hiện: 8 tuần

Giảng viên phụ trách: TS. Trương Minh Thái

Hình thức thực hiện: Nhóm 4–5 sinh viên

1. Mục tiêu dự án

Phát triển một phần mềm ở quy mô vừa (medium-scale), áp dụng các nguyên lý thiết kế phần mềm hiện đại nhằm đảm bảo:

- Tái sử dụng cao (design for reuse)
- Dễ thích ứng với thay đổi (design for change)
- Dễ bảo trì và mở rộng (maintainability & extensibility)
- Áp dụng mẫu thiết kế (design pattern)
- Lập trình đa luồng (concurrency)
- Kiểm thử và đảm bảo chất lượng (unit testing, static analysis)
- Tổ chức mã nguồn rõ ràng, có kiến trúc phần mềm hợp lý

2. Yêu cầu chức năng & kỹ thuật

1. Chủ đề dự án:

Phát triển một ứng dụng quản lý công việc nhóm với quy mô vừa, cho phép các nhóm làm việc cộng tác hiệu quả trong việc phân công, theo dõi và hoàn thành công việc.

Chức năng chính

- Quản lý dự án: Tạo, chỉnh sửa, xóa dự án; phân quyền quản lý dự án.
- Quản lý công việc: Tạo công việc, gán người thực hiện, đặt deadline, trạng thái (To Do, In Progress, Done, Cancel...).
- Quản lý người dùng: Đăng ký, đăng nhập, phân vai trò (Admin, Manager, Member).
- Thông báo: Gửi thông báo khi có thay đổi trạng thái công việc hoặc deadline sắp đến.
- Bình luận & trao đổi: Cho phép thảo luận trong từng công việc.
- Thống kê & báo cáo: Số lượng công việc theo trạng thái, tiến độ dự án.

Yêu cầu kỹ thuật

- Ngôn ngữ: Java
- IDE: Eclipse hoặc NetBeans
- Kiến trúc phần mềm:
- Mô hình đa tầng (multi-layered architecture): UI – Business Logic – Data Access
- Thiết kế UML: class diagram, sequence diagram, use-case diagram
- Thiết kế low coupling – high cohesion; tách biệt giao diện, xử lý nghiệp vụ và dữ liệu.
- Áp dụng Mẫu thiết kế (Design Patterns): Dự án yêu cầu áp dụng linh hoạt các mẫu thiết kế như Observer (quan sát dữ liệu), Strategy (chiến lược cập nhật), Factory (tạo đối tượng), và Template Method (khung sườn thuật toán) để giải quyết các bài toán về kiến trúc.
- Kiểm thử & chất lượng: Viết unit test bằng JUnit cho các lớp nghiệp vụ chính

3. Kết quả nộp

- Mã nguồn đầy đủ trên GitHub (có README hướng dẫn cài đặt và chạy thử)
- Báo cáo: tài liệu đặc tả yêu cầu (10-15 trang), tài liệu thiết kế (10–15 trang) gồm:
 - Kiến trúc hệ thống
 - UML diagrams
 - Design patterns đã áp dụng
 - Quy trình kiểm thử
 - Đánh giá khả năng mở rộng, tái sử dụng, bảo trì
- Video demo (5–10 phút) trình bày ứng dụng và các đặc điểm thiết kế
- Bài thuyết trình nhóm (trình bày trong tuần thứ 12)

4. Kế hoạch dự kiến (Mốc thời gian bắt đầu là tuần thứ 4 của học kỳ)

Dự án được chia thành 4 giai đoạn (Sprints), mỗi giai đoạn 2 tuần, mô phỏng quy trình phát triển Agile.

Giai Đoạn 1: Phân Tích & Thiết Kế Kiến Trúc (Tuần 1-2)

Mục tiêu: Xác định "Hợp đồng" và Mô hình dữ liệu (Domain Model).

Hoạt động:

- Phân tích yêu cầu và vẽ biểu đồ Use Case.
- Thiết kế biểu đồ lớp (Class Diagram) cho Core và cấu trúc Plugin.
- Định nghĩa Interface Plugin và HostContext bằng mã nguồn Java.

Kết quả: Tài liệu thiết kế v1.0 và bộ khung Interfaces.

Giai Đoạn 2: Xây Dựng Core Framework (Tuần 3-4)

Mục tiêu: Xây dựng "Container" chạy được.

- Hiện thực lớp PluginLoader sử dụng Java Reflection để quét và nạp lớp.
- Xây dựng cửa sổ chính DashboardFrame với hệ thống Menu và Layout.
- Hiện thực cơ chế xử lý lỗi tập trung.

Kết quả: Một ứng dụng Dashboard trông có thể nạp một "Hello World" Plugin đơn giản.

Giai Đoạn 3: Phát Triển Plugins & Đa Luồng (Tuần 5-6)

Mục tiêu: Chứng minh tính mở rộng và khả năng xử lý song song.

- Phát triển Plugins.
- Áp dụng SwingWorker và Timer để xử lý cập nhật dữ liệu nền.
- Hiện thực giao diện người dùng chi tiết cho từng Widget.

Kết quả: Hệ thống hoàn chỉnh về mặt chức năng.

Giai Đoạn 4: Kiểm Thử, Tối Ưu & Tài Liệu (Tuần 7-8)

Mục tiêu: Đảm bảo chất lượng và đóng gói.

- Viết Unit Test và Integration Test.
- Rà soát mã nguồn (Code Review) để đảm bảo tuân thủ quy tắc đặt tên và tính đóng gói (Information Hiding).
- Viết tài liệu hướng dẫn sử dụng và báo cáo tổng kết.

Kết quả: Sản phẩm cuối cùng, mã nguồn, và báo cáo.