ỨNG DỤNG DESIGN PATTERNS TRONG MÔ HÌNH QUẢN LÝ HỆ THỐNG

Lớp: CTK44-PM

Môn học: Mẫu thiết kế

Nhóm: 10

Giảng viên: Thầy Nguyễn Minh Hiệp

THÀNH VIÊN NHÓM 10

|  |  |  |  |
| --- | --- | --- | --- |
| STT | MSSV | Họ và tên | Lớp |
| **1** | 2011356 | Hoàng Nghĩa Minh Bảo | CTK44-PM |
| **2** | 2012365 | Nguyễn Minh Long | CTK44-PM |
| **3** | 2011433 | Lê Văn Tài | CTK44-PM |
| **4** | 2011417 | Hồ Trương Huệ Nhật | CTK44-PM |
| **5** | 2012374 | Lưu Ngọc Quốc | CTK44-PM |
| **6** | 2012349 | Mai Quang Định | CTK44-PM |
| **7** | 2011384 | Đoàn Đình Hoàng | CTK44-PM |

# 1. Giới thiệu

Hệ thống quản lý hiện đại đòi hỏi thiết kế phần mềm linh hoạt, dễ mở rộng và dễ bảo trì. Để đạt được điều này, việc áp dụng các mẫu thiết kế (Design Patterns) là cần thiết.  
Trong báo cáo này, chúng tôi xây dựng một mô hình quản lý hệ thống tổng quát, bao gồm:  
- Quản lý người dùng (Admin, Nhân viên)  
- Kết nối cơ sở dữ liệu  
- Ghi log hệ thống  
- Tích hợp dịch vụ và phân tách rõ ràng các lớp trách nhiệm

# 2. Các mẫu thiết kế áp dụng

|  |  |
| --- | --- |
| Pattern | Vai trò trong hệ thống |
| Singleton | Đảm bảo chỉ một kết nối database (DatabaseConnection) hoạt động trong toàn hệ thống |
| Factory | Tạo user (Admin, Employee) linh hoạt không phụ thuộc lớp |
| Observer | Ghi log tự động khi có sự kiện xảy ra (thêm người dùng) |
| Service Layer | Tách biệt nghiệp vụ (UserService) với phần giao diện (UI) và dữ liệu (Database) |

# 3. Sơ đồ lớp (Class Diagram)

Sơ đồ lớp mô tả các thành phần chính và mối quan hệ giữa chúng:  
- UI → UserService → UserFactory → DatabaseConnection → Logger  
- Các phần giao tiếp qua interface giúp giảm ràng buộc.

A screenshot of a computer

AI-generated content may be incorrect.

# 4. Sơ đồ tuần tự (Sequence Diagram)

Sơ đồ tuần tự mô tả quá trình thêm người dùng:  
1. UI gửi yêu cầu AddUser  
2. UserService tạo user, lưu database, ghi log  
3. Kết quả trả về UI

A black screen with white text

AI-generated content may be incorrect.

# 5. Giải thích code

- IUser, Admin, Employee: các lớp người dùng  
- UserFactory: áp dụng Factory Pattern để tạo Admin/Employee  
- DatabaseConnection: Singleton đảm bảo chỉ 1 kết nối database  
- Logger, IObserver, Subject: Observer Pattern ghi log sự kiện  
- UserService: Service Layer xử lý nghiệp vụ

# 6. Kết quả chạy thử

Database connected.  
[LOG] User 'Alice' of type 'Admin' added.  
Alice is an Admin.  
[LOG] User 'Bob' of type 'Employee' added.  
Bob is an Employee.

# 7. Kết luận

Mô hình này cho thấy sức mạnh của việc áp dụng Design Patterns:  
✅ Giảm phụ thuộc giữa các lớp  
✅ Tăng khả năng mở rộng  
✅ Dễ bảo trì  
  
Có thể mở rộng thêm:  
- Strategy Pattern: Xuất báo cáo PDF/Excel  
- Command Pattern: Undo/Redo thao tác  
- Proxy Pattern: Bảo vệ quyền truy cập tài nguyên